UNIVERSIDAD DE LA REPÚBLICA

PROYECTO DE GRADO

Diseño de una Herramienta para la Conformación de Equipos Multidisciplinarios

Autores:
Camilo SERVETTI DE BEN
Matias BANCHERO MARTINEZ

Tutores:
Dr. Ing. Franco ROBLEDO
Dr. Ing. Pablo SARTOR

Facultad de Ingeniería
11 de Marzo de 2016

UNIVERSIDAD DE LA REPÚBLICA

Resumen

Diseño de una Herramienta para la Conformación de Equipos Multidisciplinarios

Se resuelve un problema de optimización consistente en lograr equipos con máxima diversidad interna, haciendo esto para varios períodos consecutivos y controlando la repetición de parejas de integrantes en equipos de distintos períodos. Maximizar la diversidad en equipos multidisciplinarios significa encontrar una configuración de equipos en la cual todos los atributos que representan a las personas estén uniformemente distribuidos logrando equipos de similar desempeño potencial. En este proyecto se utiliza la meta-heurística GRASP-VND basada en búsquedas de entornos variables para solucionar el problema de optimización y se aplica una búsqueda tabú para minimizar la cantidad de repeticiones en los equipos formados para los distintos trimestres.

Palabras clave: GRASP, VND, Tabu Search, Equipos multidisciplinarios.

Índice

Lista de Abreviaciones

MBA	**M**aster in **B**usiness **A**dministration
VNS	**V**ariable **N**eighborhood **S**earch
GVNS	**G**eneral **V**ariable **N**eighborhood **S**earch
VND	**V**ariable **N**eighborhood **D**escent
SGP	**S**ocial **G**olfer **P**roblem
RAE	**R**eal **A**cademia **E**spañola

Lista de Símbolos

N	cantidad de estudiantes
G	cantidad de grupos
M	cantidad de estudiantes en cada grupo donde $M = \frac{N}{G}$ (entero)
K	cantidad de atributos
S	cantidad de trimestres
d_{ij}	distancia entre el estudiante i y el estudiante j
a_{ik}	es el atributo k para el estudiante i
$sd^c[i][g]$	diversidad entre estudiante i e integrantes del equipo g

Capítulo 1

Introducción

1.1. Descripción del Problema

Dada la dificultad e importancia que existe a la hora de conformar equipos de trabajo multidisciplinarios en el contexto de un determinado proyecto, trabajo, o estudio académico entre otros, surge la necesidad de tener una herramienta que nos facilite esta tarea.
En base a una cantidad dada de personas se pretende generar grupos diversos con una cierta cantidad de integrantes y que a su vez cumplan con ciertas restricciones. Estas restricciones van a modelar el problema objetivo en base a limitantes establecidas para individuos con determinadas características, un ejemplo puede ser del estilo que dos integrantes con ciertas cualidades no puedan pertenecer al mismo equipo.
La conformación de estos equipos tiene que ser de tal forma que se obtengan los mejores resultados globales, esto se entiende como un concepto muy amplio el cual dependerá particularmente del fin que exista detrás de estos equipos. Si lo analizamos desde distintos puntos de vista vamos a tener distintos fines los cuales pueden ser económicos, sociales o académicos entre otros.

En este trabajo vamos a centrarnos en el contexto particular de la conformación de equipos para estudiantes de MBA (Master in Business Administration). Dados los fines académicos que existen detrás de estos equipos podemos identificar que una partición es mejor que otra en la medida que la calidad del equipo sea heterogénea, entendiéndose por calidad un mix de experiencia, currículum, edad etc.

Resulta deseable que los estudiantes integren equipos con distintas personas en donde la diferencia de las características personales de cada alumno que integra un grupo ayude a enriquecer la experiencia de los demás alumnos del grupo y así lograr que los estudiantes obtengan aprendizajes de los demás.
Se considera positivo que los grupos no queden desnivelados a nivel académico, es decir, no se querrán equipos donde los que obtuvieron mejores resultados en las pruebas están todos en un mismo grupo y los que obtuvieron notas más bajas estén en otro equipo.

Para cumplir con estos fines necesitaremos de una definición que abarque estos conceptos de homogeneidad y calidad, siendo este un problema difícil y de importancia para poder llegar a obtener buenos resultados. Un factor para determinar que una partición es mejor que otra será comparando ciertos atributos propios de cada persona, esto refiere a que haya hombres y mujeres, profesiones diversas, nacionalidades diversas, edades diferentes y otros parámetros que enriquecen la multidisciplinariedad y diversidad del equipo. Adicionalmente existe el interés de que estos equipos roten sus integrantes una cierta cantidad de veces dado que en el curso de MBA se conforman distintos grupos en los distintos trimestres, lo cual haría que se introduzcan restricciones del estilo que dos estudiantes no puedan estar juntos repetidas veces en los distintos grupos que formarán durante la maestría.

1.2. Objetivos Buscados

Buscaremos definir el término diversidad y obtener ciertos atributos de las personas que nos sean útiles para enfocar la calidad de los resultados al escenario planteado. Para esto necesitaremos definir una ponderación de atributos y una manera de relacionarlos y medirlos. También se deberán discutir las restricciones existentes en el problema planteado.

Se pretende dar una solución a este problema siendo modelado como un problema de Investigación de Operaciones y desarrollar un algoritmo adecuado para llegar a una buena solución y que a su vez sea eficiente.

También pretendemos que este proyecto brinde una solución amplia que aplique a problemas de esta índole permitiendo una buena adaptación a cada problema en particular. Esto puede ser posible dado que como entrada vamos a tener diferentes configuraciones propuestas por el usuario con valores parametrizables.

Existen diferentes metodologías para la resolución de este problema entre las cuales podemos remarcar técnicas, GRASP, greedy, algoritmos evolutivos, programación lineal o búsquedas en entornos variables como las más relevantes. Como resultado implementaremos una herramienta la cual, en base al modelo desarrollado y la aplicación de una de estas técnicas, resuelva el problema de forma eficiente.

Dado que no podemos hacer un fuerte estudio psicológico de las personas que van a formar estos grupos, pretendemos tener los mejores resultados en base a la información a la que podemos acceder. Parte de este estudio pretende identificar cuál información es relevante y qué ponderación tiene que tener para que se vea positivamente reflejada en los resultados finales.

1.3. Resultados Esperados

Se espera que la herramienta desarrollada otorgue mejores soluciones que las obtenidas mediante el proceso de formación de grupos manual. Se pretende obtener mejoras en el tiempo invertido, la calidad de la solución y la cantidad de repeticiones entre estudiantes para los diferentes trimestres. Se espera obtener los grupos para cada trimestre en una sola ejecución del programa, que los grupos estén balanceados entre sí y tengan la mayor diversidad posible entre sus integrantes. Como objetivo secundario, pero no menos importante, se pretende obtener resultados cuyas repeticiones entre estudiantes en los grupos formados sean lo menor posible.

1.4. Antecedentes

Problemas similares al de este proyecto fueron tratados en (Bhadurya, Mightyb y Damar, 2000), (Desrosiers, Mladenović y Villeneuve, 2005) y (Baker y Benn, 2001).

En (Bhadurya, Mightyb y Damar, 2000) se presenta un modelo simplificado el cual requiere de un alto grado de similitud entre los equipos y utiliza el problema de los filósofos comensales para asignar estudiantes a los diferentes grupos.

En (Desrosiers, Mladenović y Villeneuve, 2005) se modela el problema con programación lineal. El artículo usa un centroide para representar a cada grupo de entidades. Se proponen dos enfoques para el modelado, primero el min-sum que minimiza la suma de distancias entre los centroides de dos grupos y luego la distancia de los centroides de los grupos respecto al centroide general el segundo enfoque es min-max el cual minimiza el máximo de dichas distancias.

En (Baker y Benn, 2001) el caso de estudio consiste en asignar a 8 tutores el total de 235 estudiantes. Este estudio combina programación lineal y es equivalente a la formulación realizada en (Desrosiers, Mladenović y Villeneuve, 2005) de min-sum.

El problema que abordado es conocido por ser NP-difícil (Feo y Khellaf, 1990) y se han desarrollado heurísticas para resolverlo. Encontramos en (Fan y col., 2011) la propuesta de un algoritmo genético híbrido para su resolución. Por su parte (Fan y col., 2011) sugiere tabú search con oscilaciones estratégicas. Además (Rodriguez y col., 2013) propone el método de la colonia de abejas artificiales.

También se propone un método basado en la heurística *general variable neighborhood search* (GVNS) en (Dragan, 2014), donde GVNS es una variante al *variable neighborhood search* (VNS) con búsqueda local determinística que usa *variable neighborhood descent* VND el cual es aplicado en el paso de la búsqueda local del VNS.

En (Brimberg, Mladenovic y Uroševic, 2015) se encuentra una extensión a este último, el cual implementa el método *shake* para mejorar la búsqueda de mejores resultados durante la heurística y luego aplica una alternativa al VNS conocido como *Skewed VNS*.

Capítulo 2

Análisis

2.1. ¿Qué entendemos por diversidad?

Para crear equipos multidisciplinarios es necesario formar grupos que contengan personas cuyas cualidades particulares sean diferente entre sí, cuanto más diferentes sean estas cualidades entre los integrantes de un mismo grupo mayor será el nivel de diversidad total obtenido.

Para establecer equipos multidisciplinarios se hace evidente la necesidad de definir qué es la diversidad. La Real Academia Española (RAE) lo define como *variedad, desemejanza, diferencia* o también como *abundancia, gran cantidad de varias cosas distintas*. La definición es clara pero parece demasiado amplia por lo que surge la necesidad de contar con una definición más particular para el escenario propuesto, en este caso una clase de estudiantes profesionales recibidos que cursan una maestría.

Pensando en la definición de la RAE notamos que cuando habla de *cosas distintas* en este caso podrían ser atributos que definen a una persona. Existen miles de atributos adjudicables a una persona por lo que nuevamente surge la necesidad de acotar aún más la definición analizando los atributos que serán de importancia en nuestro escenario. Claramente habrá algunos que serán claves dada nuestra realidad y otros que ni siquiera deberán ser tomados en cuenta. Por ejemplo, el tipo de sangre de los distintos integrantes de los grupos para nosotros será irrelevante, pero tal vez si definiéramos diversidad dentro de un contexto de grupos de donantes de sangre este atributo sería clave.

A su vez podría haber una ponderación para los diferentes atributos que consideremos teniendo algunos mayor peso que otros. De nuevo, la forma de definir esta ponderación en los atributos se ve influenciada directamente por el problema particular al que nos enfrentamos, por ejemplo, posiblemente a los efectos de nuestro escenario el atributo "título universitario" del estudiante (Ingeniero, Economista, etc.) tendrá mayor ponderación a la hora de armar los grupos que el atributo lugar de nacimiento (capital o interior), siendo posiblemente más enriquecedor para la diversidad de grupos que se junten personas de distinto título universitario que personas de distintas zonas del país.

Para agregar complejidad la ponderación podría llegar a no ser atribuida simplemente a un atributo sino ser adjudicada a un conjunto de atributos.

Por ejemplo, consideremos los parámetros edad y años de experiencia laboral. Tal vez sea mejor considerar los años de experiencia de trabajo y la edad como un solo atributo que considerarlos por separado, porque tal vez a los efectos de tratarse de una clase de maestría un ingeniero de 20 años con 5 años de experiencia laboral no esté muy distanciado (en términos de diversidad) de un ingeniero de 30 años con 5 años de experiencia laboral. Pero seguramente podría verse ampliamente distanciado si solo se consideran las diferencias de edades sin tener en cuenta los años de experiencia producto de que los años de experiencia son los mismos.

Y el problema podría crecer aún más si se tiene en cuenta las relaciones sociales entre los hombres y las mujeres, y aspectos psicológicos como los cambios en la formas de actuar de las personas a partir de la presencia de otras.

Entonces todo esto lleva a la necesidad de definir una serie de atributos, los cuales contemplen la realidad planteada y analizar la posibilidad de establecer ponderaciones entre atributos que ayuden a representar con mayor fidelidad la diversidad buscada.

2.2. Atributos y ponderaciones

Nuestro escenario refiere a una clase con alumnos en la que se dicta un curso de MBA y en las que habrán profesionales recibidos de distintas universidades y carreras. Los estudiantes durante el curso de la maestría integrarán diferentes grupos con el fin de resolver distintas tareas y ejercicios de trabajo grupal orientados al contexto empresarial y enfocado a temas económicos, de negocio y administrativos.

Habrán hombres y mujeres de distintas edades, provenientes de distintas zonas del país y diferentes contextos sociales. Serán personas con diferentes aptitudes las cuales serán de importancia para la realización de la maestría. A su vez como los estudiantes ya son recibidos seguramente habrán muchos que posean experiencia laboral previa y/o actual de diferentes puestos de trabajo y diferentes cargos.

Todo esto será parte de los atributos buscados y de interés para nuestro problema. Entonces en nuestra definición de diversidad encontramos cuatro aspectos principales de los individuos involucrados:

- Los relacionados a condiciones sociales e individuales en los que se incluye la edad, el género, el lugar de nacimiento, etc.
- Los vinculados a los estudios y experiencias laborales, en los que se contemplan la carrera cursada, el trabajo actual, puesto de trabajo, etc.
- Los asociados a notas académicas obtenidas, que incluye el puntaje obtenido en la prueba de ingreso, la escolaridad, etc.
- Las aptitudes vinculadas a una maestría en MBA, como por ejemplo la capacidad de liderazgo, compañerismo, etc.

En el próximo capítulo, a partir de estos cuatro conjuntos definidos procederemos a desglosarlos en atributos aplicables a los alumnos.

Para nuestro proyecto, a efectos de definir los atributos nos basaremos en los datos que nos pueda otorgar el profesor y director del MBA del IEEM - Universidad de Montevideo - Pablo Sartor. Hasta el día de hoy la conformación de equipos multidisciplinarios en el IEEM se ha hecho en forma manual, basándose en los siguientes atributos:

- Edad
- Género
- Profesión
- Experiencia laboral (años)
- Experiencia post título (años)
- Test de admisión (parte cuantitativa)
- Test de admisión puntaje total
- Puntaje de admisión

Se analizará posteriormente la necesidad o no de contar con otros datos útiles como por ejemplo la institución donde se obtuvo el título universitario, si tienen hijos o no, etc..

En el caso de decidir ponderar los distintos atributos esto deberá realizarse de forma apropiada y estará fuertemente ligada a los datos con los que se cuenten y se elijan y que finalmente se opten por usar.

2.3. Restricciones

Además de la necesidad de definir diversidad y de establecer atributos y ponderaciones, se deberán establecer restricciones que harán que la solución cumpla con determinadas reglas. Estas reglas se desprenden de necesidades y demandas que el propio problema impone. Más adelante profundizaremos en las restricciones de nuestro escenario.

2.4. Análisis de los resultados

En cuanto al análisis de los resultados surge la siguiente pregunta: ¿ es posible saber con certeza si la definición de diversidad que creamos a partir de los atributos es adecuada y que los equipos obtenidos son realmente multidisciplinarios?

Contaremos con un conjunto de datos que consta de conformaciones de grupos multidisciplinarios reales realizadas por Pablo Sartor en los últimos años. Estos datos nos serán de gran utilidad para compararlos con los resultados que nuestra herramienta obtenga. Deberemos contar con una manera de medir la calidad del conjunto de soluciones de años pasados con las que se cuenta y de la misma manera medir la calidad de la solución generada por nuestro programa para luego compararlas. Es de esperar o al menos se pretenderá que la herramienta mejore considerablemente los equipos generados.

Pero esto no contesta a la pregunta planteada, sino que nos servirá como referencia para saber qué tanto podemos mejorar las soluciones creadas manualmente.

Para saber qué atributos son los adecuados será necesario realizar pruebas utilizando diferentes atributos y ver cuales generan mejores resultados académicos, pero no será posible corroborar que el conjunto de atributos elegidos sean los que mejor establezcan la realidad planteada dado que no existen definiciones propias de nuestro escenario, sino que cualquier definición será subjetiva. Entonces lo que sí podremos comparar son los resultados obtenidos con los esperados, y para que los esperados sean lo más similar a la realidad planteada es necesario crear una definición lo más acertada posible, es decir, elegir parámetros y ponderaciones entre atributos que reflejen fielmente la realidad.

Finalmente, necesitaremos evaluar de alguna manera la diversidad en los grupos generados para corroborar que los mismos estén correctamente formados. En el Capitulo 5 profundizaremos en el análisis de los resultados obtenidos.

2.5. Otro punto de vista

Luego, viendo la naturaleza de nuestro escenario nos damos cuenta de que además de componer equipos multidisciplinarios, sería conveniente que los estudiantes obtengan la experiencia más enriquecedora posible, y como enriquecedor se puede definir solamente la experiencia de trabajar con personas distintas o se puede definir el obtener mejores notas, o un una combinación de ambas.

Esto introduce a nuestro problema el concepto de performance o productividad, y fácilmente uno se puede ver tentado a traducir el problema al de definir grupos de forma tal de que la productividad sea la mayor posible. Pero de ser así el objetivo cambiaría al de crear grupos tales que todos los alumnos obtengan el mayor puntaje posible en sus notas durante la maestría. Pero podría pasar que proponiéndonos este objetivo consigamos que los alumnos obtengan el mayor puntaje, pero a costa de formar grupos con mujeres por un lado y grupo de hombres por otro, lo cual claramente difiere con el objetivo original de los equipos multidisciplinarios.

El proyecto se basará en la teoría de que la conformación de equipos multidisciplinarios ayuda a obtener los mayores provechos en la experiencia de los alumnos y así indirectamente mejorará la productividad de los mismos y por consecuente las notas finales obtenidas.

Además de todo esto, se cuenta con una amplia variedad de teorías que podrían ser de interés para ser contempladas en la solución a generar. A continuación veremos algunas de ellas.

2.5.1. Teoría de Auto-categorización (Turner)

La Teoría de la Auto-categorización o Categorización del Yo de Turner (Bárbara Scandroglio, 2008) trata de explicar cuál es el proceso que lleva a las personas a incluirse en una u otra categoría. En qué categoría se incluyen en cada momento dependerá de las circunstancias sociales en las que se encuentren. Esta teoría pone su foco de atención en cómo las personas son capaces de llegar a actuar como un grupo. Su hipótesis básica es que hay una elaboración socio-cognitiva de la identidad social que está en un nivel de abstracción superior a la hora de percibir a uno mismo y a los demás. Existen tres niveles de abstracción según la teoría, el Superordenado (El hombre como ser humano y diferente de otras especies), el Intermedio (Basado en las semejanzas y las diferencias entre los individuos) y el Subordinado (Categorizaciones individuales frente a las de otros sujetos). El comportamiento grupal se puede ver como un cambio en el nivel de abstracción, llevando a una despersonalización pero sin perder la identidad individual. Esta teoría nos sugiere un acercamiento a entender por qué la formación de grupos diversos trae como consecuencia desentendimientos y conflictos.

Si los miembros de un grupo diverso pueden ser persuadidos para alinearse al superordenado del grupo, entonces temporalmente estarán dispuestos a ceder a sus cualidades para seguir al superordinado. Esto hace que las personas no sean ellas mismas con el fin de alinearse al grupo. La teoría de la auto-categorización se contrapone con la hipótesis de la diversidad para la formación de grupos multidisciplinarios.

2.5.2. Teoría de Auto-verificación (Swann)

Esta teoría propone que las personas necesitan que otras personas nos vean como somos. Por ejemplo, las personas que se ven a sí mismas como dominantes necesitan que otros los vean como dominantes. Esta teoría asume que las personas crean vistas de ellos mismos (auto-vistas) para así poder predecir las respuestas y saber cómo actuar ante ellos. De esta forma por ejemplo las personas que se ven a sí mismas como inteligentes esperan que las demás personas noten su inteligencia. Esto lleva a que hagan actividades que requieran el uso de su inteligencia. Esto es debido a que sus propias vistas de sí mismos son las que los guían en sus acciones y hacen que representen sus cualidades. Existen estudios acerca de matrimonios o compañeros de habitación en los cuales se demuestra que las personas prefieren compañeros con auto-verificación (incluso si es negativa) y de esta forma recibiendo auto-verificación esto promueve la intimidad, la satisfacción y el compromiso en la relación. (por más detalles se puede ver Swann, Rentfrow, Guinn, 2002) («Self Verification Theory») En conclusión, las personas están en una constante necesidad de auto confirmación por parte de sus pares.

Esta teoría es precisamente opuesta a la teoría de la auto-categorización, esto se debe a la ya mencionada necesidad de los miembros de un grupo de externalizar sus propias vistas de sí mismos.

Esta teoría nos dice que los grupos tienden a funcionar mejor cuando los participantes cumplen su deseo de auto-verificación. A continuación hablaremos de 3 razones por las cuales la auto-verificación mejora el funcionamiento del grupo. Primero, tener sensaciones compartidas por el grupo hace que los miembros se sientan más conectados al grupo y más motivados permitiendo así que se involucren fuertemente con las actividades grupales. En segundo lugar, en la medida que los integrantes del grupo se sientan pertenecientes a un grupo de personas que cumplan con auto-verificación entonces esto hace que se puedan expresar más auténticamente. Esto hace que las personas puedan llegar a un conjunto de soluciones potencialmente mejor a partir de una combinación fuertemente creativa de ideas. En tercer lugar, el intento de provocar la auto-verificación de otros miembros del grupo conlleva a la explotación de recursos cognitivos. Entendemos por recursos cognitivos al cúmulo de información que se dispone gracias a un proceso de aprendizaje o a la experiencia. Aquellas personas que cumplan con sus expectativas de auto verificación por parte de otros miembros del grupo estarán dispuestas a favorecer en los resultados globales del grupo. Por todas estas razones, los procesos de auto-verificación nos pueden llevar a una mejor performance grupal especialmente en lo que respecta al desarrollo de ideas creativas.

2.5.3. Discusión planteada y conclusiones

La pregunta que nos interesa encontrar es: ¿cuál es la mejor estrategia para obtener buenos resultados a partir de la diversidad? Según la teoría de la auto-categorización, se sugiere que la clave para este asunto es formar grupos en los cuales sus miembros estén alineados con los objetivos del superordenado del grupo. A partir de que se armen grupos donde sus integrantes estén alineados, vamos a tener un proceso de des-personalización menor el cual permite que no exista una predominancia por parte de un integrante particular. Esta disminución de la predominancia de algunas personas permite que todos los integrantes se desenvuelvan más naturalmente en los trabajos grupales, alcanzando así una mejora de la performance de todo el grupo.

También enfocándonos en otras teorías como la de la auto-verificación podemos ver que existen otros caminos para encontrar valor en la diversidad. Particularmente, se puede notar que los integrantes de un grupo de estudio muestran mejor creatividad cuando los otros integrantes del grupo les proporcionan una verificación de auto-vistas. De esta forma podemos sugerir que la adopción de identidades únicas en la selección de personas en lugar de disminuir identidades únicas en un grupo, es la mejor forma de maximizar la productividad en grupos diversos.

En conclusión, podemos ver que existen otros factores para tener en cuenta al momento de definir la diversidad, éstos factores van a tener en cuenta los procesos psicológicos que inciden en las actitudes personales dentro del grupo. Mediante este punto de vista nos enfocamos en la importancia que tienen los procesos de interacción entre las personas del grupo.

Capítulo 3

Definición y Modelado del Problema

3.1. Atributos

Como mencionamos en el capítulo anterior, resulta de gran importancia contar con una definición propia de diversidad la cual estará estrechamente relacionada a los atributos que nuestro escenario contiene y a la correcta elección de los mismos.

Estos atributos son los que nos permitirán establecer las características de un estudiante y poder medir (en base a una medida de distancia) qué tan distintos son 2 estudiantes entre sí.

Para eso procederemos a categorizar los atributos encontrados y luego hacer una selección de las que serán utilizadas.

3.1.1. Categorización de atributos

A continuación se detallan todos los atributos identificados para la representación de un estudiante. La lista de atributos encontrados se mostrarán según las 4 categorías descritas en el capítulo anterior y luego se subdividirán en el tipo de variable a la que pertenece. Los tipos de variables son:

- Categóricas: también llamadas cualitativas, son aquellas que se pueden desglosar en un número contable de categorías o grupos diferentes.
- Binarias: solamente puede adoptar los valores 0 ó 1.
- Cuantitativas: aquellas que se expresan mediante un número, varían en su magnitud.
- Ordinales: son las que aceptan una jerarquización de importancia.

Atributos relacionados a experiencias laborales y académicas:

Categóricas:

- Universidad en el que se obtuvo el título (UdelaR/UM/ORT/UDE/etc.)
- Título universitario (Ing. Computación/Contador/Economista/etc.)

- Área de trabajo actual (it/rrhh/marketing/ventas/etc.)
- Nivel de jerarquía en el trabajo (administrativo/gerente/gg/etc.)

Binarias:

- ¿Tiene otros títulos obtenidos de importancia? (sí/no)

Cuantitativas:

- Años desde recibimiento
- Años de carrera
- Cantidad de trabajos previos
- Cantidad de años de trabajo pre-título
- Cantidad de años de trabajo totales

Atributos relacionados a notas académicas:

Ordinales:

- Puntaje en test de admisión (parte cuant.)
- Puntaje en test de admisión (total)

Atributos relacionados a la posición social:

Categóricas:

- Hobbies (Deporte/Viajes/etc.)
- Estado Civil (Soltero/Casado/Divorciado/etc.)

Binarias:

- Sexo
- Lugar de nacimiento (Montevideo/Interior)
- Lugar de residencia (Montevideo/Interior)

Cuantitativas:

- Cantidad de hijos
- Edad

Atributos personales:

Categóricas:

- Capacidad de liderazgo (Buena, Media, Mala)
- Rapidez de aprendizaje (Buena, Media, Mala)
- Trabajo grupal (Bueno, Medio, Malo)

- Creatividad (Bueno, Medio, Malo)
- Ingenio (Bueno, Medio, Malo)

Binarias:

- Extrovertido o introvertido

3.1.2. Selección de atributos

Como planteo inicial y para la realización de nuestro proyecto vamos a considerar un conjunto acotado de atributos que se definen como relevantes a la hora de encontrar una buena solución para nuestro problema. Tener identificado un conjunto acotado pero potente de atributos nos permite que las soluciones que encontremos tengan una buena diversidad. Partimos de la idea de que la solución encontrada en base a un número reducido de atributos representativos va a ser de más valor que una solución en la cual se modele con un número excesivo de atributos. De esta manera también se logra mejorar la performance de ejecución y reducir la cantidad de datos de entrada requeridos para la herramienta desarrollada, teniendo en cuenta que tal vez no se cuente con una extensa información de los estudiantes, o en el caso de tener que recabar la información esta sea la mínima posible. Además el análisis de resultados será mas legible y entendible de este modo.

Las variables que son de tipo categórica las sub-dividimos en variables contables para así lograr una estandarización de todas las variables y poder utilizar la función de distancia correctamente.

El subconjunto de atributos elegidos como los más representativos y los que nos llevarán a mejores soluciones son los siguientes:

- Carrera cursada (Se subdivide en ciencias sociales, naturales y exactas)
- Puntaje en el test de admisión
- Lugar de residencia (Montevideo o Interior)
- Sexo (Masculino o Femenino)
- Edad

Respecto al atributo de carrera cursada cabe destacar que es representado mediante 3 atributos contables que definen la carrera. Estos nuevos atributos representan la carrera en base a cuanta carga de ciencias sociales, ciencias naturales y ciencias exactas contiene, el puntaje es otrogado de 0 a 100. Esta representación nos permite evaluar una carrera de la misma forma que los otros atributos y así poder usar la misma función de distancia. Dada la realidad de nuestro problema notamos que tenemos una cierta predominancia de Contadores e Ingenieros por lo tanto estas carreras son las que tenemos que repartir más equitativamente en los equipos. Esto se logra haciendo un buen desglose mediante los atributos mencionados que representan estas dos carreras.

3.2. Medidas de distancia

Para poder establecer cuánta diferencia (o similitud) hay entre dos personas será necesario contar con medidas de distancias. Existen varios métodos matemáticos para definir la distancia entre dos puntos. Algunas de las métricas más conocidas como la distancia euclidiana se define como una métrica que obedece a tres axiomas. Estos axiomas son los siguientes, (d_{ab} denota la distancia entre los puntos a y b):

1. $d_{ab} = d_{ba}$
2. $d_{ab} \geq 0$ con $d_{ab} = 0$ si y sólo si $a = b$
3. $d_{ab} \leq d_{ac} + d_{cb}$

3.2.1. Distancia Euclidiana

En dos dimensiones, la distancia entre el vector $x = (x_1, x_2)$ e $y = (y_1, y_2)$ se calcula como la distancia entre dos puntos en un plano.

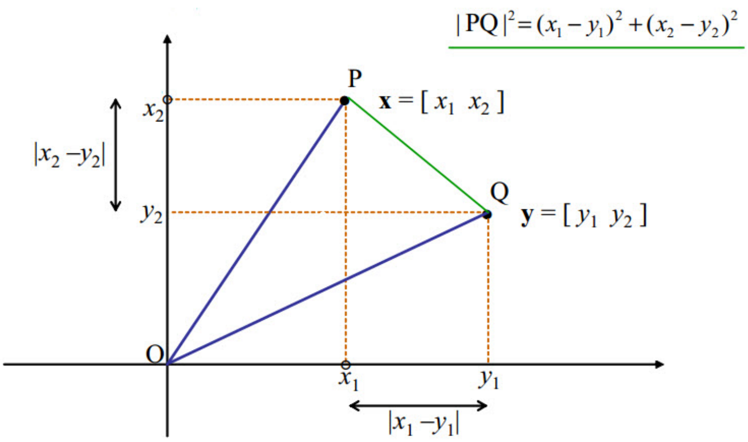

FIGURA 3.1: **Teorema de Pitágoras aplicado a 2 dimensiones**

Como se explica en la figura, usando el teorema de Pitágoras tenemos que la distancia entre dos vectores en un plano de 2 dimensiones es:

$$d_{x,y}{}^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 \tag{3.1}$$

O lo que es lo mismo:

$$d_{x,y} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{3.2}$$

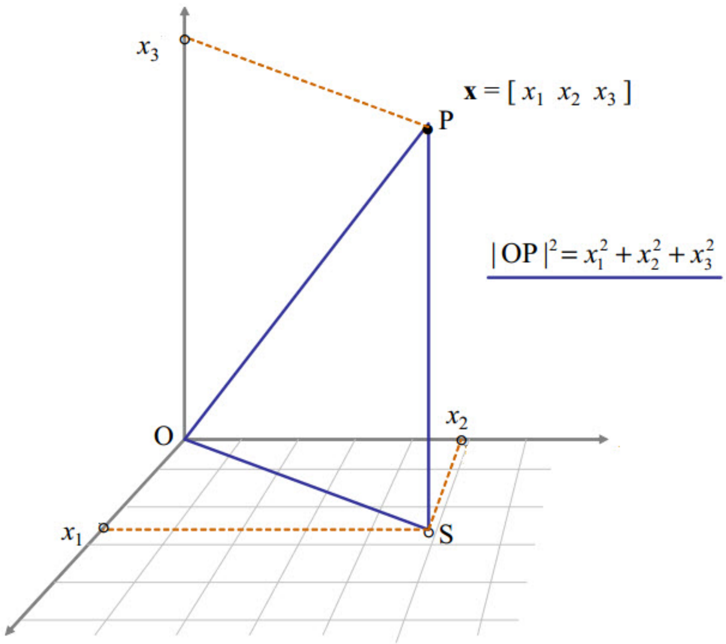

FIGURA 3.2: **Teorema de Pitágoras extendido para 3 dimensiones**

Luego para 3 dimensiones (como se ve en la figura) basta con extender la ecuación anterior y se obtiene la distancia para los vectores $x = (x_1, x_2, x_3)$ e $y = (y_1, y_2, y_3)$:

$$d_{x,y} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \tag{3.3}$$

Finalmente generalizando para J dimensiones se obtiene la definición de la distancia Euclidiana se escribe de la siguiente manera:

$$d_{x,y} = \sqrt{\sum_{j=1}^{J} (x_j - y_j)^2} \tag{3.4}$$

3.2.2. Distancia Euclidiana estandarizada

Supongamos el caso de variables con tres atributos en donde el primer atributo adquiere valores en el rango de 0 a 10, el segundo en el rango de 50 a 100 y el tercero en el rango de 2 a 3. Al aplicar la distancia Euclidiana a dos vectores dado, el segundo atributo será de mayor influencia para el resultado que el primer y tercer atributo.

Por ejemplo sea $x = [6{,}0; 51; 3{,}0]$ e $y = [1{,}9; 99; 2{,}9]$ entonces la distancia Euclidiana se calcula como:

$d_{x,y} = \sqrt{(6{,}0 - 1{,}9)^2 + (51 - 99)^2 + (3{,}0 - 2{,}9)^2}$

$d_{x,y} = \sqrt{16{,}81 + 2304 + 0{,}01} = \sqrt{2320{,}82} = 48{,}17$

Se puede ver cómo el valor del segundo atributo tiene la mayor influencia sobre los demás atributos en la generación del resultado. Lo que ocurre es que las tres variables están en escalas de valores diferentes, entonces la segunda variable domina en el cálculo de la distancia Euclidiana.

Es necesario balancear los atributos y para esto necesitaremos una estandarización. Comúnmente lo que se hace es llevar a todos los atributos a poseer una varianza de 1. La estandarización se define de la siguiente manera:

valor estandarizado = (valor original - valor promedio) / desviación estandar

Entonces la desviación estándar Euclidiana para la distancia entre dos vectores J-dimensionales puede ser escrita como:

$$d_{x,y} = \sqrt{\sum_{j=1}^{J} (\frac{x_j}{s_j} - \frac{y_j}{s_j})^2} \tag{3.5}$$

Donde s_j es la desviación estándar para la j-ésima variable.

3.2.3. Distancia Euclidiana ponderada

Reescribiendo la función de distancia Euclidiana estandarizada se tiene que:

$$d_{x,y} = \sqrt{\sum_{j=1}^{J} \frac{1}{s_j}(x_j - y_j)^2} \tag{3.6}$$

$$d_{x,y} = \sqrt{\sum_{j=1}^{J} w_j(x_j - y_j)^2} \tag{3.7}$$

Donde $w_j = \frac{1}{s_j}$ es el peso asociado a la variable j. Lo que se hace es multiplicar el peso de la variable por el cálculo usual de la distancia de Euclides para cada variable. De esta manera una variable cuya varianza sea menor tendrá un peso mayor sobre las variables de varianza mayor. Además de este peso inducido por la estandarización, es posible agregar mayor o menor peso a las variables de la forma que se desee.

3.2.4. Distancia para datos categóricos

En el caso que tengamos variables categóricas (las cuales son variables en las que sabemos qué valores pueden tomar) podemos definir la distancia de la siguiente forma. Definimos cinco variables como C1, C2, C3, C4, C5 y $a/b/c$ los valores que estas variables categóricas pueden tomar:

	C1	C2	C3	C4	C5
Ejemplo 1	a	c	c	b	a
Ejemplo 2	b	c	b	a	a

Entonces para este escenario el número de coincidencias entre los ejemples es 2 (coinciden en C2 y C5) y la cantidad de diferencias es 3 (difieren en C1, C3 y C4), por lo tanto la distancia entre estos dos ejemplos es $\frac{3}{5} = 0,6$ o $\frac{2}{5} = 0,4$ respecto a si la medimos en base a la comparación de disimilitud o similitud. Notamos que estos coeficientes son directamente proporcionales a si aplicáramos la medida de distancia Euclidiana calculada en base a transformar estas variables categóricas a variables binarias. La tabla entonces nos queda de la siguiente manera:

	C1a	C1b	C1c	C2a	C2b	C2c	C3a	C3b
Ejemplo 1	1	0	0	0	0	1	0	0
Ejemplo 2	0	1	0	0	0	1	0	1

	C3c	C4a	C4b	C4c	C5a	C5b	C5c
Ejemplo 1	1	0	1	0	1	0	0
Ejemplo 2	0	1	0	0	1	0	0

Transformaremos entonces los datos categóricos en variables binarias para luego realizar el cálculo de distancia como lo hacemos habitualmente.

3.2.5. Distancia de Mahalanobis

Existen muchas otras formas de calcular la distancia entre dos elementos. La distancia de Mahalanobis es otra forma de determinar la similitud entre dos variables aleatorias multidimensionales. Cumple con los tres axiomas numerados anteriormente y se define como:

$$d_m(\overrightarrow{x}, \overrightarrow{y}) = \sqrt{(\overrightarrow{x} - \overrightarrow{y})^T \Sigma^{-1} (\overrightarrow{x} - \overrightarrow{y})} \tag{3.8}$$

Donde Σ, es la matriz de covarianza. La distancia de Mahalanobis incorpora la correlación entre los atributos de las dos variables involucradas.

Se observa que si la matriz de covarianza es la matriz identidad la distancia de Mahalanobis se puede traducir a la función de distancia de Euclides. Si la matriz de covarianza es diagonal se traduce a la función de distancia de Euclides ponderada. La diferencia con la distancia Euclidiana radica en que tiene en cuenta la correlación entre las variables aleatorias, es decir, incorpora la dependencia entre las dos variables medidas.

3.2.6. Elección de una medida de distancia

Para calcular la disimilitud entre un par de estudiantes consideramos conveniente el uso de la distancia Euclidiana ponderada. A los efectos del alcance de este proyecto resulta una forma de medida conveniente dado que, además de estandarizar los distintos atributos los cuales tiene valores en rangos distintos impidiendo que un atributo sea determinante, se puede fácilmente extender para agregar una ponderación propia a los atributos que se consideren necesarios.

3.3. Modelado del problema

A continuación se presentan y analizan dos modelos matemáticos de programación entera. Se analizarán y compararán para luego elegir una de ellas.

3.3.1. Modelo 1

Sea:

- N la cantidad de estudiantes
- G la cantidad de grupos
- K la cantidad de atributos
- M la cantidad de estudiantes en cada grupo donde $M = \frac{N}{G}$ (entero)
- d_{ij} la distancia entre el estudiante i y el estudiante j
- $A = (a_i) = (a_{ik})$ es el atributo k para el estudiante i
- $\overrightarrow{w} = (w_k)$ guarda el peso del atributo k.
- $\overrightarrow{b} = (b_k)$ guarda el valor objetivo del atributo k. Se considera que este es un valor calculado por alguna regla específica. Este puede ser el valor promedio sobre un atributo tomando todo el conjunto de estudiantes
- $|C_j| = M$ donde C_j es el conjunto de estudiantes del grupo j
- $v_j = \frac{1}{m} \sum_{i \in C_j} a_i$ es el centroide del grupo j. Este vector tiene los valores promedios de todos los atributos dentro del conjunto de estudiantes de grupo j

Esta solución propone encontrar una partición de los estudiantes de un grupo (C_j) en la cual la distancia entre el valor del vector centroide de la clase entera (valor promedio) y el valor del vector centroide de los grupos sean lo menor posible. Esto se hace porque asumimos que el vector $\overrightarrow{v}$ es el promedio y si logramos que todos los grupos estén cercanos de él significa que los grupos están cerca de la diversidad máxima posible.

Este problema se puede formular de dos formas distintas:

- Una como un problema de *min-sum* lo que significa minimizar la sumatoria de todas las distancias de los centroides de cada grupo respecto al valor promedio. Se formaliza en la siguiente función:

  $min \sum_{1 \leq g \leq G} distancia(b, v_g)$

- Otra como un problema de *min-max* lo que significa tratar de minimizar la máxima distancia que alcanza el centroide de cada grupo al vector promedio:

  $minmax_{1 \leq g \leq G} distancia(b, v_g)$

Sea x_{ig} un valor binario el cual toma el valor 1 si el estudiante i es asignado al grupo g y 0 sino. Además sea $v_{gk}(x)$ quien representa el valor centroide para el atributo k en el grupo g el cual se calcula como $v_{gk}(x) = \frac{1}{m}\sum_{i=1}^{N} a_{ig}x_{ig}$

Entonces llegamos a que las dos soluciones son de la forma:

$$\begin{aligned}
&1) \text{ min } \sum_{1\leq g\leq G} distancia(b, v_g(x))\\
&2) \text{ min } max_{1\leq g\leq G} distancia(b, v_g(x))\\
&\quad \text{s.a.} \sum_{g=1}^{G} x_{ig} = 1 && , i = 1..N\\
&\quad\quad \sum_{i=1}^{N} x_{ig} = M && , g = 1..G\\
&\quad\quad x_{ig} \in \{0,1\} && , i = 1..N, g = 1..G
\end{aligned}$$

3.3.2. Análisis del modelo 1

Para la solución planteada como un problema de minimizar la sumatoria (*min-sum*) se puede dar el caso en que una solución tenga un equipo formado con un estudiante muy alejado del centroide y todos los otros muy cercanos a él haciendo así que se considere igualmente una buena solución. Con esta solución también se puede dar el caso que existan centroides de grupos los cuales están muy alejados del valor promedio global del problema. Dado el segundo enfoque en el cual se plantea el problema de minimizar la máxima (*min-max*) distancia del centroide de grupo al centroide global, entonces ya no tenemos el problema de que un equipo tenga el centroide muy alejado del valor global o promedio. Esto hace que podamos tener grupos más parejos entre sí y no exista un grupo el cual sea muy diferente al resto, lo cual parece mas deseable en nuestro caso.

3.3.3. Modelo 2

El objetivo es centrarse en formar particiones de estudiantes en G grupos disjuntos haciendo que la suma de la diversidad entre los estudiantes dentro de cada grupo sea maximizada.

Nuevamente tomamos x_{ig} como valor binario el cual toma el valor 1 si el estudiante i es asignado al grupo g y 0 sino, y usamos las variables definidas para el modelo 1.

Este modelo se representa como un problema de programación entera cuadrático binario:

$$
\begin{aligned}
\max & \sum_{g=1}^{G} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} x_{ig} x_{jg} & \\
\text{s.a.} & \sum_{g=1}^{G} x_{ig} = 1 & , i = 1..N \\
& \sum_{i=1}^{N} x_{ig} = M & , g = 1..G \\
& x_{ig} \in \{0,1\} & , i = 1..N, g = 1..G
\end{aligned}
$$

3.3.4. Elección del modelo

Se considera apropiado no contar con un modelo que permita la generación de soluciones en las que pueda no haber un equilibrio total entre los estudiantes de un grupo (o entre los grupos de la solución) como sucede en la propuesta de centroides del *min-sum* del modelo 1.

Se decide entonces optar por el modelo de la solución 2, el cual no contiene un enfoque de centroide sino un enfoque de concentrar la diversidad en cada grupo lo cual, por la naturaleza de su propuesta, tenderá a generar un equilibrio en la diversidad de todos sus grupos.

El problema es conocido con un problema NP-difícil por lo tanto se busca una heurística para llegar a una posible solución. Se propone usar VNS (Variable Neighborhood Search) para la resolución del problema planteado pero lo veremos en detalle en el próximo capítulo.

3.4. Restricciones

De nuestro problema se desprenden dos restricciones fundamentales.

La primera obedece a la realidad que establece que la cantidad de estudiantes no siempre es múltiplo de la cantidad de integrantes de los equipos, es decir, puede ocurrir que dado una cantidad de grupos a formar no sea posible hacer que todos los equipos tengan la misma cantidad de estudiantes, sino que probablemente se deba armar algún equipo con algún integrante más o menos, dejando equipos con distinta cantidad de integrantes.

La segunda restricción encontrada parte del hecho de que durante el transcurso de la maestría los estudiantes no conforman solamente un grupo sino tantos grupos como trimestres tenga el curso y es de interés que los estudiantes no repitan compañeros en ninguno de los equipos que conforme durante los trimestres. La idea de esta restricción es potenciar la experiencia de cada estudiante durante el curso, se pretende que todos conozcan diferentes personas y características de personas.

Esta restricción también estará sujeta a la cantidad de estudiantes, trimestres y grupos dado que en ciertos casos dicha restricción es imposible de cumplir teniendo que algunos de los estudiantes ineludiblemente repetir a algún compañero.

Es de esperar que las restricciones impuestas (en especial la segunda) afectarán en cierta medida la diversidad de los grupos en los resultados obtenidos.

Otras restricciones posibles, pero que no serán contempladas en este proyecto son por ejemplo el definir un mínimo de personas con un cierto atributo por grupo (ejemplo: tener dos mujeres mínimo por grupo) o que ciertos estudiantes con ciertas características no puedan pertenecer al mismo grupo, etc..

3.4.1. Solución al problema de grupos con distinta cantidad de estudiantes

Para contemplar el problema de posiblemente tener que repartir los estudiantes de forma tal que queden grupos con distinta cantidad de estudiantes se extiende el modelo elegido al siguiente:

$$
\begin{aligned}
\max \ & \sum_{g=1}^{G} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} x_{ig} x_{jg} & \\
\text{s.a.} \ & \sum_{g=1}^{G} x_{ig} = 1 & , i = 1..N \\
& \sum_{i=1}^{N} x_{ig} \geq a_g & , g = 1..G \\
& \sum_{i=1}^{N} x_{ig} \leq b_g & , g = 1..G \\
& x_{ig} \in \{0, 1\} & , i = 1..N, g = 1..G
\end{aligned}
$$

Analizando las restricciones del problema, vemos que la primera impone que todos los estudiantes deben pertenecer siempre a un grupo.

La segunda introduce el vector $\overrightarrow{a} = (a_g)$ el cual asigna un valor mínimo de estudiantes para el grupo g y entonces la restricción dice que la cantidad de estudiantes pertenecientes a cierto grupo debe ser mayor o igual a un valor mínimo definido para ese grupo.

De forma similar, la tercera restricción introduce el vector $\overrightarrow{b} = (b_g)$ el cual asigna un valor máximo de estudiantes para el grupo g y dice que la cantidad de estudiantes pertenecientes a cierto grupo debe ser menor o igual a un valor máximo definido para ese grupo.

De esta forma el modelo permite que existan grupos de diferente tamaño y además los mismos son predefinidos lo cual crea flexibilidad de adaptación a los casos reales.

3.4.2. Solución al problema de no repetir compañeros

Dado una cantidad N de estudiantes, una cantidad G de grupos y una cantidad S de trimestres, podemos hacer el cálculo de cuántos compañeros necesitará repetir un estudiante en un trimestre dado y así saber si al final de los trimestres habrá necesidad de generar repeticiones entre compañeros alguna vez.

Dado $M = \frac{N}{G}$ estudiantes por grupo y w_s la cantidad de compañeros disponibles para un estudiante cualquiera en el trimestre s, tenemos inicialmente que para el trimestre 1 $w_1 = N - 1$ y entones en el trimestre 2:

$$w_2 = w_1 - (M - 1) \tag{3.9}$$

Luego generalizando para el trimestre S:

$$w_S = w_{S-1} - (M - 1) \tag{3.10}$$

Entonces:

$$w_S = (N - 1) - (S - 1)(M - 1) \tag{3.11}$$

Analizando los resultados sabremos que:

- Si $w_S = 0$: en el trimestre h el estudiante habrá formado grupo con todo el resto de los estudiantes.
- Si $w_S > 0$ el estudiante no habrá formado grupo con w_S estudiantes del curso.
- Si $w_S < 0$ en el trimestre h el estudiante debe repetir grupo con al menos w_S estudiantes.

El planteo supone que los grupos siempre serán conformados por un número fijo y entero de estudiantes, en el caso de que los grupos puedan contener distintas cantidades de alumnos se pueden obtener los siguientes datos.

Sea M_{min} el grupo con menor cantidad de estudiantes posible y M_{max} el grupo con mayor cantidad de estudiantes posible, se puede obtener un mejor y peor caso para w_S:

$$w_{sMC} = (N - 1) - (S - 1)(M_{min} - 1) \tag{3.12}$$

$$w_{sPC} = (N - 1) - (S - 1)(M_{max} - 1) \tag{3.13}$$

Podemos entonces afirmar que:

- Si w_{sMC} y $w_{sPC} \geq 0$ entonces seguro no habrán estudiantes que repitan compañero.

- Si w_{sMC} y $w_{sPC} < 0$ entonces seguro que habrán estudiantes que repitan compañero.
- Si $w_{sMC} \geq 0$ y $w_{sPC} \leq 0$ entonces puede ser que un estudiante repita compañero o no.

Para el caso en que sea posible generar grupos por trimestres sin repetición de compañeros se está ante un problema de diseño combinatorio el cual fue estudiado y formulado en primera instancia por Thomas Kirkman y conocido como el *Kirkman's Schoolgirl Problem*, problema que luego fue generalizado y que se conoce como el *Social Golfer Problem* (SGP) (Triska, 2008).

El SGP en sí es un problema NP-difícil que sumada nuestra gran cantidad de parámetros se vuelve un problema cuya resolución es únicamente a través de la implementación de heurísiticas. Dicha heurística deberá en su trabajo, además de encontrar los posibles conjuntos de grupos para los trimestres, eliminar soluciones simétricas, como el caso de intercambiar grupos en un trimestre, intercambiar trimestres, etc., lo cual lo hace incluso un problema más complejo. Para nuestro trabajo optaremos por la utilización de la técnica de GRASP/VND (RESENDE y RIBEIRO, 0), (Pier Hansen, 1999) cuya aplicación en problemas similares muestra un buen desempeño en la obtención de soluciones de calidad factibles y en tiempos aceptables.

Se podría optar por contar con estos resultados para nuestra heurística. Haciendo uso de los datos pre-calculados sabríamos cuáles son las posibilidades de armados de grupos para todos los trimestres, pero esto tiene dos inconvenientes, por un lado la formación de grupos diversos quedaría fuertemente restringida a las posiblididades de armados de trimestres, por lo que la diversidad se verá altamente acotada en base a las posibilidades que se tienen, y por otro lado el desarrollo de la solución en sí es de gran porte lo cual no entra en las posiblidades del alcance del proyecto, y en el caso de optar por encontrar una solución hecha por alguien más correríamos con la incertidumbre de la calidad y correctitud de las soluciónes que el programa nos otorgue.

Entonces se plantea incorporar el uso del método *Tabu Search* a la heurística que implementaremos y que veremos en el próximo capítulo. La misma implementará un registro histórico que se denotará como *tabu list*, utilizando una matriz de tamaño $N * N$, que establecerá para cierta pareja la exclusión temporaria de armar grupo juntos o no. Cuando la heurística se estanque en la mejora de la solución entonces se evaluará la opción de liberar parejas de la matriz para que puedan formar grupo. De esta manera se irán introduciendo repeticiones paulatina y controladamente a la solución siempre y cuando las mejoras se hayan estancado y un coeficiente previamente definido lo establezcan. El cálculo de esta matriz se realizaría previo al cálculo de la formación de grupos de cada trimestre a partir de las repeticiones entre los compañeros de los trimestres anteriores.

Por la naturaleza de esta solución, es de esperar que nuestro modelo llegue a encontrar las mejores soluciones en su primera iteración y se irá degradando en cada ejecución. Es de suponer que el primer conjunto de equipos contendrá grupos con mayor diversidad que los últimos debido a que para las últimas iteraciones los conjuntos mas diversos ya habrán sido armados y por lo tanto los nuevos conjuntos estarán acotados por el armado de los grupos en los anteriores trimestres (debido a la exclusión temporaria producto de haber formado grupo anteriormente).

Sin embargo en este estudio consideramos que el enfoque planteado satisface las necesidades y se adecua a la realidad planteada dado que los primeros trimestres (en particular el primero) es el más importante de la carrera porque en estos trimestres se dictan los cursos con más trabajos de equipo y mas enriquecedores en la experiencia inter-disciplinaria. Los últimos trimestres es de esperar que los participantes se nivelen por lo tanto no es tan necesario tener una gran diversidad.

Capítulo 4

Descripción e implementación de la solución

4.1. Descripción de la heurística

4.1.1. GRASP - Variable Neighborhood Search (VNS)

La Búsqueda por Entornos Variables (Variable Neighbourhood Search, VNS) es una metaheurística para resolver problemas de optimización combinatoria y problemas globales de optimización. La idea de esta metaheurística consiste en el cambio sistemático de entorno dentro de una búsqueda local. Las Metaheurísticas son estrategias generales para diseñar procedimientos heurísticos para resolver un problema de optimización mediante un proceso de búsqueda en un cierto espacio de soluciones alternativas. Estas técnicas son capaces de explorar soluciones distantes. Cuanto más distantes sean los entornos se tiene mayor probabilidad de acercarse al óptimo global.

Los procesos de búsqueda están generalmente basados en transformaciones de las alternativas que determinan una estructura de entornos en el espacio de soluciones. Los métodos de búsquedas locales son aplicados reiteradamente quedándose en cada iteración con la solución en caso de que exista una mejora. Una solución factible x es un mínimo global del problema si no existe una solución x' tal que $f(x') < f(x)$.

Una búsqueda local descendente si obtiene una mejor solución que la que tenemos hasta el momento cambia la solución actual por la nueva solución encontrada para el entorno de búsqueda aplicado, por lo tanto corren el riesgo de quedarse atascada en un mínimo local que no sea óptimo global. Las metaheurísticas basadas en procedimientos de búsqueda local aplican distintas formas de continuar la búsqueda después de encontrar el primer óptimo local.

La metaheurística VNS se basa en aprovechar sistemáticamente tres hechos simples:

1. Un mínimo local con una estructura de entornos no lo es necesariamente con otra.
2. Un mínimo global es mínimo local con todas las posibles estructuras de entornos.

3. Para muchos problemas, los mínimos locales están relativamente próximos entre sí.

4.1.2. VNS Descendente

Una búsqueda local descendente de forma iterativa busca encontrar mejores soluciones a partir de la solución actual encontrada hasta el momento. Para encontrar estas nuevas soluciones se generan ciertas transformaciones o movimientos sobre la solución actual. La búsqueda descendente greedy consiste en reemplazar siempre la solución actual por la mejor de todas las soluciones que se pueden obtener a partir de la actual mediante transformaciones o los movimientos que se aplique. Cualquier estrategia es posible aplicarse para encontrar soluciones del entorno que mejoren la solución actual, pero todas ellas deben detenerse cuando no sea posible encontrar una mejora. La elección de los movimientos a considerar puede ser determinante en el éxito de la búsqueda local, pero difícilmente se puede determinar a priori cuál de ellas va a ser la más efectiva. Dado estas que son estrategias poco inteligentes, es conveniente combinar los distintos tipos de movimientos. En nuestra solución utilizamos movimientos de insertar (insert) un estudiante en otro grupo, hacer un intercambio (swap) de 2 estudiantes que estén en distinto grupo o hacer un intercambio entre 3 estudiantes (3-chain) que pertenezcan a 3 equipos distintos.

En la búsqueda local descendente que desarrollamos se hace un cambio de estructura de entorno cada vez que se llega a un mínimo local. Inicialmente se hacen todos los posibles *swap*, cuando el *swap* no encuentra mejora se pasan a hacer todos los *insertions* posibles. Si en alguno de los *insertions* aplicados se encuentra una mejora a la solución se vuelve a empezar la búsqueda local mediante los *swaps*.

Por tanto, la solución final proporcionada por el algoritmo es un mínimo local con respecto a cada una de las k estructuras de entornos elegidas anteriormente. Como consecuencia de ellos, la probabilidad de alcanzar un mínimo global es mayor que usando una sola estructura.

4.2. Implementación de la heurística

4.2.1. Espacio de soluciones y función objetivo

El espacio de soluciones es el conjunto de combinaciones de estudiantes en grupos siempre y cuando un estudiante pertenezca a solamente a un grupo. Una solución factible será aquella que cada grupo creado contenga un mínimo de a_g personas y un máximo de b_g personas. Representaremos a una solución actual mediante el vector $x^c[i]$ donde este valor muestra el número del grupo que integra el estudiante i.

Para mejorar la performance de los cálculos se tiene la matriz sd^c donde $sd^c[i][g]$ representa la suma de las diversidades entre la persona i y el resto de los integrantes de su equipo g. Este valor nos sirve conceptualmente para saber qué tan distinto es una persona al resto del equipo. Dado que a la hora de evaluar un cambio vamos a necesitar saber este valor y la gran cantidad de repeticiones de esta operación es que se mejora la performance teniendo estos valores ya calculados de la siguiente manera:

$$sd^c[i][g] = \sum_{j=1..N, x^c[j]=g} d_{ij} \tag{4.1}$$

Teniendo definido el vector $x^c[i]$ y la matriz sd^c podemos re-formular la función objetivo como:

$$f^c = f(x^c) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} * \chi(x^c, i, j) \tag{4.2}$$

Donde:

$$\chi(x^c, i, j) = \begin{cases} 1 & \text{si } x^c[i] = x^c[j] \\ 0 & \text{si } x^c[i] \neq x^c[j] \end{cases} \tag{4.3}$$

Si a esta función la queremos expresar utilizando la matriz sd^c llegamos a:

$$f(x^c) = \frac{1}{2} \sum_{i=1}^{N-1} sd^c[i][x^c[i]] \tag{4.4}$$

De esta manera, en el lugar de la matriz que poníamos g, ahora usamos el valor guardado en el vector x^c el cual sabemos que es el grupo del estudiante i.

Como la primera formulación ya tenía en cuenta no sumar la distancia de i a j más la de j a i entonces en este caso dividimos entre 2. Dado que en la matriz sd^c, si cuenta la distancia de i al resto del grupo y después cuenta la distancia de j al resto del grupo, entonces estaría sumando las distancia del i al j más la del j al i lo cual no era buscado.

4.2.2. Solución inicial

Para la construcción de la solución inicial se desarrolló un algoritmo en el cual primero hacemos que todos los grupos tengan la mínima cantidad de estudiantes requerida, se obtienen estudiantes aleatoriamente y se los asigna a un grupo de acuerdo a ciertos criterios que vamos a definir posteriormente. Este proceso iterativo termina cuando todos los grupos tienen la mínima cantidad de estudiantes requerida.

La segunda etapa para formar la solución inicial consiste en completar los grupos con los estudiantes que aún no fueron asignados, de tal forma que todos los grupos tengan a lo sumo la cantidad máxima definida. Este también es un proceso iterativo el cual consiste en obtener un estudiante aleatoriamente y asignarlo al grupo.

Para definir en qué grupo asigno un estudiante elegido aleatoriamente se tienen en cuenta dos factores, la cantidad de repeticiones que genera y cuanta diversidad aporta a la solución. En esta solución priorizamos minimizar las repeticiones por lo tanto siempre optamos por ingresar un estudiante al grupo que genere menos repeticiones, en caso que tengamos un empate se hace el desempate basándonos en la diversidad. Para medir la diversidad se utiliza la función de distancia ya definida previamente. Esta función se aplica para calcular la sumatoria de distancias entre el estudiante elegido y el resto de los estudiantes del grupo y seleccionado.

Definimos la distancia de un estudiante x a un grupo g como la sumatoria de las distancias a cada uno de los estudiantes del grupo g al estudiante x dividido la cantidad de estudiantes del grupo. Es necesario hacer el valor promedio o sea dividir por la cantidad de estudiantes dado que sino los grupos con mayor cantidad de estudiantes son los que tienden a tener la mayor diversidad a la hora de seleccionar un grupo.

Cabe destacar que para contemplar la necesidad de minimizar las repeticiones es necesario tener almacenada la información de cuántas repeticiones existieron en los trimestres previos y así poder hacer el cálculo.

Dada la importante necesidad de que las soluciones minimicen las repeticiones se desarrolló un algoritmo de solución inicial el cual ejecuta una cierta cantidad configurable de soluciones iniciales y se queda con la que tenga menos repeticiones. Lograr buenas soluciones iniciales es un factor muy importante para las posteriores ejecuciones de la heurística GRASP/VNS.

Sea:

- $studentGroup[s]$: el número de grupo ocupado por el estudiante s (1..N)
- $groupMax[g]$: la cantidad máxima de estudiantes que puede contener el grupo g (1..G)
- $groupMin[g]$: la cantidad mínima de estudiantes que puede contener el grupo g (1..G)
- $atrsStandard[i, j]$: el valor estandarizado del atributo j (1..K) para el estudiante i (1..N)
- $groupCount[g]$: la cantidad de estudiantes en el grupo g (1..G)

Donde:

- $assignOneRandomStudentToEachGroup()$: es una función que asigna un cada grupo un estudiante de forma aleatoria.
- $assignGroupToStudForMinRepetitions()$: es una función que asigna a un estudiante aleatorio al grupo que menor repeticiones genere a la solución. Si debe decidir entre dos grupos desempata con el que mayor diversidad genere.

Algoritmo 1 Solución Inicial

```
procedure GETINITIALSOLUTION(studentGroup, groupMax, groupMin, atrsStandard, repMatrix)
    studentVector ← {1, 2, .., N}
    groupVector ← {1, 2, .., N}
    assignOneRandomStudentToEachGroup(studentGroup, repMatrix)
    while groupVector ≠ {} do
        selGroup ← assignGroupToStudForMinRepetitions(
        studentGroup, repMatrix)
        if groupCount[selGroup] = groupMinVector[selGroup] then
            groupVector ← groupVector – selGroup
        end if
    end while
    for g ← 1 to G do
        if groupCount[g] = groupMaxVector[g] then
            groupVector ← groupVector – g
        end if
    end for
    while groupVector ≠ {} do
        selGroup ← assignGroupToStudentForMinRepetitions(
        studentGroup, repMatrix)
        if groupCount[selGroup] = groupMaxVector[selGroup] then
            groupVector ← groupVector – selGroup
        end if
    end while
end procedure
```

Algoritmo 2 Solución Inicial Menos Repeticiones

```
procedure GETINITIALSOLUTIONLESSREPETITIONS(studentGroup, groupMax, groupMin, atrsStandard, repMatrix)
    bestRep ← 999999999
    for g ← 1 to MAX_TRIES do
        currentRep ← getInitialSolution(...)
        if currentRep ≤ bestRep then
            bestRep ← currentRep
            studentGroupBest ← studentGroup
        end if
    end for
    studentGroup ← studentGroupBest
end procedure
```

4.2.3. Búsqueda local: VND-2 y VND-3

En esta sección se describen las técnicas VNS utilizadas para la solución del problema y las adaptaciones necesarias para obtener una solución factible al problema.

Nuestra búsqueda local es implementada utilizando VND, para el cual se diseñan 3 tipos de entornos (Insertion, Swap, 3-Chain). Particularmente fueron implementados dos diferentes tipos de VND, VND-2 solamente hace cambios a partir de *insertion* y *swap* y mientras que el VND-3 hace cambios a partir de *Insertion*, *Swap* y *3-Chain*.

En esta búsqueda local descendente que desarrollamos se hace un cambio de estructura de entorno cada vez que se llega a un mínimo local.

Para el caso del VND-2 comenzamos haciendo todos los posibles *swap*, cuando el *swap* no encuentra más mejoras se pasan a hacer todos los *insertions* posibles. Si en alguno de los *insertions* aplicados se encuentra una mejora a la solución se vuelve a empezar la búsqueda local mediante los *swaps*. En el caso que el *insertion* no encuentre mejoras se termina el algoritmo de búsqueda local.

Para el caso del VND-3 se hace lo mismo que mencionamos anteriormente con la diferencia de que si el *insertion* no encuentra mejoras se pasa a ejecutar la operación de *3-chain*. Cuando este *3-chain* encuentra una mejora saltamos a la operación de *swap*, en el caso que no existan más mejoras terminamos el algoritmo de búsqueda local.

Algoritmo 3 VND-2

procedure VND-2(*studentGroup*, *sd*, *solCurrent*, *atrsStandard*, *tabuMatrix*)
    endWhile $\leftarrow false$
    resSwap $\leftarrow false$
    while not *endWhile* **do**
        repeat
            resSwap $\leftarrow Swap($*studentGroup*,*sd*,*solCurrent*,*atrsStandard*,
            tabuMatrix)
        until *resSwap* $= false$
        resIns $\leftarrow Insertion($*studentGroup*,*sd*,*solCurrent*,*atrsStandard*,
        tabuMatrix)
        if *resIns* $= false$ **then**
            endWhile $\leftarrow true$
        end if
    end while
end procedure

Algoritmo 4 VND-3

```
procedure VND-3(studentGroup, sd, solCurrent, atrsStandard, tabuMatrix)
    endWhile ← false
    resSwap ← false
    while not endWhile do
        repeat
            resSwap ← Swap(studentGroup,sd,solCurrent,atrsStandard,
            tabuMatrix)
        until resSwap = false
        resIns ← Insertion(studentGroup,sd,solCurrent,atrsStandard,
        tabuMatrix)
        if resIns = false then
            res3Chain ← 3Chain(studentGroup,sd,solCurrent,atrsStandard,
            tabuMatrix)
            if res3Chain = false then
                endWhile ← true
            end if
        end if
    end while
end procedure
```

Donde:

- $tabuSearchMatrix[i][j]$: es un entero que especifica la cantidad de iteraciones en las cuales se evita juntar el estudiante i con el estudiante j en un mismo grupo.

4.2.4. Tabú Search para minimizar repeticiones

Se aplica al algoritmo general el uso de *Tabu Search*. Para esto se introduce el concepto de *tabu list* la cual se implementa (en este caso) como una matriz simétrica de enteros de tamaño $N * N$ en donde cada posición guarda una valor entero que dirá cuantas veces la pareja quedará excluida de formar parte de una solución factible. La idea es restringir en los algoritmos de búsqueda local durante una cierta cantidad de iteraciones la posibilidad de que una pareja forme un mismo grupo. Para inicializar la matriz de la *Tabu Search* se utiliza el histórico de las parejas que ya formaron grupo en trimestres anteriores y cuantas veces lo hicieron. Este valor es multiplicado por un valor pre-establecido llamado TABU SEARCH VAR para que aquellas parejas que formaron más veces un mismo grupo queden excluidas más iteraciones que aquellas que formaron menor veces grupo. Aquellas que nunca formaron grupo no se las agrega a la *tabu list*, o lo que es lo mismo, se le asigna el valor 0 a la posición correspondiente de la matriz.

Algoritmo 5 CreateTabuSearchMatrix

```
procedure CreateTabuSearchMatrix(repetitionMatrix, tabuMatrix)
    for i ← 1 to N − 1 do
        for j ← i + 1 to N do
            repetitionIJ ← repetitionMatrix[i][j]
            if repetitionIJ > 0 then
                iterations ← repetitionIJ * TABU SEARCH VAR
                tabuMatrix[i][j] ← iterations
                tabuMatrix[j][i] ← iterations
            else
                tabuMatrix[i][j] ← 0
                tabuMatrix[j][i] ← 0
            end if
        end for
    end for
end procedure
```

Cada vez que en los métodos que veremos más adelante se evalúe la posibilidad de ingresar un estudiante i a un grupo g se evaluará también si la *tabu list* lo permite. Lo que se hace es comparar al estudiante i con todos los integrantes del grupo g y si el estudiante está habilitado por la *tabu list* a formar grupo con todos los integrantes del grupo g la función retorna true. En el caso de que esté restringido a formar grupo con al menos un estudiante del grupo g devuelve false.

Algoritmo 6 UpdateTabuSearchMatrix

```
procedure UpdateTabuSearchMatrix(student, group, studentGroup,
tabuMatrix)
    res ← true
    for i ← 1 to N do
        if studentGroup[i] = group and studentGroup ≠ i and
            tabuMatrix[stud][i] > 0 then
            res ← false
            aux ← tabuMatrix[stud][i]
            tabuMatrix[stud][i] ← aux − 1
            tabuMatrix[i][stud] ← aux − 1
        end if
    end for
    return res
end procedure
```

4.2.5. Insertion

Con esta operación se obtienen las soluciones de cambiar un estudiante de un grupo a otro. Mediante el uso de la matriz sd^c es posible computar eficientemente la función objetivo resultante del cambio. Se cambia el estudiante i del grupo g_1 al grupo g_2 lo cual hace que se tengan que re calcular valores en la matriz sd^c. Dado que el estudiante i es sacado del grupo g_1 es de esperar que la sumatoria de las diversidades de g_1 disminuya, por otro lado al agregar el estudiante i al grupo g_2 esto va a hacer que la sumatoria de las diversidades en g_2 aumente.

Los cambios en la matriz sd afectan únicamente a las columnas de los grupos g_1 y g_2, todos los valores de estas columnas van a ser actualizado.

Cambio en función objetivo:

$$f(x^n) - f(x^c) = sd^c[i][g_2] - sd^c[i][g_1] \tag{4.5}$$

Algoritmo 7 Insertion

```
procedure INSERTION(studentGroup, sd, solCurrent, atrsStandard, tabuMatrix)
    res ← false
    for i ← 1 to N do
        for g ← 1 to G do
            if studentGroup[i] ≠ g
                and groupCount[g] < GROUP_MAX and
                groupCount[studentGroup[i]] > GROUP_MIN then
                diffSol ← sd[i][g] − sd[i][studentGroup[i]]
                if diffSol > 0 and updateTabuSearchMatrix(
                    i, g, studentGroup, tabuMatrix) then
                    studentGroup[i] ← g
                    solCurrent ← solCurrent + diffSol
                    updateSD(studentGroup, sd, i, g)
                    res ← true
                end if
            end if
        end for
    end for
    return res
end procedure
```

Donde:

- $groupCount[g]$: es la cantidad máxima de integrantes que puede contener el grupo g.
- $updateSD(sd, solCurrent, i, g)$: actualiza la matriz de espacio de soluciones al mover el estudiante i al grupo g.

4.2.6. Swap

Con esta operación se obtienen las soluciones de intercambiar dos estudiantes de grupo, se sacan los estudiantes i, j de los grupos g_i y g_j respectivamente. El estudiante i va a ser asignado al grupo g_j y el estudiante j al grupo g_i. Si nos basamos en la matriz sd es fácilmente calculable el nuevo valor de la función objetivo. De la misma forma que en el *insertion*, la variación de estudiantes hace que se tenga que re calcular la matriz sd.

Cambio en función objetivo:

$$f(x^n) - f(x^c) = (sd^c[i][g_j] - sd^c[i][g_i]) + (sd^c[j][g_j] - sd^c[j][g_i]) - 2d_{ij} \quad (4.6)$$

Algoritmo 8 Swap

procedure SWAP(*studentGroup*, *sd*, *solCurrent*, *atrsStandard*, *tabuMatrix*)
  res $\leftarrow false$
  for $i \leftarrow 1$ *to* N **do**
    for $j \leftarrow 1$ *to* N **do**
      if *studentGroup*$[i] \neq$ *studentGroup*$[j]$ **then**
        diffSol $\leftarrow$ *sd*$[i]$[*studentGroup*$[j]$] $+$ *sd*$[j]$[*studentGroup*$[i]$]
        $-$*sd*$[i]$[*studentGroup*$[i]$] $-$ *sd*$[j]$[*studentGroup*$[j]$] $- 2d_{i,j}$
        if *diffSol* > 0
          and $updateTabuSearchMatrix($
          i, *studentGroup*$[j]$, *studentGroup*, *tabuMatrix*)
          and $updateTabuSearchMatrix($
          j, *studentGroup*$[i]$, *studentGroup*, *tabuMatrix*) **then**
          $oldI \leftarrow$ *studentGroup*$[i]$
          $oldJ \leftarrow$ *studentGroup*$[j]$
          studentGroup$[i] \leftarrow oldJ$
          studentGroup$[j] \leftarrow oldI$
          $updateSD$(*studentGroup*, *sd*, i, *studentGroup*$[i]$)
          $updateSD$(*studentGroup*, *sd*, j, *studentGroup*$[j]$)
          solCurrent $\leftarrow$ *solCurrent* $+$ *diffSol*
          res $\leftarrow true$
        end if
      end if
    end for
  end for
  return *res*
end procedure

4.2.7. 3-Chain

Con esta operación se obtienen las soluciones de intercambiar 3 estudiantes de grupo, se sacan los estudiantes i, j y k de los grupos g_i, g_j y g_k respectivamente. El movimiento a realizarse es asignar el estudiante i al grupo g_j, el estudiante j al grupo g_k y el estudiante k al grupo g_i. Como ya mencionamos anteriormente si nos basamos en la matriz sd es fácilmente calculable el nuevo valor de la función objetivo. De la misma forma que en el *swap*, la variación de estudiantes hace que se tenga que re calcular la matriz sd.

Cambio en función objetivo:

$$\begin{aligned} f(x^n) - f(x^c) = {} & (sd^c[i][g_j] - sd^c[i][g_i]) \\ & + (sd^c[j][g_k] - sd^c[j][g_j]) \\ & + (sd^c[k][g_i] - sd^c[k][g_k]) \\ & - (d_{ij} + d_{jk} + d_{ki}) \end{aligned} \quad (4.7)$$

Algoritmo 9 3-Chain

```
procedure 3-CHAIN(studentGroup, sd, solCurrent, atrsStandard, tabuMatrix)
    res ← false
    for i ← 1 to N do
        for j ← 1 to N do
            for k ← 1 to N do
                if studentGroup[i] ≠ studentGroup[j]
                    and studentGroup[j] ≠ studentGroup[k]  then
                    diffSol ← sd[i][studentGroup[j]] + sd[][studentGroup[k]]
                    +sd[k][studentGroup[i]] − sd[i][studentGroup[i]]
                    −sd[j][studentGroup[j]] − sd[k][studentGroup[k]]
                    −2d_{i,j} − 2d_{j,k} − 2d_{k,i}
                    if diffSol > 0
                        and updateTabuSearchMatrix(
                        i, studentGroup[j], studentGroup, tabuMatrix)
                        and updateTabuSearchMatrix(
                        j, studentGroup[k], studentGroup, tabuMatrix)
                        and updateTabuSearchMatrix(
                        k, studentGroup[i], studentGroup, tabuMatrix) then
                        oldI ← studentGroup[i]
                        oldJ ← studentGroup[j]
                        oldK ← studentGroup[k]
                        studentGroup[i] ← oldJ
                        studentGroup[j] ← oldK
                        studentGroup[k] ← oldI
                        updateSD(studentGroup, sd, i, studentGroup[i])
                        updateSD(studentGroup, sd, j, studentGroup[j])
                        updateSD(studentGroup, sd, k, studentGroup[k])
                        solCurrent ← solCurrent + diffSol
                        res ← true
                    end if
                end if
            end for
        end for
    end for
    return res
end procedure
```

4.2.8. Shake

Dada la necesidad de perturbar la solución actual de alguna forma se implementa la operación de *shake*. Se define el k-entorno como la solución obtenida de aplicar k *swap* consecutivos de estudiantes elegidos aleatoriamente que pertenezcan a distintos grupos. Análogamente a los casos anteriores, luego de ejecutar esta operación se re-calcula la matriz sd y el valor de la función objetivo.

Algoritmo 10 Shake

```
procedure SHAKE(studentGroup, k, sd, solCurrent, atrsStandard, tabuMatrix)
    while k > 0 do
        randomI ← getRandom(N)
        randomJ ← getRandom(N)
        if studentGroup[randomI] <> studentGroup[randomJ] then
            if updateTabuSearchMatrix(randomI,
                studentGroup[randomJ], studentGroup, tabuMatrix) and
                updateTabuSearchMatrix(randomJ,
                studentGroup[randomI], studentGroup, tabuMatrix) then
                oldI ← studentGroup[i]
                oldJ ← studentGroup[j]
                studentGroup[i] ← oldJ
                studentGroup[j] ← oldI
                updateSD(studentGroup, sd, i, studentGroup[i])
                updateSD(studentGroup, sd, j, studentGroup[j])
                k ← k − 1
            end if
        end if
    end while
    updateSolCurrent(solCurrent, sd, studentGroup
end procedure
```

4.2.9. VND Solución Final

Para la búsqueda de la solución global se aplican todos los conceptos previamente definidos y explicados. Esta búsqueda global parte de la idea de combinar los conceptos generales de GVNS con Skewed VNS, esto nos permite una intensificación de las búsquedas locales a partir del GVNS con una buena diversificación a partir de SVNS.

Luego de obtener una buena solución a partir de búsquedas locales es necesario buscar estrategias que nos ayuden a aproximarnos a una solución que sea un óptimo global, esta es va ser logrado en base a la búsqueda sesgada que se aplica. Las operaciones de *insertion*, *swap*, *3-chain* y *shake* reciben como entrada la matriz tabu search. Esta matriz es utilizada por las operaciones a la hora de decidir posibilidades de cambios de estudiantes, si dos estudiantes no satisfacen la condición de la tabu para ser cambiados entonces no se realiza el cambio. El uso de una búsqueda tabu aplicado al contexto de nuestra solución global es lo que nos permite cumplir con la restricción de minimizar las repeticiones lo más posible.

La búsqueda global comienza ejecutando la operación de obtener la solución inicial explicada anteriormente, posteriormente le aplica un VND-2 o VND-3 para encontrar una buena solución local, a partir de esta solución encontrada y durante una cantidad de iteraciones parametrizable se ejecuta la búsqueda sesgada. La búsqueda sesgada implica que luego de hacer un k-shake y una búsqueda local, nos quedamos con la solución encontrada si se satisface la siguiente condición:

$$\frac{f(x^n)}{f(x^c)} + \alpha\rho(x^c, x^n) > 1 \wedge \frac{f(x^n)}{f(x^b)} + \alpha\rho(x^b, x^n) > 1 \tag{4.8}$$

Donde ρ: se define como la función de distancia entre dos soluciones dadas.Esta se calcula como la fracción de pares que pertenecen al mismo grupo en una solución y en la otra solución no. El factor α es un valor configurable el cual nos permite ajustar nuestra condición de acuerdo a resultados obtenidos de la ejecución del algoritmo. Si este valor no es lo suficientemente pequeño para hacer que la condición evalué menor a 1 en una cierta cantidad de casos, entonces no tendría sentido tener esta condición. Por lo tanto se configura el parámetro para que esta condición evalué verdadera el 70 % de las veces en promedio. En cuanto a la diversificación de las soluciones el algoritmo general utiliza el k-shake el cual hace k movimientos aleatorios. Donde se define un k-mínimo y un k-máximo como parámetros configurables dando la posibilidad de que el algoritmo diversifique mas o menos la solución.

Algoritmo 11 VND Solución Final

procedure VNDSOLFINAL(k_{Min}, k_{Step}, k_{Max}, t_{Max})
    $getInitialSolutionLessRepetitions(studentGroupC, groupMax,$
    $groupMin, atrsStandard, repMatrix)$
    $vnd\text{-}3(studentGroup^c, sd^c, sol^c, atrsStandard, tabuMatrix)$
    $(studentGroup^b, sd^b, sol^b) \leftarrow (studentGroup^c, sd^c, sol^c)$
    $k \leftarrow k_{Min}$
    $t \leftarrow 0$
    while $t < t_{Max}$ **do**
        $(studentGroup^n, sd^n, sol^n) \leftarrow (studentGroup^c, sd^c, sol^c)$
        $Shake(studentGroup^n, k, sd^n, solCurrent^n, atrsStandard, tabuMatrix)$
        $vnd\text{-}3(studentGroup^n, sd^n, sol^n, atrsStandard, tabuMatrix)$
        $randomJ \leftarrow getRandom(N)$
        if $compareSol(studentGroup^n, sol^n, studentGroup^c, sol^c) > 1$ **and**
            $compareSol(studentGroup^n, sol^n, studentGroup^b, sol^b) > 1$ **then**
            $(studentGroup^c, sd^c, sol^c) \leftarrow (studentGroup^n, sd^n, sol^n)$
            if $sol^c > sol^b$ **then**
                $(studentGroup^b, sd^b, sol^b) \leftarrow (studentGroup^c, sd^c, sol^c)$
            end if
            $k \leftarrow k_{Min}$
        else
            $k \leftarrow k + k_{Step}$
            if $k > k_{Max}$ **then**
                $k \leftarrow k_{Min}$
            end if
        end if
    end while
    $updateSolCurrent(solCurrent, sd, studentGroup$
end procedure

Capítulo 5

Resultados

5.1. Desarrollo y entorno de ejecución

La herramienta fue desarrollada usando el lenguaje C++, con la nomenclatura escrita en lenguaje inglés para que sea fácilmente re utilizable. En todo momento se buscó y se logró contar con un sistema que tuviera alta cohesión y bajo acoplamiento. De esta manera cada módulo se especializa en una acción específica siendo, por ejemplo, sencillo agregar una nueva búsqueda local al VND. Se buscó que sea extensible y flexible para el usuario, si bien no contiene una interfaz amigable, mediante la alteración de archivos de texto, además de poder configurar una generación entera de estudiantes, se puede aumentar o disminuir la cantidad de grupos a formar, la cantidad máxima y mínima de integrantes por grupo, la cantidad de trimestres y aumentar los atributos por estudiantes.

Los resultados que veremos a continuación fueron ejecutados usando:

- gcc-mingw32 v4.8.1
- Sistema operativo: Windows 7 Ultimate
- Hardware: Sony Vaio VPCF24C5 (Intel Core i7 2.2GHz, 8GB DDR3 SDRAM, 500GB HDD).

A menos que especifique lo contrario, el juego de datos utilizado para las siguientes pruebas es una generación de estudiantes de MBA real la cual se detalla en el Anexo B.

5.2. Ajuste de parámetros

Dada la complejidad de la meta heurística desarrollada y la necesidad de darle flexibilidad, se agregaron una serie de parámetros internos lo cuales son configurables y ajustados de acuerdo a los mejores resultados obtenidos. Entendemos como mejores resultados a las soluciones que generen mayor diversidad y la vez tengan una cantidad moderada de repeticiones. Estos parámetros actúan independientemente sobre 3 factores de nuestra solución, la solución inicial, la *tabú search* y el SGVNS.

Solución inicial:

- GLOBAL INITIAL SOLUTION ITERATOR: define la cantidad de veces que se ejecuta la solución inicial con el objetivo de quedarse con la de menor repeticiones.

Tabú search:

- GLOBAL TABU SEARCH ON: define si se usa la *tabu search* o no, los valores posibles son (true, false).
- GLOBAL TABU SEARCH VAR: define, junto a la cantidad de repeticiones de grupo acarreadas entre dos estudiantes, un valor entero con el cual ése par de estudiantes serán excluidos por la *tabu search* de formar grupo durante un determinado período. Dado el valor p como la cantidad de veces que el estudiante i repitió con el j, definimos el valor i, j en la tabú como: p * GLOBAL TABU SEARCH VAR.

SGVNS:

- GLOBAL VND3 ON: define si se usa o no el VND-3, los valores posibles son (true,false). En el caso que este parámetro sea false, por defecto se usa el algoritmo de VND-2.
- GLOBAL K MIN: define la cantidad inicial con la que se inicializa la variable k del SGVNS. La variable k es utilizada por el *shake* para definir cuantos movimientos se van a hacer.
- GLOBAL K MAX: define la cantidad máxima alcanzable para la variable k del SGVNS. La variable k es utilizada por el *shake* para definir cuantos movimientos se van a hacer.
- GLOBAL K STEP: Define la cantidad a incrementar en cada paso del algoritmo a la variable k.
- GLOBAL T MAX: Define la cantidad de iteraciones del bucle principal del SGVNS.
- GLOBAL ALPHA: define el valor α utilizado en la condición explicada en Capítulo 4

$$\frac{f(x^n)}{f(x^c)} + \alpha\rho(x^c, x^n) > 1 \wedge \frac{f(x^n)}{f(x^b)} + \alpha\rho(x^b, x^n) > 1 \tag{5.1}$$

5.2.1. Definiendo valores

GLOBAL INITIAL SOLUTION ITERATOR

En cuanto a la solución inicial el parámetro GLOBAL INITIAL SOLUTION ITERATOR es quien define cuantas veces se ejecuta la solución inicial para seleccionar la que tenga menor cantidad de repeticiones posibles. En base a una serie de ejecuciones variando el valor de este parámetro y comparando la cantidad de repeticiones obtenidas llegamos a que el valor ideal para este parámetro es GLOBAL INITIAL SOLUTION ITERATOR = 100. Notamos en base a las pruebas realizadas que incrementar este valor por encima de 100 solamente tiene como resultado un n mayor tiempo de ejecución sin obtener mejores resultados en cuanto a las repeticiones.

Lograr un buen ajuste para este parámetro es importante dado que el algoritmo de SGVNS comenzara su ejecución en base a la cantidad de repeticiones obtenidas por la solución inicial, de todas formas es posible que el SGVNS pueda bajar el número de repeticiones debido a los movimiento de estudiantes que se realicen y la aplicación de la *tabú search*.

GLOBAL VND3 ON

El parámetro GLOBAL VND3 ON se define prendido (true) dado que se obtienen mejores resultados porque posee un método de búsqueda más respecto al VND-2 que es el que la herramienta usa por defecto.

GLOBAL ALPHA

El parámetro GLOBAL ALPHA utilizado en la condición del SGVNS previamente explicada fue ajustado de acuerdo a la necesidad de que la condición evalué verdadera en una cantidad adecuada de veces. En base a estadísticas de ejecución del fragmento del código que evalúa la condición como verdadera consideramos un ajuste del parámetro GLOBAL ALPHA= 0.005. Con este valor nos aseguramos que la condición evalué verdadera en aproximadamente un 60 % de las veces lo cual nos permite explorar sobre nuevas soluciones en el 60 % para así llegar a buenos resultados globales respecto a diversidad y repeticiones.

GLOBAL T MAX

El parámetro GLOBAL T MAX define la cantidad de repeticiones del bucle principal de SGVNS, este parámetro está ligado a los parámetros de la *tabu search* dado que cuantas más veces se ejecute el algoritmo se realizarán mayor cantidad liberaciones en la tabú. De acuerdo a esta consideración si aumentamos el GLOBAL T MAX también aumentaremos los parámetros de la tabú.
Inicialmente se busca definir GLOBAL T MAX de forma independiente a la *tabu search* para saber cuantas iteraciones son necesarias para obtener una buena solución. Para esto se apaga el uso de la *tabú search* estableciendo el parámetro GLOBAL TABU SEARCH ON en false.
Para las pruebas se busca variar el valor de GLOBAL T MAX para encontrar el punto en que la mejora de la solución se estabiliza. Se ejecutó la solución 5 veces para obtener un promedio de la calidad de la solución que se obtiene. Para esto usamos el algoritmo de la función objetivo del Capítulo 4. Se obtuvieron los siguientes resultados.

T MAX:	100	300	500	700	900
Ejecución 1:	520.255	520.333	520.449	520.390	520.618
Ejecución 2:	520.211	520.159	520.488	520.602	520.463
Ejecución 3:	520.075	520.463	520.499	520.511	520.394
Ejecución 4:	520.091	520.495	520.347	520.458	520.531
Ejecución 5:	520.353	520.556	520.596	520.505	520.468
Promedio:	**520.197**	**520.401**	**520.476**	**520.493**	**520.495**
Tiempo total (m):	4,2	14,3	21,3	29,5	38,5

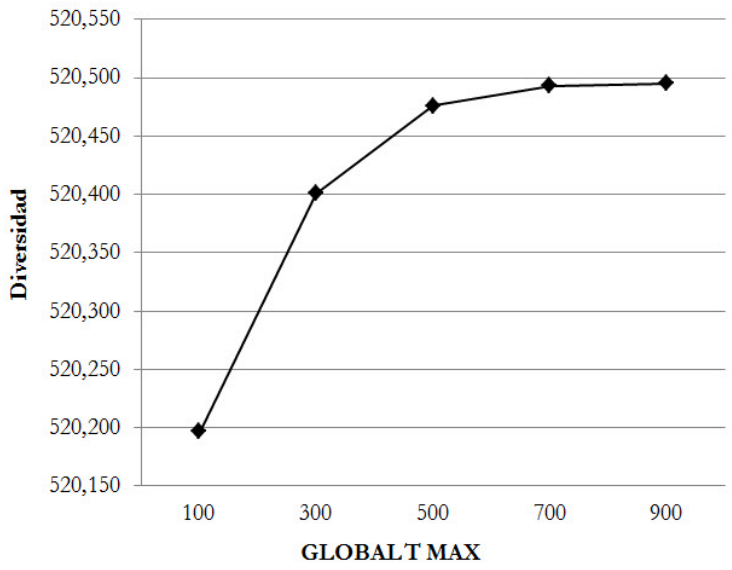

FIGURA 5.1: **Relación entre promedio de diversidad y GLOBAL T MAX**

Como se puede ver en la figura 5.1 a partir del GLOBAL T MAX definido en 500 se comienza a producir una meseta en la calidad de las soluciones. Por lo tanto aumentar el valor del parámetro no producirá soluciones considerablemente mayores. Debido a esto y a que el tiempo de ejecución aproximado para dicha configuración está en el rango de los 20 minutos (el cuál aumentará luego con el uso de la *tabu search*) consideramos este el valor apropiado para la ejecución de la solución.

GLOBAL K MIN, GLOBAL K MAX y GLOBAL K STEP

Necesitaremos ajustar los valores de GLOBAL K MIN, GLOBAL K MAX y GLOBAL K STEP utilizados por el *shake* de tal forma que esta función pueda obtener una buena diversificación de las soluciones pero a la vez haciéndolo de una forma moderada sin modificar completamente y de forma aleatoria la solución actual.
Realizamos pruebas con GLOBAL T MAX = 500 para establecer los valores apropiados.

Nro de Prueba	GLOBAL K MIN	GLOBAL K STEP	GLOBAL K MAX
1	1	1	1
2	1	1	2
3	1	1	3
4	2	2	4
5	1	2	20

Nro de Prueba:	1	2	3	4	5	6
Ejecución 1:	520.452	520.468	520.449	520.323	520.499	520.295
Ejecución 2:	520.309	520.618	520.488	520.297	520.451	520.423
Ejecución 3:	520.587	520.357	520.499	520.329	520.205	520.228
Ejecución 4:	520.587	520.401	520.347	520.370	520.618	520.374
Ejecución 5:	520.380	520.411	520.596	520.341	520.314	520.334
Promedio:	**520.463**	**520.451**	**520.476**	**520.332**	**520.417**	**520.331**

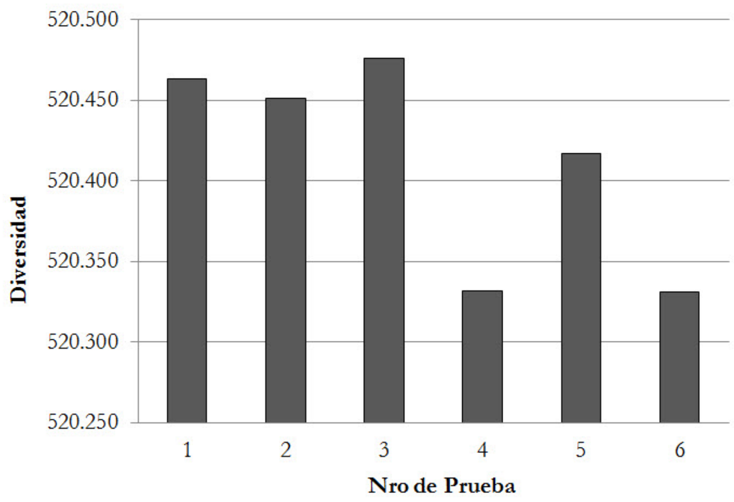

FIGURA 5.2: **Relación entre promedio de diversidad y Nro. de Prueba de los GLOBAL K**

Como se puede ver en la gráfica de la figura 5.2 las mejores soluciones son alcanzadas cuando GLOBAL K MIN = GLOBAL K STEP = 1 y GLOBAL K MAX = 3. Viendo el log que la aplicación produce ocurre que dado el valor fijado para GLOBAL ALPHA (0,005) el valor máximo que la aplicación alcanza termina siendo 3, es decir con las últimas pruebas nunca alcanzó los k máximos impuestos a pesar de establecer el GLOBAL K STEP mayor a 1.

GLOBAL TABU SEARCH VAR

El parámetro GLOBAL TABU SEARCH VAR se utiliza, junto a un histórico de repeticiones entre estudiantes para definir un valor con el que se excluirá temporalmente de las soluciones a una pareja de estudiantes, este valor se guarda en una *tabu list* como se explicó en el capitulo anterior.
Para las siguientes pruebas se fijaron los parámetros que ya establecimos. Se realizaron 3 ejecuciones para cada configuración de parámetros, se muestra

a continuación un conjunto reducido de los resultados promedios obtenidos de las ejecuciones con el fin de no sobrecargar las tablas y gráficas. Las pruebas completas se encuentran en el anexo A.

La siguiente tabla muestra la cantidad de repeticiones para cada trimestre obtenida para las distintas configuraciones del parámetro GLOBAL TABU SEARCH VAR:

TABU VAR	Trimestre1	Trimestre2	Trimestre3	Trimestre4	Trimestre5
5.000	0	0	8	46	98
25.000	0	0	3	30	74
45.000	0	0	1	21	62
85.000	0	0	1	18	56
125.000	0	0	1	19	51
165.000	0	0	2	19	53
205.000	0	0	1	17	51
245.000	0	0	2	15	51
285.000	0	0	2	18	51

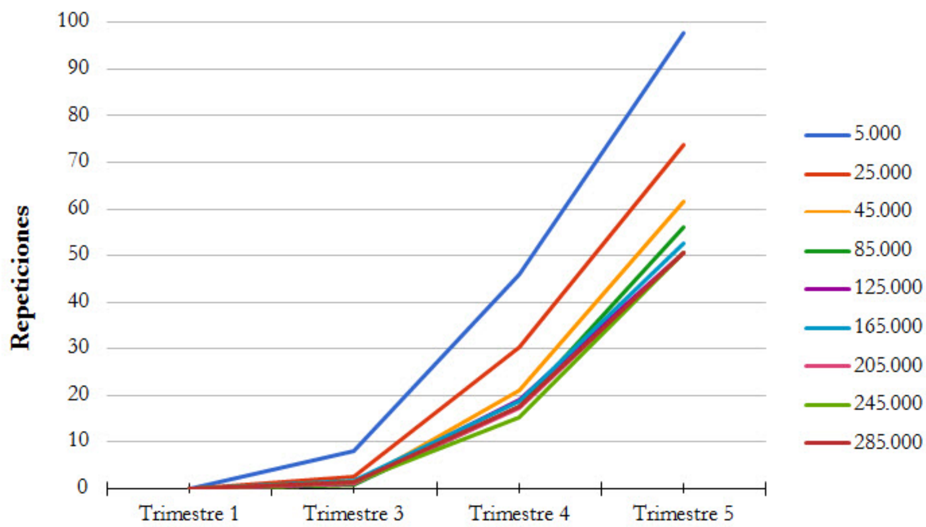

FIGURA 5.3: **Repeticiones obtenidas para los distintos GLOBAL TABU SEARCH VAR por trimestre**

Dados estos datos obtenemos la gráfica de la figura 5.3. Se aprecia, analizando principalmente los resultados en el trimestre 4 y 5, que cuando el valor del parámetro es pequeño la cantidad de repeticiones que se obtienen es considerablemente alto, luego a medida que el valor del parámetro aumenta la cantidad de repeticiones disminuye. Esto era de esperarse dado que cuanto mayor sea el valor con el que se restringe la formación de grupo de dos compañeros en la *tabu list* menor va a ser la posibilidad de que el algoritmo asigne esos dos estudiantes en un mismo grupo.

Los resultados obtenidos respecto a la diversidad ese muestran en la tabla siguiente y en la posterior gráfica de la figura 5.5.

TABU VAR	Trimestre1	Trimestre2	Trimestre3	Trimestre4	Trimestre5
5.000	520.361	517.095	514.505	518.161	518.183
25.000	520.432	516.713	505.648	516.364	515.823
45.000	520.348	516.876	504.219	511.087	514.185
85.000	520.388	516.858	499.710	511.402	512.208
125.000	520.384	516.802	500.061	509.246	510.535
165.000	520.457	516.726	497.551	503.810	510.159
205.000	520.442	517.253	501.225	504.345	508.620
245.000	520.421	517.113	497.066	500.503	507.821
285.000	520.424	517.944	501.948	501.173	506.785

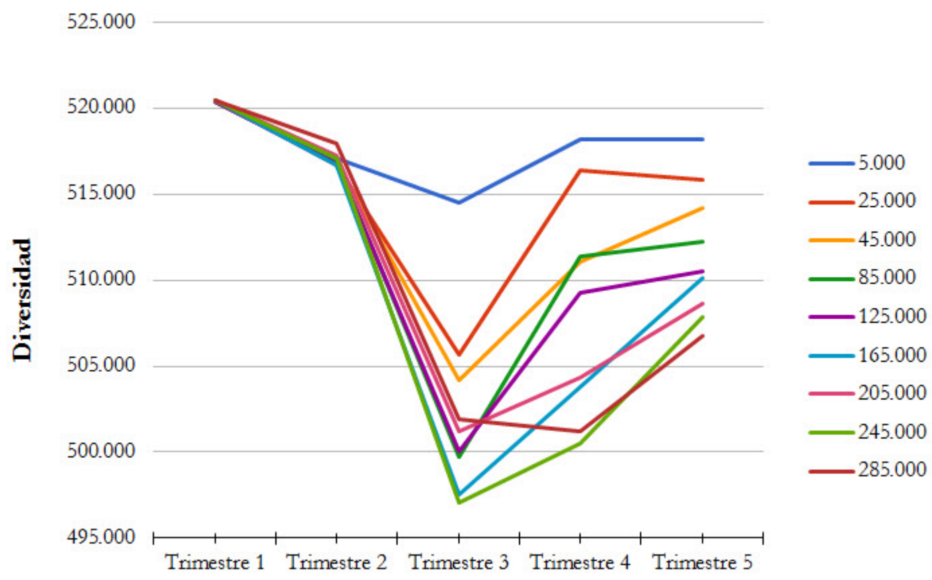

FIGURA 5.4: **Diversidad obtenida para los distintos GLOBAL TABU SEARCH VAR por trimestre**

En este caso vemos que a medida que el valor de GLOBAL TABU SEARCH VAR aumenta la calidad de la solución empeora, esto se produce debido a que el parámetro al aumentar reduce las posibilidades de los armados de grupo debido a que la restricción en las repeticiones juegan un papel mayor. También se puede ver que luego del valor 125.000 los resultados obtenidos comienzan a estancarse tanto en las repeticiones como la diversidad, y si analizamos más detalladamente los resultados obtenidos se ve que por momentos se tienen resultados realmente buenos y que no siempre son así debido al grado de aleatoriedad que la herramienta posee tanto en la creación de la solución inicial como en el método *shake*. Por lo que se deduce que es conveniente realizar la ejecución varias veces para optar por la que mejores resultados obtenga.

La decisión del valor optado para el parámetro dependerá de los intereses de quien solicita los resultados, a medida que el parámetro crece se tiene mayor impacto en la reducción de las repeticiones pero como contrapartida la diversidad también se reduce, cuando las repeticiones no son un problema cuanto mas bajo o nulo sea el parámetro mayor será la libertad del programa para obtener diversidad pero a costa de mayor repeticiones. En este caso, dado que la finalidad de la herramienta es un grupo de MBA, es prioritario mantener controladas las repeticiones entonces tomamos como un valor adecuado para el parámetro el GLOBAL TABU SEARCH VAR = 265.000 dado que con este se obtuvieron los mejores resultados en cuanto a una baja cantidad de repeticiones y el tiempo de ejecución es prudente, aunque si se buscan aún mejores soluciones es conveniente elegir un parámetro muy alto para aumentar las posibilidades de tener buenas soluciones teniendo en cuenta que el tiempo de ejecución aumentará.

Finalmente graficamos los tiempos de ejecución obtenidos:

TABU SEARCH VAR	Tiempo (minutos)
5.000	21
45.000	25
85.000	30
125.000	33
165.000	37
205.000	42
245.000	46
285.000	49

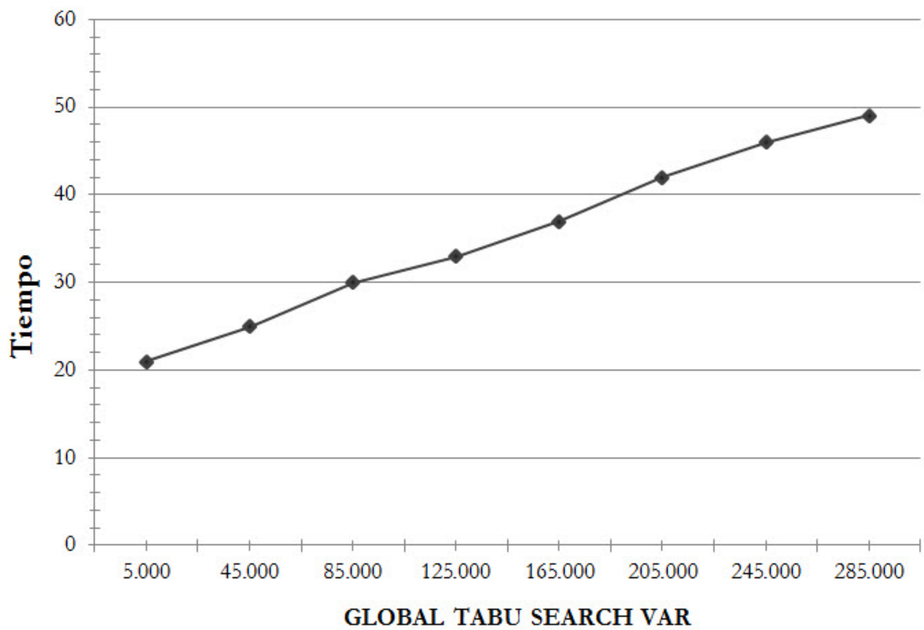

FIGURA 5.5: **Relación entre tiempo de ejecución y GLOBAL TABU SEARCH VAR**

Como se ve en los resultados el tiempo de ejecución aumenta a medida que el tamaño de la variable GLOBAL TABU SEARCH VAR aumenta. Este tiempo extra que ocurre a medida que aumenta el parámetro se debe a la naturaleza del método *shake* en el cual se busca de forma aleatoria un par de personas para intercambiarlas de grupo, y dado que este intercambio no siempre es posible porque es evaluado a partir de la *tabu list* termina ocurriendo que la búsqueda de esta pareja aleatoria demore un cierto tiempo porque se debe evaluar una pareja aleatoria tras de otra hasta que se llegue

a una pareja que no se está (o que terminó saliendo) de la *tabu list* y por lo tanto permita el intercambio.
Este fue el motivo que impulsó a que el parámetro GLOBAL T MAX fuera definido como la cantidad de iteraciones y no un temporizador para la ejecución dado que si fuera un temporizador sería necesario ajustarlo distinto en cada ordenador en el que se ejecute el algoritmo. De esta manera, sea cual sea el ordenador utilizado las operaciones ocurridas en la ejecución será la misma y solo lo que variará será el tiempo de ejecución.

5.3. Análisis de resultados

Luego de hacer el ajuste de parámetros y ejecutar las pruebas para la generación 2016 podemos ver cómo las soluciones obtenidas por la herramienta son muy diversas y con pocas repeticiones. A continuación vamos a hacer un recorrido por cada uno de los atributos utilizados en la herramienta para poder ver qué tanto se diferencia con el valor promedio de ese atributo en toda la generación.

Si analizamos el atributo de edad para cada grupo podemos ver cómo el valor promedio otorgado por la herramienta siempre está en el entorno de 30.7 +-(2.5) mientras que el valor promedio de la generación es de 30.7. Respecto al sexo donde el valor 1 representa femenino y 0 masculino vemos que el valor promedio otorgado por la herramienta es de 0.372 +-(0.206) mientras que el valor promedio de la generación es de 0.372. Si analizamos el puntaje en la prueba de ingreso vemos que el valor promedio otorgado por la herramienta está en el entorno de 393.8+-(78) mientras que el valor promedio de la generación es de 393.8. Si analizamos la localidad (Montevideo = 1/Interior = 0) vemos que el valor promedio otorgado por la herramienta esta en el entorno de 0.98+-(0.2) mientras que el promedio de la generación es de 0.98. Particularmente dada esta generación en la cual tenemos 1 solo estudiante del interior este atributo no hace la diferencia en cuanto a la diversidad, de todas formas se lo incluyó dado que este fue considerado como relevante a la hora de definir los atributos. Los resultados de una ejecución para la generación 2016 se encuentra en el Anexo B.

Analizando los resultados y diferenciando por trimestre podemos ver como en el tercer trimestre, en el cual la función de diversidad evalúa peor, es que se dan los valores más alejados al valor promedio por atributo. En el tercer trimestre tenemos este valor mínimo de diversidad dado que es el último trimestre hasta el cual el algoritmo encuentra soluciones sin la necesidad de introducir muchas repeticiones. Esto significa que hasta este trimestre la *tabu search* tiene peso a la hora de impedir que los estudiantes tengan repeticiones sin la necesidad de relajar la condición de repeticiones. En los siguientes dos trimestres se nota como el algoritmo tiene la necesidad de introducir repeticiones para llegar a la obtención de soluciones, por lo tanto al relajar la condición de repeticiones es esperable que se puedan obtener valores mas altos respecto a la diversidad.

La siguiente tabla muestra gráficamente los resultados promedios obtenidos por atributos:

Atributos	Prom.Gral	G.Min.Prom.	G Max.Pro.
Edad	30.7	29.5	32.428
Sexo	0.372	0.142	0.5
Puntaje admisión	393.7	338.667	452.571
Montevideo/Interior	0.98	0.833	1.0

Prom.Gral: promedio general de la generación
G.Min.Prom: grupo con el mínimo promedio retornado por la herramienta.
G.Max.Prom: grupo con el máximo promedio retornado por la herramienta.

Si observamos los atributos de profesiones podemos ver como estas son distribuidas diversamente en todos los grupos. En nuestra generación con predominancia de Contadores e Ingenieros tenemos la necesidad de que estas profesiones no invadan un grupo, dados los resultados obtenidos podemos ver como ningún grupo cuenta con mas de 3 contadores de los 6 o 7 integrantes y como ningún grupo cuenta con más de 2 ingenieros de los 6 o 7 integrantes.
Los resultados analizados se encuentran en el anexo B.

5.4. Herramienta vs. soluciones manuales

Para hacer una evaluación de la herramienta desarrollada se hacen comparaciones entre la distribución de equipos con la herramienta desarrollada y un conjunto de soluciones manuales. Las soluciones manuales constan de formaciones de grupos reales realizadas para generaciones de estudiantes en años anteriores las cuales fueron realizadas por el director y profesor del MBA en la IEEM - Universidad de Montevideo. Para hacer la comparación nos enfocamos específicamente en el valor de la función objetivo y la cantidad de repeticiones generadas por la solución.
Para hacer la evaluación de los grupos de generaciones anteriores, se cargan en la herramienta los grupos que se generaron manualmente para cada trimestre para así obtener una medida para cada solución utilizando las mismas métricas que usa el programa para generar las suyas propias.

A continuación se detallan comparativamente los resultados obtenidos para 3 generaciones anteriores. Para la obtención de la solución de la herramienta nos quedamos con la mejor de 5 ejecuciones.

Generación MBA 2013/2014-A

MBA 1314-B	Rep. Manual	Sol. Manual	Rep. Herramienta	Sol. Herramienta
Trimestre 1	0	169.971	0	218.373
Trimestre 2	0	180.625	0	216.253
Trimestre 3	9	185.499	1	202.261
Trimestre 4	23	163.451	14	205.851
Trimestre 5	28	138.875	43	217.207

Graficando los resultados obtenemos las siguientes gráficas:

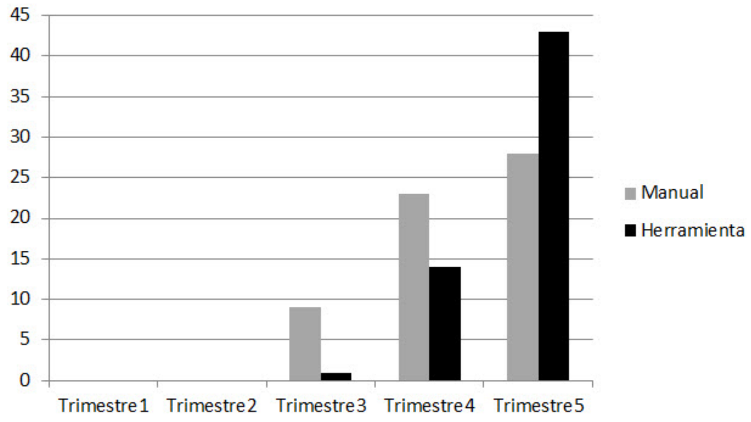

FIGURA 5.6: **Repeticiones MBA 2013/2014 A manual vs. herramienta**

Se puede ver en la figura 5.6 que en el trimestre 3 y 4 la herramienta genera menor número de repeticiones y en el trimestre 5 empeora respecto a la solución manual.

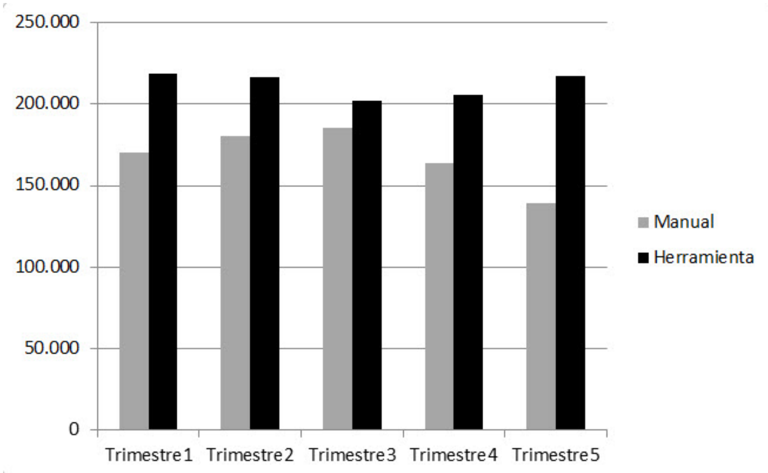

FIGURA 5.7: **Diversidad MBA 2013/2014 A manual vs. herramienta**

Se puede ver en la figura 5.7 que en todos los trimestres la diversidad obtenida por la herramienta es pareja y siempre por encima de la solución manual.

Generación MBA 2013/2014-B

MBA 1314-B	Rep. Manual	Sol. Manual	Rep. Herramienta	Sol. Herramienta
Trimestre 1	0	144.589	0	216.064
Trimestre 2	4	162.507	11	215.924
Trimestre 3	15	150.288	30	215.111
Trimestre 4	36	162.355	62	205.612
Trimestre 5	54	134.89	99	211.583

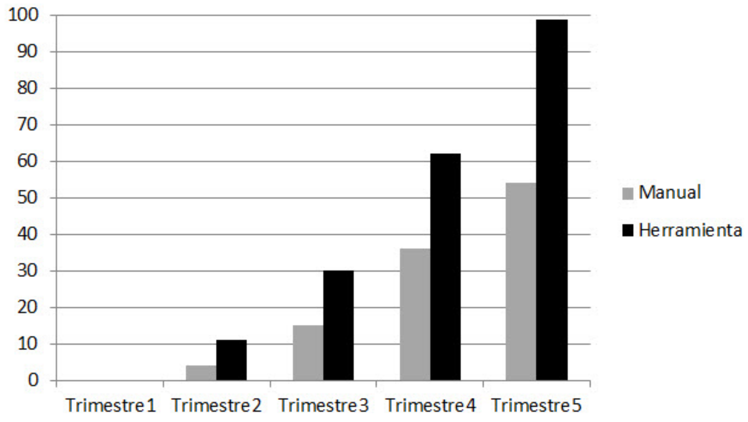

FIGURA 5.8: **Repeticiones MBA 2013/2014 B manual vs. herramienta**

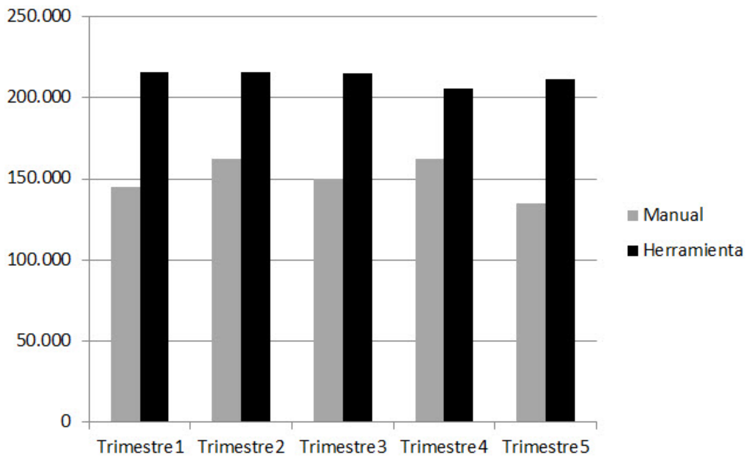

FIGURA 5.9: **Diversidad MBA 2013/2014 B manual vs. herramienta**

Analizando las gráficas de la figuras 5.8 y 5.9 vemos que en las repeticiones la herramienta pierde en todos los trimestres, vale considerar que dicha generación esta compuesta por tan solo 21 estudiantes lo cual hace que sea difícil establecer grupos para todos los trimestres generando pocas repeticiones. En cuanto a la diversidad nuevamente la herramienta muestra soluciones considerablemente mejores que las realizadas manualmente.

Generación MBA 2014/2015-A

MBA 1415-A	Rep. Manual	Sol. Manual	Rep. Herramienta	Sol. Herramienta
Trimestre 1	0	324.987	0	420.797
Trimestre 2	1	329.884	0	418.919
Trimestre 3	4	330.118	0	412.187
Trimestre 4	21	347.421	6	407.046
Trimestre 5	38	334.633	27	407.393

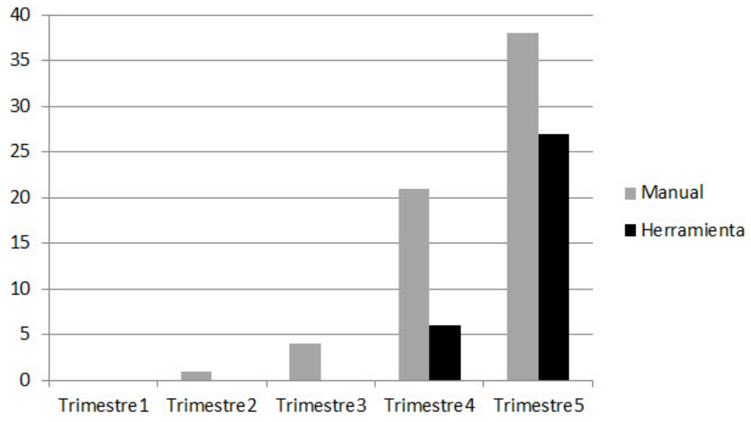

FIGURA 5.10: **Repeticiones MBA 2014/2015 A manual vs. herramienta**

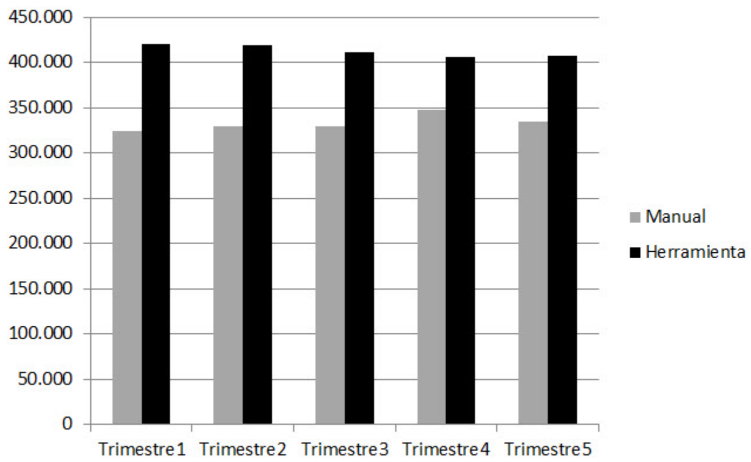

FIGURA 5.11: **Diversidad MBA 2014/2015 A manual vs. herramienta**

En la figuras 5.10 y 5.11 vemos que en este caso las repeticiones que la herramienta produce son menores a las producidas manualmente, también en cuanto a la diversidad la solución generada por la herramienta supera a la generada a mano.

Analizando todas las tablas comparativas podemos ver que las soluciones generadas por la herramienta son mejores que las manuales en lo que respecta a la diversidad. Por otro lado podemos ver que respecto a las repeticiones la herramienta no siempre nos ofrece mejores soluciones respecto a las manuales motivo de no mayor importancia dado que este no es el objetivo primario de la conformación de equipos multidisciplinarios.

Capítulo 6

Conclusiones

De acuerdo a los buenos resultados obtenidos respecto a diversidad y repeticiones podemos comprobar la correctitud del algoritmo desarrollado. Los resultados obtenidos mediante la herramienta superaron notoriamente las soluciones generadas manualmente por lo tanto podemos concluir que el uso de la herramienta va a tener buenos resultados en las generaciones del MBA.

Dado nuestro análisis de resultados podemos ver cómo cada grupo está muy aproximado al centroide de la generación. Donde el centroide de la generación representa un estudiante con los valores promedios de todos los atributos de la generación. Por lo tanto como todos los grupos se acercan a este valor concluimos que tenemos grupos diversos.

El tiempo de ejecución de la herramienta dependerá de los datos de entrada dada una determinada generación y también dependerá del ajuste de parámetros del programa. Es de esperar que si tenemos una generación con mayor número de estudiantes y con mayor cantidad de atributos sobre los estudiantes la ejecución de la herramienta va a tomar más tiempo. Un aumento en los valores para los parámetros GLOBAL TABU SEARCH VAR y GLOBAL T MAX implica un aumento en el tiempo de ejecución.

Respecto a la generación de repeticiones podemos concluir en base a los resultados obtenidos que la solución inicial es la que introduce la mayor cantidad de repeticiones. Por lo tanto para mitigar este problema se puede desarrollar un algoritmo que mejore las repeticiones en la generación de la solución inicial.
Otra opción podría ser usar una heurística como la del problema de los golfistas (Triska, 2008) para conocer anticipadamente los posibles grupos a formar de forma tal de que ningún par de alumnos repitan grupo. A partir de estos grupos conocidos nos quedamos con aquellos que generen más diversidad, de esta manera la solución inicial no introduce repeticiones y gracias al uso de la *tabu search* se disminuye durante la ejecución del SGVND la generación de repeticiones. Con la aplicación de esta técnica nos enfocamos mas en el problema de las repeticiones que en el de diversidad por lo tanto es de esperar que los resultados en cuanto a diversidad no sean tan buenos.

En los casos en los cuales la *tabu search* tiene liberaciones, osea se relaja la restricción de introducir repeticiones, vemos mejoras en la solución encontrada respecto a diversidad. Esto se debe a que en la búsqueda de soluciones se pueden considerar como válidas soluciones que antes estaban limitadas para no generar repeticiones. De esta forma vemos como se introducen repeticiones pero a cambio se aumenta la diversidad.

De acuerdo a las ejecuciones realizadas concluimos cómo un buen ajuste de parámetros nos lleva a encontrar mejores soluciones en cuanto a repeticiones y diversidad. También podemos ver como existe una relación entre los parámetros GLOBAL TABU SEARCH VAR y GLOBAL T MAX. Si aumentamos uno de estos parámetros implica que ese aumente el otro.
El parámetro GLOBAL TABU SEARCH VAR el cual es relevante a la hora de introducir las repeticiones tiene una cota superior. Esto significa que aumentarlo infinitamente no va a lograr que la solución no tenga repeticiones. Esto se debe a que la solución inicial introduce repeticiones y el algoritmo en base a los movimientos realizados de estudiantes no esta diseñado solamente para disminuir las repeticiones. Dentro de las posibles soluciones intermedias encontradas por el algoritmo solamente consideramos aquellas que evalúen mejor la función de diversidad o las que sean relativamente distintas y la función de diversidad evalué en el entorno de la última mejor solución encontrada.
En base a las ejecuciones realizadas concluimos que un valor adecuado para GLOBAL TABU SEARCH VAR es de 265000, si elegimos un valor superior no vamos a tener mayores beneficios en cuanto a disminuir las repeticiones.

Dada la combinación recomendada de parámetros el algoritmo obtiene muy buenas soluciones pero se recomienda hacer varias ejecuciones hasta encontrar la mejor dentro de una serie de 5 o 10 ejecuciones.
Este comportamiento se explica dada la aleatoriedad del algoritmo a la hora de armar la solución inicial y hacer *shakes* para introducir variaciones aleatorias en una determinada solución.

Finalmente podemos concluir como el algoritmo desarrollado es fácilmente aplicable y extensible a otras áreas de estudio. En el desarrollo del programa se tuvo en cuenta la posibilidad de agregar nuevos atributos y configurar todos los parámetros anteriormente mencionados.

Apéndice A

Resultados de ejecuciones

REPETICIONES - Ejecución 1					
TABU_SEARCH_VAR	T1	T2	T3	T4	T5
5.000	0	0	10	49	103
25.000	0	0	1	30	71
45.000	0	0	2	14	53
65.000	0	0	2	17	60
85.000	0	0	0	15	50
105.000	0	0	0	22	58
125.000	0	0	2	13	52
145.000	0	0	2	15	51
165.000	0	0	2	28	59
185.000	0	0	1	22	48
205.000	0	0	4	24	49
225.000	0	0	0	14	48
245.000	0	0	2	8	49
265.000	0	0	1	29	54
285.000	0	0	3	21	53

DIVERSIDAD - Ejecución 1						
TABU_SEARCH_VAR	T1	T2	T3	T4	T5	Tiempo (s)
5.000	520.393	516.808	515.455	518.396	518.316	1265
25.000	520.344	516.868	503.570	516.884	514.673	1386
45.000	520.230	516.924	506.298	499.338	514.237	1505
65.000	520.451	516.476	498.686	512.554	512.677	1657
85.000	520.356	516.881	496.856	508.148	512.504	1782
105.000	520.470	516.597	502.037	514.177	512.489	1848
125.000	520.336	516.453	506.152	507.400	510.769	2127
145.000	520.317	517.089	497.244	503.179	509.825	2382
165.000	520.370	517.136	493.862	512.677	508.063	2588
185.000	520.324	516.277	506.764	512.497	507.908	2358
205.000	520.451	517.469	502.668	501.959	501.963	2422
225.000	520.336	517.105	506.827	505.663	510.888	2584
245.000	520.360	517.040	491.405	483.237	512.042	2703
265.000	520.307	516.872	503.941	511.271	507.096	2811
285.000	520.488	517.109	500.325	507.224	506.666	2884

FIGURA A.1: **Ejecución 1**

REPETICIONES - Ejecución 2					
TABU_SEARCH_VAR	T1	T2	T3	T4	T5
5.000	0	0	5	32	84
25.000	0	0	3	39	79
45.000	0	0	1	20	64
65.000	0	0	4	15	58
85.000	0	0	2	18	60
105.000	0	0	1	21	52
125.000	0	0	2	21	51
145.000	0	0	0	24	57
165.000	0	0	3	10	43
185.000	0	0	11	32	66
205.000	0	0	0	25	55
225.000	0	0	0	23	57
245.000	0	0	1	16	48
265.000	0	0	0	8	42
285.000	0	0	1	13	49

DIVERSIDAD - Ejecución 2						
TABU_SEARCH_VAR	T1	T2	T3	T4	T5	Tiempo (s)
5.000	520.390	516.929	514.145	516.239	518.456	1238
25.000	520.378	516.734	507.920	515.986	515.957	1346
45.000	520.488	516.761	498.138	514.084	514.545	1509
65.000	520.482	516.634	499.925	482.226	514.068	1657
85.000	520.379	516.883	496.882	510.258	513.375	1801
105.000	520.295	517.090	504.178	513.129	510.526	2020
125.000	520.276	517.168	495.328	506.690	509.041	1933
145.000	520.618	517.370	506.692	510.631	510.441	2152
165.000	520.325	516.405	502.146	486.908	513.929	2037
185.000	520.370	516.948	516.276	512.205	510.223	2143
205.000	520.507	517.340	500.852	511.324	509.952	2596
225.000	520.281	517.098	494.424	510.389	508.562	2643
245.000	520.269	517.258	499.105	501.317	504.255	2796
265.000	520.396	516.391	504.039	487.457	504.824	2929
285.000	520.394	517.271	502.631	489.418	509.595	2965

FIGURA A.2: **Ejecución 2**

REPETICIONES - Ejecución 3					
TABU_SEARCH_VAR	T1	T2	T3	T4	T5
5.000	0	0	10	54	105
25.000	0	0	3	21	68
45.000	0	0	0	25	68
65.000	0	0	1	22	63
85.000	0	0	1	22	57
105.000	0	0	1	27	60
125.000	0	0	0	23	51
145.000	0	0	1	19	48
165.000	0	0	1	18	59
185.000	0	0	1	24	49
205.000	0	0	0	4	47
225.000	0	0	0	22	56
245.000	0	0	2	16	49
265.000	0	0	2	16	51
285.000	0	0	0	18	50

DIVERSIDAD - Ejecución 3						
TABU_SEARCH_VAR	T1	T2	T3	T4	T5	Tiempo (s)
5.000	520.144	516.791	514.753	519.531	518.173	1300
25.000	520.420	516.932	507.577	514.907	516.980	1378
45.000	520.363	516.785	503.198	515.741	512.374	1496
65.000	520.435	516.746	502.158	513.758	514.002	1607
85.000	520.357	516.315	500.814	515.254	511.104	1837
105.000	520.384	516.660	500.454	512.977	511.654	1841
125.000	520.321	516.840	506.871	508.156	512.507	1969
145.000	520.618	517.497	501.304	509.864	510.454	2068
165.000	520.731	516.942	497.621	511.821	509.823	2192
185.000	520.587	516.625	496.619	513.557	508.642	2369
205.000	520.464	517.211	499.589	494.605	512.241	2467
225.000	520.470	516.300	500.777	506.393	504.236	2505
245.000	520.517	517.341	504.645	505.369	510.354	2711
265.000	520.495	516.727	493.567	499.796	505.600	2819
285.000	520.517	520.517	497.290	510.350	504.926	2930

FIGURA A.3: **Ejecución 3**

REPETICIONES - Ejecución 4					
TABU_SEARCH_VAR	T1	T2	T3	T4	T5
5.000	0	0	7	49	99
25.000	0	0	3	31	77
45.000	0	0	0	25	62
65.000	0	0	2	23	54
85.000	0	0	0	16	58
105.000	0	0	0	22	49
125.000	0	0	1	19	49
145.000	0	0	1	28	62
165.000	0	0	1	19	50
185.000	0	0	3	25	55
205.000	0	0	0	16	51
225.000	0	0	0	22	50
245.000	0	0	1	21	56
265.000	0	0	0	7	34
285.000	0	0	2	18	50

DIVERSIDAD - Ejecución 4						
TABU_SEARCH_VAR	T1	T2	T3	T4	T5	Tiempo (s)
5.000	520.517	517.852	513.668	518.476	517.787	1318
25.000	520.587	516.316	503.524	517.678	515.680	1419
45.000	520.312	517.032	509.243	515.184	515.584	1481
65.000	520.370	517.240	504.177	514.392	511.135	1586
85.000	520.459	517.353	504.288	511.947	511.849	1807
105.000	520.449	517.390	497.718	513.481	513.207	1895
125.000	520.603	516.746	491.892	514.738	509.822	1953
145.000	520.488	517.307	495.289	510.579	508.865	2090
165.000	520.401	516.420	496.576	503.832	508.821	2119
185.000	520.303	516.637	506.309	501.624	509.283	2378
205.000	520.345	516.992	501.789	509.492	510.323	2496
225.000	520.364	516.524	496.773	512.521	504.999	2516
245.000	520.536	516.812	493.107	512.087	504.633	2806
265.000	520.361	516.762	503.581	493.789	509.762	2856
285.000	520.295	516.878	507.546	497.701	505.953	2914

FIGURA A.4: **Ejecución 4**

REPETICIONES - Promedio					
TABU_SEARCH_VAR	T1	T2	T3	T4	T5
5.000	0	0	8	46	98
25.000	0	0	3	30	74
45.000	0	0	1	21	62
65.000	0	0	2	19	59
85.000	0	0	1	18	56
105.000	0	0	1	23	55
125.000	0	0	1	19	51
145.000	0	0	1	22	55
165.000	0	0	2	19	53
185.000	0	0	4	26	55
205.000	0	0	1	17	51
225.000	0	0	0	20	53
245.000	0	0	2	15	51
265.000	0	0	1	15	45
285.000	0	0	2	18	51

RESULTADO - Promedio						
TABU_SEARCH_VAR	T1	T2	T3	T4	T5	Tiempo (s)
5.000	520.361	517.095	514.505	518.161	518.183	1.280
25.000	520.432	516.713	505.648	516.364	515.823	1.382
45.000	520.348	516.876	504.219	511.087	514.185	1.498
65.000	520.435	516.774	501.237	505.733	512.971	1.627
85.000	520.388	516.858	499.710	511.402	512.208	1.807
105.000	520.400	516.934	501.097	513.441	511.969	1.901
125.000	520.384	516.802	500.061	509.246	510.535	1.996
145.000	520.510	517.316	500.132	508.563	509.896	2.173
165.000	520.457	516.726	497.551	503.810	510.159	2.234
185.000	520.396	516.622	506.492	509.971	509.014	2.312
205.000	520.442	517.253	501.225	504.345	508.620	2.495
225.000	520.363	516.757	499.700	508.742	507.171	2.562
245.000	520.421	517.113	497.066	500.503	507.821	2.754
265.000	520.390	516.688	501.282	498.078	506.821	2.854
285.000	520.424	517.944	501.948	501.173	506.785	2.923

FIGURA A.5: **Ejecución Promedio**

Apéndice B

Archivos de salida de ejecución

B.1. Solución MBA 2016

Conformacion Grupos de Estudio

** TRIMESTRE 1 **

GRUPO 1:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
75916	0	29	Ing. Ind. Mec.	637	1
74402	1	29	Contador	611	1
133806	0	33	Ing. Computacion	315	1
152043	1	27	Contador	238	1
79795	0	28	Contador	380	1
152751	1	30	Quimico Farm.	336	1
151723	0	42	Lic. Marketing	199	1
Promedio:	0.428571	31.1429		388	1

GRUPO 2:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
112163	0	37	Ing. Quimico	410	1
144527	0	29	Contador	602	1
119962	1	30	Arquitecto	319	1
83914	0	32	Contador	236	1
79211	1	27	Contador	380	1
152528	0	28	Contador	396	1
Promedio:	0.333333	30.5		390.5	1

GRUPO 3:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
75747	0	29	Contador	361	1
132406	0	37	Ing. Ind. Mec.	368	1
151997	1	28	Lic. Neg. Inter.	361	1
117953	1	30	Contador	311	1
115201	0	30	Contador	539	1
123928	0	27	Ing. Quimico	456	1
Promedio:	0.333333	30.1667		399.333	1

GRUPO 4:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
73887	0	29	Contador	486	1
7480	0	37	Lic. Adm. Empresa	477	1
103087	0	31	Ing. Quimico	463	1
120359	1	29	Arquitecto	313	1
149961	1	30	Contador	315	1
125127	0	27	Contador	317	1
Promedio:	0.333333	30.5		395.167	1

GRUPO 5:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
145937	0	33	Ing. Computacion	313	1
74910	0	30	Contador	428	1
77883	1	27	Contador	231	1
144539	1	29	Lic. Economia	447	1
151465	0	37	Lic. Neg. Inter.	484	1
125630	0	30	Ing. Quimico	528	1
Promedio:	0.333333	31		405.167	1

GRUPO 6:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
77005	0	30	Ing. Ind. Mec.	315	1
83509	0	27	Lic. Economia	620	1
144837	1	28	Economista	493	1
3130	0	34	Ing. Quimico	620	1
143895	0	33	Contador	366	1
116934	1	28	Contador	189	1
119350	1	35	Dr. Derecho	188	1
Promedio:	0.428571	30.7143		398.714	1

GRUPO 7:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
76576	0	30	Ing. Telecom.	623	1
46401	1	30	Lic. Adm. Empresa	280	1
146552	0	31	Contador	405	1
145386	0	29	Ing. Computacion	259	1
142472	0	42	Contador	313	0
75615	1	29	Lic. Com. Social	373	1
74497	1	29	Ing. Alimentos	495	1
Promedio:	0.428571	31.4286		392.571	0.857143

GRUPO 8:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
149824	0	26	Contador	382	1
151548	0	33	Ing. Computacion	507	1
149856	1	28	Lic. Comunicacion	275	1
151659	0	29	Quimico Farm.	412	1
92098	0	35	Contador	306	1
152491	1	30	Ing. Telecom.	405	1
Promedio:	0.333333	30.1667		381.167	1

```
************************************************** TRIMESTRE 2 **************************************************
```

GRUPO 1:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
144527	0	29	Contador	602	1
146552	0	31	Contador	405	1
152043	1	27	Contador	238	1
92098	0	35	Contador	306	1
120359	1	29	Arquitecto	313	1
125630	0	30	Ing. Quimico	528	1
Promedio:	0.333333	30.1667		398.667	1

GRUPO 2:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
112163	0	37	Ing. Quimico	410	1
77005	0	30	Ing. Ind. Mec.	315	1
74402	1	29	Contador	611	1
73887	0	29	Contador	486	1
77883	1	27	Contador	231	1
151997	1	28	Lic. Neg. Inter.	361	1
142472	0	42	Contador	313	0
Promedio:	0.428571	31.7143		389.571	0.857143

GRUPO 3:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
46401	1	30	Lic. Adm. Empresa	280	1
132406	0	37	Ing. Ind. Mec.	368	1
75916	0	29	Ing. Ind. Mec.	637	1
149856	1	28	Lic. Comunicacion	275	1
152528	0	28	Contador	396	1
143895	0	33	Contador	366	1
Promedio:	0.333333	30.8333		387	1

GRUPO 4:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
145937	0	33	Ing. Computacion	313	1
83509	0	27	Lic. Economia	620	1
151659	0	29	Quimico Farm.	412	1
117953	1	30	Contador	311	1
74497	1	29	Ing. Alimentos	495	1
125127	0	27	Contador	317	1
151723	0	42	Lic. Marketing	199	1
Promedio:	0.285714	31		381	1

GRUPO 5:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
74910	0	30	Contador	428	1
83914	0	32	Contador	236	1
75615	1	29	Lic. Com. Social	373	1
152491	1	30	Ing. Telecom.	405	1
3130	0	34	Ing. Quimico	620	1
149961	1	30	Contador	315	1
123928	0	27	Ing. Quimico	456	1
Promedio:	0.428571	30.2857		404.714	1

GRUPO 6:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
76576	0	30	Ing. Telecom.	623	1
75747	0	29	Contador	361	1
7480	0	37	Lic. Adm. Empresa	477	1
144539	1	29	Lic. Economia	447	1
152751	1	30	Quimico Farm.	336	1
116934	1	28	Contador	189	1
Promedio:	0.5	30.5		405.5	1

GRUPO 7:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
151548	0	33	Ing. Computacion	507	1
144837	1	28	Economista	493	1
119962	1	30	Arquitecto	319	1
145386	0	29	Ing. Computacion	259	1
79795	0	28	Contador	380	1
151465	0	37	Lic. Neg. Inter.	484	1
Promedio:	0.333333	30.8333		407	1

GRUPO 8:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
149824	0	26	Contador	382	1
133806	0	33	Ing. Computacion	315	1
103087	0	31	Ing. Quimico	463	1
79211	1	27	Contador	380	1
115201	0	30	Contador	539	1
119350	1	35	Dr. Derecho	188	1
Promedio:	0.333333	30.3333		377.833	1

```
*************************************************** TRIMESTRE 3 **************************************************

GRUPO 1:
          Nombre/Id  |         Sexo  |         Edad  |            Carrera  |  Puntaje Test  |     Localidad
              73887  |            0  |           29  |           Contador  |           486  |             1
              92098  |            0  |           35  |           Contador  |           306  |             1
             152528  |            0  |           28  |           Contador  |           396  |             1
             151723  |            0  |           42  |     Lic. Marketing  |           199  |             1
             116934  |            1  |           28  |           Contador  |           189  |             1
             123928  |            0  |           27  |       Ing. Quimico  |           456  |             1

          Promedio:  |     0.166667  |         31.5  |                     |       338.667  |             1

GRUPO 2:
          Nombre/Id  |         Sexo  |         Edad  |            Carrera  |  Puntaje Test  |     Localidad
             112163  |            0  |           37  |       Ing. Quimico  |           410  |             1
              46401  |            1  |           30  |  Lic. Adm. Empresa  |           280  |             1
              75747  |            0  |           29  |           Contador  |           361  |             1
              74910  |            0  |           30  |           Contador  |           428  |             1
             103087  |            0  |           31  |       Ing. Quimico  |           463  |             1
              79795  |            0  |           28  |           Contador  |           380  |             1

          Promedio:  |     0.166667  |      30.8333  |                     |           387  |             1

GRUPO 3:
          Nombre/Id  |         Sexo  |         Edad  |            Carrera  |  Puntaje Test  |     Localidad
             146552  |            0  |           31  |           Contador  |           405  |             1
             119962  |            1  |           30  |         Arquitecto  |           319  |             1
             151659  |            0  |           29  |      Quimico Farm.  |           412  |             1
             144539  |            1  |           29  |      Lic. Economia  |           447  |             1
             143895  |            0  |           33  |           Contador  |           366  |             1
             115201  |            0  |           30  |           Contador  |           539  |             1

          Promedio:  |     0.333333  |      30.3333  |                     |       414.667  |             1

GRUPO 4:
          Nombre/Id  |         Sexo  |         Edad  |            Carrera  |  Puntaje Test  |     Localidad
              76576  |            0  |           30  |      Ing. Telecom.  |           623  |             1
              83509  |            0  |           27  |      Lic. Economia  |           620  |             1
             133806  |            0  |           33  |   Ing. Computacion  |           315  |             1
             149856  |            1  |           28  |  Lic. Comunicacion  |           275  |             1
              83914  |            0  |           32  |           Contador  |           236  |             1
             151997  |            1  |           28  |    Lic. Neg. Inter.  |           361  |             1
             125630  |            0  |           30  |       Ing. Quimico  |           528  |             1

          Promedio:  |     0.285714  |      29.7143  |                     |       422.571  |             1
```

GRUPO 5:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
149824	0	26	Contador	382	1
75916	0	29	Ing. Ind. Mec.	637	1
77883	1	27	Contador	231	1
144837	1	28	Economista	493	1
75615	1	29	Lic. Com. Social	373	1
125127	0	27	Contador	317	1
Promedio:	0.5	27.6667		405.5	1

GRUPO 6:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
145937	0	33	Ing. Computacion	313	1
132406	0	37	Ing. Ind. Mec.	368	1
74402	1	29	Contador	611	1
144527	0	29	Contador	602	1
145386	0	29	Ing. Computacion	259	1
149961	1	30	Contador	315	1
119350	1	35	Dr. Derecho	188	1
Promedio:	0.428571	31.7143		379.429	1

GRUPO 7:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
151548	0	33	Ing. Computacion	507	1
142472	0	42	Contador	313	0
79211	1	27	Contador	380	1
120359	1	29	Arquitecto	313	1
117953	1	30	Contador	311	1
3130	0	34	Ing. Quimico	620	1
152751	1	30	Quimico Farm.	336	1
Promedio:	0.571429	32.1429		397.143	0.857143

GRUPO 8:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
77005	0	30	Ing. Ind. Mec.	315	1
152043	1	27	Contador	238	1
7480	0	37	Lic. Adm. Empresa	477	1
152491	1	30	Ing. Telecom.	405	1
151465	0	37	Lic. Neg. Inter.	484	1
74497	1	29	Ing. Alimentos	495	1
Promedio:	0.5	31.6667		402.333	1

TRIMESTRE 4

GRUPO 1:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
144527	0	29	Contador	602	1
149856	1	28	Lic. Comunicacion	275	1
103087	0	31	Ing. Quimico	463	1
3130	0	34	Ing. Quimico	620	1
144539	1	29	Lic. Economia	447	1
151723	0	42	Lic. Marketing	199	1
Promedio:	0.333333	32.1667		434.333	1

GRUPO 2:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
149824	0	26	Contador	382	1
46401	1	30	Lic. Adm. Empresa	280	1
83509	0	27	Lic. Economia	620	1
119962	1	30	Arquitecto	319	1
7480	0	37	Lic. Adm. Empresa	477	1
123928	0	27	Ing. Quimico	456	1
Promedio:	0.333333	29.5		422.333	1

GRUPO 3:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
112163	0	37	Ing. Quimico	410	1
75747	0	29	Contador	361	1
133806	0	33	Ing. Computacion	315	1
144837	1	28	Economista	493	1
142472	0	42	Contador	313	0
152491	1	30	Ing. Telecom.	405	1
152528	0	28	Contador	396	1
Promedio:	0.285714	32.4286		384.714	0.857143

GRUPO 4:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
132406	0	37	Ing. Ind. Mec.	368	1
75916	0	29	Ing. Ind. Mec.	637	1
146552	0	31	Contador	405	1
151997	1	28	Lic. Neg. Inter.	361	1
79211	1	27	Contador	380	1
79795	0	28	Contador	380	1
116934	1	28	Contador	189	1
Promedio:	0.428571	29.7143		388.571	1

GRUPO 5:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
77005	0	30	Ing. Ind. Mec.	315	1
73887	0	29	Contador	486	1
145386	0	29	Ing. Computacion	259	1
75615	1	29	Lic. Com. Social	373	1
117953	1	30	Contador	311	1
125630	0	30	Ing. Quimico	528	1
Promedio:	0.333333	29.5		378.667	1

GRUPO 6:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
76576	0	30	Ing. Telecom.	623	1
74910	0	30	Contador	428	1
151548	0	33	Ing. Computacion	507	1
152043	1	27	Contador	238	1
125127	0	27	Contador	317	1
119350	1	35	Dr. Derecho	188	1
Promedio:	0.333333	30.3333		383.5	1

GRUPO 7:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
145937	0	33	Ing. Computacion	313	1
74402	1	29	Contador	611	1
83914	0	32	Contador	236	1
151659	0	29	Quimico Farm.	412	1
120359	1	29	Arquitecto	313	1
115201	0	30	Contador	539	1
Promedio:	0.333333	30.3333		404	1

GRUPO 8:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
77883	1	27	Contador	231	1
92098	0	35	Contador	306	1
151465	0	37	Lic. Neg. Inter.	484	1
74497	1	29	Ing. Alimentos	495	1
152751	1	30	Quimico Farm.	336	1
149961	1	30	Contador	315	1
143895	0	33	Contador	366	1
Promedio:	0.571429	31.5714		361.857	1

```
************************************************** TRIMESTRE 5 **************************************************

GRUPO 1:
    Nombre/Id |     Sexo |     Edad |            Carrera | Puntaje Test |  Localidad
        74910 |        0 |       30 |           Contador |          428 |          1
        73887 |        0 |       29 |           Contador |          486 |          1
       133806 |        0 |       33 |   Ing. Computacion |          315 |          1
       120359 |        1 |       29 |         Arquitecto |          313 |          1
       144539 |        1 |       29 |      Lic. Economia |          447 |          1
        74497 |        1 |       29 |     Ing. Alimentos |          495 |          1

    Promedio: |      0.5 |  29.8333 |                    |          414 |          1

GRUPO 2:
    Nombre/Id |     Sexo |     Edad |            Carrera | Puntaje Test |  Localidad
       132406 |        0 |       37 |     Ing. Ind. Mec. |          368 |          1
       149856 |        1 |       28 |  Lic. Comunicacion |          275 |          1
       152043 |        1 |       27 |           Contador |          238 |          1
         3130 |        0 |       34 |       Ing. Quimico |          620 |          1
       152528 |        0 |       28 |           Contador |          396 |          1
        79795 |        0 |       28 |           Contador |          380 |          1
       149961 |        1 |       30 |           Contador |          315 |          1

    Promedio: | 0.428571 |  30.2857 |                    |      370.286 |          1

GRUPO 3:
    Nombre/Id |     Sexo |     Edad |            Carrera | Puntaje Test |  Localidad
        77005 |        0 |       30 |     Ing. Ind. Mec. |          315 |          1
       144837 |        1 |       28 |         Economista |          493 |          1
       146552 |        0 |       31 |           Contador |          405 |          1
       119962 |        1 |       30 |         Arquitecto |          319 |          1
         7480 |        0 |       37 |  Lic. Adm. Empresa |          477 |          1
        92098 |        0 |       35 |           Contador |          306 |          1

    Promedio: | 0.333333 |  31.8333 |                    |      385.833 |          1

GRUPO 4:
    Nombre/Id |     Sexo |     Edad |            Carrera | Puntaje Test |  Localidad
        76576 |        0 |       30 |      Ing. Telecom. |          623 |          1
       112163 |        0 |       37 |       Ing. Quimico |          410 |          1
       149824 |        0 |       26 |           Contador |          382 |          1
        77883 |        1 |       27 |           Contador |          231 |          1
       145386 |        0 |       29 |   Ing. Computacion |          259 |          1
       151723 |        0 |       42 |     Lic. Marketing |          199 |          1
       115201 |        0 |       30 |           Contador |          539 |          1

    Promedio: | 0.142857 |  31.5714 |                    |      377.571 |          1
```

GRUPO 5:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
46401	1	30	Lic. Adm. Empresa	280	1
144527	0	29	Contador	602	1
151997	1	28	Lic. Neg. Inter.	361	1
152491	1	30	Ing. Telecom.	405	1
143895	0	33	Contador	366	1
125127	0	27	Contador	317	1
Promedio:	0.5	29.5		388.5	1

GRUPO 6:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
151659	0	29	Quimico Farm.	412	1
142472	0	42	Contador	313	0
152751	1	30	Quimico Farm.	336	1
125630	0	30	Ing. Quimico	528	1
123928	0	27	Ing. Quimico	456	1
119350	1	35	Dr. Derecho	188	1
Promedio:	0.333333	32.1667		372.167	0.833333

GRUPO 7:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
74402	1	29	Contador	611	1
151548	0	33	Ing. Computacion	507	1
83914	0	32	Contador	236	1
103087	0	31	Ing. Quimico	463	1
117953	1	30	Contador	311	1
116934	1	28	Contador	189	1
Promedio:	0.5	30.5		386.167	1

GRUPO 8:

Nombre/Id	Sexo	Edad	Carrera	Puntaje Test	Localidad
75747	0	29	Contador	361	1
145937	0	33	Ing. Computacion	313	1
75916	0	29	Ing. Ind. Mec.	637	1
83509	0	27	Lic. Economia	620	1
79211	1	27	Contador	380	1
75615	1	29	Lic. Com. Social	373	1
151465	0	37	Lic. Neg. Inter.	484	1
Promedio:	0.285714	30.1429		452.571	1

```
***** RESULTADOS: Total de Repeticiones en cada Trimestre *****

TRIMESTRE 1: 0
TRIMESTRE 2: 0
TRIMESTRE 3: 0
TRIMESTRE 4: 13
TRIMESTRE 5: 41

***** RESULTADOS: Puntaje de la Solución en cada Trimestre *****

TRIMESTRE 1: 520.421
TRIMESTRE 2: 516.906
TRIMESTRE 3: 492.115
TRIMESTRE 4: 499.193
TRIMESTRE 5: 487.006

********** DATOS DE INTERÉS **********

>>> Parámetros del sistema:

    GLOBAL_VND3_ON: true
    GLOBAL_T_MAX: 500
    GLOBAL_TABU_SEARCH_ON: true
    GLOBAL_TABU_SEARCH_VAR: 53000
    GLOBAL_ALPHA: 0.005
    GLOBAL_K_MIN: 1
    GLOBAL_K_STEP: 1
    GLOBAL_K_MAX: 3
    GLOBAL_INITIAL_SOLUTION_ITERATOR: 100

>>> Parámetros definidos por el usuario:

    Cantidad de estudiantes: 51
    Cantidad de grupos: 8
    Cantidad mínima de estudiantes por grupo: 6
    Cantidad máxima de estudiantes por grupo: 7
    Cantidad de atributos por estudiante: 7
    Cantidad de semestres: 5

>>> Estadisticas de la solución en cada trimestre:

    TRIMESTRE 1: Inserts: 9 | Swaps: 9318 | TreeChains: 71 | Shakes:
    TRIMESTRE 2: Inserts: 92 | Swaps: 2720 | TreeChains: 237 | Shakes
    TRIMESTRE 3: Inserts: 4 | Swaps: 492 | TreeChains: 1 | Shakes: 99
    TRIMESTRE 4: Inserts: 12 | Swaps: 1097 | TreeChains: 0 | Shakes:
    TRIMESTRE 5: Inserts: 7 | Swaps: 835 | TreeChains: 10 | Shakes: 9

>>> Tiempo de ejecución: 3029.84 seg
```

Bibliografía

Baker, B. M. y C. Benn (2001). «Assigning pupils to tutor groups in a comprehensive school». En: *Journal of the Operational Research Society* 62. URL: http://www.palgrave-journals.com/jors/journal/v52/n6/abs/2601135a.html.

Bhadurya, Joyendu, E. Joy Mightyb y Hario Damar (2000). «Maximizing workforce diversity in project teams: a network flow approach». En: *Omega* 28. URL: http://www.sciencedirect.com/science/article/pii/S0305048399000377.

Brimberg, Jack, Nenad Mladenovic y Dragan Uroševic (2015). «Solving the maximally diverse grouping problem by skewed general variable neighborhood search». En: *Information Sciences* 295. URL: http://www.doiserbia.nb.rs/Article.aspx?ID=0354-02431300003U&AspxAutoDetectCookieSupport=1#.VsuILlvhBD8.

Bárbara Scandroglio, Jorge S. López Martínez y Mª Carmen San José Sebastián. «Self Verification Theory». En: *Information Sciences* 0. URL: http://homepage.psy.utexas.edu/homepage/faculty/swann/docu/svt%20lange%20et%20al.pdf.

– (2008). «La Teoría de la Identidad Social: una síntesis crítica de sus fundamentos, evidencias y controversias». En: *Information Sciences* 0. URL: http://www.psicothema.com/pdf/3432.pdf.

Desrosiers, J., N. Mladenović y D. Villeneuve (2005). «Design of balanced MBA student teams». En: *Journal of the Operational Research Society* 56. URL: http://www.palgrave-journals.com/jors/journal/v56/n1/full/2601775a.html.

Dragan, Uroševic (2014). «Variable neighborhood search for maximum diverse grouping problem». En: *Yugoslav Journal of Operations Research* 24. URL: http://www.doiserbia.nb.rs/Article.aspx?ID=0354-02431300003U&AspxAutoDetectCookieSupport=1#.VsuILlvhBD8.

Fan, Z. P. y col. (2011). «A hybrid genetic algorithmic approach to the maximally diverse grouping problem». En: *Journal of the Operational Research Society* 62. URL: http://www.palgrave-journals.com/jors/journal/v62/n1/full/jors2009168a.html.

Feo, Thomas A. y Mallek Khellaf (1990). «A class of bounded approximation algorithms for graph partitioning». En: *Networks* 20. URL: http://onlinelibrary.wiley.com/doi/10.1002/net.3230200205/abstract?systemMessage=Wiley+Online+Library+will+be+unavailable+on+Saturday+27th+February+from+09:00-14:00+GMT+/+04:00-09:00+EST+/+17:00-22:00+SGT+for+essential+maintenance.++Apologies+for+the+inconvenience..

Pier Hansen, Nenad Mladenovic (1999). «Variable neighborhood search:Principles and applications». En: *Information Sciences* 0. URL: https://www.google.com.uy/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0ahUKEwiv5ujUpNrMAhUBkZAKHZVwDBYQFgg7MAM&url=

https%3A%2F%2Fwww.researchgate.net%2Ffile.PostFileLoader.html%3Fid%3D54674dbcd3df3ef6218b4685%26assetKey%3DAS%253A273635015626763%25401442251079922&usg=AFQjCNH-hjFA_VfS.

RESENDE, MAURICIO G.C. y CELSO C. RIBEIRO (0). «Greedy Randomized Adaptive Search Procedures». En: *Information Sciences* 0. URL: http://www.optimization-online.org/DB_FILE/2001/09/371.pdf.

Rodriguez, F. J. y col. (2013). «An artificial bee colony algorithm for the maximally diverse grouping problem». En: *Information Sciences* 230. URL: http://www.sciencedirect.com/science/article/pii/S0020025513000121.

Triska, Markus (2008). «Solution Methods for the Social Golfer Problem». En: *Information Sciences* 295. URL: http://www.metalevel.at/mst.pdf.