

Algoritmos Evolutivos

Informe final

Andrés Kent
Facultad de Ingeniería
Universidad de la República
CI: 4.762.138-0
Email: andreskent7@gmail.com

Daniel Susviela
Facultad de Ingeniería
Universidad de la República
CI: 4.894.156-5
Email: d.susviela@gmail.com

Resumen—El objeto de estudio del siguiente informe es el empleo de la técnica de algoritmos evolutivos en el problema de asignación horaria de las clases de una universidad, teniendo en cuenta salones con capacidad limitada, variedad de orientaciones y de cursos. Se busca la mejor configuración paramétrica para el algoritmo, para luego contrastar en una instancia con datos reales el desempeño de éste contra un algoritmo tradicional.

1. Descripción del problema

Los centros académicos siempre enfrentan al comienzo de cada ciclo académico al problema de asignar horarios de clase que se dictan. Teniendo en cuenta la disponibilidad horaria de cada salón, sus capacidades, y a la vez intentar dar la mayor cantidad de opciones a los alumnos el problema se vuelve rápidamente complejo. Además de estas variables, la mayoría de los cursos tienen múltiples clases por semana, y varios profesores que dictan el mismo curso. La propuesta de este equipo es la resolución de este problema con las siguientes consideraciones.

Para definir el problema de forma más concreta se necesita la definición de ciertos conceptos:

- **Salón:** Espacio físico en el cual se desarrolla una clase.
- **Capacidad de un salón:** Cantidad de alumnos que entran en un salón.
- **Clase:** Sesión en la que se imparte una lección sobre un curso.
- **Curso:** Conjunto de clases sobre una materia que está estructurada y sigue un programa, especialmente dentro de un plan de estudios. En el presente informe se referencia este concepto también como materia.
- **Orientación:** Una carrera académica que se define como el conjunto de estudios, repartidos en cursos, que capacitan para ejercer una profesión.
- **Año de un curso:** Año en el cual se dicta un curso en la currícula sugerida de una orientación.

Por otra parte para atacar un problema así de complejo se utilizan las siguientes abstracciones de la realidad:

- Se asume que los salones están a priori disponibles para su uso por las clases todo el día.

- Un día se divide en turnos. En esta realidad se manejan tres turnos por día, mañana, tarde y noche. A su vez entran dos clases por turno. Por lo tanto, en un día no pueden haber más de seis clases en el mismo salón.
- Los cursos a lo sumo dan dos clases por semana por tipo de clase. Dado un curso que exige esta carga horaria semanal, las dos clases que se dan en la misma semana se les denomina par.
- Por el punto anterior se tienen las siguientes 4 categorizaciones para las clases.
 - **Teórica:** Siempre vienen de a pares en una semana. En una clase teórica se imparten conocimientos teóricos sobre el objeto de estudio.
 - **Práctica:** Siempre vienen de a pares en una semana. En una clase practica se imparten conocimientos de cómo aplicar la teoría en la realidad del objeto de estudio.
 - **Teórica simple:** Ídem a la teórica con la diferencia que esta clase no tiene un par en una semana.
 - **Práctica simple:** Ídem a la practica con la diferencia que esta clase no tiene un par en una semana.
- Los cursos pueden ofrecer más de una opción por clase y puede tener clases de todos los tipos en una semana. Por ejemplo; un curso puede ofrecer 5 opciones de pares de clases teóricas en una semana, 3 opciones de pares de clases prácticas y una clase práctica simple (que puede ser asociada en la realidad a una clase de consulta).
- Los alumnos se distribuyen de forma uniforme entre todas las opciones de clases de un tipo para un curso.
- La cantidad de alumnos total que cursan un curso es inversamente proporcional al año del mismo.

Con todo esto el problema es dar la mejor asignación de los horarios en una semana de todas las clases correspondientes a todos los cursos. Una asignación de horarios es mejor que otra si distribuye todas las clases de un curso de forma uniforme entre los turnos de los días y además minimiza la superposición de horarios entre cursos de la misma orientación.

Por otro lado se tienen las siguientes restricciones:

- Las clases que tienen un par deben estar ambas en el mismo turno.
- Las clases que tienen un par deben estar separadas por más de un día entre ellas.
- No puede haber clases asignadas a salones en los cuales no tienen capacidad suficiente para albergar la proporción de estudiantes que asisten a dicha clase¹.

2. Justificación del uso de AE

Los problemas de construcción de cronogramas con diversas restricciones han sido probados de ser en su mayoría NP-Complejos [1], por lo que realizar una búsqueda exhaustiva de la solución óptima al problema en un tiempo razonable no es posible. El problema en cierta forma es análogo al de un grafo, donde cada vértice representa un curso y las aristas unen cursos que no pueden ser asignados a la misma hora, siendo el objetivo encontrar el número cromático del grafo [2]. Es sabido que el problema de coloración de grafos es NP-Completo, y de ahí se desprende que el problema de asignación de horarios también lo es. Por esta razón surge el interés de la utilización de algoritmos evolutivos, los cuales han demostrado ser una alternativa más que viable para encontrar buenas soluciones a este tipo de problemas.

3. Formulación matemática

Sea el conjunto de los días de la semana $D = \{\text{lunes, martes, miércoles, jueves, viernes}\}$, sea $T = \{\text{mañana, tarde, noche}\}$ el conjunto de turnos en los cuales se da clase y sea M el conjunto de materias. Por otro lado sea O el conjunto de orientaciones, sea C el conjunto de clases y sea S el conjunto de salones.

Para cada elemento $m \in M$ existe al menos un elemento $c \in C$ relacionados por la función *pertenece_curso* : $C \rightarrow M$. Por otro lado las orientaciones O también se relacionan con los elementos del conjunto M por la función *pertenece_orientación* : $M \rightarrow O$. Sea también la función *cantidad_opciones* : $C \rightarrow \mathbb{N}$ que devuelve para una clase la cantidad de alternativas para esa clase. Sea la función *año_curso* : $C \rightarrow 1, \dots, N$ la cual retorna el año de un curso en la currícula sugerida de una orientación. Finalmente sea el conjunto S de salones, la función *capacidad* : $S \rightarrow \mathbb{N}$ que devuelve la capacidad de un salón y el parámetro $A[O]$ que representan los alumnos que cursan la orientación O .

El problema se puede reducir a optimizar la función objetivo *se_dicta* : $C \rightarrow T \times D \times S$ que para una clase le asigna un turno, un día y un salón. La optimización de dicha función viene dada por:

$$\min : v. \sum_{m \in M} \phi(m) + u. \sum_{c_1, c_2 \in C, c_1 \neq c_2} \psi(c_1, c_2) \quad (1)$$

Sujeto a las siguientes restricciones:

1. Se entra en detalle sobre esto en la formulación matemática del problema

- $(\exists c_1, c_2, \in C)$
 $(c_1 \neq c_2 \wedge se_dicta(c_1) = se_dicta(c_2) \rightarrow$
 $\nexists c_3 \in C, c_3 \neq c_1 \wedge c_3 \neq c_2 \wedge se_dicta(c_3) = se_dicta(c_1))$
 Se dictan a lo sumo dos clases por turno por día en un mismo salón.
- $\forall c \in C, \exists t \in T, d \in D, s \in S / se_dicta(c) = (t, d, s)$
 La función *se_dicta* debe tener una imagen para todo elemento del Dominio
- $\forall c \in C, se_dicta(c) = (_, _, s) \rightarrow$
 $\sum_{a \in A[(p_m(p_c(c)))]} ac_o(c)^{-1} \lambda \leq capacidad(s)$
 $\lambda \in \mathbb{R}$

La función *se_dicta* asigna salones que son suficientemente grandes para alojar al menos una porción de los alumnos que cursan la asignatura a las clases. Se utiliza la abreviación *c_o* para la función *cantidad_opciones* y las abreviaciones *p_m*, y *p_c* para las funciones *pertenece_materia* y *pertenece_curso*. Con λ siendo el factor de decadencia de estudiantes² por año multiplicado por un factor estimado de 0,8 de proporción de asistencia a clase.

Donde los parámetros u y v ponderan la medida de interés entre ambas componentes de la función objetivo. Por otro lado ϕ es una función que para un tipo de clase fijo mide la distribución de las clases de ese tipo a lo largo de los turnos de una semana. Para distribuciones mas uniformes retorna valores mas bajos y para distribuciones mas dispares retorna valores altos. Esta definida de la siguiente forma:

$$\phi(m) = \sum_{t \in \text{tipo}} maxClases(m, t) - minClases(m, t)$$

Donde la función *maxClases* retorna el máximo de ocurrencias de las clases del tipo t en un mismo turno y *minClases* es análoga pero retorna el mínimo de ocurrencias. Esta función se desea minimizar ya que las soluciones que tengan valores bajos de esta componente tienen una distribución de los turnos de forma uniforme, lo cual es deseable así se presentan mas opciones para los alumnos, los cuales tienen sus propios horarios también.

Además se tiene ψ que es una función que mide la superposición de clases de una misma orientación en el mismo turno para el mismo día. Formalmente queda dada por:

$$\psi(c_1, c_2) = \begin{cases} \sum_{o \in O(c_1)} A[o] : Log(c_1, c_2) \\ 0 & : \text{en otro caso} \end{cases}$$

Donde la función cumple que *val(Log)* : $C \times C \rightarrow \{0, 1\}$ y esta dada por:

$$Log(c_1, c_2) = p_o(p_c(c_1)) = p_o(p_c(c_2)) \\ \wedge (\exists t \in T, d \in D)(s_d(c_1) = (t, d, _) \wedge s_d(c_2) = (t, d, _)) \\ \wedge (\exists c_3 \in C)((\exists t \in T, d \in D)(s_d(c_1) = (t, d, _) \\ \wedge s_d(c_3) = (t, d, _)))$$

2. El factor de decadencia por año fue estimado en base a las actas de exámenes de las materias específicas de cada carrera

Se utiliza la notación $O(c)$ para denotar el conjunto de orientaciones de la clase c .

Luego de varias simulaciones se eligió el valor de u y de v para llevar los valores de los resultados de las dos funciones al mismo orden de magnitud. Luego se tomo un múltiplo de cada uno de ellos en base a preferencia. Finalmente con el fin de lograr una mejor visualización de los números se los llevo al rango de las centenas, por lo que se dividió entre cien ambos. Por tanto los valores quedaron dados por $u = 2,0 \times 10^{-6}$ y $v = 1,0 \times 10^{-2}$.

Por otra para poder lidiar con la restricción de que dos pares no pueden estar en días consecutivos, se utiliza la estrategia de penalización al fitness. Al contrario de las otras restricciones se permite en cierto grado esta restricción ya que da lugar a exploración en el espacio de soluciones y las soluciones que violan esta restricción no son imposibles de implementar en la realidad. Esto no es el caso con las otras restricciones, una clase no puede estar ubicada en un salón que tiene una capacidad de por ejemplo diez veces menor y por otro lado no se pueden dar dos clases en simultáneo en el mismo salón (restricción de dos clases por turno). En las soluciones donde ocurra esto, se aumenta el fitness en un valor fijo el cual fue elegido en base a resultados empíricos. De todas formas, en caso de que dos pares estén en el mismo día, esto sí será considerado no factible, por lo que no se permitirá formar una solución con esa característica.

4. Representación de una solución

Para modelar una solución se construye una matriz donde a cada salón le corresponden tres columnas, una por cada turno, mientras que a cada día de la semana le corresponden dos filas, una por cada clase a desarrollarse ese día, en ese salón durante el turno correspondiente. A cada entrada de la matriz se le asigna una clase que se dictará en ese salón, ese día durante ese turno.

La representación de los pares se hace también en esta matriz. Por cada celda de las anteriores se agrega una debajo la cual contiene el índice de la entrada de la matriz correspondiente al par asociado.

En resumen se tiene una matriz con 20 filas, y tres veces la cantidad de salones en cantidad de columnas. La mitad horizontal superior de la matriz se utiliza para denotar la asignación semanal de los cursos en las combinaciones *salon - turno - dia* y la mitad de abajo se utiliza para mantener la información del par. En caso de ser una clase sin par (teórico simple o práctico simple), el índice se apunta a la clase misma. Un ejemplo visual de esto se tiene en la Figura 1 de los anexos.

5. Estrategia de resolución

5.1. Operadores

Debido a la complejidad de la representación, es necesario definir los operadores de mutación y de cruzamiento de forma específica para este problema. Estos deben asegurar

soluciones factibles como resultado y además deben mantener la información de los pares de forma consistente. Se opta por esta estrategia para resolver la problemática de las soluciones no factibles ya que la corrección de soluciones que violan las restricciones a la hora de evaluar las soluciones además de ser altamente difícil de implementar tiene un costo computacional muy elevado. Por otra parte el operador de selección que se utiliza es el operador de selección por torneo binario común.

5.1.1. Mutación. El operador de mutación esta definido de la siguiente forma:

1. De izquierda a derecha y de arriba hacia abajo, se toma un gen (que corresponde a una clase de un curso que se dicta un día, en determinado turno y determinado salón) que corresponda a la mitad superior horizontal de la solución (es decir, la mitad que contiene las clases, NO la mitad que contiene la referencia a los pares) y se sortea si el gen debe mutar según una función de distribución uniforme. Todos los genes pueden ser elegibles con idéntica probabilidad.
2. Se elige la mutación que se le aplicará al gen. Una mutación puede ser cualquiera de las siguientes: $G = \{\text{cambio_día}, \text{cambio_turno}, \text{cambio_salón}\}$
3. Se elige otro gen víctima junto con el cual se aplicará la mutación. La elección de estos genes víctima depende del tipo de mutación que se este aplicado. Por otro lado, se debe asegurar que el gen víctima entre en el salón en el que se encuentra el gen original y viceversa, lo cual se calcula en función de la capacidad del salón, la cantidad de alumnos anotados al curso y la cantidad de alternativas para esa clase. En caso de que la mutación sea imposible ya que no existen genes víctima para ese gen original, con el fin de mantener la factibilidad, esta se aborta y no se aplica para ese gen.

Más formalmente; Sea $p_0 \in (0, 1)$, $p_0 \in \mathbb{R}$, $G = \{\text{cambio_día}, \text{cambio_turno}, \text{cambio_salón}\}$ y sean las variables aleatorias $X_0, X_1, X_2 \sim U[0, 1]$. Sean además los rangos $R = [\frac{i-1}{3}, \frac{i}{3}] \forall i \in [1, \dots, 3]$ y la función $map : G \rightarrow R$ una función biyectiva que para cada tipo de mutación asigna un rango. Se dice entonces que si un gen que puede mutar, lo hace con probabilidad $P(X_0 \leq p_0)$. De esto se desprende que la probabilidad de una mutación en particular sobre un gen que puede mutar es $P(\text{mutación}) = P(X_0 \leq p_0)P(X_1 \in \text{map}(\text{mutación}))$. Asimismo sean los rangos $[\frac{i-1}{n}, \frac{i}{n}] \forall i \in [1, \dots, n]$ con n igual a la cantidad de genes víctima, la función $mapV$ que asigna un rango a un gen víctima, si este es utilizado para mutarse con probabilidad $P(X_2 \in \text{mapV}(\text{gen}))$.

Las mutaciones de G están definidas de la siguiente forma:

- **cambio_día:** Un gen intercambia el día con otro gen dentro del mismo salón, y dentro del mismo turno. Se debe procurar que el segundo gen no comparta día con el par del primer gen ni viceversa.
- **cambio_turno:** Un gen y su par intercambian su turno con otro gen y su par dentro del mismo día, y el mismo

salón. Si uno de los dos genes no tiene par y el otro sí, el par se ubica en el mismo salón y día que estaba, cambiando únicamente de turno al turno en que se ubicó su compañero. Es importante notar que este es el único tipo de mutación donde se intercambian también los pares, debido a que por restricción los pares siempre deben coincidir en turno.

- **cambio_salón:** Un gen intercambia su salón con otro gen dentro del mismo día y el mismo turno. Se debe procurar, como se mencionó anteriormente, que los salones puedan acomodar los estudiantes asociados a los genes mutados.

En la Figura 2 al final del informe se muestra un ejemplo de cada tipo de mutación.

5.1.2. Cruzamiento. El operador de cruzamiento está definido de la siguiente forma:

1. Sean dos padres que se toman como entrada. Se le asigna a uno el rol de “madre” y a otro el de “padre”.
2. Se sortea el tipo de cruzamiento a realizar con una variable aleatoria de distribución uniforme tal como se hace en la mutación. Un tipo de cruzamiento puede ser uno de los siguientes $Cr = \{cruzamiento_dia, cruzamiento_turno, cruzamiento_salon\}$
3. Se genera el hijo a partir del cruzamiento sorteado
4. Se intercambian los roles de madre y padre
5. Se repiten los pasos 2 y 3 para la creación del segundo hijo

La importancia de los roles de madre y padre se detalla en la explicación de los tipos de cruzamiento:

- **cruzamiento_día:** Se sortea al azar un día a ser cruzado. Se crea un individuo, el cual se llamará hijo, de forma tal que es una copia idéntica de la madre. De este hijo se eliminan todas las celdas del día a cruzar. También se eliminan todos los pares de dichas celdas. Luego al hijo se le insertan todas las clases de ese día del padre junto con sus pares. El caso en el que un par se intente insertar en una celda ocupada se llama colisión. La resolución de las colisiones está detallada mas adelante. Luego de realizada la inserción, se eliminan las clases sobrantes y se agregan las faltantes en cualquier lugar disponible factible. Los dos resultados son nuevos individuos. En el raro caso de que una colisión no pueda ser resuelta el cruzamiento se aborta y se retorna como hijo una copia de la madre. En la Figura 3 se presenta un ejemplo de este cruzamiento.
- **cruzamiento_turno:** Es análogo al cruzamiento anterior, solo que se sortea un turno del padre a ser insertado en el hijo. El turno elegido se inserta en su totalidad, para todos los días y salones. Nuevamente en el raro caso de que no se pueda resolver la colisión esta se aborta y se retorna como hijo una copia de la madre.
- **cruzamiento_salón:** Análogo a los casos anteriores, pero se inserta en este caso un salón entero, para todos los días y turnos. Nuevamente en el raro caso de que no se pueda resolver la colisión esta se aborta y se retorna como hijo una copia de la madre.

5.2. Resolución de colisiones

En caso de colisiones al intercambiar clases entre individuos, se resuelven de la siguiente forma:

1. En primer lugar, se busca otro salón donde se pueda dictar la clase, manteniendo el turno y el día. Este salón debe poder albergar a los estudiantes de la clase que se quiera ubicar.
2. Si no se encuentra otro salón, se mantiene el salón original pero se elige otro día, manteniendo el turno. Este nuevo día no puede coincidir con el día de su compañero.
3. En caso de aún no encontrar una ubicación para el nuevo par, se elige otro salón y otro día de la semana para ubicarlo, manteniendo siempre el turno.

5.2.1. Selección. La selección de los individuos se hace mediante una selección por torneo binaria.

5.3. Población inicial

La generación de individuos se hace mediante el siguiente algoritmo.

1. Se toma una clase y se evalúan las opciones factibles donde esta clase puede ser ubicada. En caso de tener un par también se evalúan sus opciones.
2. Se toma una de ellas al azar y se inserta en la solución
3. En caso de no tener opciones de celdas libres, se intenta mover una de las clases ya asignadas con su par a otro salón.
4. En el raro caso de no poder asignar la clase en ningún lado se aborta la solución y se comienza de nuevo.
5. Repetir los pasos 1 y 2 con los pasos alternativos 3 y 4 cuando sean pertinentes hasta tener el individuo completo

5.4. Aclaración sobre los abortos

Durante la descripción de los operadores como de la creación de la población inicial, se tomaron varias precauciones para que los abortos ocurran lo menos posible. Según datos obtenidos en las ejecuciones de los algoritmos, la tasa de abortos en todos los casos no supera el 1% por lo que su inclusión no solo no afecta el desempeño de manera significativa sino que además mantienen la consistencia de los individuos en el sistema.

5.4.1. Implementación y arquitectura. Para la implementación de estos operadores se utilizó la biblioteca de jMetal en su versión 6. En cuanto la arquitectura se precisa un módulo por operador, un módulo del problema que implementa la función de fitness y un módulo que corra el problema, que tiene la configuración paramétrica del problema. Para el manejo de las instancias del problema y de la estructura de las soluciones se implementa una clase de ayuda la cual maneja las estructuras de las soluciones. Con el fin de asegurar la buena programación de esta clase

la cual es vital, y de los operadores, se implementa una clase que prueba casos de prueba. Debido a que jMetal no ofrece una representación matricial de soluciones, se usó una representación de vector de enteros, formado por la concatenación de las columnas de la matriz.

6. Evaluación experimental

6.1. Obtención de los datos de prueba

Para probar el algoritmo se evaluó el mismo con datos de la realidad, en particular con las asignaturas y los salones de la Facultad de Ingeniería de la Universidad de la República. Debido a la alta complejidad del problema, para mantener los tiempos de ejecución en tiempos razonables, se toman los cursos de primer semestre de primer a tercer año de todas las orientaciones que ofrece la facultad. Se obtuvieron los datos de capacidades de salones así como las cantidades de alumnos por orientación en el año 2017 [3] [4]. Las cantidades de clases y su naturaleza se obtuvieron a partir del cronograma de horarios del 2017 y de los sitios de EVA de cada curso y de la misma forma se obtuvieron los años correspondientes en las currículas sugeridas de cada una de las orientaciones [6] [5]. Los datos obtenidos se adjuntan en los anexos del presente informe.

Con el fin de analizar el algoritmo frente al problema real también se variaron las mismas variables.

6.2. Generación de instancias

6.2.1. Instancias de la configuración paramétrica.

Partiendo de los datos obtenidos, se construyó una instancia reducida del problema, la cual cuenta con tan solo 4 orientaciones y 20 salones. Esta instancia reducida luego fue sometida a un proceso de variación, donde a cada salón se le sumó un porcentaje de su capacidad y a cada carrera se le restó una porción de su cantidad de alumnos inscriptos. A partir de este proceso se obtuvieron entonces 3 instancias reducidas del problema las cuales fueron empleadas en la configuración paramétrica.

En detalle las carreras que utilizan son:

- Agrimensura
- Ingeniería Civil
- Ingeniería en Producción
- Ingeniería Eléctrica

Por otra parte se descartan de la realidad los siguientes salones:

- Salón: UDELAR A: 21
- Salón: UDELAR B: 21
- Salón: UDELAR C: 21
- Salón: UDELAR D: 21
- Salón: BIBLIOTECA: 10
- Salón: 312: 42
- Salón: 314: 44
- Salón: 315: 30
- Salón: 401: 45
- Salón: 402: 36

- Salón: SW: 16

Por otro lado las cantidades fueron alteradas de la siguiente forma para las instancias:

- Instancia 1: Datos reales sin modificación
- Instancia 2: -10% de alumnos por carrera, $+10\%$ de capacidad en los salones
- Instancia 3: -20% de alumnos por carrera, $+20\%$ de capacidad en los salones

6.2.2. Instancias para la evaluación del algoritmo. Se generan tres instancias mediante las mismas transformaciones de la sección anterior aplicadas en datos de la realidad, los cuales se encuentran en los anexos del presente informe.

6.3. Variaciones paramétricas

Los operadores definidos en la sección 5 toman varios parámetros configurables. La toma de distintos valores impacta fuertemente en el algoritmo, por esto mismo para determinar cual de las configuraciones paramétricas es mejor se realizara un estudio estadístico sobre el desempeño de las variaciones en las tres instancias del problema definidas anteriormente.

Se estudian entonces las siguientes probabilidades de cruzamiento:

- 0.25
- 0.5
- 0.75

Por otro lado las probabilidades de mutación a analizar son:

- 0.01
- 0.005
- 0.001

Se ejecuta cada combinación unas treinta veces sobre las distintas instancias del problema estipuladas en la sección anterior. Con esto para cada instancia se obtiene la función de distribución de su fitness y aplicando un estudio de test de Friedman se puede determinar si hay una distribución que es mejor. Para hallar luego cual es, se utiliza un test post-hoc de Dunn con corrección de Bonferroni. La configuración que tenga mejor puntaje entre las tres instancias sera tomada como la mejor. Con el fin de simplificar la notación se numeraran las distintas configuraciones paramétricas:

1. $P(\text{mutación}) = 0.001$; $P(\text{cruzamiento}) = 0.25$
2. $P(\text{mutación}) = 0.001$; $P(\text{cruzamiento}) = 0.5$
3. $P(\text{mutación}) = 0.001$; $P(\text{cruzamiento}) = 0.75$
4. $P(\text{mutación}) = 0.005$; $P(\text{cruzamiento}) = 0.25$
5. $P(\text{mutación}) = 0.005$; $P(\text{cruzamiento}) = 0.5$
6. $P(\text{mutación}) = 0.005$; $P(\text{cruzamiento}) = 0.75$
7. $P(\text{mutación}) = 0.01$; $P(\text{cruzamiento}) = 0.25$
8. $P(\text{mutación}) = 0.01$; $P(\text{cruzamiento}) = 0.5$
9. $P(\text{mutación}) = 0.01$; $P(\text{cruzamiento}) = 0.75$

6.3.1. Análisis estadístico para la primer instancia reducida. Se define entonces el test de Hipótesis para el test de Friedman, donde la hipótesis nula h_0 es que no hay diferencia significativa entre los datos para asegurar que una

distribución se comporta de manera diferente a las demás. Se toma un valor de $\alpha = 5\%$ por lo que si el factor obtenido es menor que este umbral se rechaza dicha hipótesis.

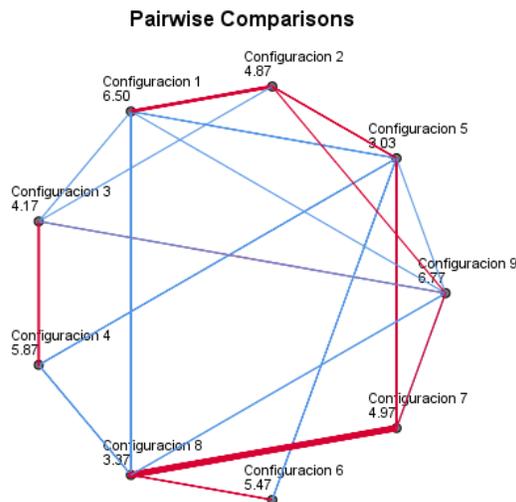
El resultado para el test de Friedman, para la configuración paramétrica de la primera instancia fueron los siguientes:

Resultado del p-valor.

$$p - \text{valor} = 5.8896 \times 10^{-9} \quad (2)$$

Se está en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn en conjunto con la corrección de Bonferroni.

Se presenta entonces el gráfico de comparaciones dos a dos.



Each node shows the sample number of successes.

Gráfico 1: Comparación dos a dos de la primera instancia.

En el gráfico se pueden apreciar las configuraciones con su ranking promedio. En azul se presentan las aristas para las cuales la diferencia estadística es tal que permite afirmar que las distribuciones son diferentes, mientras que las rojas muestran que las distribuciones no presentan ningún indicio de ser diferentes.

De este gráfico se puede afirmar que las configuraciones 1, 4, 6, 9 no pueden ser las óptimas para esta instancia ya que la configuración 5 tiene diferencias significativas con estas y es la que rankea mejor en promedio, por lo que su distribución es mejor³.

3. Las gráficas de distribución de rankings se encuentran en los anexos del informe

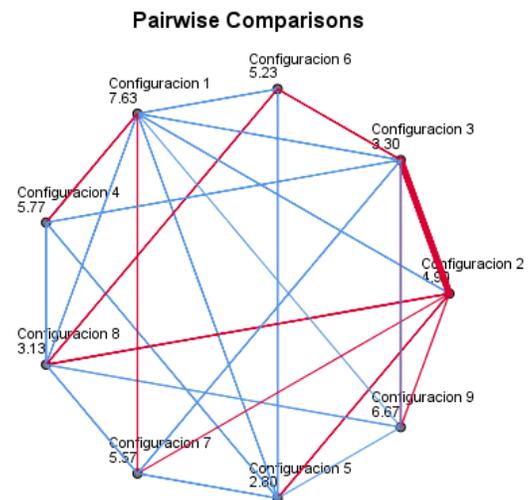
Observando la tabla asociada a los datos⁴ se aprecia que efectivamente no hay más configuraciones que puedan ser descartadas, por lo que se procede a la siguiente instancia.

6.3.2. Análisis estadístico para la segunda instancia reducida. Los resultados obtenidos para el test de Friedman, para la configuración paramétrica de la segunda instancia fue el siguiente p-valor.

$$p - \text{valor} = 1.4264 \times 10^{-15} \quad (3)$$

Nuevamente se está en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn⁴. En las gráficas de promedio de ranqueo³ se puede observar una vez más que la configuración que mejor rankea es la 5.

Se presenta entonces el gráfico de comparaciones dos a dos.



Each node shows the sample number of successes.

Gráfico 2: Comparación dos a dos de la segunda instancia.

En este caso se pueden descartar además de las anteriores la configuración 7. Además se puede ver cierto alejamiento de las configuraciones 3 y 8 al ahora tener más diferencias con las otras configuraciones, esto se aprecia con el aumento de las aristas azules que parten de estos nodos.

Hasta el momento, las configuraciones candidatas a ser la óptima son: 2, 3, 5, 8. Se procede a la última instancia.

6.3.3. Análisis estadístico para la tercera instancia reducida. Finalmente los resultados obtenidos para el test de

4. La tabla entera de datos se encuentra en los anexos del informe

Friedman, para la configuración paramétrica de la tercera instancia es el siguiente:

$$p - \text{valor} = 2.1678 \times 10^{-11} \quad (4)$$

Nuevamente se esta en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn. Nuevamente la configuración que rankea mejor es la 5.³

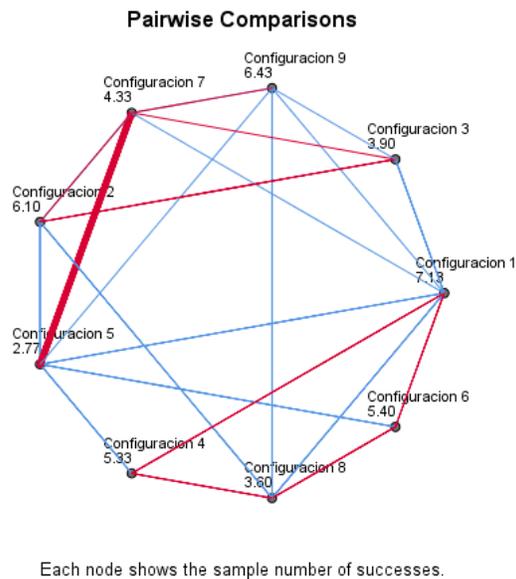


Gráfico 3: Comparación dos a dos de la tercera instancia.

A partir de este gráfico se puede entonces descartar la configuración 2 ya que se revela que se tiene una diferencia significativa con la 5 la cual como se menciono anteriormente fue la que rankeo mejor en general para esta tercera instancia. Por ende de las tres configuraciones restantes, se tomara la 5, ya que no solo fue la que rankeo primera de forma consistente, sino que es la que lleva a menos operaciones de las tres y por ende es la que debería ejecutar mas rápidamente.

6.3.4. Configuración del tamaño de la población. Ahora resta configurar el tamaño de población ideal. Se procede con una estrategia similar a la anterior, sobre tres instancias se prueban tres tamaños los cuales son 50, 100, 200.

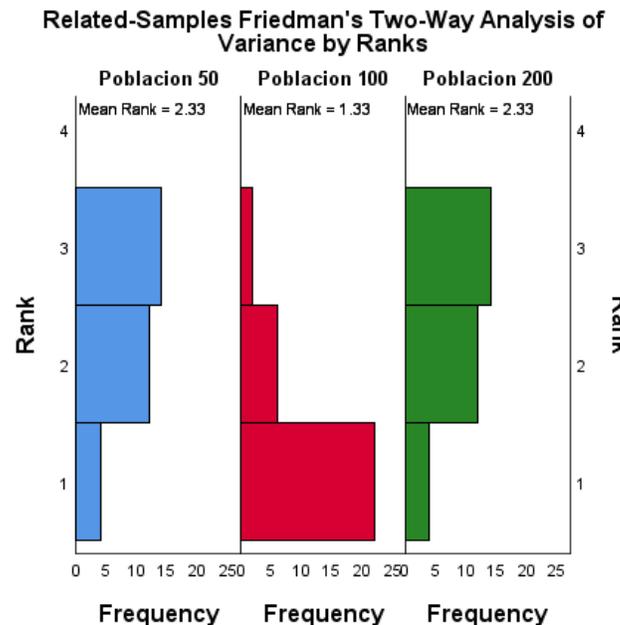
Para la instancia 1 se obtiene el siguiente resultado del p-valor.

$$p - \text{valor} = 0,000045 \quad (5)$$

Se está en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn y los resultados son los siguientes.

Muestra 1-	Comparaciones dos a dos				
	Test	Error	Test	Sig.	Bonferroni
Poblacion 100- Poblacion 50	1.0000000	0.2581989	3.8729833	0.0001075	0.0003225
Poblacion 100- Poblacion 200	-1.0000000	0.2581989	-3.8729833	0.0001075	0.0003225
Poblacion 50- Poblacion 200	0.0000000	0.2581989	0.0000000	1.0000000	1.0000000

De la tabla se puede apreciar que el tamaño de población de 100 individuos tiene una distribución bastante diferente a la de 50 y 200. A continuación la tabla de rankings.



Mirando la tabla de distribución de rankings podemos afirmar que es además la que mejor rankea de las tres por lo que en definitiva esta cantidad de individuos es la mejor para esta instancia.

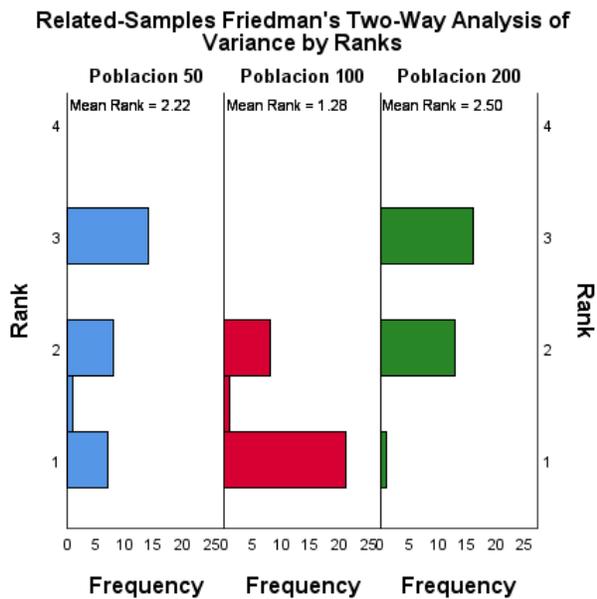
Se analiza ahora el caso para la instancia 2 y se obtiene el siguiente valor de p:

$$p - \text{valor} = 0,000005 \quad (6)$$

Se esta en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn. Los resultados se muestran a continuación:

Comparaciones dos a dos					
Muestra 1-	Test	Error	Test	Sig.	Bonferroni
Poblacion 100-	0.9333333	0.25819889	3.6147845	0.00030060	0.00090179
Poblacion 100-	-1.2166667	0.25819889	-4.7121297	0.00000245	0.00000735
Poblacion 50-	-0.2833333	0.25819889	-1.0973453	0.27249048	0.81747145

Una vez mas de la tabla se puede apreciar que el tamaño de población de 100 individuos tiene una distribución bastante diferente a la de 50 y 200. A continuación la tabla de rankings.



Por ultimo para la instancia 3 se obtiene entonces el siguiente resultado del p-valor después de aplicar el test de Friedman.

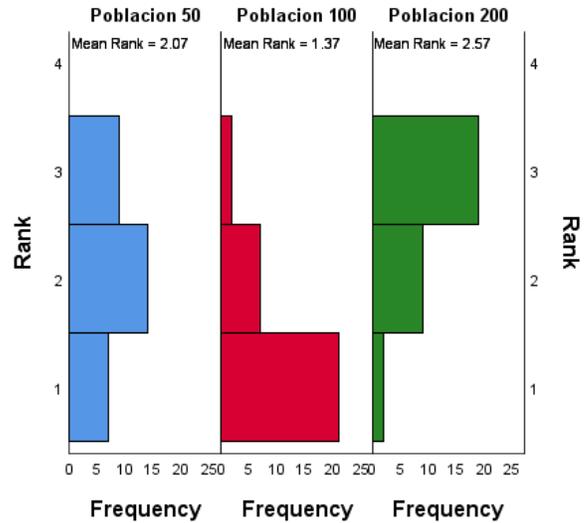
$$p - valor = 0,000018 \quad (7)$$

Se esta en condiciones entonces de rechazar la hipótesis nula y aceptar que para la primera instancia las configuraciones tienen distribuciones diferentes. Se aplica entonces el test post-hoc de Dunn cuyos resultados se aprecian en la siguiente tabla.

Comparaciones dos a dos						
Muestra 1-	Muestra 2	Test	Error standard	Test estandard	Sig.	Bonferroni
Poblacion 100-	Poblacion 50	0.7000000	0.25819889	2.7110883	0.00670628	0.02011883
Poblacion 100-	Poblacion 200	-1.2000000	0.25819889	-4.6475800	0.00000336	0.00001008
Poblacion 50-	Poblacion 200	-0.5000000	0.25819889	-1.9364917	0.05280751	0.15842253

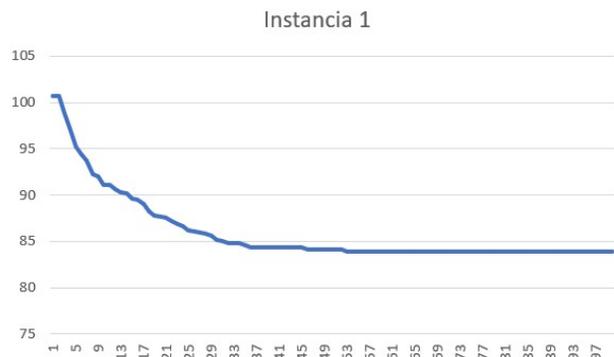
Por tercera vez consecutiva se tiene que la distribución que se distingue de las demás es justamente la de población 100. Mirando la gráfica de los rankings nuevamente se aprecia que esta es la mejor distribución.

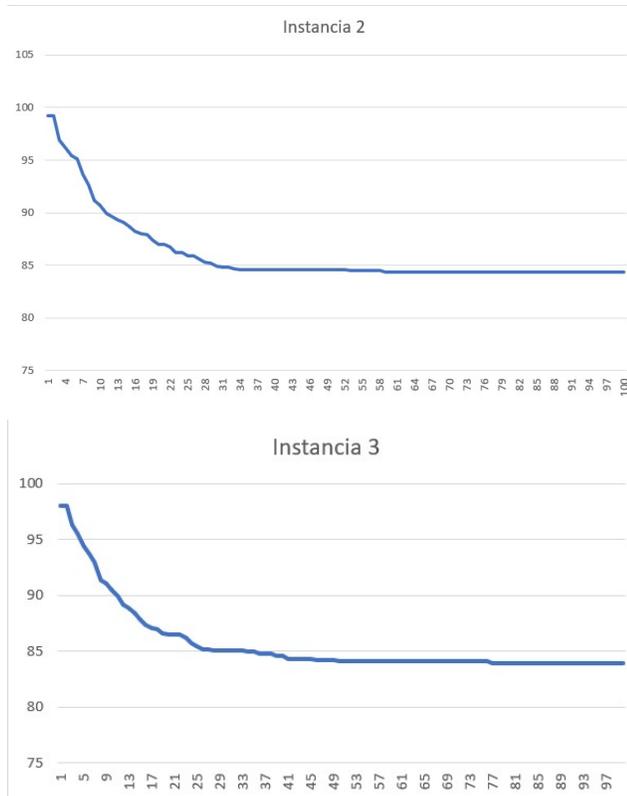
Related-Samples Friedman's Two-Way Analysis of Variance by Ranks



Luego como esta cantidad de individuos fue la que obtuvo mejores resultados de manera consistente en todas las pruebas se puede afirmar que es la óptima.

6.3.5. Optimización del numero de generaciones. En todas las pruebas paramétricas se ejecutó el algoritmo con una población de 100 individuos. Se procedió entonces a graficar la evolución del mejor individuo por generaciones para las tres instancias de prueba, con el objetivo de decidir si era conveniente aumentar o disminuir el número de generaciones. Se utilizó la configuración paramétrica obtenida anteriormente, es decir, 0,005 la probabilidad de mutación y 0,5 la de cruzamiento. Los resultados fueron los siguientes:





En las tres gráficas los resultados son consistentes, el algoritmo no encuentra soluciones considerablemente mejores en el tramo final de la ejecución, por lo que no es necesario aumentar el número de generaciones.

6.4. Comparación con otras técnicas o soluciones

Se plantea como línea base de comparación un algoritmo de Backtracking, el cual toma sus decisiones según un algoritmo Greedy. Es un algoritmo que para cada una de las clases y sus pares (si los hubiese) calcula todas las opciones donde pueden ir en la solución que va construyendo y toma siempre la opción que minimiza el fitness. En caso de que no tenga opciones realiza backtracking e intenta rehacer su decisión pasada con la siguiente mejor, terminando una vez que ubique todas las clases en el cronograma. La comparación se hace tanto a nivel de resultados como en tiempos de ejecución. Para que la comparación sea posible entre las dos metodologías se emplea el mismo hardware. Las especificaciones del mismo son:

- CPU: Intel Core i7-8700k (6 núcleos a 3.7MHz)
- RAM: 2x8 GB Kingston DDR4 2666MHz
- MOTHER: Gigabyte Z370 AORUS
- SO: Windows 10 Education 64 bits.

A continuación se muestran los resultados del algoritmo de Backtracking realizando sus decisiones entre las opciones de forma Greedy. Se muestran tanto tiempos de ejecución como valor de fitness para las tres instancias mencionadas anteriormente.

- Instancia 1: Fitness 369.466376. Tiempo de ejecución 09h34m16s

- Instancia 2: Fitness 367.474964. Tiempo de ejecución 11h54m37s
- Instancia 3: Fitness 364.671344. Tiempo de ejecución 17h19m41s

Por otro lado se corre el algoritmo evolutivo 30 veces para la configuración 5 para 100 individuos en las tres instancias. Los resultados obtenidos son:

- Instancia 1: Fitness promedio: 357.501908 Desviación estandar de: 1.645548023. Tiempo de ejecución promedio 0h17m45s
- Instancia 2: Fitness promedio: 356.450736. Desviación estandar de: 1.605275376. Tiempo de ejecución promedio 0h18m14s
- Instancia 3: Fitness promedio: 352.896576. Desviación estandar de: 1.700019622. Tiempo de ejecución promedio 0h18m37s

Se presenta un gráfico para asistir a la comparación visual de la comparación de fitness. En rojo se presentan los resultados del algoritmo evolutivo mientras que en azul los valores correspondientes al algoritmo de backtracking. El eje horizontal denota las distintas instancias.

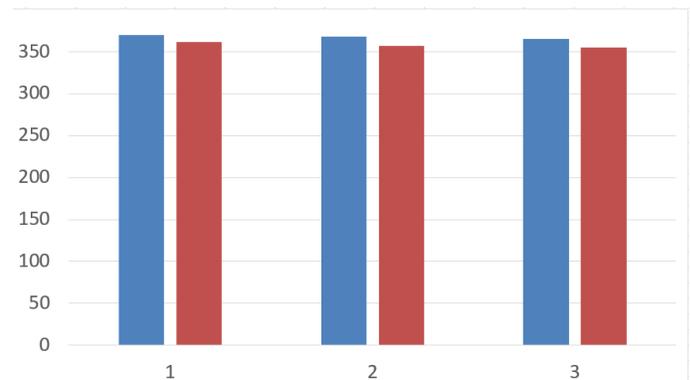


Gráfico 4: Comparación de fitness.

Como se puede apreciar en este gráfico el algoritmo evolutivo se desempeñó mejor que la línea base. Esta diferencia puede deberse a que el algoritmo evolutivo explora más el espacio de soluciones que el algoritmo Greedy, lo que le permite llegar a una mejor solución. Se realiza un test estadístico para determinar si esta diferencia es significativa como para poder afirmar que el algoritmo realmente es mejor. Para cada instancia se calcula la siguiente fórmula:

$$\max\{\sigma_A, \sigma_B\} \leq |f_{avg}(A) - f_{avg}(B)| \quad (8)$$

Por tanto se tienen las siguientes tres inecuaciones:

- $\max\{\sigma_A, \sigma_B\} = 1,645548023 \leq 11,964468$
- $\max\{\sigma_A, \sigma_B\} = 1,605275376 \leq 11,024228$
- $\max\{\sigma_A, \sigma_B\} = 1,700019622 \leq 11,774768$

Por lo que se puede decir que efectivamente el algoritmo evolutivo retorna soluciones con mejores valores de fitness.

Sin embargo la mayor ganancia del Algoritmo Evolutivo no es su calidad de solución, sino la velocidad con la que obtiene una solución mejor que su contra parte. La razón

por la cual el algoritmo de Backtracking demora tanto en ejecutar se puede deber al alto costo de evaluación de las soluciones intermedias. Por cada opción de clase, el algoritmo debe considerar todas las alternativas en la cual esa clase puede ir, lo cual en la mayoría casos llegan a ser cerca de las 6000 en la primera instancia. Habiendo un total de 44 cursos, con un promedio de 20 opciones, se estaría estimando cerca de las 5×10^6 de evaluaciones, asumiendo el mejor caso en el que el algoritmo no deshace sus pasos. El algoritmo evolutivo dada su naturaleza, solo precisa evaluar toda la población una vez por generación. Dado que se tiene una población de 100 individuos, en 100 generaciones el total de 100000 evaluaciones totales. La función de fitness implementada tiene un alto costo computacional debido a que la función ϕ la que requiere de mas computo ya que tiene orden aproximado de $O(\frac{n^3}{2})$ con n el tamaño de la matriz.

6.5. Independencia de las evaluaciones

Debido al uso de la clase JMetalRandom que ofrece jMetal para la generación de números pseudoaleatorios, la cual utiliza como generador una instancia de la clase JavaRandomGenerator, que toma como semilla del generador los milisegundos transcurridos desde el 1ro de enero de 1970 hasta hoy, se puede afirmar que las evaluaciones son independientes entre sí.

7. Conclusiones

7.1. Mejoras posibles

Al proyecto se le pueden realizar diversas mejoras, tanto en su planteamiento como en la calidad del código las cuales llevarían a no solo mejorías en la calidad de las soluciones, sino que también podrían abarcar casos todavía más realistas (como puede ser la inclusión de factores externos los cuales no se tuvieron en cuenta).

En cuanto al planteamiento el problema se presta para ser multiobjetivo y también considerar las medidas de interés de otros agentes como las horas puentes de los profesores. También se podría tener en cuenta la cercanía de las clases de un año procurando que los alumnos no pierdan tiempo cambiando de salón. Asimismo se podría penalizar el uso de ciertos salones ya que puede que en alguna realidad existan salones preferenciales o con horarios restringidos. Otra mejora a considerar sería la posibilidad de representar clases que se dictan de a tríos.

En cuanto al código se intentó reutilizar la mayor cantidad de funciones posibles para la implementación de esta tarea. Se incluye una clase de prueba de estas funciones, pero los mismos no son exhaustivos y muchos casos bordes poco comunes no se les realiza ninguna prueba. La utilidad de los tests es fundamental si se quiere la mantenibilidad del código y si se busca implementar las mejoras del planteamiento discutidas anteriormente. En cuanto al algoritmo de backtracking se podrían paralelizar muchas de sus instrucciones, en especial las de evaluación. De esta forma

el algoritmo seria mas rápido, y la diferencia no seria tan acentuada.

7.2. Pensamientos finales

De esta experiencia se puede afirmar que para el problema de asignación de horarios en la Facultad de Ingeniería con las leves modificaciones realizadas el algoritmo evolutivo se desempeñó de mejor forma según la función de fitness planteada por la linea base del algoritmo Greedy, por lo cual la técnica es buena. Además la gran diferencia en tiempos de ejecución permite decir que es un algoritmo más eficiente computacionalmente.

Por otra parte la mejor configuración paramétrica para las instancias reducidas es tener una probabilidad de mutacion del 0,005 y una probabilidad de cruzamiento igual a 0,5. El algoritmo respondió mejor a las instancias de prueba con una población de 100 individuos y 100 generaciones.

Referencias

- [1] T. B. Cooper and J. H. Kingston, *The Complexity of Timetable Construction Problems*, Lecture Notes in Computer Science, Springer Verlag, Vol.1153, (1996), pp.283–295.
- [2] A. Chaudhuri and K. De , *Fuzzy Genetic Heuristic for University Course Timetable Problem*, Birla Institute of Technology Mesra (2010), pp.3–4.
- [3] *Estadísticas Básicas 2017 de la Universidad de la República*, Dirección General de Planeamiento, Universidad de la República, Montevideo, Uruguay
- [4] *Autoevaluación de la Carrera de Ingeniería Eléctrica*, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.
- [5] *Carreras de Grado*, Sitio web de la Facultad de Ingeniería de la Universidad de la Republica.
<https://www.fing.edu.uy/ensenanza/carreras-de-grado>
Consultado el 4 de Noviembre del 2019.
- [6] *Portal Eva Facultad de Ingeniería*
<https://eva.fing.edu.uy/>
Consultado el 4 de Noviembre del 2019.

8. Anexos

8.1. Datos

Las carreras de la Facultad de Ingeniería según el año 2017 con sus inscriptos son:

0. Agrimensura: 39
1. Ingeniería Civil: 259
2. Ingeniería de Producción: 96
3. Ingeniería Eléctrica: 218
4. Ingeniería en Computación: 728
5. Ingeniería Industrial Mecánica: 242
6. Ingeniería Naval: 21
7. Lic. en Ingeniería Biológica: 28
8. Ingeniería de Alimentos ⁶: 102
9. Ingeniería en Sistemas de Comunicación⁵: 50
10. Lic. en Ciencias de la Atmósfera: 41
11. Ingeniería Química ⁶: 206
12. Tecnólogo Mecánico: 91

Se obvian las siguientes carreras:

- Ingeniería Forestal: Se dicta en Tacuarembó
- Lic. en Recursos Hídricos y Riego: Se dicta en Salto
- Tecnólogo Cárnico: Se dicta en Tacuarembó
- Tecnólogo en Cartografía: Carece de currícula sugerida.
- Tecnólogo Minero: Se dicta en Treinta y Tres
- Tecnólogo en Telecomunicaciones: Se dicta en Rocha
- Tecnólogo Informático: Carece de currícula sugerida.

Las clases obtenidas con sus carreras asociadas se presentan a continuación. Con el fin de simplificar la notación se presentan las orientaciones con su identificador de número utilizado anteriormente. Se dividen además por año. Primero:

- Cálculo 1: 0, 1, 2, 3, 4, 5, 6, 9, 10, 11
- GAL 1: 0, 1, 2, 3, 4, 5, 6, 9, 10 11
- Física 1: 0, 1, 2, 3, 4, 5, 6, 9, 10 11, 12
- Taller DCRG: 0, 1, 2, 5, 6, 10
- Tallere: 3, 9

Segundo:

- Cálculo 3: 0, 1, 3,5, 6, 9, 10 11
- Probabilidad y Estadística: 0, 1, 2, 3, 4, 5, 6, 9, 10 11
- Sistemas de Referencia: 0
- Ciencia, Tecnología y Sociedad: 0, 1, 2, 4
- Mecánica Newtoniana: 1, 2, 3, 5, 6, 10
- Física Experimental: 1, 2, 3, 5, 6, 11
- Principios de la Química General: 1, 2, 11
- Matemática Discreta 2: 4
- Lógica: 4
- Programación 2: 4, 9
- Introducción a la Ciencias de los Materiales: 2, 5, 6, 12

Tercero:

5. Carrera nueva que comenzó en el año 2018, sus datos fueron estimados en base a las actas de los exámenes de las materias específicas de la orientación

6. No se tomaron en cuenta los cursos que se imparten en la Facultad de Química

- Teoría de Errores 2: 0
- Agrimensura legal: 0
- Fotointerpretación: 0
- Topografía: 0
- Materiales y Ensayos: 1
- Elasticidad: 1
- IIO: 1, 2, 4
- Gestión de Calidad: 2
- AGPI: 2, 4
- Comportamiento Mecánico de los Materiales: 2, 5, 6, 12
- Funciones de Variable Compleja: 3, 9
- Señales y Sistemas: 3, 9
- Introducción a los Microprocesadores: 3
- Programación para Ingenieros Eléctricos: 3
- Electrónica Fundamental: 3, 9
- Sistemas Operativos: 4
- Programación 4: 4
- Teoría de Lenguajes: 4
- Elementos de la Mecánica de Fluidos: 5
- Metalurgia de la Transformación: 6
- Termodinámica Aplicada: 7, 8, 11
- Fenómenos de Transporte: 7, 8, 11
- Matemática Discreta 1: 9
- Antenas y Propagación: 9
- Elementos de la Meteorología y el Clima: 10

Los siguientes cursos se denotan como especiales ya que pueden ser cursados tanto en segundo como en tercero en las orientaciones que lo precisan.

- Física 2: 0, 1, 3, 5, 6, 10, 12
- Física 3: 0, 1, 3, 5, 6, 9, 10 11

Por otra parte se listan las cantidades de opciones por tipo de clase de la Facultad de Ingeniería, datos obtenidos de la asignación de horarios del primer semestre de 2017⁷. Se numeran los tipos del 0 al 3 para simplificar la notación, siendo Teórico el cero, Practico el uno, Teórico simple el dos y Practico simple el tres.

- Cálculo 1
 - 0. 5⁸
 - 1. 12
 - 2. 0
 - 3. 0
- GAL 1
 - 0. 5
 - 1. 12
 - 2. 0
 - 3. 0
- Física 1
 - 0. 5
 - 1. 15
 - 2. 5

7. Las clases de la realidad denominadas “Teórico Prácticas” o “de Enfoque” fueron catalogadas como Teórico si tienen par o Teórico simple si no.

8. En la realidad, estas clases son tríos, se toma por simplicidad que son dúos

- 3. 0
- Taller DCRG
 - 0. 0
 - 1. 0
 - 2. 7
 - 3. 2
- TallerIne
 - 0. 0
 - 1. 0
 - 2. 5
 - 3. 0
- Cálculo 3
 - 0. 3
 - 1. 6
 - 2. 0
 - 3. 0
- Probabilidad y Estadística
 - 0. 4
 - 1. 10
 - 2. 0
 - 3. 0
- Sistemas de Referencia
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Ciencia, Tecnología y Sociedad
 - 0. 0
 - 1. 0
 - 2. 0
 - 3. 1
- Mecánica Newtoniana
 - 0. 2
 - 1. 0
 - 2. 12
 - 3. 0
- Principios de la Química General
 - 0. 0
 - 1. 0
 - 2. 1
 - 3. 1
- Matemática Discreta 2
 - 0. 2
 - 1. 7
 - 2. 0
 - 3. 0
- Lógica
 - 0. 1
 - 1. 0
 - 2. 2
 - 3. 0
- Programación 2
 - 0. 1
 - 1. 2

- 2. 0
- 3. 0
- Introducción a las Ciencias de los Materiales
 - 0. 1⁸
 - 1. 0
 - 2. 0
 - 3. 0
- Teoría de Errores
 - 0. 0
 - 1. 0
 - 2. 1
 - 3. 1
- Agrimensura Legal
 - 0. 1
 - 1. 0
 - 2. 1
 - 3. 0
- Topografía
 - 0. 0
 - 1. 0
 - 2. 0
 - 3. 1
- Materiales y Ensayos
 - 0. 1⁸
 - 1. 0
 - 2. 0
 - 3. 0
- Elasticidad
 - 0. 1⁸
 - 1. 0
 - 2. 2
 - 3. 0
- IIO
 - 0. 1
 - 1. 0
 - 2. 6
 - 3. 0
- Gestión de Calidad
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- AGPI
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Comportamiento de los Materiales
 - 0. 0
 - 1. 0
 - 2. 1
 - 3. 1
- Funciones de Variable Compleja
 - 0. 1
- 1. 0
- 2. 2
- 3. 0
- Señales y Sistemas
 - 0. 1
 - 1. 0
 - 2. 2
 - 3. 0
- Programación para Ingenieros Eléctricos
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Electrónica Fundamental
 - 0. 1
 - 1. 0
 - 2. 2
 - 3. 0
- Sistemas Operativos
 - 0. 1
 - 1. 0
 - 2. 3
 - 3. 0
- Programación 4
 - 0. 1
 - 1. 1
 - 2. 0
 - 3. 0
- Teoría de Lenguajes
 - 0. 1
 - 1. 0
 - 2. 2
 - 3. 0
- Elementos de la Mecánica de Fluidos
 - 0. 1⁸
 - 1. 0
 - 2. 3
 - 3. 0
- Electrotécnica
 - 0. 0
 - 1. 0
 - 2. 3
 - 3. 1
- Metalurgia de la Transformación
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Termodinámica Aplicada
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Fenómenos de Transporte

- 0. 1
- 1. 0
- 2. 1
- 3. 0
- Matemática Discreta 1
 - 0. 1
 - 1. 3
 - 2. 0
 - 3. 0
- Antenas y Propagación
 - 0. 1
 - 1. 0
 - 2. 1
 - 3. 0
- Elementos de Meteorología y Clima
 - 0. 1
 - 1. 0
 - 2. 0
 - 3. 0
- Física 2
 - 0. 2
 - 1. 7
 - 2. 0
 - 3. 0
- Física 3
 - 0. 1
 - 1. 12
 - 2. 0
 - 3. 0

Finalmente los salones disponibles con sus capacidades son los siguientes:

- Salón: 31: 50
- Salón: 101: 55
- Salón: 102: 55
- Salón: 115: 42
- Salón: 116: 25
- Salón: 301: 130
- Salón: 303: 110
- Salón: 305: 60
- Salón: 307: 350
- Salón: 309: 32
- Salón: 310: 32
- Salón: 311: 60
- Salón: 501: 130
- Salón: 601: 110
- Salón: ACTOS: 300
- Salón: 502: 50
- Salón: 703: 45
- Salón: 705: 40
- Salón: 720: 40
- Salón: 722: 50
- Salón: 725: 40
- Salón: 727: 50
- Salón: A01: 300
- Salón: A11: 100

- Salón: A12: 220
- Salón: A21: 140
- Salón: A22: 220
- Salón: B01: 380
- Salón: B11: 90
- Salón: B12: 250
- Salón: B21: 70
- Salón: B22: 70
- Salón: C01: 200
- Salón: C11: 100
- Salón: C12: 220
- Salón: C21: 100
- Salón: C22: 220
- Salón: UDELAR A: 21
- Salón: UDELAR B: 21
- Salón: UDELAR C: 21
- Salón: UDELAR D: 21
- Salón: BIBLIOTECA: 10
- Salón: 312: 42
- Salón: 314: 44
- Salón: 315: 30
- Salón: 401: 45
- Salón: 402: 36
- Salón: SW: 16

Es importante destacar que se considera el uso de los salones del Edificio Polifuncional José Luis Massera como si fueran parte de la Facultad de Ingeniería y que ninguna otra Facultad de la Universidad de la República los utiliza, al igual que tampoco se considera el uso de los mismos para diversos eventos.

8.2. Imágenes

(Comienzan en la próxima página)

	A	B	C	D	E	F	G	H	I	J
1		303			305			307		
2		M	T	N	M	T	N	M	T	N
3	Lu	Fis 1 P	Gal 1 P	Prog 4 P	Gal 1 P		Prog 2 P	Prog 2 P	Fis 1 T	Prog 2 T
4					MDisc 2 P		MecN P	Gal 1 T	Cal 1 T	MDisc 2 T
5	Ma	Cal 1 P	MDisc 2 P			MecN P		MDisc 2 T	Gal 1 T	Fis 1 T
6						Fis 1 P	Fis 1 P		Prog 4 T	Cal 1 T
7	Mi	MecN P	Gal 1 P	Gal 1 P		Cal 1 P	Prog 2 P	MecN T	Fis 1 T	
8							MDisc 2 P			
9	Ju	Cal 1 P		Prog 4 P		MecN P		MDisc 2 T	Gal 1 T	Prog 2 T
10		Gal 1 P							Prog 4 T	Fis 1 T
11	Vi	Fis 1 P	MDisc 2 P	MDisc 2 P	MDisc 2 P	Cal 1 P	MecN P	MecN T	Cal 1 T	MDisc 2 T
12		MecN P	Fis 1 P	Fis 1 P	Prog 2 P		Gal 1 P	Gal 1 T		Cal 1 T
13		B11	C7	D9	B10		G7	E12	I7	I9
14					E11		G11	H12	I11	I11
15		B9	C11			F9		H9	I9	I10
16						C12	D12		I10	I12
17		B12	C3	G12		F11	G3	H11	I3	
18							D11			
19		B5		D3		F5		H5	I5	I3
20		E3							I6	I5
21		B3	C5	G8	E4	F7	G4	H7	I4	I4
22		B7	F6	G5	H3		D7	H4		I6

Figura 1. Ejemplo de representación de una solución.

SOLUCIÓN VIEJA									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu			MDisc 2 P	Gal 1 P	Prog 2 P	Fis 1 P	Gal 1 T	MecN T	Cal 1 T
			Prog 2 P						Prog 2 T
Ma	Prog 2 P	Cal 1 P		Fis 1 P	MDisc 2 P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P			MecN P		Fis 1 T		
Mi	MDisc 2 P	Gal 1 P	Gal 1 P	MecN P		Fis 1 P	MDisc 2 T	Prog 4 T	Cal 1 T
	Cal 1 P		MDisc 2 P						Fis 1 T
Ju	Prog 2 P	Cal 1 P		Fis 1 P	MecN P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P					Fis 1 T	MecN T	
Vi	MDisc 2 P	Gal 1 P	Gal 1 P	MecN P	Prog 2 P		MDisc 2 T	Prog 4 T	Prog 2 T
	Cal 1 P		Prog 2 P	Gal 1 P	MDisc 2 P		Gal 1 T		Fis 1 T

SOLUCIÓN NUEVA									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu			MDisc 2 P	Gal 1 P	MDisc 2 P	Fis 1 P	Gal 1 T	MecN T	Cal 1 T
			Prog 2 P						Prog 2 T
Ma	Prog 2 P	Cal 1 P		Fis 1 P	Prog 2 P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P			MecN P		Fis 1 T		
Mi	Gal 1 P	MDisc 2 P	Gal 1 P	MecN P		Fis 1 P	MDisc 2 T	Prog 4 T	Cal 1 T
	Cal 1 P		MDisc 2 P						Fis 1 T
Ju	Prog 2 P	MecN P		Fis 1 P	Cal 1 P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P					Fis 1 T	MecN T	
Vi	Gal 1 P	MDisc 2 P	Gal 1 P	MecN P	Prog 2 P		MDisc 2 T	Prog 4 T	Prog 2 T
	Cal 1 P		Prog 2 P	Gal 1 P	MDisc 2 P		Gal 1 T		Fis 1 T

Figura 2. Ejemplo de mutación. En azul mutación de salón, en rojo mutación de turno y en amarillo mutación de día.

PADRE 1									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu			MDisc 2 P	Gal 1 P	Prog 2 P	Fis 1 P	Gal 1 T	MecN T	Cal 1 T
			Prog 2 P						Prog 2 T
Ma	Prog 2 P	Cal 1 P		Fis 1 P	MDisc 2 P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P			MecN P		Fis 1 T		
Mi	MDisc 2 P	Gal 1 P	Gal 1 P	MecN P		Fis 1 P	MDisc 2 T	Prog 4 T	Cal 1 T
	Cal 1 P		MDisc 2 P						Fis 1 T
Ju	Prog 2 P	Cal 1 P		Fis 1 P	MecN P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
		Fis 1 P					Fis 1 T	MecN T	
Vi	MDisc 2 P	Gal 1 P	Gal 1 P	MecN P	Prog 2 P		MDisc 2 T	Prog 4 T	Prog 2 T
	Cal 1 P		Prog 2 P	Gal 1 P	MDisc 2 P		Gal 1 T		Fis 1 T

PADRE 2									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu	Fis 1 P	Gal 1 P	Prog 4 P	Gal 1 P		Prog 2 P	Prog 2 P	Fis 1 T	Prog 2 T
				MDisc 2 P		MecN P	Gal 1 T	Cal 1 T	MDisc 2 T
Ma	Cal 1 P	MDisc 2 P			MecN P	Gal 1 P	MDisc 2 T	Gal 1 T	Fis 1 T
					Fis 1 P	Fis 1 P	Prog 4 T	Cal 1 T	
Mi	MecN P	Gal 1 P	Gal 1 P		Cal 1 P	Prog 2 P	MecN T	Fis 1 T	
						MDisc 2 P			
Ju	Cal 1 P		Prog 4 P		MecN P		MDisc 2 T	Gal 1 T	Prog 2 T
	Gal 1 P							Prog 4 T	Fis 1 T
Vi	Fis 1 P	MDisc 2 P	MDisc 2 P	MDisc 2 P	Cal 1 P	MecN P	MecN T	Cal 1 T	MDisc 2 T
	MecN P	Fis 1 P	Fis 1 P	Prog 2 P			Gal 1 T		Cal 1 T

HIJO 1									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu	MDisc 2 P	Gal 1 P	MDisc 2 P	Gal 1 P	Prog 2 P	Fis 1 P	Prog 2 T	MecN T	Cal 1 T
			Prog 2 P				MecN T	Fis 1 T	Prog 2 T
Ma	Prog 2 P	Cal 1 P		Fis 1 P	MDisc 2 P	Prog 4 P	Cal 1 T	Gal 1 T	MDisc 2 T
	Cal 1 P	Fis 1 P			MecN P	Gal 1 P	Fis 1 T	Prog 4 T	Gal 1 T
Mi	MecN P	Gal 1 P	Gal 1 P		Cal 1 P	Prog 2 P	MecN T	Fis 1 T	
					MDisc 2 P				
Ju	Prog 2 P			Fis 1 P	Prog 4 P		Cal 1 T	Gal 1 T	MDisc 2 T
	MecN P	Fis 1 P			MDisc 2 P		Fis 1 T	MecN T	Cal 1 T
Vi	MDisc 2 P	Gal 1 P	Gal 1 P	MecN P	Prog 2 P	MDisc 2 P	MDisc 2 T	Prog 4 T	Prog 2 T
	Cal 1 P	Cal 1 P	Prog 2 P	Gal 1 P	MecN P		Prog 2 T		Gal 1 T

HIJO 2									
	303			305			307		
	M	T	N	M	T	N	M	T	N
Lu	Fis 1 P	Gal 1 P	Prog 4 P	Gal 1 P		Prog 2 P	Prog 2 P	Fis 1 T	Prog 2 T
	Cal 1 P		Fis 1 P	MDisc 2 P		MecN P	Gal 1 T	Cal 1 T	MDisc 2 T
Ma	Cal 1 P	MDisc 2 P			MecN P	Gal 1 P	MDisc 2 T	Gal 1 T	Fis 1 T
					Fis 1 P	Fis 1 P	MecN T	Prog 4 T	Cal 1 T
Mi	MDisc 2 P	Gal 1 P	Gal 1 P		Cal 1 P	Prog 2 P	MecN T	Fis 1 T	
	Cal 1 P		MDisc 2 P		MDisc 2 P				Fis 1 T
Ju	Cal 1 P	MDisc 2 P	Prog 4 P	MecN P	Cal 1 P	Prog 2 P	MDisc 2 T	Gal 1 T	Prog 2 T
	Gal 1 P	Fis 1 P	Gal 1 P		MDisc 2 P		Prog 4 T	Fis 1 T	
Vi	Fis 1 P	Gal 1 P	MDisc 2 P	MDisc 2 P	Cal 1 P	MecN P	MecN T	Cal 1 T	MDisc 2 T
	MecN P	Fis 1 P	Fis 1 P	Prog 2 P			Gal 1 T		Cal 1 T

Figura 3. Ejemplo de cruzamiento_día para el día miércoles. En amarillo los pares afectados, en rojo las clases eliminadas y en verde los pares regenerados.

Comparaciones dos a dos primera instancia (config parametrica)

Muestra 1- Muestra 2	Test	Error standard	Test estandard	Sig.	Bonferroni
Configuracion 5- Configuracion 8	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 5- Configuracion 3	1.13333333	0.70710678	1.60277537	0.10898426	1.00000000
Configuracion 5- Configuracion 2	1.83333333	0.70710678	2.59272486	0.00952189	0.34278808
Configuracion 5- Configuracion 7	-1.93333333	0.70710678	-2.73414622	0.00625423	0.22515215
Configuracion 5- Configuracion 6	-2.43333333	0.70710678	-3.44125300	0.00057903	0.02084497
Configuracion 5- Configuracion 4	2.83333333	0.70710678	4.00693843	0.00006151	0.00221439
Configuracion 5- Configuracion 1	3.46666667	0.70710678	4.90260702	0.00000095	0.00003405
Configuracion 5- Configuracion 9	-3.73333333	0.70710678	-5.27973063	0.00000013	0.00000466
Configuracion 8- Configuracion 3	0.80000000	0.70710678	1.13137085	0.25789904	1.00000000
Configuracion 8- Configuracion 2	1.50000000	0.70710678	2.12132034	0.03389485	1.00000000
Configuracion 8- Configuracion 7	1.60000000	0.70710678	2.26274170	0.02365162	0.85145820
Configuracion 8- Configuracion 6	2.10000000	0.70710678	2.96984848	0.00297947	0.10726080
Configuracion 8- Configuracion 4	2.50000000	0.70710678	3.53553391	0.00040695	0.01465027
Configuracion 8- Configuracion 1	3.13333333	0.70710678	4.43120250	0.00000937	0.00033735
Configuracion 8- Configuracion 9	-3.40000000	0.70710678	-4.80832611	0.00000152	0.00005479
Configuracion 3- Configuracion 2	0.70000000	0.70710678	0.98994949	0.32219881	1.00000000
Configuracion 3- Configuracion 7	-0.80000000	0.70710678	-1.13137085	0.25789904	1.00000000
Configuracion 3- Configuracion 6	-1.30000000	0.70710678	-1.83847763	0.06599206	1.00000000
Configuracion 3- Configuracion 4	-1.70000000	0.70710678	-2.40416306	0.01620954	0.58354349
Configuracion 3- Configuracion 1	2.33333333	0.70710678	3.29983165	0.00096743	0.03482742
Configuracion 3- Configuracion 9	-2.60000000	0.70710678	-3.67695526	0.00023603	0.00849724
Configuracion 2- Configuracion 7	-0.10000000	0.70710678	-0.14142136	0.88753708	1.00000000
Configuracion 2- Configuracion 6	-0.60000000	0.70710678	-0.84852814	0.39614391	1.00000000

Configuracion 2- Configuracion 4	-1.00000000	0.70710678	-1.41421356	0.15729921	1.00000000
Configuracion 2- Configuracion 1	1.63333333	0.70710678	2.30988215	0.02089468	0.75220846
Configuracion 2- Configuracion 9	-1.90000000	0.70710678	-2.68700577	0.00720957	0.25954455
Configuracion 7- Configuracion 6	0.50000000	0.70710678	0.70710678	0.47950012	1.00000000
Configuracion 7- Configuracion 4	0.90000000	0.70710678	1.27279221	0.20309179	1.00000000
Configuracion 7- Configuracion 1	1.53333333	0.70710678	2.16846080	0.03012364	1.00000000
Configuracion 7- Configuracion 9	-1.80000000	0.70710678	-2.54558441	0.01090950	0.39274194
Configuracion 6- Configuracion 4	0.40000000	0.70710678	0.56568542	0.57160764	1.00000000
Configuracion 6- Configuracion 1	1.03333333	0.70710678	1.46135401	0.14391831	1.00000000
Configuracion 6- Configuracion 9	-1.30000000	0.70710678	-1.83847763	0.06599206	1.00000000
Configuracion 4- Configuracion 1	0.63333333	0.70710678	0.89566859	0.37042979	1.00000000
Configuracion 4- Configuracion 9	-0.90000000	0.70710678	-1.27279221	0.20309179	1.00000000
Configuracion 1- Configuracion 9	-0.26666667	0.70710678	-0.37712362	0.70608174	1.00000000

Comparaciones dos a dos segunda instancia (config parametrica)

Muestra 1- Muestra 2	Test	Error standard	Test estandard	Sig.	Bonferroni
Configuracion 5- Configuracion 8	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 5- Configuracion 3	-0.50000000	0.70710678	-0.70710678	0.47950012	1.00000000
Configuracion 5- Configuracion 2	-2.10000000	0.70710678	-2.96984848	0.00297947	0.10726080
Configuracion 5- Configuracion 6	-2.43333333	0.70710678	-3.44125300	0.00057903	0.02084497
Configuracion 5- Configuracion 7	-2.76666667	0.70710678	-3.91265752	0.00009129	0.00328629
Configuracion 5- Configuracion 4	-2.96666667	0.70710678	-4.19550024	0.00002723	0.00098017
Configuracion 5- Configuracion 9	-3.86666667	0.70710678	-5.46829244	0.00000005	0.00000164
Configuracion 5- Configuracion 1	-4.83333333	0.70710678	-6.83536555	0.00000000	0.00000000
Configuracion 8- Configuracion 3	-0.16666667	0.70710678	-0.23570226	0.81366372	1.00000000
Configuracion 8- Configuracion 2	-1.76666667	0.70710678	-2.49844396	0.01247399	0.44906351
Configuracion 8- Configuracion 6	-2.10000000	0.70710678	-2.96984848	0.00297947	0.10726080
Configuracion 8- Configuracion 7	-2.43333333	0.70710678	-3.44125300	0.00057903	0.02084497
Configuracion 8- Configuracion 4	-2.63333333	0.70710678	-3.72409571	0.00019602	0.00705660
Configuracion 8- Configuracion 9	-3.53333333	0.70710678	-4.99688792	0.00000058	0.00002097
Configuracion 8- Configuracion 1	-4.50000000	0.70710678	-6.36396103	0.00000000	0.00000001
Configuracion 3- Configuracion 2	1.60000000	0.70710678	2.26274170	0.02365162	0.85145820
Configuracion 3- Configuracion 6	-1.93333333	0.70710678	-2.73414622	0.00625423	0.22515215
Configuracion 3- Configuracion 7	-2.26666667	0.70710678	-3.20555074	0.00134804	0.04852954
Configuracion 3- Configuracion 4	-2.46666667	0.70710678	-3.48839345	0.00048593	0.01749357
Configuracion 3- Configuracion 9	-3.36666667	0.70710678	-4.76118566	0.00000192	0.00006929
Configuracion 3- Configuracion 1	4.33333333	0.70710678	6.12825877	0.00000000	0.00000003
Configuracion 2- Configuracion 6	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 2- Configuracion 7	-0.66666667	0.70710678	-0.94280904	0.34577859	1.00000000

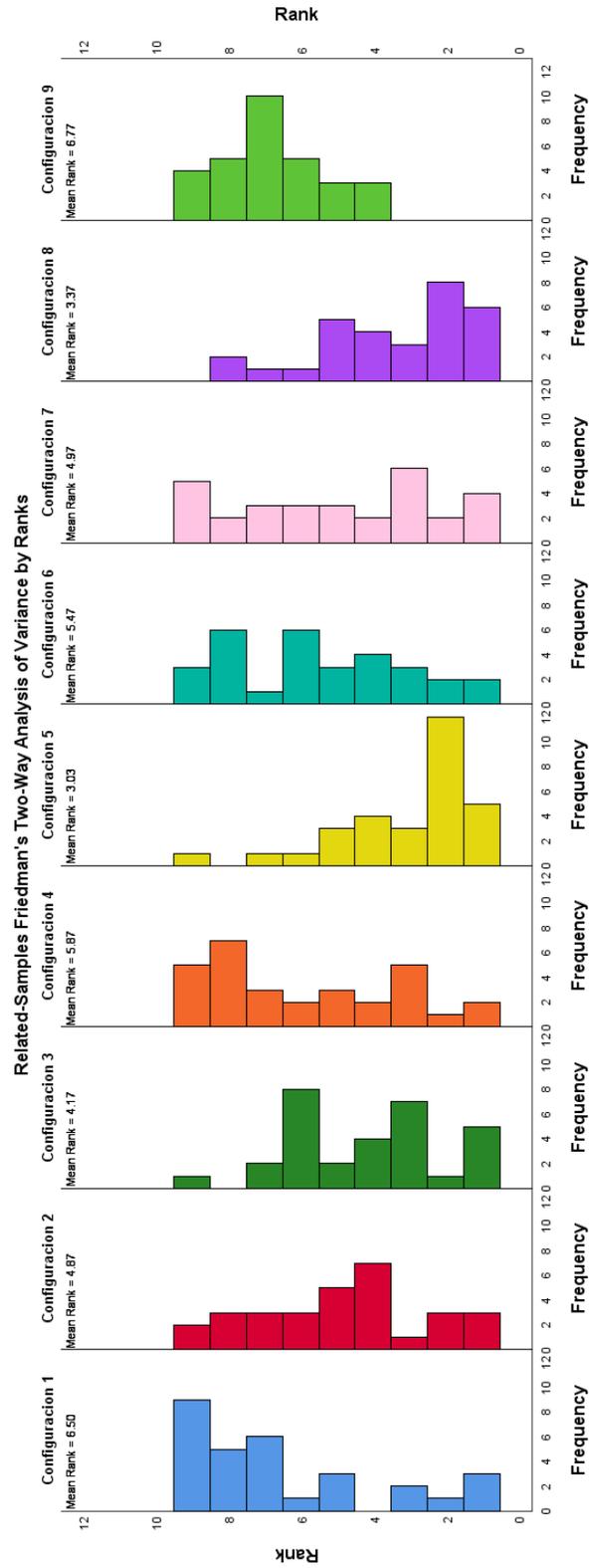
Configuracion 2- Configuracion 4	-0.86666667	0.70710678	-1.22565175	0.22032975	1.00000000
Configuracion 2- Configuracion 9	-1.76666667	0.70710678	-2.49844396	0.01247399	0.44906351
Configuracion 2- Configuracion 1	2.73333333	0.70710678	3.86551707	0.00011085	0.00399075
Configuracion 6- Configuracion 7	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 6- Configuracion 4	0.53333333	0.70710678	0.75424723	0.45070078	1.00000000
Configuracion 6- Configuracion 9	-1.43333333	0.70710678	-2.02703944	0.04265838	1.00000000
Configuracion 6- Configuracion 1	2.40000000	0.70710678	3.39411255	0.00068851	0.02478650
Configuracion 7- Configuracion 4	0.20000000	0.70710678	0.28284271	0.77729741	1.00000000
Configuracion 7- Configuracion 9	-1.10000000	0.70710678	-1.55563492	0.11979493	1.00000000
Configuracion 7- Configuracion 1	2.06666667	0.70710678	2.92270803	0.00347002	0.12492062
Configuracion 4- Configuracion 9	-0.90000000	0.70710678	-1.27279221	0.20309179	1.00000000
Configuracion 4- Configuracion 1	1.86666667	0.70710678	2.63986532	0.00829390	0.29858033
Configuracion 9- Configuracion 1	0.96666667	0.70710678	1.36707311	0.17160239	1.00000000

Comparaciones dos a dos tercera instancia (config parametrica)

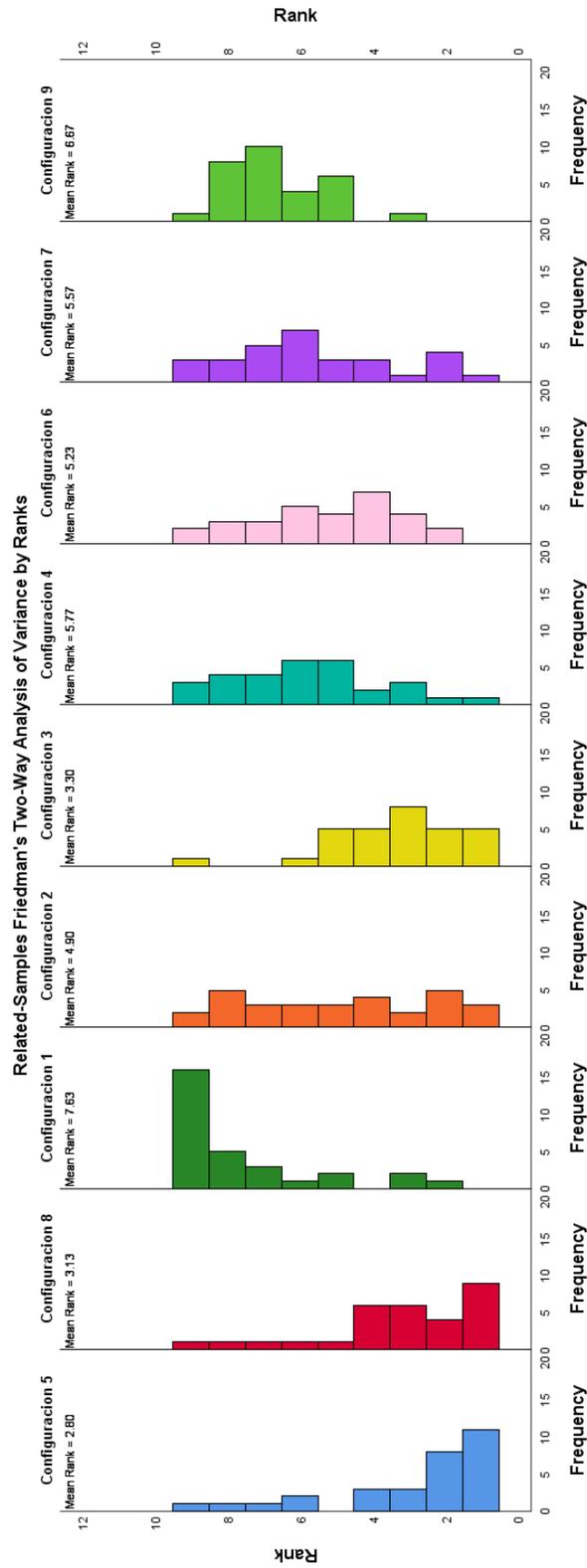
Muestra 1- Muestra 2	Test	Error standard	Test estandard	Sig.	Bonferroni
Configuracion 5-Configuracion 8	-0.83333333	0.70710678	-1.17851130	0.23859283	1.00000000
Configuracion 5-Configuracion 3	-1.13333333	0.70710678	-1.60277537	0.10898426	1.00000000
Configuracion 5-Configuracion 7	-1.56666667	0.70710678	-2.21560125	0.02671882	0.96187764
Configuracion 5-Configuracion 4	-2.56666667	0.70710678	-3.62981481	0.00028362	0.01021049
Configuracion 5-Configuracion 6	-2.63333333	0.70710678	-3.72409571	0.00019602	0.00705660
Configuracion 5-Configuracion 2	-3.33333333	0.70710678	-4.71404521	0.00000243	0.00008742
Configuracion 5-Configuracion 9	-3.66666667	0.70710678	-5.18544973	0.00000022	0.00000776
Configuracion 5-Configuracion 1	-4.36666667	0.70710678	-6.17539922	0.00000000	0.00000002
Configuracion 8-Configuracion 3	-0.30000000	0.70710678	-0.42426407	0.67137324	1.00000000
Configuracion 8-Configuracion 7	-0.73333333	0.70710678	-1.03708995	0.29969394	1.00000000
Configuracion 8-Configuracion 4	-1.73333333	0.70710678	-2.45130351	0.01423399	0.51242356
Configuracion 8-Configuracion 6	-1.80000000	0.70710678	-2.54558441	0.01090950	0.39274194
Configuracion 8-Configuracion 2	-2.50000000	0.70710678	-3.53553391	0.00040695	0.01465027
Configuracion 8-Configuracion 9	-2.83333333	0.70710678	-4.00693843	0.00006151	0.00221439
Configuracion 8-Configuracion 1	-3.53333333	0.70710678	-4.99688792	0.00000058	0.00002097
Configuracion 3-Configuracion 7	-0.43333333	0.70710678	-0.61282588	0.53999148	1.00000000
Configuracion 3-Configuracion 4	-1.43333333	0.70710678	-2.02703944	0.04265838	1.00000000
Configuracion 3-Configuracion 6	-1.50000000	0.70710678	-2.12132034	0.03389485	1.00000000
Configuracion 3-Configuracion 2	2.20000000	0.70710678	3.11126984	0.00186285	0.06706247
Configuracion 3-Configuracion 9	-2.53333333	0.70710678	-3.58267436	0.00034009	0.01224340
Configuracion 3-Configuracion 1	3.23333333	0.70710678	4.57262385	0.00000482	0.00017340
Configuracion 2-Configuracion 6	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 2-Configuracion 7	-0.66666667	0.70710678	-0.94280904	0.34577859	1.00000000

Configuracion 2- Configuracion 4	-0.86666667	0.70710678	-1.22565175	0.22032975	1.00000000
Configuracion 2- Configuracion 9	-1.76666667	0.70710678	-2.49844396	0.01247399	0.44906351
Configuracion 2- Configuracion 1	2.73333333	0.70710678	3.86551707	0.00011085	0.00399075
Configuracion 6- Configuracion 7	-0.33333333	0.70710678	-0.47140452	0.63735189	1.00000000
Configuracion 6- Configuracion 4	0.53333333	0.70710678	0.75424723	0.45070078	1.00000000
Configuracion 6- Configuracion 9	-1.43333333	0.70710678	-2.02703944	0.04265838	1.00000000
Configuracion 6- Configuracion 1	2.40000000	0.70710678	3.39411255	0.00068851	0.02478650
Configuracion 7- Configuracion 4	0.20000000	0.70710678	0.28284271	0.77729741	1.00000000
Configuracion 7- Configuracion 9	-1.10000000	0.70710678	-1.55563492	0.11979493	1.00000000
Configuracion 7- Configuracion 1	2.06666667	0.70710678	2.92270803	0.00347002	0.12492062
Configuracion 4- Configuracion 9	-0.90000000	0.70710678	-1.27279221	0.20309179	1.00000000
Configuracion 4- Configuracion 1	1.86666667	0.70710678	2.63986532	0.00829390	0.29858033
Configuracion 9- Configuracion 1	0.96666667	0.70710678	1.36707311	0.17160239	1.00000000

Grafica de rankings por configuración para la primera instancia



Grafica de rankings por configuración para la segunda instancia



Grafica de rankings por configuración para la tercera instancia

