

Universidad Nacional de La Pampa

Facultad de Ingeniería

Trabajo Final para alcanzar el grado de Ingeniero en Sistemas

Secuenciamiento de tareas mediante Algoritmos Genéticos

Estudiante:

A.P. Emiliano Dueñas

Directora:

Dra. Gabriela Minetti

Agradecimientos

A mi familia: Gladys, Rolando y Verónica, que fueron el apoyo en esta etapa; a mi novia Verónica, que estuvo día a día acompañándome y a Gabriela, que fue mi guía en la investigación.

Contenido

Capítulo 1: Introducción.....	4
1.1 Planteamiento.....	4
1.2 Descripción del Problema.....	5
Capítulo 2: Algoritmos Genéticos.....	6
2.1 Introducción a las metaheurísticas.....	6
2.2. Clasificación de las metaheurísticas.....	6
2.3 Algoritmos Evolutivos.....	9
2.3.1 Algoritmos Genéticos.....	9
2.3.1.1 Representación de un individuo.....	11
2.3.1.2 Inicialización de la Población.....	13
2.3.1.3 Selección.....	14
2.3.1.4 Operadores evolutivos.....	16
2.3.1.4.1 Recombinación o Cruce.....	16
2.3.1.4.2 Mutación.....	22
2.3.1.5 Estrategias de Reemplazo.....	26
2.3.1.6 Condición de Corte.....	27
Capítulo 3: Experimentación.....	28
3.1 Algoritmo Genético implementado.....	28
3.2 Metodología experimental.....	31
3.3 Análisis de Resultados.....	32
Capítulo 4: Conclusiones.....	41
Bibliografía.....	43
Anexos.....	44
Anexo 1: Salida ejemplo del AG1.....	44
Anexo 2: Resultados del análisis del diagrama de cajas.....	56
Anexo 3: Función principal del AG.....	58
Anexo 4: Script de ejecución del AG.....	65

Capítulo 1: Introducción

En este capítulo se explica los antecedentes y motivaciones que llevaron a plantear este proyecto final de la carrera Ingeniería en Sistemas de la Facultad de Ingeniería de la Universidad Nacional de La Pampa. Esto incluye una breve introducción a los Algoritmos Genéticos, la descripción del problema abordado y el objetivo principal de este trabajo.

1.1 Planteamiento

Algunos problemas no pueden ser resueltos con sistemas de producción o métodos tradicionales debido al extenso espacio de búsqueda en el cual se halla la solución, ni tampoco es necesario conocer de qué forma se resuelve el problema [1], en su mayoría son problemas de optimización combinatoria, donde generalmente el tiempo para encontrar la mejor solución es exponencial razón por la cual se clasifican como problemas No Polinomiales Completos (NP-completo) [2]. Para resolver estos casos se pueden utilizar algoritmos evolutivos (*Evolutionary Algorithms*, EAs). Estos algoritmos se inspiran en el proceso de evolución de los seres vivos, tienen la ventaja de adaptarse a situaciones cambiantes y generar posibles soluciones suficientemente buenas en forma rápida. Se encuadra entre las técnicas algorítmicas pertenecientes a las metaheurísticas poblacionales [3]. Dentro de los algoritmos evolutivos podemos encontrar los Algoritmos Genéticos (*Genetic Algorithms*, GAs) [4], que simulan la teoría darwiniana de la evolución.

Uno de estos problemas es el abordado en este trabajo: Problema de Secuenciamiento de Tareas (*Flow-Shop Sequencing Problem*, FSSP). Este es un problema clásico de la planificación de trabajos, típico de la optimización combinatoria, que se presenta en talleres muy tecnificados con equipamiento específico para cada tarea [5] [6][7][8][9].

Los Algoritmos Genéticos poseen una gran variedad de posibles configuraciones de las cuales depende directamente el desempeño del algoritmo y la calidad de las soluciones. En el presente trabajo se presenta una explicación detallada del problema; una introducción a las metaheurísticas, sus ventajas y desventajas, además de su clasificación; una descripción a la familia de algoritmos al cual pertenece, así como también, una especificación de la metaheurística en estudio, su diseño e implementación para resolver el Problema de Secuenciamiento de Tareas. A la vez se ha analizado y comparado diferentes opciones de diseño y configuraciones paramétricas: generación de población inicial, selección de reproductores, operadores de cruce, operadores de mutación y reemplazo de población, con diferentes configuraciones. Finalmente, se enuncian las conclusiones de este trabajo y los posibles trabajos futuros.

El objetivo de este trabajo es diseñar e implementar distintos Algoritmos Genéticos para resolver eficientemente el FSSP y analizar el desempeño de los mismos. Esto implica diseñar, implementar y analizar distintas estrategias para generar la población inicial, para seleccionar los padres, para realizar el cruce y la mutación y para reemplazar la población. Además, para alcanzar el objetivo planteado es necesario estudiar diferentes configuraciones paramétricas de las opciones de diseño mencionadas anteriormente.

1.2 Descripción del Problema

En el Problema de Secuenciamiento de Tareas [10] un conjunto de tareas j , que poseen m operaciones donde cada operación requiere una máquina diferente, deben seguir el mismo orden en una rutina de procesamiento para un conjunto de máquinas m con el objetivo de minimizar el tiempo de salida de la última tarea en la última máquina denominado *makespan* (en la Figura 1 se muestra el diagrama de Gantt para la siguiente secuencia de tareas: 4, 2, 1 y 3). En otras palabras,

- Cada tarea tiene que procesarse sobre todas las máquinas en el orden 1, 2, ..., m .
- Cada máquina procesa solo una tarea a la vez.
- Cada tarea se procesa en una máquina a la vez.
- Los tiempos de puesta a punto para las operaciones son independientes de la secuencia y son incluidos en el tiempo de procesamiento.
- La secuencia de operaciones de las tareas son las mismos para cada máquina.

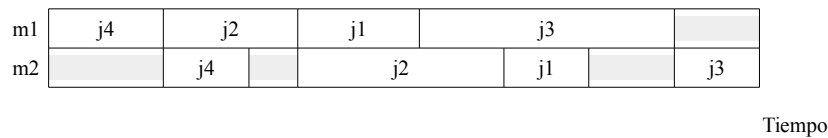


Figura 1: Diagrama de Gantt para un FSSP para 4 tareas y 2 máquinas.

Cuando el número de máquinas es mayor o igual a tres, el FSSP se transforma en un problema No Polinomial Completo conforme se incrementa el número de tareas. Existen tres diferentes categorías en las cuales podemos clasificar el problema:

- Determinista
- Estocástico
- Difuso

En este trabajo se aborda el caso determinista, donde se conoce el tiempo de procesamiento de las tareas, para ello utilizaremos las instancias de Taillard [11]. Eric Taillard propuso un conjunto de instancias con el fin de establecer un conjunto base de comparación (*benchmarks*).

Si denotamos $c(j_i, k)$ como el tiempo de procesamiento de la tarea j en la máquina m y denotamos $\{j_1, j_2, \dots, j_n\}$ como una permutación de tareas, podemos calcular el tiempo de procesamiento del problema con las j tareas y m máquinas de la siguiente manera:

$$\begin{aligned}
 c(j_1, 1) &= t_{j_1 1} \\
 c(j_1, k) &= c(j_1, k-1) + t_{j_1 k}, \text{ con } k=2, \dots, m \\
 c(j_i, 1) &= c(j_{i-1}, 1) + t_{j_i 1}, \text{ con } i=2, \dots, n \\
 c(j_i, k) &= \max\{c(j_{i-1}, k), c(j_i, k-1)\} + t_{j_i k}, \text{ con } i=2, \dots, n, k=2, \dots, m
 \end{aligned}$$

Finalmente el tiempo de salida de la última tarea en la última máquina (*makespan*) se calcula como:

$$c_{max} = c(j_n, m)$$

Capítulo 2: Algoritmos Genéticos

En este capítulo se introduce el concepto de metaheurísticas, profundizando especialmente en los algoritmos genéticos que son de interés para el presente trabajo.

2.1 Introducción a las metaheurísticas

Las metaheurísticas representan una familia de técnicas de optimización aproximadas que ganó mucha popularidad en las últimas décadas y se encuentran entre las técnicas más prometedoras y exitosas [3]. Además proporcionan soluciones "aceptables" en un plazo razonable para la solución de problemas difíciles y complejos en diferentes disciplinas, tales como: Física, Biología, Ingeniería, Bioinformática y Economía. Esto explica el importante crecimiento del interés por el dominio de la metaheurística. A diferencia de los algoritmos de optimización exactos, las metaheurísticas no garantizan la optimalidad de las soluciones obtenidas.

Estos algoritmos impulsan heurísticas básicas, ya sea una heurística constructiva a partir de una solución nula y la adición de elementos para construir una buena o bien una heurística de búsqueda local a partir de una solución completa, modificando de forma iterativa algunos de sus elementos a fin de lograr una solución mejor. En otras palabras, las metaheurísticas son algoritmos aproximados de propósito general, consistentes en procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.

Estas características proporcionan grandes ventajas, ya que un mismo algoritmo puede ser utilizado para resolver diferentes problemas, ya que se pueden modificar y adaptar fácilmente. Esto a su vez hace que la implementación de las metaheurísticas no sea compleja. Otra ventaja es que ejecutarlos paralelamente es sencillo, se puede ejecutar múltiples algoritmos sin complicaciones. A su vez son algoritmos no deterministas, por lo que dos ejecuciones del mismo algoritmo seguramente retornaran diferentes soluciones proporcionando soluciones aceptables para un mismo problema. Estas ventajas hacen que sean algoritmos exitosos para resolver problemas complejos cuando no hay un método exacto de resolución, o éste requiere mucho tiempo de cálculo y memoria, ó cuando no se necesita la solución óptima, basta con una de buena calidad.

Las técnicas metaheurísticas también presentan desventajas a la hora de ser utilizadas, a saber: son algoritmos aproximados, es decir que la solución que se encuentra es de buena calidad y no la mejor, generalmente existe un número importante de parámetros que necesitan ser configurados y poseen poca base teórica.

2.2. Clasificación de las metaheurísticas

Hay diferentes formas de clasificar y describir las metaheurísticas. Dependiendo de las características que se seleccionen se pueden obtener diferentes taxonomías: basadas en la naturaleza y no basadas en la naturaleza, con memoria o sin ella, con una o varias estructuras de vecindario, etc. Una de las clasificaciones más populares (Figura 2) las divide en metaheurísticas basadas en trayectoria y basadas en población [3]. Las primeras manipulan en cada paso un único

elemento del espacio de búsqueda, mientras que las segundas trabajan sobre un conjunto de ellos (población).

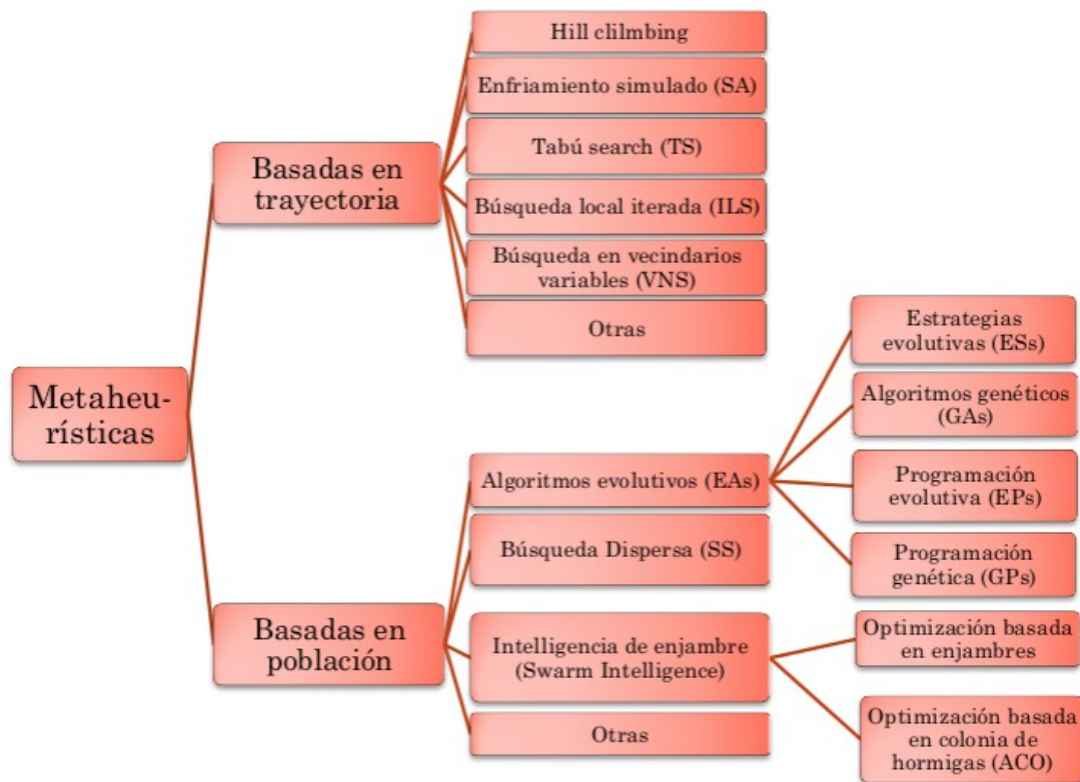


Figura 2: Clasificación de metaheurísticas basadas en trayectoria y basadas en población.

2.2.1 Metaheurísticas basadas en trayectoria

Estos algoritmos aplican iterativamente procesos de generación y reemplazo a partir de la solución (Figura 3). En la etapa de generación, un grupo de soluciones candidatas son generadas a partir de la solución s . Este grupo de soluciones candidatas $C(s)$ generalmente es obtenido a través de transformaciones locales de la solución. En la etapa de reemplazo, una solución es seleccionada desde el grupo de soluciones candidatas $C(s)$ para reemplazar a la solución actual s . Este proceso itera hasta que se cumple el criterio de parada. Las etapas de generación y reemplazo pueden ser sin memoria. En este caso, los dos procesos se basan sólo en la solución actual. Si son con memoria, se guarda un historial de las soluciones para ser usadas en la generación del grupo de soluciones candidatas y para la selección de la nueva solución [3].

La principal característica de estas técnicas es que parten de una solución y, mediante la exploración del vecindario, van actualizando la solución actual, formando una trayectoria. La mayoría de estos algoritmos surgen como extensiones de los métodos simples de búsqueda local a los que se les añade algún mecanismo para escapar de los mínimos locales.

Esto implica la necesidad de una condición de parada más elaborada que la de encontrar un mínimo local. Normalmente se termina la búsqueda cuando se alcanza un número máximo predefinido de iteraciones, se encuentra una solución con una calidad aceptable, o se detecta un

estancamiento del proceso.

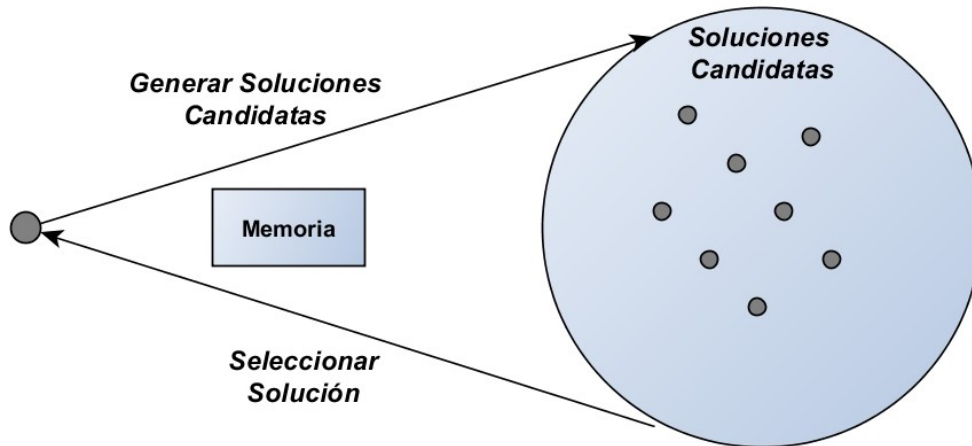


Figura 3: Metaheurísticas basadas en trayectoria.

Metaheurísticas basadas en población

Las técnicas basadas en población (Figura 4) se inician con un conjunto de soluciones iniciales, denominado población inicial. Luego iterativamente van aplicando métodos de generación de nueva población y métodos de reemplazo. La etapa de generación de nueva población, crea una nueva población a partir de la población anterior. En la etapa de reemplazo la población es reemplazada por la nueva población, este reemplazo puede ser total o parcial. Este proceso itera hasta que se cumpla la condición de parada del algoritmo. Estas metaheurísticas también pueden ser sin memoria, es decir que las etapas de generación y reemplazo se basan sólo en la población actual; o bien pueden ser con memoria, donde se almacena cierto historial de las poblaciones anteriores para ser utilizados en la etapa de generación y reemplazo [3].

Estas metaheurísticas trabajan con un conjunto de soluciones (población) que iterativamente son mejoradas a través de un proceso inteligente de exploración del espacio de búsqueda, a diferencia de los métodos basados en trayectoria, que únicamente manipulan una solución del espacio de búsqueda por iteración.

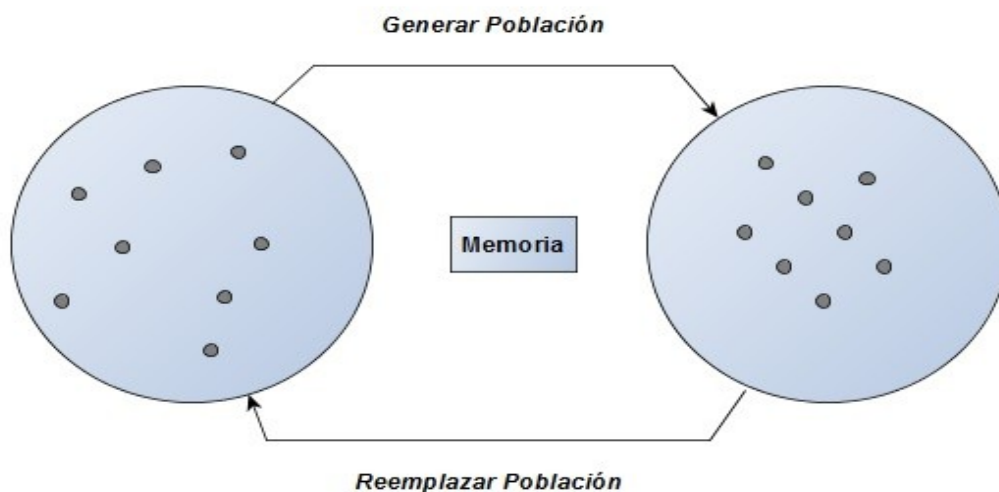


Figura 4: Metaheurísticas basadas en población.

2.3 Algoritmos Evolutivos

Los algoritmos evolutivos, que son una subclase de los métodos basada en población, están inspirados en la teoría de la evolución natural. Esta familia de técnicas sigue un proceso iterativo y estocástico que opera sobre una población de soluciones, denominadas en este contexto individuos. La población inicial es generada normalmente de forma aleatoria, pero se podría generar con ayuda de una heurística de construcción [3].

El esquema general de un algoritmo evolutivo comprende tres fases principales: selección, reproducción y reemplazo. El proceso completo es repetido hasta que se cumpla un cierto criterio de terminación (normalmente después de un número dado de iteraciones). En la fase de selección se escogen generalmente los individuos más aptos de la población actual para ser posteriormente recombinados en la fase de reproducción. Los individuos resultantes de la recombinación se alteran mediante un operador de mutación. Finalmente, a partir de la población actual y/o los mejores individuos generados (de acuerdo a su valor de fitness) se forma la nueva población, dando paso a la siguiente generación del algoritmo.

2.3.1 Algoritmos Genéticos

Los algoritmos genéticos, una variante de los Algoritmos Evolutivos, fueron desarrollados por John H. Holland a principios de la década de 1960, inspirándose en el proceso observado en la evolución natural de los seres vivos [4]. Los algoritmos genéticos se basan en la supervivencia de los individuos más aptos. La ventaja de estos algoritmos, es el paralelismo implícito en la búsqueda de la mejor solución ya que trabaja en base a un conjunto de posibles soluciones. A este conjunto de soluciones se lo denomina población, el cual irá evolucionando por medio de operadores de selección, cruzamiento, mutación y reemplazo. En la Figura 3 puede observarse el ciclo de vida de un algoritmo genético.

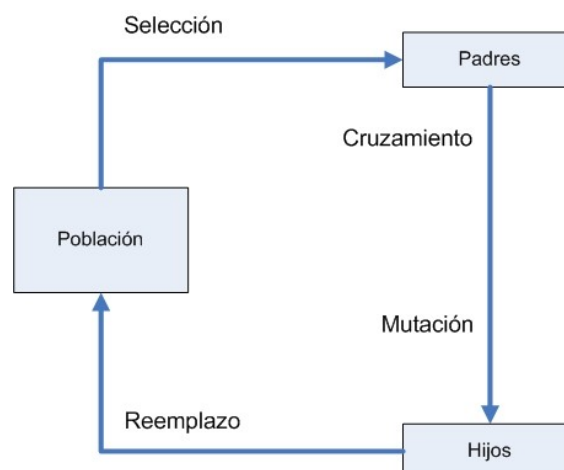


Figura 4: Esquema Evolutivo Genético.

En la siguiente pseudocódigo (Algoritmo 1) puede observarse un pseudocódigo básico que representa el funcionamiento del algoritmo genético.

```

Generar(P(0)) ; / * Población Inicial * /
t = 0;
While no se cumpla condición de corte(P(t)) Do
  Evaluar(P(t)) ;
  P'(t) = Selección(P(t)) ;
  P'(t) = Reproducción(P'(t));
  Evaluar(P'(t)) ;
  P(t + 1) = Reemplazar(P(t), P'(t)) ;
  t = t + 1;
End While
Output El mejor individuo encontrado.

```

Algoritmo 1: Pseudocódigo de un Algoritmo Genético.

El Algoritmo Genético trabaja sobre individuos que representan potenciales soluciones al problema, codificados de acuerdo a un mecanismo prefijado [3]. Los algoritmos genéticos no trabajan directamente sobre las soluciones del problema en cuestión, sino que lo hacen sobre una abstracción de los objetos solución, usualmente denominadas cromosomas por analogía con la evolución natural biológica. Un cromosoma es un vector de genes, mientras que el valor asignado a un gen se denomina alelo. El genotipo denota al conjunto de genes que definen a un individuo (suelen utilizarse indistintamente los términos genotipo, cromosoma e individuo). El genotipo sometido al medio ambiente se denomina fenotipo. Representa un punto del espacio de soluciones del problema. Informalmente, el fenotipo es una característica del individuo.

La operativa del algoritmo genético comienza con una etapa de inicialización de los individuos, que puede ser completamente aleatoria, muestreando al azar diferentes secciones del espacio de soluciones, o guiada de acuerdo a características del problema a resolver. El algoritmo genético podría inclusive tomar como población inicial individuos resultantes como salida de algún otro algoritmo heurístico de resolución que permitiera calcular buenas soluciones iniciales aproximadas para el problema.

La evolución propiamente dicha se lleva a cabo en el ciclo que genera nuevos individuos a partir de la población actual mediante un procedimiento de aplicación de operadores estocásticos (basados en probabilidades). En este ciclo se distinguen cuatro etapas:

- **Evaluación:** etapa que consiste en asignar un valor de adecuación a cada individuo en la población. Este valor evalúa que tan bien resuelve cada individuo el problema en cuestión, y es utilizado para guiar el mecanismo evolutivo.
- **Selección:** proceso que determina candidatos adecuados, de acuerdo a sus valores de *fitness*, para la aplicación de los operadores evolutivos con el objetivo de engendrar la siguiente generación de individuos.
- **Operadores evolutivos:** etapa que genera un conjunto de descendientes a partir de los individuos seleccionados en la etapa anterior, denominada recombinación, y luego los somete a una probabilidad de mutación. Se recomienda que la probabilidad de mutación tome valores chicos ($P_m \in [0.001, 0.01]$). Algunas estrategias inicializan la probabilidad de mutación en $P_m=1/k$, donde k es el número de variables de decisión.

- Reemplazo: mecanismo que realiza el recambio poblacional, sustituyendo individuos de la generación anterior por descendientes creados en la etapa anterior.

Diversas políticas para la selección y el reemplazo de individuos permiten modificar las características del algoritmo genético. Aplicando políticas adecuadas es posible privilegiar los individuos más adaptados en cada generación (estrategias de elitismo), aumentar la presión selectiva sobre individuos mejor adaptados, generar un número reducido de descendientes en cada generación (modelos de estado estacionario), y muchas otras variantes. Con estos operadores se determina el modo de perpetuar buenas características, que se asumen con aquellas presentes en los individuos más adaptados.

2.3.1.1 Representación de un individuo

La representación (o codificación) de individuos solución resulta importante para el desempeño de los algoritmos genéticos y qué codificación utilizar depende del problema que se quiere resolver. Según Fang [12] las normas básicas a tener en cuenta para el diseño de una codificación óptima se pueden resumir en:

- 1- Cada solución del problema tendrá su correspondiente cadena codificada.
- 2 - Para cada cadena codificada producida mediante los operadores genéticos existirá su correspondiente solución decodificada.
- 3 - Codificación y solución se corresponderán una a una, es decir no habrá redundancia.
- 4 - La codificación deberá permitir heredar de padres a hijos las características inherentes a los primeros.

Existen distintos tipos de codificaciones de las soluciones, a saber: binarias, simbólicas, permutación y la real.

Codificación binaria: es la codificación más extendida debido a que los primeros algoritmos genéticos utilizaron este tipo de codificación [3]. En este caso, cada cromosoma (Figura 5) es una cadena de bits (0 o 1). Se pueden representar muchos cromosomas incluso con un número reducido de genes. Sin embargo esta opción no es la mejor para muchos problemas y en algunas ocasiones es necesario realizar correcciones después de la reproducción y/o mutación.

1	0	1	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

Figura 5: Cromosoma en codificación binario.

Codificación por Permutación: las soluciones de problemas de asignación, problemas de secuenciación y mezclas de los mismos pueden ser representados por esquemas de permutación [3]. Un modelo de permutación de estos problemas se describe generalmente de la siguiente manera:

Un conjunto m de recursos están disponibles para llevar a cabo n tareas. Asignar todas las tareas a los recursos y secuenciar la asignación de las tareas para cada recurso de modo tal que se minimice la función costo. Solución representada por una permutación de tareas $\{1,2,\dots,n\}$ (Figura 6), que está particionada en m por los números $\{n_k\}$.

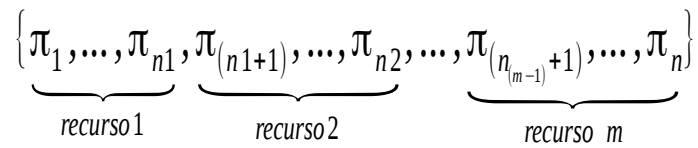


Figura 6: Representación por permutación.

Hay algunos casos especiales de este modelo. Cuando $m=1$ las tareas deben ser asignadas a un sólo recurso. Si la tarea i precede a la tarea j el costo será c_{ij} unidades. Luego la función que se deberá minimizar será:

$$\sum_{i=1}^{n-1} c_{(\pi_i, \pi_{(i+1)})}$$

El otro caso extremo es cuando $n=m$. Se asume que cada recurso debe resolver sólo una tarea. El problema de asignación debe minimizar la función:

$$\sum_{i=1}^m c_{(i, \pi_i)}$$

Donde c_{ij} representa el costo de asignar el recurso i a la tarea j .

Codificación Real: este tipo de codificación será utilizada en caso de resolución de problemas en el que se requiera del uso de valores de cifrado complejos como podría ser en el uso de números reales (Figura 7), cuya codificación con números binarios sería muy compleja [3]. En esta codificación cada cromosoma es una cadena de valores relacionados con el problema a estudiar, pudiendo ser desde números decimales, cadenas de caracteres o incluso una combinación de varios de ellos. Su aplicación es muy buena en ciertos problemas concretos. Por el contrario para la utilización de esta codificación, normalmente es necesario desarrollar nuevas técnicas de reproducción y mutación específicas hacia la resolución del problema.

3,34	0,5	1,34	0,7	9	0,21	7,5	4,3	1,2	1,3
------	-----	------	-----	---	------	-----	-----	-----	-----

Figura 7: Cromosoma en codificación real.

Codificación en Árbol: este tipo de codificación se utiliza principalmente en el desarrollo de programas o expresiones para programación genética [3]. Cada cromosoma será en este caso un árbol con ciertos objetos (Figura 8).

En este método, los cambios aleatorios pueden generarse cambiando el operador, alterando el valor de un cierto nodo del árbol o simplemente sustituyendo un subárbol por otro.

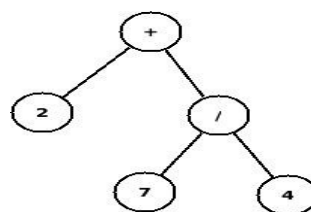


Figura 8: Representación en Árbol.

2.3.1.2 Inicialización de la Población

Según Michalewicz [1] la forma más eficaz para obtener los individuos que formarán parte de la primera iteración es la generación aleatoria. De esta forma, se podrá obtener una población inicial uniformemente distribuida por el espacio de búsqueda mejorando la exploración. Sin embargo, la calidad media de las soluciones generadas de esta forma será baja necesitando más iteraciones para ser mejoradas.

Otra posible forma de inicialización, será utilizar una o varias heurísticas más sencillas capaces de encontrar diferentes soluciones que pasarán a ser parte de la población inicial. El problema que se encuentra con este método es que se generan soluciones en zonas muy determinadas del espacio de búsqueda perdiendo poder exploratorio. Por otro lado, el beneficio será que se parte de soluciones de algún modo mejoradas.

Naturalmente, esta capacidad de exploración inicial dependerá de la relación entre el número de soluciones potenciales totales y el número de individuos que formarán parte de la población. Por lo tanto, siguiendo este razonamiento, cuanto menor sea esta proporción, mayor será el poder exploratorio del algoritmo. Pero por otra parte, cuanto mayor sea el número de soluciones con la que se trabaja menor será la velocidad de búsqueda, retrasando la convergencia del proceso y necesitando un mayor número de generaciones, iteraciones, lo que provocará unas mayores necesidades computacionales y temporales. Por tanto, esta decisión será un compromiso entre exploración y explotación del espacio de búsqueda.

Existen distintas técnicas de inicialización de la población inicial, entre ellas a saber: al azar, heurística NEH y una variación a la NEH, denominada NEHH.

Inicialización al azar: cada individuo es generado al azar de manera tal que el individuo sea válido en la codificación del problema.

Heurística NEH: la heurística NEH propuesta por Nawaz, Enscore, Ham [14] es un procedimiento constructivo de secuenciación muy eficiente que proporciona buenas secuencias para el FSSP, con el objetivo de minimizar el instante máximo de finalización de los trabajos [10].

El algoritmo NEH consiste en:

1. Ordenar las j tareas de forma descendente de acuerdo al tiempo de procesamiento.
2. Tomar las primeras 2 tareas y ordenarlas de forma tal que se minimice el *makespan*, como si fueran las únicas 2 tareas.
3. Para $k=3$ a j hacer: poner la k tarea en los k posibles lugares y dejarla en la posición que minimice el *makespan*.

Heurística NEHH: variación del NEH para que sea una heurística no determinista, propuesta por Ruiz et al. [15], consiste en una modificación después de ordenar las tareas en el paso 2, en la cual se toman 2 tareas al azar de la lista y se las coloca en la primera y segunda posición. De esta manera, se obtiene una heurística no determinista.

2.3.1.3 Selección

Esta etapa es una de los principales componentes de los Algoritmos Evolutivos. Según El-Ghazali Talbi [3] “el mejor es un individuo que tenga la mayor probabilidad de ser padre”.

Para garantizar la diversidad de la población todos los individuos deben tener posibilidad de ser seleccionados, aún los de peor *fitness* (aptitud de un individuo para resolver el problema en cuestión) pueden aportar valioso material genético.

Los individuos son evaluados de acuerdo a una función de aptitud que toma en cuenta la adecuación de cada solución al problema que se intenta resolver. La complejidad de dicha función está influenciada por la representación de los cromosomas. La función de *fitness* tiene una influencia importante en el mecanismo del AG, por un lado, actúa como una caja negra para el proceso evolutivo, y por el otro guía el mecanismo de exploración. El *fitness* es asignado a los individuos de dos maneras diferentes:

- Asignación proporcional de *fitness* en la que los *fitness* absolutos se asocian con los individuos.
- Asignación basada en ranking en la que el *fitness* relativo esta asociado con los individuos. Por ejemplo, un rango en la población se asocia a cada individuo de acuerdo a su rango en una clasificación decreciente de los individuos.

Existen diferentes estrategias para seleccionar los padres por su *fitness*. Algunas de las más conocidas y utilizadas son: Selección de Ruleta, Muestreo Estocástico Universal, Selección por Torneo y Selección Basada en Ranking.

Selección de Ruleta: es la estrategia de selección más común (Figura 9). Esta le asigna a cada individuo una probabilidad de ser seleccionado de acuerdo a su *fitness* [3]. Sea N el número de individuos existentes y f_i el desempeño del i-ésimo individuo. La probabilidad asociada a su selección está dada por:

$$p_i = f_i / \left(\sum_{j=1}^n f_j \right)$$

Esta selección permite que los mejores individuos sean elegidos con una mayor probabilidad, pero al mismo tiempo permite a los peores individuos ser elegidos, lo cual puede ayudar a mantener la diversidad de la población, en contraste con la selección por truncamiento.

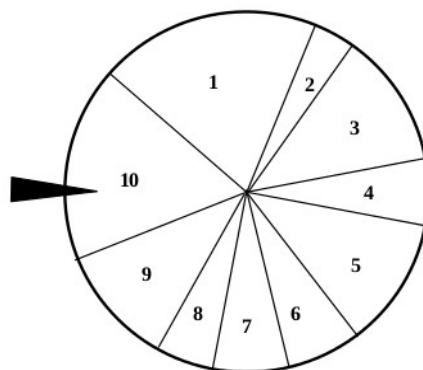


Figura 9: Selección por Ruleta.

Un problema de la selección de ruleta se presenta cuando existe una pequeña fracción de la población (en el límite, sólo un individuo) que posee una medida de desempeño excesivamente superior al resto. Esto provoca pérdida de diversidad y puede conducir a convergencia prematura pues la mayor parte de los individuos seleccionados será una copia de los pocos predominantes.

Muestreo Estocástico Universal: este método se puede utilizar para reducir el error de selección de la estrategia de Ruleta. Este método consiste en una Selección de Ruleta que se le colocan alrededor μ punteros equidistantes en lugar de sólo uno [3]. En esta estrategia un solo giro de la ruleta seleccionará simultáneamente todos los individuos μ para la reproducción.

Selección por Torneo: este método consiste en seleccionar aleatoriamente un grupo de k individuos, el parámetro k es conocido como el tamaño del grupo del torneo [3]. El torneo es aplicado entre los k individuos del grupo donde se seleccionará el mejor de todos de acuerdo a su *fitness* (Figura 10).

Los beneficios de este tipo de selección son la velocidad de aplicación, dado que no es necesario evaluar ni comparar la totalidad de la población, y la capacidad de prevenir, en cierto grado, la convergencia prematura. La principal desventaja es la necesidad de establecer el parámetro correspondiente al tamaño del subconjunto.

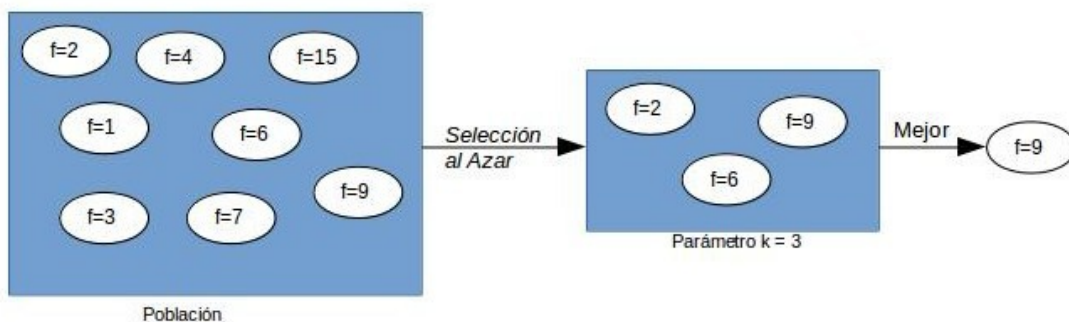


Figura 10: Selección por Torneo, tomando $k = 3$.

Selección Basada en Ranking: en lugar de usar el *fitness* de un individuo, se utiliza un ranking de individuos (Figura 11). Esta función da prioridad a individuos con mayor ranking [3]. El ranking se hace en base a la siguiente fórmula:

$$P(i) = \frac{2-s}{\mu} - \frac{2r(i)(s-1)}{\mu(\mu-1)}$$

Donde s es la precisión de selección ($1.0 < s \leq 2.0$), μ es el tamaño de la población, y $r(i)$ es el *fitness* asociado con el individuo i . Mientras más grande sea la precisión de selección s , más importancia se le da a los mejores individuos.

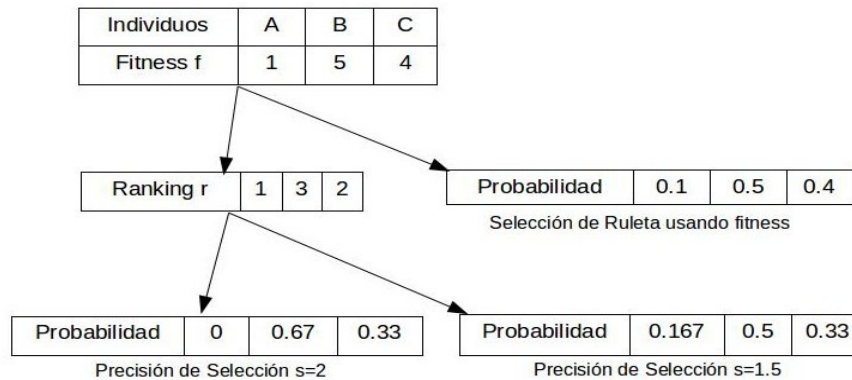


Figura 11: Selección por Basada en Ranking [3].

2.3.1.4 Operadores evolutivos

Una vez seleccionado el conjunto de padres, se arman las parejas para luego combinar sus cromosomas utilizando un operador de cruce y posteriormente se mutan los hijos resultantes de acuerdo a un algoritmo de mutación.

Los operadores evolutivos determinan el modo en que el algoritmo explora el espacio de soluciones del problema. Los diferentes operadores y las particularidades en su modo de aplicación dan características peculiares a las distintas variantes de los algoritmos genéticos. Por su lado, los Operadores de Cruce permiten combinar características de dos o más individuos con la idea de obtener descendientes mejor adaptados y los Operadores de Mutación introducen diversidad mediante modificaciones aleatorias, son los dos operadores evolutivos más difundidos.

2.3.1.4.1 Recombinación o Cruce

Es un operador binario o en algunos casos n-ario. El papel de los operadores de cruce es heredar algunas características de los dos padres para generar la descendencia. El diseño de estos operadores dependerá principalmente de la representación utilizada.

Para el diseño de los Operadores de Cruce se deben tener en cuenta los siguientes puntos:

- **Heredabilidad:** principal característica de un Operador de Cruce. El operador debe heredar material genético de ambos padres. Si dos individuos generan siempre descendientes iguales, se lo denomina un Operador de Cruce Puro o se dice que tiene fuerte heredabilidad.
- **Validez:** el operador debe siempre proporcionar soluciones válidas. Cuando se trata de problemas de optimización con restricciones no siempre se obtienen soluciones válidas.

La probabilidad de cruce p_c ($p_c \in [0, 1]$) representa la proporción de padres sobre los cuales los Operadores de Cruce serán aplicados. El mejor valor p_c esta relacionado con otros

parámetros tales como el tamaño de la población, la probabilidad de mutación y el método de selección. Las probabilidades de cruce más usadas normalmente varían entre [0.45, 0.95]. Puede resultar útil aplicar técnicas adaptativas para la probabilidad de cruce.

Para las representaciones lineales excluyendo las permutaciones, los operadores de cruce bien conocidos son los operadores de cruce en 1 punto, su forma general es el operador de cruce en n puntos y el cruce uniforme.

Estos operadores fueron propuestos originalmente para representaciones binarias. En el cruce 1 punto, el cruce se sitúa en una posición k seleccionada al azar y dos hijos son creados a través del intercambio de los segmentos de los padres considerándose la posición k . Por lo general, para seleccionar el punto k se utiliza una distribución al azar uniforme.

En el uso de cruce uniforme, dos individuos pueden ser recombinados sin importar el tamaño de los segmentos. Cada elemento de los hijos es seleccionado de forma azarosa desde ambos padres. Cada padre participa de forma igualitaria en la creación de los descendientes.

Para representaciones de valores reales, además de los operadores mencionados se utilizan normalmente dos tipos de operadores de cruce: recombinación basada en la media y recombinación basada en los padres.

En la recombinación basada en la media, los hijos son generados según la proximidad de sus padres a la media. Los operadores de cruce que se pueden utilizar son:

- **Cruce Intermedio:** este operador intenta promediar el valor de los elementos de los padres [3]. Dados dos padres p_1 y p_2 , el cruce aritmético crea el hijo o usando la media ponderada:

$$o_i = \alpha x_{1i} + (1 - \alpha)x_{2i}$$

Donde o_i represente el gen i del hijo o , x_{1i} representa el gen i del padre 1, x_{2i} representa el gen i del padre 2 y α es un valor al azar entre [0,1].

- **Cruce Geométrico:** este operador genera un hijo o de la siguiente manera [3]:

$$o = (\sqrt{x_{11}x_{21}}, \dots, \sqrt{x_{1n}x_{2n}})$$

Donde o representa al hijo, x_{1i} representa el gen i del padre 1 y x_{2i} representa el gen i del padre 2.

- **Cruce de Distribución Normal Unimodal (Unimodal Normal Distribution Crossover, UNDX) :** este cruce utiliza μ padres (c^1, \dots, c^μ). Se calcula la media $g = (x_1, \dots, x_n)$ de todos los padres, excepto el último, x_n representa la media de todos los genes n de los $\mu - 1$ padres, y los vectores $d^i = c^i - g$, para $i = 1, \dots, \mu - 1$. Sean los cosenos $e^i = d^i / |d^i|$, para $i = 1, \dots, \mu - 1$. Se calcula la longitud D del vector $c^\mu - g$ ortogonal a los e^i calculados. Se crean los vectores e^j , para $j = \mu, \dots, n$, con n la dimensión del problema, que forman una base ortonormal al subespacio que generan los vectores e^i , con $i = 1, \dots, \mu - 1$. Entonces, el descendiente y se obtiene con la siguiente ecuación:

$$y = g + \sum_{i=1}^{\mu} w_i |d^i| e^i + \sum_{i=\mu}^n v_i D e^i$$

Donde w_i y v_i son variables aleatorias que siguen una distribución normal con media cero y varianza σ_ζ^2 y σ_η^2 , respectivamente (parámetros del operador) [3].

En la recombinación basada en los padres, los hijos son generados cercanos a sus padres. A cada padre se le asigna una probabilidad igualitaria de crear un hijo en su vecindario. Los operadores de cruce que se pueden utilizar son:

- **Cruce binario Simulado (Simulated Binary Crossover, SBX):** un número al azar μ_i es generado entre $[0,1]$. Tomando la función de probabilidad de distribución:

$$P(\beta) = \begin{cases} \left(\frac{1}{2}(\eta+1)\beta_i^\eta\right) & \text{si } \beta_i \leq 1 \\ \left(\frac{1}{2}(\eta+1)\beta_i^{\frac{1}{\eta+2}}\right) & \text{otro caso} \end{cases}$$

El índice de distribución η proporciona mas probabilidad de generar hijos cercanos a sus padres. El valor de βq_i es tomado de forma tal que el área bajo la curva de probabilidad desde 0 a βq_i es igual al número al azar μ_i generado:

$$\beta q_i = \begin{cases} (2\mu_i^{\frac{1}{\eta+1}}) & \text{si } \mu_i \leq \frac{1}{2} \\ \left(\left(\frac{1}{2(1-\mu_i)}\right)^{\frac{1}{\eta+1}}\right) & \text{otro caso} \end{cases}$$

Los dos descendientes y^1 e y^2 se obtienen de la siguiente manera:

$$y_i^1 = \frac{1}{2}[(1 + \beta q_i)x_i^1 + (1 - \beta q_i)x_i^2]$$

$$y_i^2 = \frac{1}{2}[(1 - \beta q_i)x_i^1 + (1 + \beta q_i)x_i^2]$$

donde x^1 y x^2 son los padres [3].

- **Cruce Centrada en los Padres (Parent Centric Crossover, PCX):** dado un conjunto de padres (x^1, \dots, x^μ), se calcula la media g . Después, para cada descendiente a crear, se selecciona aleatoriamente uno de los padres del conjunto (x^p) con igual probabilidad para todos. Se calcula el vector $d^p = x^p - g$. Para cada uno de los $(\mu - 1)$ padres restantes, se obtienen las distancias D_i perpendiculares a la recta d^p y se calcula su media \bar{D} . Entonces el descendiente y se calcula como muestra la siguiente fórmula:

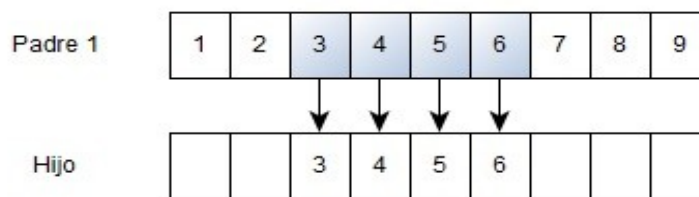
$$y = x^p + \omega_\xi |d^p| + \sum_{i=1, i \neq p}^{\mu} \omega_\eta \bar{D} e^i$$

donde e^i es un conjunto de $(\mu-1)$ bases ortonormales que generan el subespacio perpendicular a d^p , y ω_ξ y ω_η son variables aleatorias que siguen una distribución normal con media cero y varianzas σ_ξ^2 y σ_η^2 , respectivamente (parámetros del operador) [3].

La aplicación de operadores de cruce clásicos a permutaciones generarán soluciones que no son permutaciones, es decir soluciones no viables. Por ello, se diseñaron operadores de cruce de la siguiente manera:

- Cruce de orden (Order Crossover, OX):** se seleccionan 2 puntos de cruce al azar. Del padre 1 se copiará en el hijo en la misma posición los elementos entre los 2 puntos de cruce. Del padre 2, se partirá desde el segundo punto de cruce y tomará los elementos que no se encuentren en el hijo y los insertará en el hijo a partir del segundo punto de cruce. Este operador se lo denomina operador de cruce puro (Figura 12). Otra alternativa sería empezar a llenar desde el primer punto de cruce del padre 2, pero el operador no será puro. Del padre 1, el orden relativo, la adyacencia y las posiciones absolutas se conservan. Del padre 2, sólo el orden relativo se conserva [10].

1 - El hijo hereda del padre 1 elementos dentro de los 2 puntos seleccionados



2 - El hijo hereda del padre 2 elementos restantes del padre 2 a partir del segundo punto de cruce

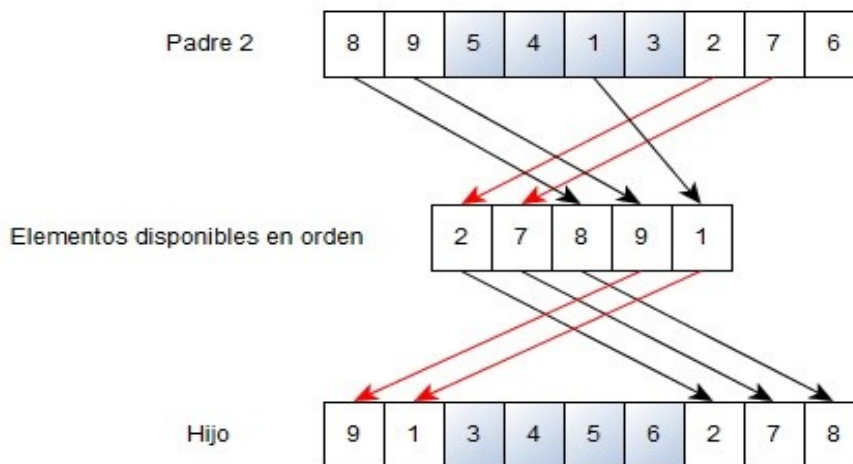
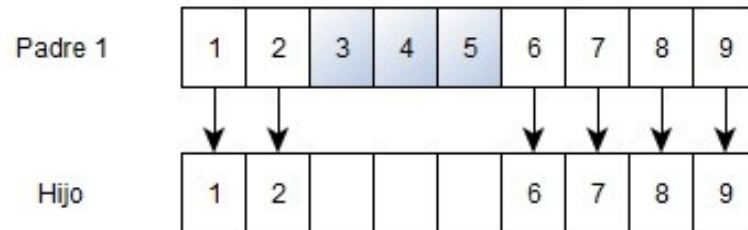


Figura 12: Cruce de Orden [10].

- Cruce en dos puntos (2 Points Crossover, 2PX):** en este operador se seleccionan dos puntos al azar (Figura 13). Los elementos por fuera de los dos puntos seleccionados son heredados del padre 1 al hijo, y los otros elementos son puestos en el orden en que

aparecen en el otro padre, si ya se encuentra el elemento se lo ignora y se coloca el siguiente [3].

1 - El hijo hereda del padre 1 elementos fuera de los 2 puntos seleccionados



2 - El hijo hereda del padre 2 elementos restantes en el orden del padre 2

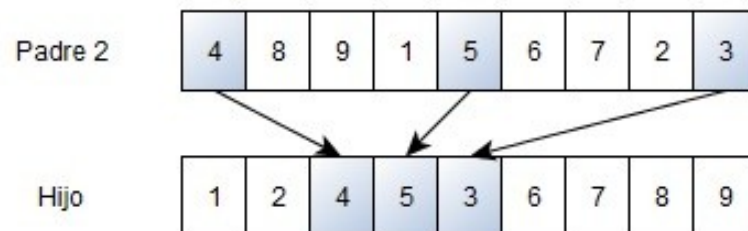


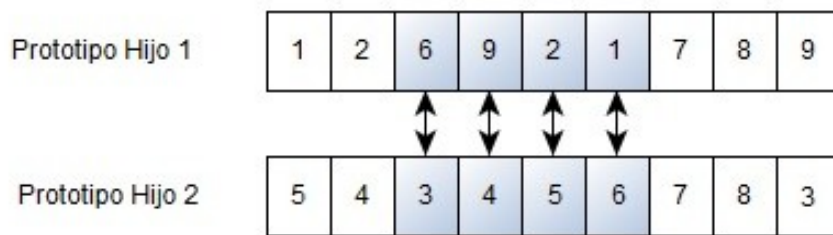
Figura 13: Cruce en dos puntos [10].

- Cruce Parcialmente Mapeado (Partial-Mapped Crossover, PMX):** Este operador puede ser visto como una extensión del 2PX, pero que utiliza un método para reparar las ilegalidades que pueden ser producidas. En esencia es un 2PX con un método de mejora de ilegalidades. Primero se seleccionan dos posiciones al azar, definiendo dos subpermutaciones a partir de los dos padres, llamada secciones de mapeo. Como segundo paso, se intercambian entre los padres estas subpermutaciones generando dos prototipos de hijos. El tercer paso consiste en determinar la relación de mapeo entre las dos secciones de mapeo. Y por último, se legalizan los hijos con la relación de mapeo [10] (Figura 14).
- Cruce de Ciclo (Cycle Crossover, CX):** Este operador selecciona los elementos desde uno de los padres que definan un ciclo el cual es determinado por las posiciones de los elementos entre los padres (Figura 15). Se copian los elementos del ciclo en el hijo respetando las posiciones que tenían en uno de los padres, luego se determinan los elementos restantes del hijo eliminando los elementos del ciclo del segundo padre y se completa el hijo con los elementos restante del segundo padre [10].

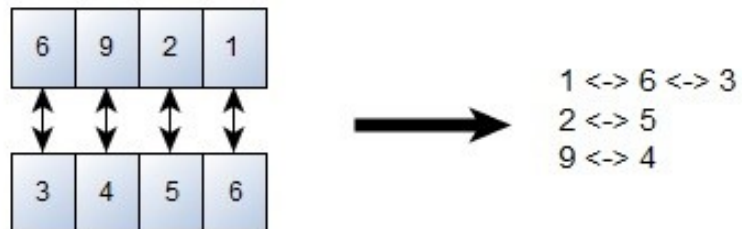
1 - Definir secciones de mapeo al azar



2 - Intercambiar las secciones de mapeo entre los padres



3 - Determinar las relación de mapeo

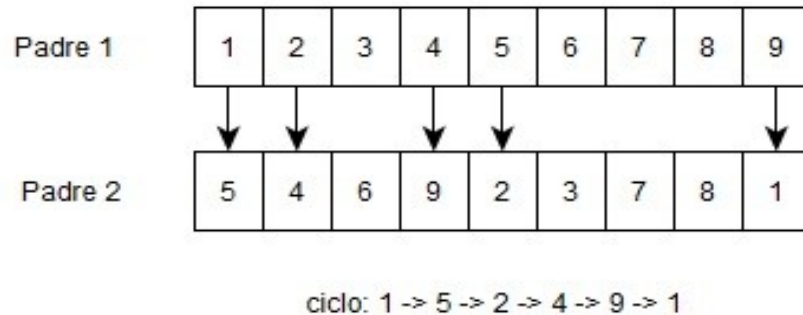


4 - Legalizar los hijos con la relación de mapeo



Figura14: Cruce Parcialmente Mapeado [10].

1 - Encontrar el ciclo definido por los padres



2 - Copiar los elementos del ciclo en el hijo



3 - Determinar los elementos restantes para el hijo



4 - Completar el hijo



Figura 15: Cruce de Ciclo [10].

2.3.1.4.2 Mutación

Tras el cruce, tiene lugar la mutación. Si nos referimos en términos de evolución, la mutación se manifiesta de forma extraordinaria [3]. Las mutaciones suelen en promedio ser beneficiosas ya que contribuyen a la diversidad genética de la especie, evitando caer en óptimos locales. Por lo tanto, la mutación consiste en modificar ciertos genes de forma aleatoria atendiendo a la probabilidad de mutación establecida con anterioridad. La mutación depende de la codificación y de la reproducción. Si se abusa de la mutación podemos caer en el uso del algoritmo genético como una simple búsqueda aleatoria. Por lo tanto antes de aumentar las mutaciones, conviene estudiar otras soluciones que aporten diversidad a la población como

podría ser el aumento del tamaño de la población o garantizar la aleatoriedad de la población inicial. En general, se recomiendan valores pequeños para la probabilidad de mutación ($p_m \in [0.001, 0.01]$). Algunas estrategias utilizan como probabilidad $1/k$ donde k representa el número de variables de decisión.

Para el caso de una codificación binaria, la mutación consiste simplemente en la inversión del gen mutado que corresponderá con un bit. En el caso de una codificación numérica, la mutación podría consistir en sustituir un número por otro o intercambiar un número por otro que está en otra posición del cromosoma. En el caso de codificación por valor directo en el que por ejemplo se utilicen números reales, la mutación puede consistir simplemente en modificar el valor en unos decimales. Por último, en una codificación en árbol, la mutación podría radicar en el cambio de operador, de un número o incluso en la mutación de una rama entera.

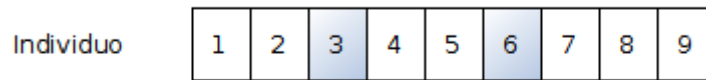
A la hora de diseñar o usar un operador de mutación debemos tener en cuenta:

- Ergodicidad: el operador de mutación debe permitir alcanzar todas las soluciones del espacio de búsqueda.
- Validez: el operador debe siempre proporcionar soluciones válidas. Cuando se trabaja con problemas de optimización con restricciones no siempre se obtienen soluciones válidas.
- Localidad: el cambio producido por el operador debe ser mínimo. El tamaño de la mutación es importante y debe ser controlable. Esta es la principal propiedad de un operador de mutación. Localidad es el efecto en la solución después de que se produzca un cambio en el genotipo. Cuando se producen pequeños cambios en el genotipo y el *fitness* revela pequeños cambios, se dice que la mutación tiene una fuerte localidad. Por lo cual, el algoritmo hará una búsqueda significativa en el espacio de búsqueda del problema. Una localidad débil se caracteriza por tener un alto cambio en el *fitness* cuando se produce un pequeño cambio en el genotipo. En casos extremos, terminará siendo una búsqueda al azar dentro del espacio de búsqueda.

La mutación en los algoritmos evolutivos se relaciona con los operadores de vecindad de metaheurísticas basadas en una simple solución. Por lo tanto, las definiciones de la estructura de vecindad para representaciones tradicionales (por ejemplo, discretos lineales) pueden ser reutilizados como operadores de mutación.

- Mutación en representaciones binarias: la mutación normalmente utilizada se define como el operador golpe brusco: invertir el valor de un gen.
- Mutación en representación discreta: consiste en cambiar el valor asociado a un gen por otro valor en el alfabeto.
- Mutación en permutaciones: la mutación en representaciones basadas en orden son realizadas mediante operadores de intercambio (Figura 16), inversión o inserción (Figura 17).

1 - Se seleccionan 2 elementos al azar



2 - Se intercambian estos dos elementos entre si

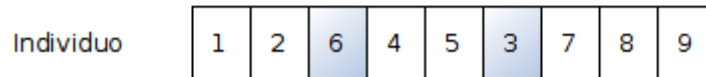
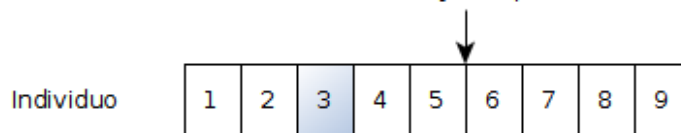


Figura 16: Mutación por Intercambio.

1 - Se selecciona 1 elemento al azar y una posición



2 - Se inserta el elemento, en la posición seleccionada, desplazando los elementos restantes

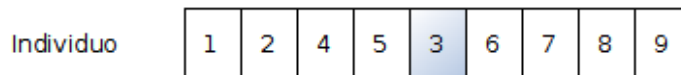


Figura 17: Mutación por Inserción.

En la programación genética donde se utilizan árboles como representaciones, se definirán distintas formas de mutación:

- Agrandar: Se selecciona un nodo terminal al azar y es reemplazado por un subárbol generado al azar (Figura 18).
- Encoger: Se selecciona un nodo interno al azar y es reemplazado por un nodo terminal generado al azar (Figura 18).
- Intercambiar: Se selecciona un nodo interno al azar, dos de sus subárboles son seleccionados al azar y se intercambian sus posiciones dentro del árbol (Figura 19).
- Ciclo: Se selecciona un sólo nodo (interno o terminal) al azar y es reemplazado por un nodo con el mismo número de argumentos (Figura 19).

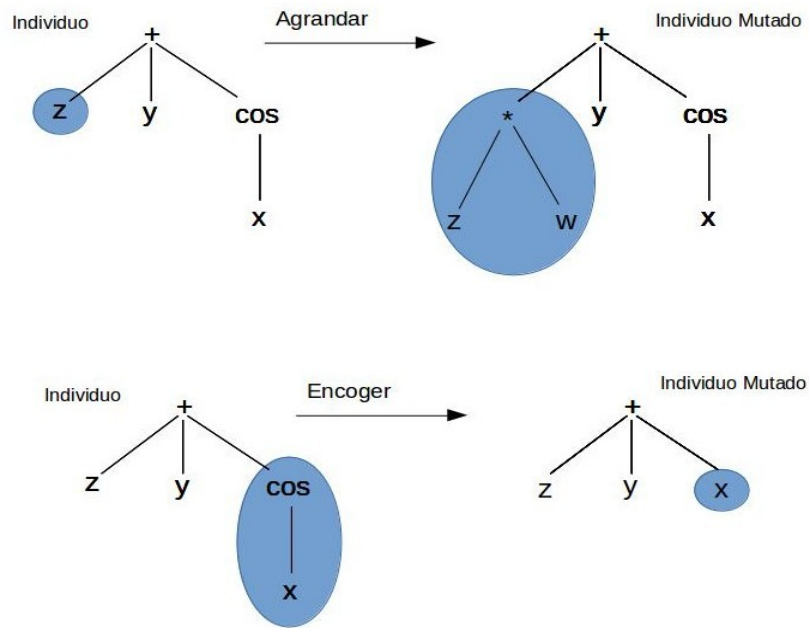


Figura 18: Mutación en Representación de Árbol mediante operadores de Agrandar y Encoger [3].

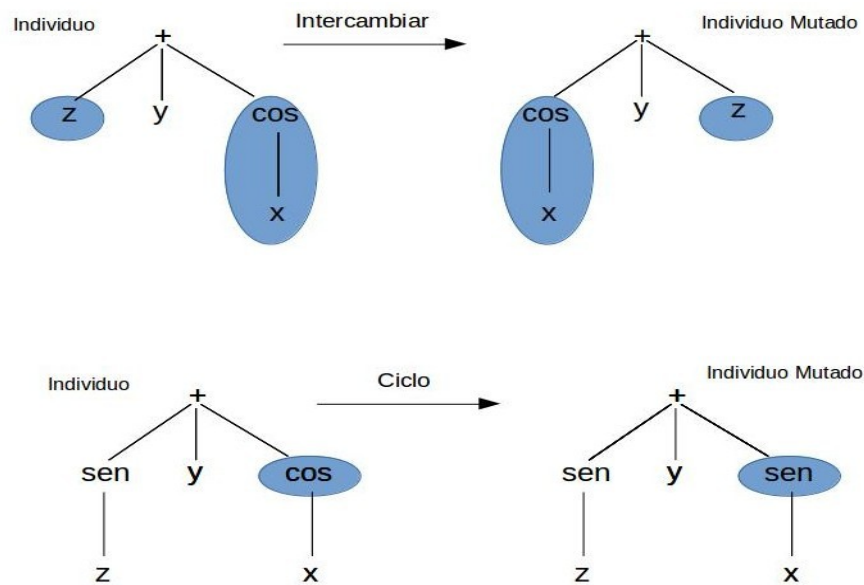


Figura 19: Mutación en Representación de Árbol mediante operadores de Intercambiar y Ciclo [3].

Para vectores de valores reales, hay muchos operadores de mutación distintos. La clase de operadores más usados tienen la forma:

$$x' = x + M$$

Donde M es una variable al azar. El valor de M puede ser tomado de las siguientes formas:

- Mutación uniforme al azar: se genera una variable uniforme al azar en el intervalo $[a, b]$. El parámetro a es en general $-b$. Los descendientes son generados entre el intervalo $x + U(-b, b)$ donde b representa una constante definida por el usuario.
- Mutación de distribución normal: se utiliza una distribución Gaussiana (o normal) $M = N(0, \sigma)$, donde $N(0, \sigma)$ es un vector de números al azar Gaussianos independientes con la media 0 y la desviación estándar σ . Es el esquema de mutación más popular en las estrategias de evolución y programación evolutiva. El parámetro σ es crucial y debe ser seteado correctamente.
- Mutación polinomial: en una mutación polinomial, los descendientes se generan de la siguiente manera:

$$x'_i = x_i + (x_i^u - x_i^L) \delta_i$$

donde x_i^u representa el límite superior y x_i^L representa el límite inferior para x_i . El parámetro δ_i es calculado por la probabilidad de distribución polinomial:

$$P(\delta) = 0.5(\eta_m + 1) * (1 - (|\delta|)^{\eta_m})$$

$$\delta_i = \begin{cases} (2r)^{\frac{1}{\eta_m + 1}} - 1 & \text{si } r_i < 0.5 \\ 1 - (2(1 - r_i))^{\frac{1}{\eta_m + 1}} & \text{otro caso} \end{cases}$$

donde η_m representa el índice de distribución ($\eta_m \approx 20$ es usado generalmente) y r_i es un número al azar entre $[0, 1]$.

- Existen otros operadores de mutación como la distribución de Cauchy y la distribución de Laplace.

2.3.1.5 Estrategias de Reemplazo

La fase de reemplazo se refiere a la selección sobreviviente de los padres y los descendientes. Como el tamaño de la población es constante, permite retirar individuos de acuerdo a una estrategia de selección dada. Las estrategias de sustitución extremas:

- **Reemplazo generacional:** el reemplazo se realizará sobre toda la población de tamaño μ . Los descendientes reemplazarán a los padres. Esta estrategia fue propuesta por J. Holland [2] para ser aplicada a algoritmos genéticos canónicos.
- **Reemplazo de estado estacionario:** en cada generación de un algoritmo evolutivo, se genera un solo descendiente [3]. Por ejemplo, reemplaza el peor individuo de la población de los padres.
- **Reemplazo Elitista:** consiste en la selección de los mejores individuos entre los padres y los descendientes. Esto conduce a una convergencia más rápida y una convergencia

prematura. A veces, la selección de los malos individuos es necesario para evitar el problema de error de muestreo [3].

- **Reemplazo Mu + Lambda ($\mu+\lambda$):** en este reemplazo μ padres producen λ descendientes reduciéndose nuevamente la población a μ individuos. Esta reducción a μ individuos se puede realizar mediante diferentes técnicas de selección de individuos [3].

2.3.1.6 Condición de Corte

La condición de corte de la fase iterativa del algoritmo genético usualmente toma en cuenta la cantidad de generaciones procesadas, deteniendo el ciclo evolutivo al alcanzar un número prefijado de generaciones [3].

Otras alternativas consideran la variación de los valores de *fitness*, deteniendo el ciclo evolutivo cuando el proceso se estanca y no obtiene mejoras considerables en los valores de *fitness*, o estimaciones del error cometido respecto al valor óptimo del problema o una aproximación, en caso de conocerse, implicando un alto grado de complejidad a la hora de elaborar la condición de corte.

Capítulo 3: Experimentación

En este capítulo se introducen y explican los algoritmos genéticos implementados para resolver el FSSP. También se describe la metodología de experimentación y, por último, los resultados obtenidos de las diferentes experimentaciones.

3.1 Algoritmo Genético implementado

El lenguaje en el cual se programó el algoritmo fue en C++ [13], por ser un lenguaje que permite obtener códigos ejecutables que hacen que los tiempos de ejecución disminuyan, además de permitir el uso de la memoria dinámica. A su vez, por ser orientado a objetos permite un análisis y diseño del problema con esta metodología (Figura 20).

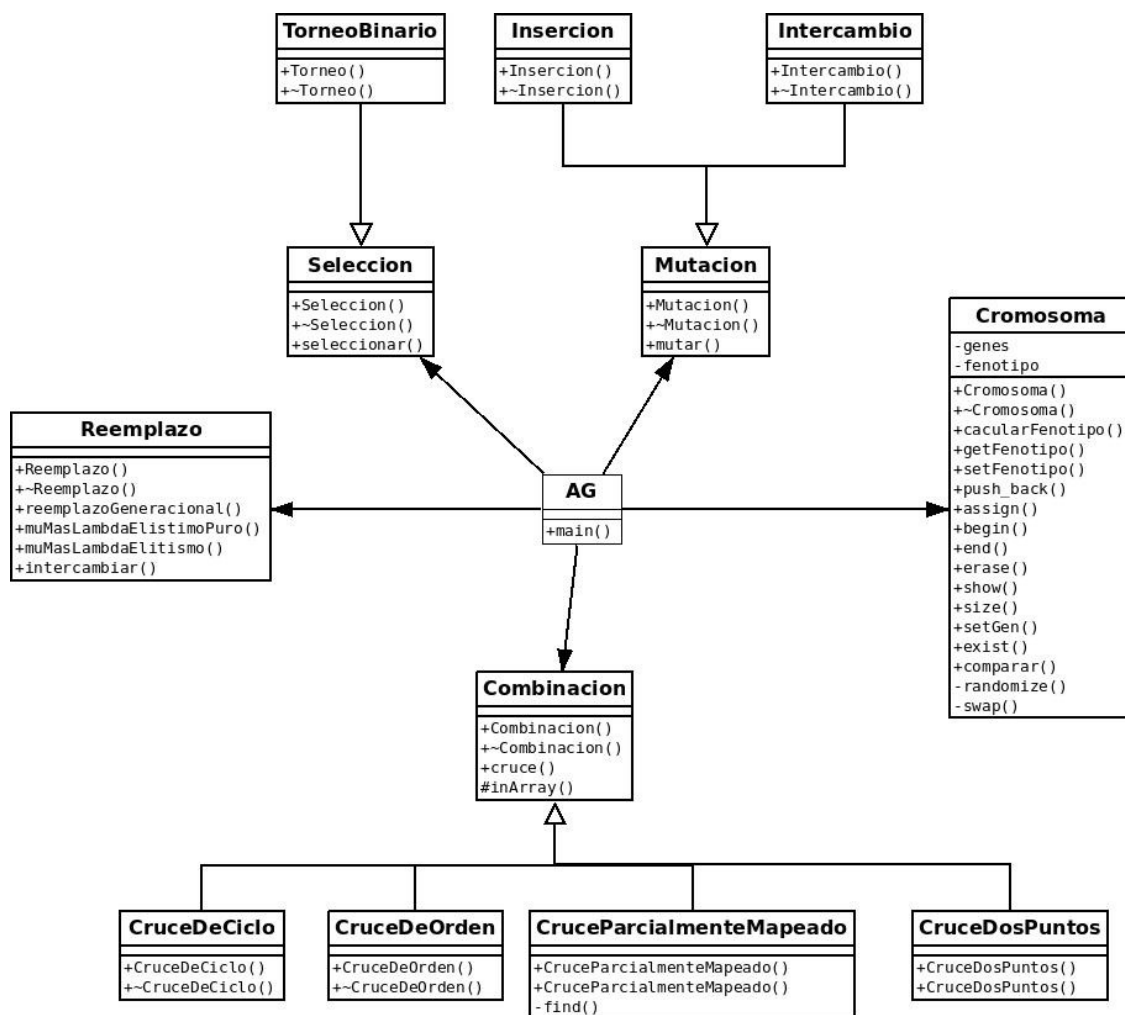


Figura 20: Diagrama de clases del algoritmo implementado

Con el objetivo de resolver el FSSP se diseñan e implementan una serie de algoritmos genéticos basados en la representación por permutación. Este tipo de representación es útil para

resolver el FSSP, ya que esencialmente es un problema de permutación de tareas, podemos representar la permutación de tareas como la permutación de genes dentro de un cromosoma. Por ejemplo, una representación de una secuencias de 4 tareas es:

4	2	1	3
---	---	---	---

Figura 21: Representación de una permutación de tareas del FSSP.

El orden de la secuencia de tareas, indica el orden en el cual se van a procesar en las diferentes máquinas. Por ejemplo el diagrama de Gantt representando estas 4 tareas procesadas en 2 máquinas sería:

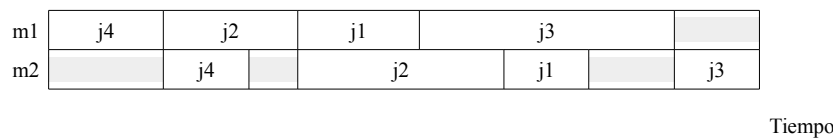


Figura 22: Diagrama de Gantt para un FSSP para 4 tareas y 2 máquinas.

En el Capítulo 2 se explicó que es necesario escoger los operadores genéticos (cruce y mutación) según la codificación (o representación) usada. Dado que los GAs propuestos usan la permutación como codificación, los operadores elegidos son: Cruce de Orden, Cruce de Ciclo, Cruce en Dos Puntos y Cruce Parcialmente Mapeado; y los siguientes operadores de mutación: Intercambio e Inserción.

En tanto que, tres formas diferentes de generar la población son implementadas en los GAs propuestos. La primera de ellas es la forma tradicional, es decir, toda la población es generada en forma aleatoria. La segunda forma genera la mitad de la población al azar y al otra mitad usando la metaheurística NEHH (explicada en el Capítulo 2). La tercera y última manera de generar la población utiliza NEHH para crear todos los individuos de la población.

Luego de evaluar el desempeño de los algoritmos genéticos con 32, 64, 128 y 256 individuos, se elige una población de 64 soluciones. Esto se debe a que establece la mejor relación entre calidad y esfuerzo computacional.

Una vez generada la población inicial, se seleccionan los individuos (padres) para aplicar los operadores genéticos mencionados, esta selección se realizará mediante Torneo Binario. Luego de aplicar los operadores genéticos se debe reemplazar la población para poder continuar el ciclo nuevamente. El reemplazo de la población se realiza de tres formas diferentes: Recambio Generacional, Recambio $\mu + \lambda$ con Elitismo (seleccionando los 64 mejores individuos) y $\mu + \lambda$ con Elitismo utilizando selección mediante Torneo Binario (seleccionando el mejor individuo y los restantes 63 son seleccionados por Torneo Binario). Por último la condición de corte fué configurada en 1000 iteraciones, determina la cantidad de iteraciones que se realizan para terminar el ciclo. Todas estas variantes implementadas dan origen a diferentes versiones de GAs como se resumen en la Tabla 1.

Algoritmo	Inicialización de Pob. Inicial	Selección de Padres	Operador de Cruce	Operador de Mutación	Recambio
AG 1	100% Azar	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 2	50% NEHH + 50% Azar	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 3	100% NEHH	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 4	100% Azar	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 5	50% NEHH + 50% Azar	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 6	100% NEHH	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 7	100% Azar	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Elitismo
AG 8	50% NEHH + 50% Azar	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Elitismo
AG 9	100% NEHH	Torneo Binario	OX	Intercambio	$\mu+\lambda$ con Elitismo
AG 10	100% Azar	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Elitismo
AG 11	50% NEHH + 50% Azar	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Elitismo
AG 12	100% NEHH	Torneo Binario	OX	Inserción	$\mu+\lambda$ con Elitismo
AG 13	100% Azar	Torneo Binario	OX	Intercambio	Generacional
AG 14	50% NEHH + 50% Azar	Torneo Binario	OX	Intercambio	Generacional
AG 15	100% NEHH	Torneo Binario	OX	Intercambio	Generacional
AG 16	100% Azar	Torneo Binario	OX	Inserción	Generacional
AG 17	50% NEHH + 50% Azar	Torneo Binario	OX	Inserción	Generacional
AG 18	100% NEHH	Torneo Binario	OX	Inserción	Generacional
AG 19	100% Azar	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 20	50% NEHH + 50% Azar	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 21	100% NEHH	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 22	100% Azar	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 23	50% NEHH + 50% Azar	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 24	100% NEHH	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 25	100% Azar	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Elitismo
AG 26	50% NEHH + 50% Azar	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Elitismo
AG 27	100% NEHH	Torneo Binario	CX	Intercambio	$\mu+\lambda$ con Elitismo
AG 28	100% Azar	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Elitismo
AG 29	50% NEHH + 50% Azar	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Elitismo
AG 30	100% NEHH	Torneo Binario	CX	Inserción	$\mu+\lambda$ con Elitismo
AG 31	100% Azar	Torneo Binario	CX	Intercambio	Generacional
AG 32	50% NEHH + 50% Azar	Torneo Binario	CX	Intercambio	Generacional
AG 33	100% NEHH	Torneo Binario	CX	Intercambio	Generacional
AG 34	100% Azar	Torneo Binario	CX	Inserción	Generacional
AG 35	50% NEHH + 50% Azar	Torneo Binario	CX	Inserción	Generacional
AG 36	100% NEHH	Torneo Binario	CX	Inserción	Generacional
AG 37	100% Azar	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 38	50% NEHH + 50% Azar	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 39	100% NEHH	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 40	100% Azar	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 41	50% NEHH + 50% Azar	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 42	100% NEHH	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 43	100% Azar	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Elitismo
AG 44	50% NEHH + 50% Azar	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Elitismo

AG 45	100% NEHH	Torneo Binario	PMX	Intercambio	$\mu+\lambda$ con Elitismo
AG 46	100% Azar	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Elitismo
AG 47	50% NEHH + 50% Azar	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Elitismo
AG 48	100% NEHH	Torneo Binario	PMX	Inserción	$\mu+\lambda$ con Elitismo
AG 49	100% Azar	Torneo Binario	PMX	Intercambio	Generacional
AG 50	50% NEHH + 50% Azar	Torneo Binario	PMX	Intercambio	Generacional
AG 51	100% NEHH	Torneo Binario	PMX	Intercambio	Generacional
AG 52	100% Azar	Torneo Binario	PMX	Inserción	Generacional
AG 53	50% NEHH + 50% Azar	Torneo Binario	PMX	Inserción	Generacional
AG 54	100% NEHH	Torneo Binario	PMX	Inserción	Generacional
AG 55	100% Azar	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 56	50% NEHH + 50% Azar	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 57	100% NEHH	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Torneo Binario
AG 58	100% Azar	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 59	50% NEHH + 50% Azar	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 60	100% NEHH	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Torneo Binario
AG 61	100% Azar	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Elitismo
AG 62	50% NEHH + 50% Azar	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Elitismo
AG 63	100% NEHH	Torneo Binario	2PX	Intercambio	$\mu+\lambda$ con Elitismo
AG 64	100% Azar	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Elitismo
AG 65	50% NEHH + 50% Azar	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Elitismo
AG 66	100% NEHH	Torneo Binario	2PX	Inserción	$\mu+\lambda$ con Elitismo
AG 67	100% Azar	Torneo Binario	2PX	Intercambio	Generacional
AG 68	50% NEHH + 50% Azar	Torneo Binario	2PX	Intercambio	Generacional
AG 69	100% NEHH	Torneo Binario	2PX	Intercambio	Generacional
AG 70	100% Azar	Torneo Binario	2PX	Inserción	Generacional
AG 71	50% NEHH + 50% Azar	Torneo Binario	2PX	Inserción	Generacional
AG 72	100% NEHH	Torneo Binario	2PX	Inserción	Generacional

Tabla 1: GAs propuestos para la resolución del FSSP.

La función de evaluación (*fitness* del individuo) utilizada para realizar la comparación entre dos individuos es el *makespan*, descrito en el Capítulo 1. Por lo tanto, entre dos individuos el de menor *makespan* será el mejor.

3.2 Metodología experimental

En esta sección, se describe la metodología utilizada para llevar a cabo las experimentaciones con cada algoritmo de la Tabla 1 para resolver las instancias de Taillard (OR-Library [17]) más complejas del problema en cuestión. La complejidad para FSSP está dado por el número de jobs y de máquinas. En consecuencia se eligieron las instancias con 200 tareas y 10 máquinas y 500 tareas y 20 máquinas, cuyos mejores valores conocidos están disponibles en la OR-Library ([10676, 10616] y [26527, 26315], respectivamente). Para las experimentaciones se utiliza una PC con las siguientes características de software y hardware: procesador Intel Core i5-3317U, 2,6GHz, memoria RAM de 4GB DDR3 y Ubuntu 12.04.2 LTS (núcleo 3.13.0-45-generic) como sistema operativo.

Dadas las características del equipo sobre el que se hizo la experimentación, se realizaron ejecuciones, de diferentes algoritmos, en paralelo. Los resultados de cada algoritmo se insertaban en archivos CSV (Anexo 1). Luego por cada archivo se tabularon los datos con el fin de realizar diagramas de caja [16], estos diagramas permiten analizar rápidamente un conjunto de datos. Con la información obtenida de este análisis se confeccionó la tabla 2 del Anexo 2. En base a estos datos se determinan las mejores configuraciones para los algoritmos propuestos, como puede observarse en la tabla 2. De esta manera se seleccionaron los algoritmos AG1, AG2, AG3, AG4, AG5, AG6, AG19, AG20, AG21, AG22, AG23 y AG24, con las siguientes probabilidades de cruce 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85 y 0.9; y de mutación 0.001, 0.005, 0.01, 0.05 y 0.1. Cabe aclarar que los algoritmos seleccionados tienen en común el método de reemplazo $\mu + \lambda$ con Elitismo y el de selección mediante Torneo Binario.

El análisis de los resultados se lleva a cabo desde dos puntos de vista: la calidad de los resultados y el esfuerzo computacional empleado por los algoritmos para hallar dichos resultados. En cuanto a la calidad, se estudia la diferencia relativa de las soluciones encontradas con respecto a la mejor solución conocida, como muestra la siguiente ecuación:

$$DR = (se - me) / me$$

donde DR es la distancia relativa, *se* es la solución encontrada y *me* es la mejor solución publicada en la literatura para una determinada instancia. Si esta diferencia es cero o es muy próxima a cero significa que el algoritmo encontró las mejores soluciones conocidas para esa instancia. En cambio si la diferencia relativa se aleja de cero, es posible medir el error cometido por el algoritmo al intentar hallar la mejor solución.

En cuanto al esfuerzo computacional se realiza un análisis sobre el tiempo que tarda el AG en encontrar la mejor solución y en completar la búsqueda. También se analiza la cantidad de evaluaciones que realiza. El objetivo es hallar el algoritmo que obtenga la mejor calidad de resultados con el menor esfuerzo computacional.

3.3 Análisis de Resultados

En esta sección se presentan y analizan los resultados obtenidos de los algoritmos genéticos implementados para resolver el FSSP. El análisis se aborda desde dos perspectivas diferentes: la calidad y el esfuerzo computacional.

Debido a la gran cantidad de datos obtenidos en la ejecución de los algoritmos, para analizar los resultados se los agrupa primero según el método utilizado para inicializar la población inicial: Azar, NEHH+Azar y NEHH (Figuras 22 a 27). Luego se analiza cuál de los métodos usados para generar la población inicial obtienen mejores resultados tanto en calidad como en esfuerzo computacional. De este conjunto (Tablas 2 a 8) de datos se analizará el operador de cruce, operador de mutación y configuración para la cual se obtiene mejores resultados evaluando calidad y esfuerzo computacional.

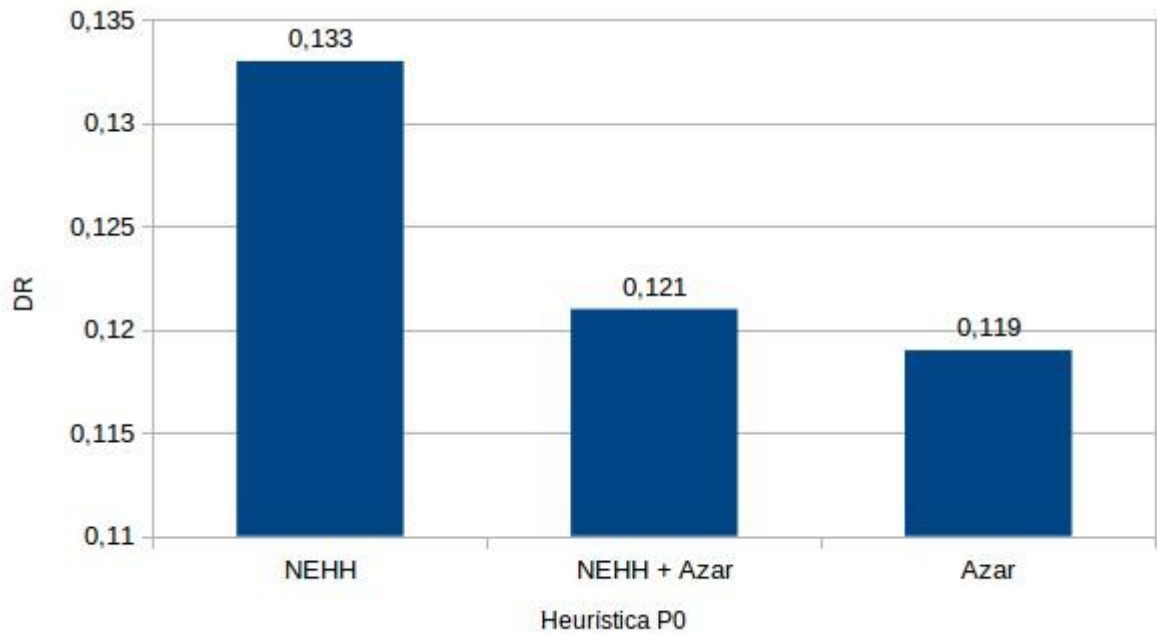


Figura 23: Desvío Relativo promedio de los mejores fitness en la instancia 200x10

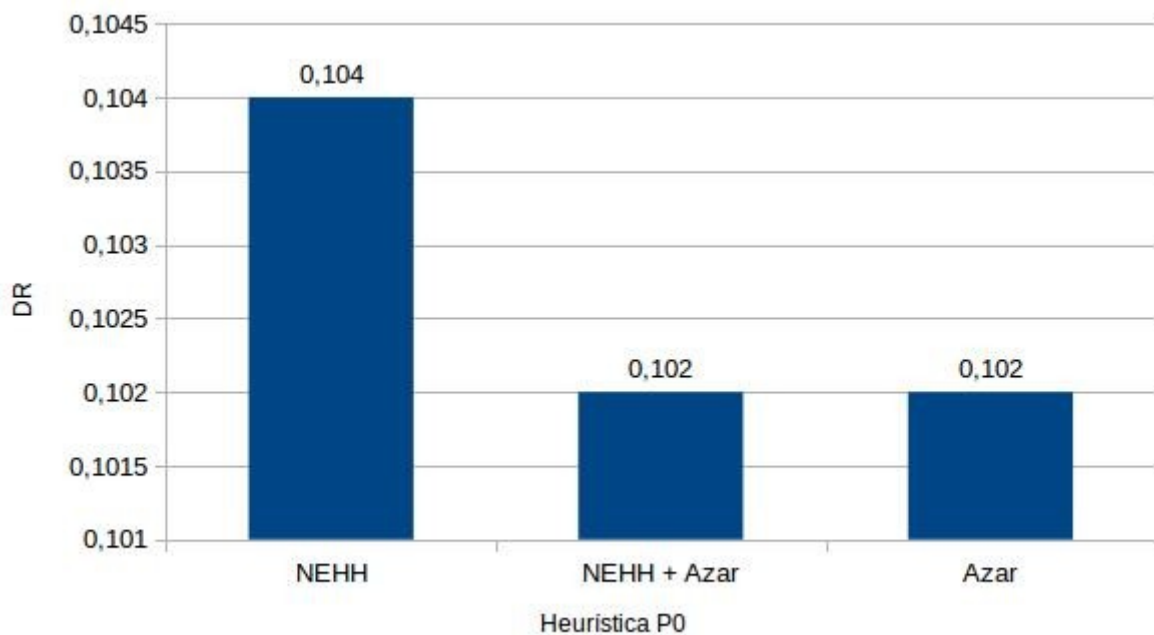


Figura 24: Desvío Relativo promedio de los mejores fitness en la instancia 500x20

La evaluación de calidad del algoritmo para resolver la instancia de 200 tareas y 10 máquinas realizada a partir de la Figura 23, sugiere que los resultados de mejor calidad (DR próximo a 0) se obtienen cuando el algoritmo inicializa la población al azar. En cuanto al análisis del algoritmo para resolver la instancia de 500 tareas y 20 máquinas que se puede realizar a partir de la Figura 24, es que se obtienen resultados de mejor calidad cuando el algoritmo inicializa la población utilizando las estrategias de inicio: Azar y NEHH+Azar.

También se observa que cuando el algoritmo crea una población inicial usando sólo NEHH el operador de mutación no es capaz de generar la diversidad suficiente durante la búsqueda, y en consecuencia el algoritmo converge a óptimos locales.

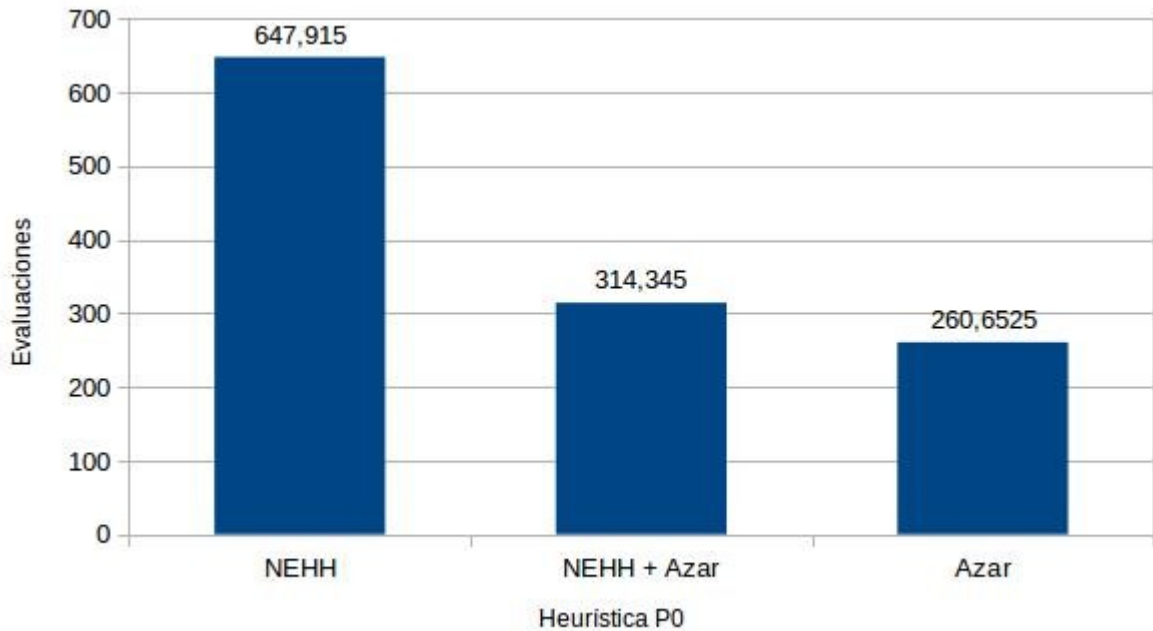


Figura 25: Cantidad de evaluaciones promedio de los mejores fitness en la instancia 200x10

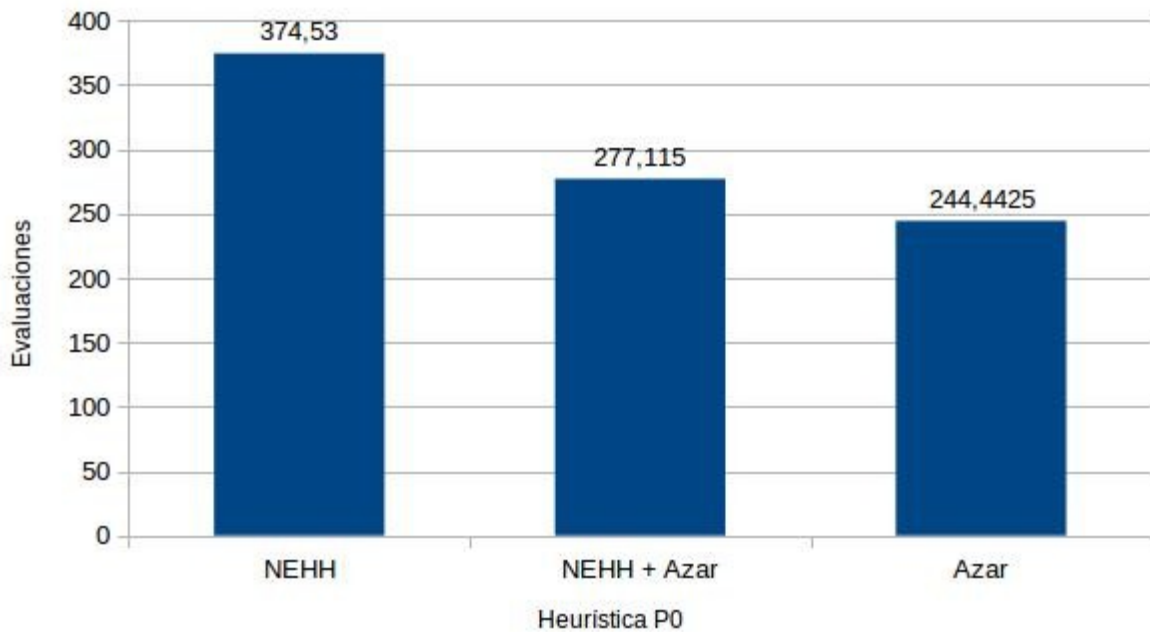


Figura 26: Cantidad de evaluaciones promedio de los mejores fitness en la instancia 500x20

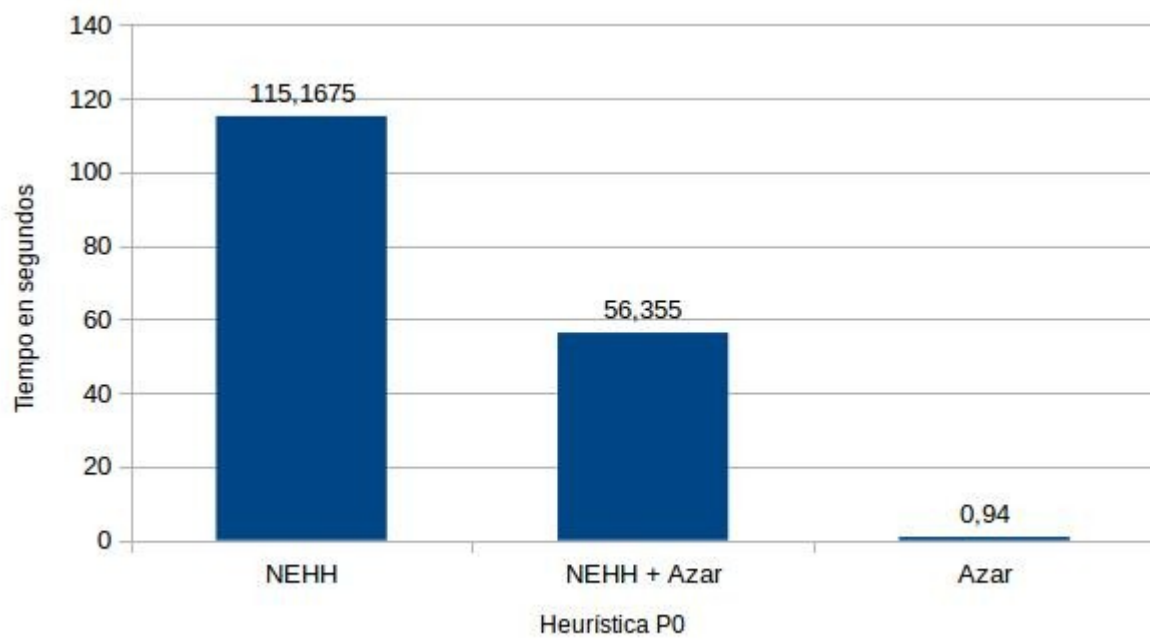


Figura 27: Tiempo en promedio de los mejores fitness en la instancia 200x10

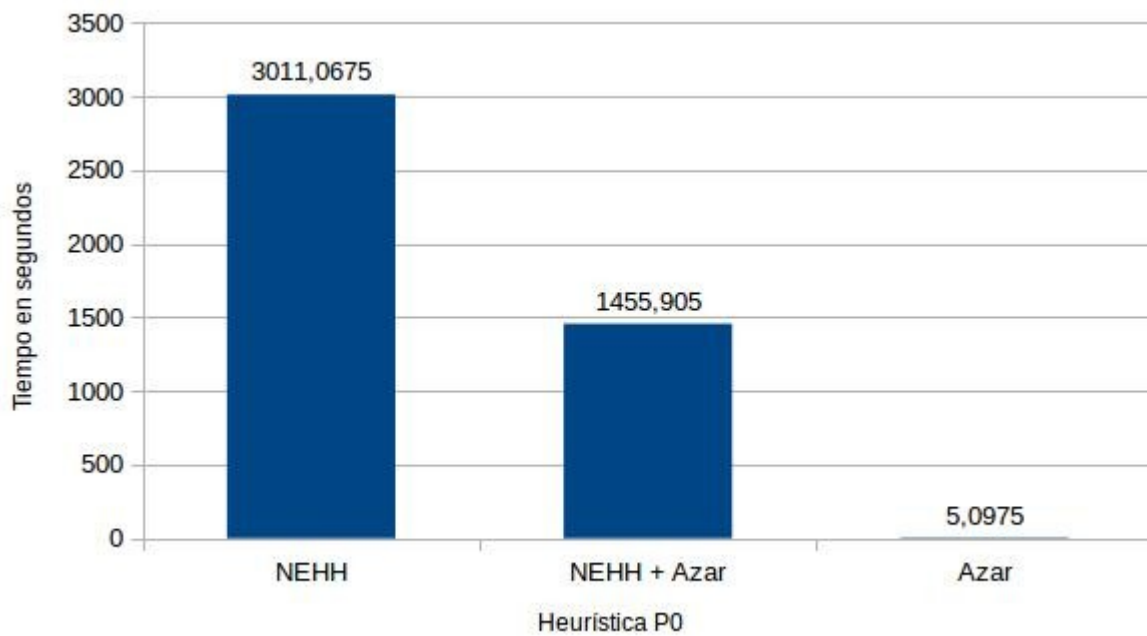


Figura 28: Tiempo en promedio de los mejores fitness en la instancia 500x20

En cuanto a la evaluación del esfuerzo computacional para hallar soluciones de calidad, la forma de inicializar el algoritmo que requiere menos esfuerzo computacional es la que genera aleatoriamente la totalidad de la población al azar. Esto se debe a que la heurística NEHH utiliza un algoritmo que compara todos los elementos del individuo para poder generarlo.

El primer análisis de calidad del algoritmo nos indica que la mejor forma de inicializar la población depende de la complejidad de la instancia, para instancias de complejidad moderada el Azar es la forma adecuada, pero para instancias complejas el Azar o el NEHH+Azar, proporcionan resultados de calidad.

En las Tablas 2, 3, y 4 se muestran los valores de los mejores DR promedio para cada probabilidad de cruce, resaltados con negrita, los valores de los mejores DR promedio para cada probabilidad de mutación, celda rellena, y por último un recuadro al mejor valor de DR promedio encontrado. A partir de estos datos se analiza cuáles son los mejores operadores de cruce y mutación, y sus probabilidades para cada instancia. Se puede observar que independientemente del método utilizado para generar la población inicial, el operador de cruce que mejores resultados proporciona, desde el punto de vista de calidad, es el Cruce de Ciclo. Esto se debe a que el FSSP es un problema de permutación de tareas y este operador mantiene los ciclos definidos entre los padres, obteniendo así mejores resultados. A su vez, de este análisis se obtienen que los mejores resultados se hallan con las probabilidades de cruce 0,8 a 0,9 y las probabilidades de mutación 0,05 a 0,1. En relación al operador de mutación que obtiene mejores resultados para la instancia de 200 tareas y 10 máquinas, es el de Intercambio y en la instancia de 500 tareas y 20 máquinas es el operador de Inserción, aunque estas diferencias no son estadísticamente significativas. El hecho de que un operador de mutación permite obtener mejores soluciones que otro indica que es capaz de realizar una mejor exploración del espacio de búsqueda.

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	0,124	0,121	0,120	0,119	0,117	0,124	0,119	0,120	0,120	0,116
OX	0,55	0,122	0,120	0,122	0,120	0,118	0,121	0,120	0,123	0,118	0,118
OX	0,6	0,119	0,120	0,121	0,119	0,117	0,120	0,122	0,123	0,119	0,118
OX	0,65	0,123	0,122	0,120	0,118	0,120	0,123	0,122	0,121	0,118	0,118
OX	0,7	0,123	0,122	0,122	0,120	0,120	0,125	0,120	0,122	0,116	0,116
OX	0,75	0,121	0,121	0,124	0,119	0,119	0,122	0,121	0,121	0,118	0,117
OX	0,8	0,122	0,120	0,122	0,119	0,116	0,125	0,123	0,123	0,120	0,116
OX	0,85	0,123	0,120	0,123	0,119	0,118	0,123	0,121	0,121	0,118	0,118
OX	0,9	0,120	0,123	0,123	0,119	0,116	0,121	0,123	0,122	0,119	0,116
CX	0,5	0,117	0,121	0,120	0,119	0,116	0,118	0,120	0,119	0,117	0,117
CX	0,55	0,117	0,118	0,117	0,117	0,115	0,118	0,119	0,119	0,118	0,117
CX	0,6	0,118	0,115	0,117	0,118	0,116	0,117	0,118	0,119	0,115	0,117
CX	0,65	0,118	0,118	0,119	0,118	0,116	0,119	0,121	0,119	0,115	0,117
CX	0,7	0,117	0,118	0,115	0,118	0,117	0,121	0,118	0,118	0,117	0,115
CX	0,75	0,121	0,118	0,117	0,117	0,116	0,122	0,119	0,118	0,115	0,116
CX	0,8	0,118	0,117	0,117	0,117	0,115	0,121	0,117	0,117	0,117	0,116
CX	0,85	0,120	0,119	0,119	0,116	0,115	0,118	0,119	0,119	0,114	0,115
CX	0,9	0,119	0,118	0,117	0,115	0,116	0,119	0,117	0,120	0,117	0,116

Tabla 2: Valores de DR promedio de los mejores fitness de la instancia 200x10 con heurística de P0 Azar

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	0,104	0,104	0,104	0,102	0,102	0,104	0,103	0,103	0,103	0,102
OX	0,55	0,104	0,103	0,105	0,103	0,101	0,104	0,103	0,103	0,103	0,101
OX	0,6	0,103	0,103	0,103	0,102	0,101	0,104	0,102	0,103	0,101	0,101
OX	0,65	0,103	0,104	0,103	0,102	0,102	0,104	0,104	0,104	0,103	0,103
OX	0,7	0,104	0,103	0,103	0,102	0,102	0,104	0,104	0,102	0,103	0,101
OX	0,75	0,106	0,103	0,104	0,102	0,101	0,104	0,104	0,104	0,103	0,101
OX	0,8	0,102	0,104	0,103	0,102	0,102	0,104	0,104	0,103	0,102	0,102
OX	0,85	0,104	0,105	0,102	0,102	0,102	0,105	0,103	0,103	0,102	0,100
OX	0,9	0,104	0,104	0,104	0,103	0,101	0,105	0,104	0,104	0,103	0,101
CX	0,5	0,102	0,102	0,102	0,102	0,101	0,103	0,101	0,102	0,102	0,101
CX	0,55	0,103	0,102	0,102	0,102	0,101	0,103	0,103	0,104	0,100	0,101
CX	0,6	0,103	0,102	0,100	0,101	0,101	0,101	0,105	0,102	0,101	0,101
CX	0,65	0,102	0,101	0,102	0,101	0,101	0,101	0,103	0,101	0,100	0,101
CX	0,7	0,102	0,102	0,102	0,100	0,100	0,103	0,101	0,103	0,101	0,101
CX	0,75	0,101	0,101	0,101	0,101	0,101	0,103	0,101	0,103	0,103	0,101
CX	0,8	0,101	0,102	0,100	0,101	0,100	0,102	0,103	0,102	0,099	0,101
CX	0,85	0,103	0,101	0,100	0,101	0,101	0,101	0,103	0,100	0,100	0,100
CX	0,9	0,102	0,102	0,102	0,099	0,100	0,103	0,103	0,102	0,100	0,099

Tabla3: Valores de DR promedio de los mejores fitness de la instancia 500x20 con heurística de P0 Azar

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	0,104	0,103	0,102	0,102	0,102	0,103	0,102	0,102	0,102	0,101
OX	0,55	0,104	0,104	0,103	0,102	0,102	0,103	0,102	0,102	0,102	0,099
OX	0,6	0,102	0,105	0,104	0,101	0,103	0,102	0,102	0,101	0,102	0,101
OX	0,65	0,104	0,103	0,103	0,102	0,101	0,103	0,102	0,102	0,102	0,101
OX	0,7	0,103	0,104	0,102	0,102	0,101	0,104	0,104	0,102	0,102	0,101
OX	0,75	0,104	0,103	0,103	0,101	0,102	0,102	0,103	0,100	0,102	0,101
OX	0,8	0,103	0,103	0,104	0,101	0,102	0,104	0,101	0,103	0,100	0,101
OX	0,85	0,104	0,103	0,103	0,100	0,101	0,103	0,103	0,103	0,102	0,100
OX	0,9	0,103	0,102	0,104	0,102	0,102	0,103	0,101	0,104	0,103	0,102
CX	0,5	0,104	0,105	0,104	0,103	0,102	0,103	0,102	0,101	0,099	0,100
CX	0,55	0,104	0,104	0,104	0,103	0,102	0,103	0,103	0,101	0,099	0,098
CX	0,6	0,104	0,104	0,103	0,103	0,102	0,103	0,102	0,102	0,100	0,099
CX	0,65	0,105	0,104	0,102	0,104	0,103	0,102	0,102	0,100	0,100	0,099
CX	0,7	0,105	0,104	0,104	0,102	0,103	0,102	0,102	0,101	0,100	0,100
CX	0,75	0,106	0,105	0,103	0,101	0,102	0,102	0,102	0,101	0,101	0,098
CX	0,8	0,103	0,104	0,104	0,102	0,102	0,101	0,102	0,102	0,100	0,099
CX	0,85	0,104	0,103	0,102	0,101	0,101	0,101	0,103	0,099	0,098	0,097
CX	0,9	0,106	0,102	0,102	0,103	0,101	0,102	0,103	0,102	0,099	0,100

Tabla 4: Valores de DR promedio de los mejores fitness de la instancia 500x20 con heurística de P0 NEHH+Azar

Analizando particularmente la instancia 500x20 (Tablas 3 y 4), se observa que el DR más próximo a cero, y por lo tanto una solución de mejor calidad, se obtiene inicializando la población con la técnica NEHH+Azar.

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	98,2	173,0	152,8	330,9	363,6	128,3	179,8	265,7	376,3	404,4
OX	0,55	138,3	171,2	229,4	406,6	438,5	180,0	155,0	375,3	345,7	490,1
OX	0,6	124,9	164,5	193,4	242,0	392,4	109,5	166,5	168,9	454,6	374,3
OX	0,65	152,4	182,2	261,7	389,5	392,6	196,1	225,0	308,3	411,0	438,7
OX	0,7	138,1	202,0	190,6	267,4	515,2	132,6	213,6	159,8	424,3	520,4
OX	0,75	203,1	167,9	223,0	363,0	398,8	103,4	146,0	235,5	337,9	425,9
OX	0,8	134,8	149,6	203,3	395,0	376,6	153,5	155,1	189,2	361,5	424,8
OX	0,85	179,3	254,2	231,3	295,1	463,2	119,9	233,8	296,9	380,4	391,4
OX	0,9	136,2	164,0	212,0	448,3	416,0	140,8	199,4	239,5	393,4	510,5
CX	0,5	213,4	165,9	217,0	400,5	280,9	157,2	162,9	260,6	411,8	368,5
CX	0,55	220,8	187,2	188,0	250,0	367,9	134,3	157,8	142,3	347,1	365,0
CX	0,6	204,3	207,7	145,4	251,3	210,4	150,1	217,9	202,4	300,9	409,6
CX	0,65	232,5	169,7	160,0	338,2	303,7	140,5	131,6	167,5	326,6	457,7
CX	0,7	203,4	195,0	157,7	324,4	459,0	179,8	172,7	208,7	427,3	384,2
CX	0,75	216,6	170,2	265,1	277,9	346,2	167,1	146,8	183,8	360,9	410,9
CX	0,8	296,9	172,6	196,0	248,4	283,1	228,4	154,2	158,5	315,8	457,1
CX	0,85	162,0	168,8	196,2	192,7	326,5	178,1	146,3	250,2	351,5	452,9
CX	0,9	213,5	192,7	198,7	352,4	371,4	187,9	184,5	156,8	372,0	397,1

Tabla 5: Cantidad de evaluaciones promedio para encontrar la mejor solución en la instancia 200x10 con heurística de PO Azar

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	165,4	139,3	183,5	259,7	384,5	179,9	133,7	181,1	285,3	469,1
OX	0,55	160,5	170,9	120,9	317,4	427,2	139,2	165,1	130,1	293,3	382,6
OX	0,6	184,5	141,1	210,9	308,2	395,2	163,6	113,0	163,6	303,2	387,5
OX	0,65	187,7	203,8	199,2	274,1	372,5	148,7	167,9	183,7	361,2	528,2
OX	0,7	133,0	200,1	170,3	237,8	328,4	108,5	158,7	154,7	299,3	439,7
OX	0,75	123,7	207,9	141,5	275,6	389,8	129,7	224,4	316,5	314,8	358,3
OX	0,8	154,8	169,7	189,8	491,7	341,4	124,1	161,1	294,6	238,7	400,1
OX	0,85	149,0	152,4	230,1	269,9	353,9	106,9	186,5	176,8	351,2	540,2
OX	0,9	146,6	135,5	189,6	301,4	313,8	130,6	121,1	190,1	442,4	392,7
CX	0,5	268,8	189,1	180,0	280,0	365,5	176,1	146,2	162,9	411,5	225,8
CX	0,55	194,8	137,5	232,1	322,0	388,0	175,2	138,0	195,3	254,8	459,7
CX	0,6	235,7	176,7	197,9	256,8	338,9	240,1	208,6	198,6	338,4	358,2
CX	0,65	324,6	222,1	225,4	213,4	348,6	160,5	173,7	170,7	295,0	402,0
CX	0,7	122,7	179,5	294,5	256,1	309,5	160,1	157,7	155,1	222,7	311,1
CX	0,75	235,4	237,9	196,1	264,3	292,9	211,9	162,7	248,5	319,9	388,7
CX	0,8	195,0	203,8	224,5	253,8	228,1	222,1	215,2	150,1	262,8	356,6
CX	0,85	207,0	212,8	213,3	362,7	357,1	223,5	170,2	196,4	310,9	415,9
CX	0,9	245,9	197,9	203,9	161,8	326,3	191,3	162,9	176,0	271,2	519,0

Tabla 6: Cantidad de evaluaciones promedio para encontrar la mejor solución en la instancia 500x20 con heurística de PO Azar

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	0,023	0,326	0,260	0,930	1,072	0,127	0,347	0,687	1,177	1,272
OX	0,55	0,145	0,322	0,601	1,371	1,537	0,362	0,224	1,298	1,083	1,738
OX	0,6	0,106	0,255	0,427	0,676	1,399	0,029	0,342	0,338	1,735	1,373
OX	0,65	0,257	0,401	0,856	1,497	1,542	0,508	0,615	1,068	1,653	1,760
OX	0,7	0,237	0,602	0,541	0,925	2,343	0,160	0,665	0,348	1,839	2,380
OX	0,75	0,601	0,366	0,732	1,551	1,752	0,027	0,300	0,734	1,384	1,933
OX	0,8	0,154	0,291	0,620	1,906	1,691	0,334	0,309	0,559	1,595	2,085
OX	0,85	0,431	0,945	0,852	1,215	2,422	0,091	0,849	1,236	1,865	1,931
OX	0,9	0,251	0,455	0,762	2,457	2,188	0,225	0,624	0,888	2,006	2,880
CX	0,5	0,546	0,318	0,574	1,480	0,847	0,284	0,286	0,802	1,551	1,337
CX	0,55	0,656	0,476	0,460	0,776	1,429	0,188	0,272	0,195	1,315	1,451
CX	0,6	0,562	0,594	0,199	0,834	0,624	0,230	0,649	0,532	1,110	1,852
CX	0,65	0,799	0,355	0,315	1,436	1,223	0,190	0,125	0,400	1,385	2,236
CX	0,7	0,638	0,535	0,280	1,419	2,395	0,472	0,421	0,609	2,166	1,856
CX	0,75	0,855	0,431	1,076	1,186	1,712	0,427	0,214	0,505	1,798	2,225
CX	0,8	1,514	0,378	0,557	1,072	1,290	0,892	0,313	0,310	1,559	2,685
CX	0,85	0,323	0,413	0,600	0,637	1,678	0,525	0,170	1,030	1,925	2,716
CX	0,9	0,797	0,606	0,682	2,008	2,163	0,572	0,550	0,294	2,209	2,334

Tabla 7: Tiempo promedio para encontrar la mejor solución de la instancia 200x10 con heurística de P0 Azar

	Op. Mutación	Intercambio					Inserción				
	Prob. Mut.	0,001	0,005	0,01	0,05	0,1	0,001	0,005	0,01	0,05	0,1
Op. Cruce	Prob. Cruce										
OX	0,5	1,754	1,030	2,083	3,792	6,971	2,123	0,966	2,025	4,557	9,095
OX	0,55	1,664	1,958	0,713	5,854	8,751	0,925	1,760	0,878	5,196	7,536
OX	0,6	2,400	1,206	3,385	6,049	8,612	2,093	0,348	1,586	5,806	8,429
OX	0,65	2,614	3,516	3,200	5,451	8,165	1,664	2,077	2,770	7,903	13,393
OX	0,7	0,986	3,305	2,325	4,348	7,208	0,559	1,716	1,695	6,428	11,201
OX	0,75	0,811	3,726	1,101	6,199	10,552	1,074	4,366	7,776	7,457	9,143
OX	0,8	1,944	2,263	2,895	14,661	9,078	0,651	2,378	7,086	4,713	10,766
OX	0,85	2,190	2,038	4,915	6,637	10,147	0,148	3,032	2,740	9,903	17,425
OX	0,9	1,582	1,623	3,677	8,687	9,482	0,787	0,421	3,812	14,596	11,889
CX	0,5	5,172	2,735	2,302	5,237	7,776	2,313	1,206	1,695	9,122	3,413
CX	0,55	2,966	0,927	4,130	6,920	9,120	2,686	1,045	2,752	4,855	11,504
CX	0,6	4,445	2,474	2,839	4,992	7,795	4,651	3,675	3,283	7,972	8,694
CX	0,65	8,480	4,314	4,601	3,617	9,272	2,217	2,598	2,437	6,895	11,200
CX	0,7	0,713	2,873	8,011	5,698	7,901	2,101	1,899	1,868	4,374	8,354
CX	0,75	5,730	5,597	3,501	6,740	7,450	4,121	2,555	5,986	8,939	12,139
CX	0,8	3,942	3,805	4,859	6,374	4,832	4,901	4,704	1,690	6,511	11,351
CX	0,85	5,053	4,733	4,561	12,331	11,559	5,442	3,327	3,739	9,822	14,894
CX	0,9	6,650	3,864	4,263	2,122	10,200	4,028	2,467	2,835	7,841	20,526

Tabla 8: Tiempo promedio para encontrar la mejor solución de la instancia 500x20 con heurística de P0 Azar

En las Tablas 5 y 6, se muestran los valores de la menor cantidad de evaluaciones promedio para cada probabilidad de cruce, resaltados con negrita, los valores de la menor cantidad de evaluaciones promedio para cada probabilidad de mutación, celda rellena, y por

último un recuadro a la menor cantidad de evaluaciones promedio encontrada. Y en las Tablas 7 y 8, se muestran los valores de los mejores tiempos promedios para cada probabilidad de cruce, resaltados con negrita, los valores de los mejores tiempos promedios para cada probabilidad de mutación, celda rellena, y por último un recuadro al mejor valor del tiempo promedio encontrada. Del análisis de esfuerzo computacional obtenido de las Tablas 5 a 8, se puede observar que el menor esfuerzo computacional se realiza cuando se usa el operador de Cruce de Orden y con probabilidades de cruce y mutaciones bajas, pero no es posible precisar cuál operador de mutación requiere menor esfuerzo computacional. Esto último, es debido a que los operadores de mutación seleccionados poseen una complejidad algorítmica similar, obteniendo de esta manera tiempos y cantidad de comparaciones similares en las diferentes configuraciones. En tanto que, el Cruce de Orden debido a su menor complejidad algorítmica, requiere un menor esfuerzo. Con respecto a la probabilidad de cruce baja, requiere menor esfuerzo computacional que las altas debido a que se producen menos cruces de padres, de igual manera con probabilidades bajas de mutación, hay menos mutaciones de individuos. Pero probabilidades bajas, como se mencionó previamente, se traducen en una reducción de la calidad de las soluciones.

Finalmente, la mejor configuración, entre las propuestas en el informe, para resolver el problema dependerá de la perspectiva que tomemos, en lo que respecta a la calidad los mejores resultados se obtienen inicializando la población con la heurística NEHH+Azar, utilizando el operador de Cruce de Ciclo con probabilidades altas (0,8 a 0,9) para obtener los hijos y finalmente mutando los hijos con el operador de mutación de Inserción con probabilidades altas (0,05 a 0,1). Aunque estas combinaciones requieran un mayor esfuerzo computacional que las restantes, el tiempo empleado por estos algoritmos es bajo ya que no superan los 3 segundos para el caso de 200 tareas y los 22 segundos para el caso de 500 tareas. En otras palabras, permiten obtener resultados de calidad en tiempos razonables.

Capítulo 4: Conclusiones

En este capítulo se resume el trabajo realizado y explican las conclusiones sobre los resultados de las diferentes ejecuciones de los algoritmos genéticos para resolver el FSSP. Además, se plantean trabajos futuros que se pueden realizar a tomando como base el presente trabajo.

En este trabajo se presentó una solución al Problema de Secuenciamiento de Tareas, este problema se lo clasifica como No Polinomial Completo, por lo que su resolución es muy compleja para algoritmos exactos. Debido a esta complejidad se planteó una resolución mediante metaheurísticas, que son técnicas de optimización aproximadas, es decir que logran resultados óptimos en tiempos aceptables para la complejidad del problema. Estas técnicas se las puede agrupar según la cantidad de elementos del espacio de búsqueda de soluciones que manipulan, están las Basadas en Trayectorias, que manipulan un único elemento, y las Basadas en Población, que manipulan un conjunto de elementos; estos elementos son soluciones al problema que deseamos resolver, en este caso soluciones al FSSP. Entre las técnicas Basadas en Población se hallan los Algoritmos Evolutivos que están inspirados en la teoría de la evolución natural, dentro de esta clasificación se encuentran los Algoritmos Genéticos que se basan en la supervivencia de los individuos más aptos.

El lenguaje de programación seleccionado para realizar el problema fue C++ (Anexo 3), inicialmente se había pensado en Java pero fue descartado debido que genera código intermedio que debe ser interpretado por una máquina virtual el cual es portable, pero ineficiente. Una de las características de C++ es el manejo de memoria dinámica, permitiendo realizar algoritmos eficientes y eficaces. Programar algoritmos utilizando memoria dinámica es complejo, debido al manejo de direcciones de memoria, reserva de memoria y liberación de la misma. Para evitar que quede alguna variable sin liberar el espacio en memoria que se asigno, se elaboró un *script* en *bash* (Anexo 4) el cuál iteraba la ejecución del algoritmo con las diferentes configuraciones paramétricas, evitando de esta forma la iteración dentro del algoritmo. Cuando una iteración termina el sistema operativo es el responsable de liberar los recursos que no fueron liberados, por lo que se puede garantizar que la siguiente ejecución del algoritmo no apila espacio de memoria sin utilizar. Una vez finalizada la ejecución del algoritmo, los resultados se guardan en un archivo csv (*Comma Separated Values*) para luego ser analizados, utilizando planillas de cálculo. En el Anexo 1 se muestra la salida de todas las ejecuciones de una configuración paramétrica.

La solución al FSSP se encontró al implementar Algoritmos Genéticos, que buscan la mejor solución partiendo de conjunto de posibles soluciones. A este conjunto de soluciones se lo denomina población inicial, el cual irá evolucionando por medio de operadores de selección, cruzamiento, mutación y reemplazo. Como el FSSP, es un problema de permutación de tareas, la forma en la cuál se representan las soluciones es mediante permutaciones de números enteros, esta representación define a su vez los operadores evolutivos que se pueden utilizar. La población con la cuál se trabajó es una población constante de 64 individuos.

La configuración del algoritmo dependerá de las diferentes técnicas que se utilizan para inicializar la población, seleccionar los individuos a cruzar, cruzar los individuos para obtener nuevos, mutar estos nuevos individuos y finalmente reemplazarlos en la población. Para

inicializar la población se utilizaron tres técnicas diferentes el Azar, que genera individuos de forma aleatoria, NEHH, que genera individuos a través de una heurística proporcionando individuos con cierta calidad y NEHH +Azar, que es la combinación de las anteriores donde el 50% de la población se genera con una técnica y el restante 50% con la otra. De esta población se seleccionan individuos para cruzar y obtener nuevas soluciones, esta selección se realiza mediante Torneo Binario, el cuál consiste en seleccionar al azar dos individuos y se selecciona al mejor de ambos, este procedimiento se realiza dos veces y se obtienen los dos padres a cruzar. Este cruce consiste en obtener una nueva solución a partir de dos anteriores, las técnicas utilizadas fueron Cruce de Orden, Cruce en Dos Puntos, Cruce de Ciclo y Cruce Parcialmente Mapeado. Estos nuevos individuos generados pueden sufrir una mutación, es decir, sufrir alteraciones de algunos de sus genes, las técnicas de mutación implementadas fueron Inserción e Intercambio. Tanto el cruce como la mutación, son afectadas por probabilidades de éxito, es decir que el operador de cruce produzca un hijo y a su vez que el mismo sufra una mutación. Una vez obtenidos los nuevos individuos (hijos), se reemplazan algunos individuos de la población por estos. Se implementaron tres técnicas de reemplazo: Recambio Generacional (los hijos son la nueva población), Recambio $\mu + \lambda$ con Elitismo (seleccionando los mejores individuos) y $\mu + \lambda$ con Elitismo utilizando selección mediante Torneo Binario (seleccionando el mejor individuo y los restantes son seleccionados por Torneo Binario). Estos operadores evolutivos se aplican sucesivamente hasta que se cumpla la condición de corte, la cuál es por cantidad de iteraciones, esta cantidad se fijó en 1000.

Debido a la gran cantidad de datos obtenidos en la ejecución de los algoritmos, para analizar los resultados se los agrupó primero según el método utilizado para inicializar la población: Azar, NEHH+Azar y NEHH. Luego se analizó cuál de los métodos usados para generar la población inicial obtienen mejores resultados tanto en calidad como en esfuerzo computacional.

A partir del análisis de los resultados se deduce que, los algoritmos genéticos que generan la población con la heurística NEHH+Azar y utilizan el operador de Cruce de Ciclo y el de mutación por inserción, ambos con probabilidades altas, permiten obtener las soluciones de alta calidad. Si bien el esfuerzo computacional de los mismos es mayor al del resto de las combinaciones algorítmicas propuestas, se encuentra dentro de lo que en la literatura se conoce como tiempos razonables de ejecución. Por lo que se los considera apropiados para resolver este tipo de problemas.

Como trabajo futuro se propone a partir de los resultados obtenidos en este trabajo, probar nuevas configuraciones paramétricas. Las nuevas configuraciones sugeridas son variar el tamaño de la población, pudiendo trabajar con población constante o variable, probar diferentes métodos de selección de padres y cambiar la condición de corte, puede cambiarse la cantidad de iteraciones o bien, utilizar otro método de corte, por ejemplo cuando se alcanza un individuo de cierta calidad. El objetivo de probar nuevas configuraciones es tratar de lograr soluciones de mayor calidad y que requieran menor esfuerzo computacional.

Bibliografía

- [1] - Michalewicz, Zbigniew. Genetic Algorithms + Data Structures = Evolution Programs. Springer. 1996.
- [2] - Peter Brucker. Scheduling Algorithms. Springer. 2006.
- [3] - El-Ghazali Talbi. Metaheuristics from design to implementation. John Wiley & Sons. 2009.
- [4] - J. H. Holland. Adaptation in Natural and Artificial Systems. The MIT Press. 1975.
- [5] - Zhigang Lian, Xingsheng Gu, Bin Jiao. A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan. Elsevier. Volume 35, Issue 5. 2008.
- [6] - Mohammad Kazem Sayadi, Reza Ramezani, Nader Ghaffari-Nasab. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. International Journal of Industrial Engineering Computations. Volume 1 Issue 1. 2010.
- [7] - I-Hong Kuoa, Shi-Jinn Hornga, Tzong-Wann Kaod, Tsung-Lieh Lina, Cheng-Ling Lee, Takao Teranoc, Yi Panf. An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model. Elsevier. Volume 36, Issue 3. 2009.
- [8] - Bo Liu, Ling Wang, Yi-Hui Jin. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. Elsevier . Volume 35. Issue 9. 2008.
- [9] - Zhigang Lian, Xingsheng Gua, Bin Jiao. A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. Elsevier. Volume 175. Issue 1.2006.
- [10] - Mitsuo Gen, Runwei Cheng. Genetic Algorithms and Engineering Design. John Wiley & Sons. 1997.
- [11] - Taillard, E. Benchmarks for basic scheduling problems. ORWP. 1989.
- [12] - Hsiao-Lan Fang. Genetic Algorithms in Timetabling and Scheduling. University of Edinburgh. 1994
- [13] - Peter Class . Tutorial de C++: o el diario de Peter Class. Free Document. 2004
- [14] - Nawaz, M., Enscore Jr, E., Ham, I., A heuristic algorithm for the m-machine n-job flow-shop sequencing problem. Omega. The International Journal of Management Science.
- [15] – Ruben Ruiz. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. European Journal of Operational Research vol 165. 2005.
- [16] – Enrique Montenegro. Diagrama de Caja o Boxplot.
<http://curso-estadistica-probabilidades.blogspot.com.ar/2009/10/diagrama-de-caja-o-boxplot.html>. Última consulta 20/10/2015.
- [17] - Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud. Scheduling instances.
<http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>. Última consulta 20/10/2015.

Anexos

Anexo 1: Salida ejemplo del AG1

A continuación se muestra a modo de ejemplo la salida obtenida del procesamiento de 200 tareas en 10 máquinas utilizando el AG1:

Cant. Tareas	Cant. Maq.	Población	Iteraciones	Pc	Pm	Op. X	Op. Mut.	H. P0	H. Reemplazo	M° Fitness	T° M° Fitness	Generación M°	Cant. Eval. M°	M° PI	Promedio PI	Promedio PF	T° Total	Cant. Total Eval	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	12139	0,01	1	64	12139	12584,31	12301	4,22	1116	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11926	0,02	2	102	12150	12565,22	12430	4,03	1134	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11872	0,03	6	135	12204	12533,98	12217	3,99	1131	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11920	0,02	4	125	12059	12560,98	12354	3,71	1147	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	12021	0,02	2	105	12066	12550,72	12346	3,67	1141	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11987	0,02	4	123	12038	12472,77	12453	3,66	1132	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	12045	0,03	6	131	12113	12571,45	12208	4,14	1145	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11998	0,06	10	151	12095	12503,38	12145	4,23	1141	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11947	0,02	3	107	12045	12526,72	12777	4,22	1129	
200	10	64	1000	0,5	0,001	OX	Inter	Azar	M+L c/E.	11972	0,03	4	118	12182	12536,34	12608	4,21	1117	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	11917	0,02	2	101	12145	12498,28	12673	4,03	1126	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	11971	1,66	449	575	12013	12567,72	12226	3,66	1126	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12001	0,01	1	89	12127	12519,81	12700	3,68	1123	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12035	0,02	3	106	12054	12493,12	12467	3,65	1123	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12102	0,02	3	110	12144	12523,47	12574	3,68	1140	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	11878	0,01	1	97	12133	12536,16	12191	3,93	1138	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12020	0,01	1	64	12020	12494,25	12585	4,23	1144	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12093	0,01	1	64	12093	12526,61	12701	4,16	1130	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	11932	0,11	24	145	12018	12538,64	12212	4,26	1121	
200	10	64	1000	0,5	0,005	OX	Inter	Azar	M+L c/E.	12004	0,02	2	97	12212	12542,03	12472	4,2	1126	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	12084	0,04	6	120	12134	12551,27	12438	4,1	1119	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11981	0,02	2	104	12097	12505,95	12247	4,2	1139	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11897	1,41	335	461	12125	12549,69	12124	4,07	1126	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	12037	0,74	172	295	12157	12509,48	12485	4,22	1123	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	12026	0,03	4	125	12031	12549,39	12353	3,94	1134	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11902	0,03	5	124	12107	12542,66	12976	3,67	1129	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11899	0,01	2	100	11979	12554,38	12217	4,16	1129	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11910	0,04	7	115	11953	12514,83	12809	4,22	1112	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11992	0,12	26	147	12150	12558,52	12554	4,28	1121	
200	10	64	1000	0,5	0,010	OX	Inter	Azar	M+L c/E.	11850	0,02	2	98	11980	12506,3	12594	4,25	1109	

200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	12077	0,03	3	123	12084	12545,58	12592	3,7	1135
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11925	2,21	604	736	12083	12538,8	12797	3,64	1132
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11939	3,94	934	1077	12066	12510,89	12509	4,23	1143
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11825	0,02	2	108	11973	12555,56	12210	4,21	1135
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	12011	0,03	4	121	12120	12566,44	12597	4,24	1147
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11908	0,15	32	170	12161	12553,56	12672	3,98	1138
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11966	2,76	653	788	12110	12492,19	12443	4,24	1135
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11893	0,02	3	107	12152	12552,22	12554	4,27	1116
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11978	0,01	1	64	11978	12488,39	12408	4,18	1144
200	10	64	1000	0,5	0,050	OX	Inter	Azar	M+L c/E.	11961	0,03	4	125	12060	12553,08	12626	4,21	1142
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11890	0,24	54	181	12096	12569,45	12743	4,27	1127
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11932	0,73	174	298	12149	12536,03	12199	4,2	1124
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11904	3,39	795	924	12145	12594,86	12267	4,24	1129
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11934	2,58	619	742	11975	12506,72	12508	4,2	1123
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11971	0,02	2	106	12022	12540,17	12170	3,99	1121
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11949	2,72	635	760	11985	12523,89	12451	4,27	1125
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	12008	0,02	3	105	12233	12524,98	12307	4,23	1131
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	12005	1,42	336	454	12098	12512,56	12455	4,21	1118
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	11869	0,01	2	102	12012	12523,31	12404	3,96	1113
200	10	64	1000	0,5	0,100	OX	Inter	Azar	M+L c/E.	12051	0,02	3	121	12132	12497,72	12347	3,67	1135
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11882	2,47	612	743	12034	12527,12	12050	4,03	1131
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	12106	0,02	3	112	12156	12553,59	12450	4,05	1122
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	12110	0,02	4	113	12145	12518,48	12488	4,01	1117
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11926	0	1	64	11926	12534,48	12286	4,03	1142
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11990	0,11	25	141	12028	12539,28	12308	4,02	1116
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11876	0,02	4	126	12088	12523,52	12270	4,03	1134
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	12118	0,01	1	83	12132	12527,41	12686	4,02	1132
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11994	0,01	2	97	12084	12564,8	12578	4,29	1113
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	12007	0	1	64	12007	12508,72	12348	4,48	1142
200	10	64	1000	0,55	0,001	OX	Inter	Azar	M+L c/E.	11878	0,03	4	115	12147	12541,02	12689	4,56	1113
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	12026	0,04	6	122	12117	12529,41	12376	4,64	1133
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	11816	0,01	1	64	11816	12543,11	12360	4,62	1118
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	12066	4,23	928	1046	12128	12531,12	12294	4,56	1118
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	11928	0,04	6	138	12062	12497,16	12792	4,58	1135
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	11920	0,01	1	84	12083	12507,72	12519	4,45	1128
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	11890	2,98	641	770	12029	12529,12	12463	4,63	1129
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	12016	0,01	1	64	12016	12521,2	12832	4,59	1100
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	12064	0,05	7	140	12185	12517,94	12407	4,31	1145
200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	11961	2,63	562	699	12026	12491,39	12336	4,65	1137

200	10	64	1000	0,55	0,005	OX	Inter	Azar	M+L c/E.	12004	0,07	10	148	12178	12574,55	12593	4,64	1138
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	12002	0,02	2	103	12131	12538,28	12431	4,6	1134
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11965	0,01	1	87	12067	12506,25	12302	4,59	1140
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11938	0,01	1	90	12032	12531,62	12214	4,64	1134
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11852	4,17	897	1031	12092	12556,09	12421	4,66	1134
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	12028	0,02	2	101	12159	12498,11	12693	4,29	1140
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	12027	0,01	1	64	12027	12533,67	12512	4,59	1134
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11956	0,01	1	85	12062	12514,39	12946	4,63	1130
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11905	0,03	3	120	12130	12551,3	12455	4,23	1140
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	12043	0,01	2	102	12069	12508	12534	4,05	1122
200	10	64	1000	0,55	0,010	OX	Inter	Azar	M+L c/E.	11889	0,01	2	106	12165	12514,5	12892	4,03	1127
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11905	1,36	331	456	12007	12496,44	12454	4,07	1125
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11928	3,19	783	947	11960	12506,38	13061	4,08	1164
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11933	0,6	148	268	12120	12502,7	12503	4,01	1120
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	12018	3,69	915	1061	12025	12551,34	12883	4,03	1146
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11937	0,02	4	127	11983	12513,88	12402	4,44	1130
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11969	0,02	3	111	12113	12516,78	12636	4,64	1118
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11961	4,19	921	1048	12162	12543,38	12722	4,55	1127
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	12001	0,01	1	87	12054	12497,66	12670	4,66	1135
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	12051	4,24	928	1061	12169	12559,47	12533	4,57	1133
200	10	64	1000	0,55	0,050	OX	Inter	Azar	M+L c/E.	11833	0,04	5	122	11964	12534,22	12887	4,59	1119
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11926	0,02	2	112	12095	12526,7	12554	4,59	1150
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11842	3,21	687	840	12139	12606,61	12903	4,62	1153
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11850	3,38	724	877	12044	12538,97	12437	4,69	1153
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	12000	1,41	296	419	12021	12545,06	12581	4,65	1123
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11974	0,02	1	92	11984	12510,95	12605	4,63	1131
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11925	1,64	346	480	12105	12538,17	12511	4,69	1134
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11855	0,06	10	142	12024	12539,55	12182	4,65	1132
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11922	0,01	1	64	11922	12518,94	12539	4,69	1114
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11963	0,01	1	87	12245	12557,23	12383	4,67	1118
200	10	64	1000	0,55	0,100	OX	Inter	Azar	M+L c/E.	11953	0,07	12	142	12047	12506,19	12650	4,7	1130
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	11927	0,01	1	64	11927	12517,22	12672	5,07	1133
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	12057	0,01	1	64	12057	12517,92	12249	5,12	1142
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	12089	0,02	2	115	12106	12508,22	12617	5,05	1146
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	12017	0,03	3	119	12061	12560,02	12247	5,08	1128
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	12003	0,04	4	126	12045	12518,41	12444	4,74	1156
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	12057	1,89	376	506	12089	12523,44	12649	5,01	1130
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	11948	0,02	1	89	12019	12516,59	12394	5,08	1147
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	11904	2,72	547	690	12142	12542,14	12018	4,96	1143

200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	11953	0,01	1	64	11953	12486,77	12724	5,09	1148
200	10	64	1000	0,6	0,001	OX	Inter	Azar	M+L c/E.	11786	0,04	5	131	12078	12492,16	12122	5	1128
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11897	0,04	5	128	12052	12486,94	12674	5	1137
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11897	0,08	12	160	12009	12509,44	12930	5,01	1148
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11966	0,01	1	64	11966	12484,38	12426	5,17	1148
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	12066	0,03	4	116	12119	12541	12400	4,97	1121
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11946	0,01	1	93	12081	12544,47	12441	5,14	1131
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	12049	0,03	4	125	12082	12499,42	12212	5,02	1136
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11923	0,01	1	64	11923	12546,09	12582	4,97	1128
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	12033	0,02	1	91	12110	12556,06	12551	5	1134
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	11920	0,05	7	138	11937	12496,05	12700	5,08	1132
200	10	64	1000	0,6	0,005	OX	Inter	Azar	M+L c/E.	12056	0,01	1	64	12056	12500,25	12400	5,02	1130
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12024	0,01	1	64	12024	12556,02	12657	4,97	1127
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	11953	0,07	10	163	12084	12506,03	12478	4,74	1155
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12014	0,01	1	88	12022	12482,72	12868	4,99	1126
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	11945	0,02	2	106	12008	12522,28	12652	4,92	1150
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12053	0,02	3	122	12112	12545,8	12517	4,4	1137
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12094	0,02	4	120	12118	12486	12545	4,4	1130
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12018	0,61	139	275	12100	12542,2	12414	4,42	1136
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	11910	0,01	2	107	12181	12551,55	12605	4,38	1136
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12010	0,02	3	120	12064	12512,44	12601	4,4	1133
200	10	64	1000	0,6	0,010	OX	Inter	Azar	M+L c/E.	12086	0,4	90	233	12110	12518,28	13005	4,39	1143
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11881	0,02	3	125	12033	12541,23	12627	4,36	1158
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	12019	0,24	51	178	12129	12476,5	12231	4,41	1127
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	12048	3,52	718	850	12195	12562,52	12329	4,95	1132
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11915	1,44	279	417	11999	12502,78	12584	5,09	1138
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11938	2,98	589	735	11993	12521,86	12672	5,04	1146
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11894	0,02	1	94	12115	12497,67	12725	5,08	1128
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11960	0,02	1	93	12034	12500,91	12508	5,14	1146
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	12010	0,03	3	125	12106	12553,97	12448	5,09	1144
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	11979	2,3	456	587	12125	12568,47	12275	5,05	1131
200	10	64	1000	0,6	0,050	OX	Inter	Azar	M+L c/E.	12013	2,7	530	672	12203	12541,47	12496	5,07	1142
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11940	0,01	1	64	11940	12520,86	12804	5,13	1131
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11943	4,35	865	1000	12128	12499,45	12617	5	1135
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11856	0,02	1	94	12021	12549,73	12274	5,17	1143
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11962	0,22	40	175	12164	12566,06	12472	4,73	1135
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11936	3,41	782	904	12073	12506,23	12620	4,38	1122
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11964	0,01	1	92	12067	12527,61	12294	4,41	1135
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11835	0,04	7	143	12027	12531,47	12432	4,39	1146

200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11860	0,01	1	91	12111	12545,73	12942	4,38	1133
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11917	0,16	34	175	12011	12522,69	12647	4,39	1141
200	10	64	1000	0,6	0,100	OX	Inter	Azar	M+L c/E.	11992	0	1	64	11992	12521,8	12694	4,39	1135
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11983	1,5	315	434	12008	12527,83	12473	4,73	1119
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	12036	0,02	2	110	12052	12535,8	12853	4,77	1142
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11969	0	1	64	11969	12545,31	12439	4,77	1127
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11998	0	1	64	11998	12563,92	12806	4,73	1127
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	12018	1,42	296	414	12194	12588,31	12122	4,73	1118
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11931	1,79	375	518	12129	12507,92	12213	4,77	1143
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11964	0,29	59	188	12141	12542,91	12316	4,73	1129
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11940	0	1	64	11940	12536,78	12332	4,73	1127
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	11882	0,11	22	147	12077	12555,14	12533	4,76	1125
200	10	64	1000	0,65	0,001	OX	Inter	Azar	M+L c/E.	12049	0	1	64	12049	12533,22	12401	4,73	1133
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11855	0,02	3	121	12134	12496,88	12041	4,72	1135
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11956	0,03	4	125	12000	12548,09	12671	4,74	1122
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11891	0,02	2	110	12116	12551,59	12372	5,15	1129
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11891	0,03	4	130	11996	12525,7	12775	5,58	1134
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11868	0,03	3	122	12057	12536,92	12923	5,54	1151
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11773	0,03	2	113	12127	12535,77	12649	5,11	1142
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11957	0,02	2	107	12104	12588,3	12391	5,4	1127
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11960	0,05	6	133	12169	12518,08	12174	5,44	1135
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	12060	0,04	4	141	12105	12534,67	12367	5,43	1164
200	10	64	1000	0,65	0,005	OX	Inter	Azar	M+L c/E.	11941	0,03	3	112	12025	12483,48	12374	5,5	1130
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	11948	0,04	4	124	12210	12561,66	12382	5,44	1133
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	12047	0,01	1	64	12047	12503,94	12537	5,42	1132
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	11946	0,03	3	115	12226	12543,34	12746	5,43	1132
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	12015	2,11	389	538	12028	12556,77	12343	5,42	1149
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	12023	3,1	559	703	12101	12543,47	12753	5,48	1144
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	11997	0,03	4	118	12115	12499,8	12861	5,28	1135
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	12044	0,02	2	109	12096	12554,5	12321	4,77	1140
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	12027	0,02	3	112	12153	12579,64	13203	4,85	1121
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	11974	0	1	64	11974	12540,05	12479	5,19	1144
200	10	64	1000	0,65	0,010	OX	Inter	Azar	M+L c/E.	11956	0,02	1	93	12042	12526,06	12474	5,49	1125
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11824	2,22	400	542	12159	12513,77	12746	5,48	1142
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11922	0,03	3	119	12122	12536,38	12528	5,51	1129
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	12034	3,85	706	825	12104	12549,66	12245	5,41	1119
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	12043	4,73	988	1120	12084	12518,02	12331	4,79	1132
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11891	4,36	922	1065	12116	12553,11	12286	4,73	1143
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11846	3,59	683	828	12021	12514,98	13088	5,33	1145

200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11954	3,49	634	782	12165	12493,89	12327	5,49	1148
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	12041	2,22	410	531	12113	12526,19	12585	5,4	1121
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11869	0,05	5	129	11971	12517,69	12562	5,6	1131
200	10	64	1000	0,65	0,050	OX	Inter	Azar	M+L c/E.	11929	0,03	2	118	12139	12514,83	12550	5,24	1148
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11902	0,25	43	189	12246	12546,34	12825	5,54	1146
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11963	1,94	358	494	12131	12532,33	12464	5,43	1136
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11876	2,82	524	663	12079	12519,84	12223	5,44	1139
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11961	3,54	647	783	12116	12532,44	12423	5,5	1136
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11989	4,22	764	901	12209	12595,58	12763	5,53	1137
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11961	2,18	405	536	12121	12502,23	12282	5,43	1131
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11999	0,91	162	302	12042	12469	12691	5,5	1140
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11993	0,06	9	149	12196	12555,67	12508	5,5	1140
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11946	0,01	1	89	12061	12554,17	12858	5,52	1135
200	10	64	1000	0,65	0,100	OX	Inter	Azar	M+L c/E.	11895	4,89	887	1049	12057	12542,48	12303	5,52	1162
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	12096	0,01	1	64	12096	12551,47	12445	5,65	1166
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	11979	0,06	8	152	12144	12562,69	12220	5,9	1150
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	12037	0,01	1	64	12037	12532,22	12386	5,92	1145
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	11870	0,03	3	119	12130	12542,95	12401	5,89	1155
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	11952	0,01	1	64	11952	12541,42	12467	5,96	1136
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	12025	1,29	217	361	12103	12544,34	12276	5,89	1144
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	11997	0,05	6	136	12138	12525,5	12038	5,96	1139
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	12049	0,03	3	117	12099	12565,2	12875	5,97	1141
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	11998	0,04	4	126	12082	12565,62	12498	5,91	1137
200	10	64	1000	0,7	0,001	OX	Inter	Azar	M+L c/E.	12075	1,3	217	361	12089	12499,17	12422	5,93	1144
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11920	0,7	116	250	12152	12544,25	12375	5,92	1134
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11817	0,01	1	64	11817	12512,78	12334	5,84	1122
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	12046	0,63	105	241	12136	12568,67	12446	5,9	1136
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11983	1,22	204	343	12143	12528,69	12344	5,86	1139
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	12093	3,35	568	723	12124	12545,25	12282	5,64	1155
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11872	0,01	1	64	11872	12512,97	12583	6,03	1122
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11995	0,02	2	113	12067	12516,67	12693	5,95	1144
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	11900	0,02	1	96	12013	12552,89	12685	5,9	1153
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	12110	0,03	3	120	12146	12596,83	12404	5,59	1152
200	10	64	1000	0,7	0,005	OX	Inter	Azar	M+L c/E.	12016	0,02	2	109	12089	12503,34	12409	5,9	1140
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11991	0,03	3	125	12163	12550,78	12912	5,94	1135
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11954	1,91	323	451	11968	12526,58	12812	5,84	1128
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	12019	0,04	4	121	12206	12531,42	12465	5,84	1122
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	12040	0,05	6	149	12163	12519,08	12678	5,85	1161
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11986	0,1	14	170	12122	12493,11	12468	5,84	1156

200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11951	0,11	17	151	12134	12531,62	12362	5,83	1134
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	12031	0,04	4	130	12159	12549,97	12415	5,83	1131
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	12054	0,03	2	119	12110	12482,81	12320	5,77	1154
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11950	2,41	410	562	11999	12513,12	12859	5,84	1152
200	10	64	1000	0,7	0,010	OX	Inter	Azar	M+L c/E.	11953	3,15	538	680	12047	12530,34	12932	5,83	1142
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11946	3,94	671	813	12135	12528,5	12264	5,87	1142
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11925	4,41	659	798	12057	12570,17	12869	6,67	1139
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	12019	4,02	691	823	12144	12586,59	12188	5,8	1132
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	12035	0,03	2	109	12131	12520,44	12209	5,74	1136
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11943	5,37	915	1057	11974	12549,69	12749	5,88	1142
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11950	1,76	303	443	12012	12562,2	12328	5,8	1140
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11968	4,33	738	875	12088	12543,61	12779	5,84	1137
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11930	0,02	2	115	11994	12538,61	12260	5,71	1142
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11912	0,02	2	109	12128	12557,53	12574	5,89	1144
200	10	64	1000	0,7	0,050	OX	Inter	Azar	M+L c/E.	11887	0,03	3	114	12125	12539,33	12502	5,95	1127
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11885	0,02	2	107	12172	12551,05	12432	5,74	1128
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11937	0,47	77	215	12014	12554,88	12129	5,95	1138
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11925	0,12	18	144	12139	12521,47	12310	5,94	1126
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	12014	0,01	1	64	12014	12508,91	12380	5,91	1167
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11906	0,01	1	64	11906	12553,41	12452	6,02	1118
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	12005	0,03	3	123	12033	12516,73	12622	5,93	1131
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11965	3,12	523	672	12003	12551,5	12319	5,93	1149
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11835	5,91	990	1115	12072	12544,94	11994	5,96	1125
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11866	0,06	7	140	12004	12511,19	12597	5,91	1137
200	10	64	1000	0,7	0,100	OX	Inter	Azar	M+L c/E.	11874	0,05	6	137	12125	12579,88	12560	5,88	1134
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	11873	0,64	100	237	12070	12503,22	11949	6,26	1137
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	12081	0	1	64	12081	12504,31	12318	5,79	1140
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	11964	0,04	6	145	12095	12518,31	12446	5,48	1140
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	12075	0,03	4	133	12084	12517,81	12533	5,49	1144
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	11974	0,02	2	108	12177	12509,5	12523	6,25	1122
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	11939	0,04	4	128	12127	12581,72	12573	5,86	1128
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	12098	0,01	1	101	12178	12547,19	12292	5,86	1148
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	12028	0,06	10	172	12120	12517,72	12331	5,46	1168
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	11936	0,03	3	127	12098	12549,05	12649	5,49	1145
200	10	64	1000	0,75	0,001	OX	Inter	Azar	M+L c/E.	12049	0	1	64	12049	12482,39	12509	5,85	1148
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11973	5,35	844	987	12054	12477,42	12283	6,34	1143
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11932	0,62	97	229	12020	12541,09	12488	6,28	1132
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	12013	0,04	4	132	12055	12511,95	12355	6,28	1136
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	12020	4,69	804	962	12127	12518,92	12381	5,92	1158

200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11946	0,29	43	183	12047	12538,41	12584	6,31	1140
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	12048	0,04	4	125	12066	12549,34	12443	6,27	1145
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11865	0,02	1	96	12006	12533,62	12712	6,32	1160
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11884	0,1	13	144	11992	12564,31	12466	6,32	1131
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11975	0,01	1	64	11975	12530,12	12315	6,13	1138
200	10	64	1000	0,75	0,005	OX	Inter	Azar	M+L c/E.	11928	0,04	4	138	12111	12542,23	12393	6,35	1154
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	11755	0,03	3	129	12159	12593,8	12613	6,39	1147
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	12004	0,02	1	92	12157	12584,69	12798	6,3	1138
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	12047	0,02	2	109	12141	12485,25	12604	6,48	1133
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	12052	0,01	1	64	12052	12546,5	12178	6,33	1147
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	11994	0,03	3	116	12134	12539,56	12549	6,31	1125
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	11993	0,06	7	164	12044	12500,05	12427	6,39	1162
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	11985	0,95	148	294	12154	12546,2	12248	6,34	1146
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	12087	0,04	4	134	12112	12523,95	12718	6,31	1140
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	12018	0,03	3	128	12057	12522,58	12766	6,3	1152
200	10	64	1000	0,75	0,010	OX	Inter	Azar	M+L c/E.	11955	3,01	477	634	12082	12511,56	12562	6,32	1157
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	12005	0,27	40	169	12169	12536,14	12921	6,25	1129
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	12016	4,68	746	894	12110	12539,08	12459	6,27	1148
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11929	0,02	1	95	12011	12559,53	12535	6,32	1160
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	12018	0,11	15	180	12206	12594,72	12356	6,22	1166
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11904	1,1	173	303	12023	12555,55	12145	6,21	1130
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11984	0,01	1	64	11984	12578,69	12676	6,19	1149
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11972	5,15	825	943	12145	12484,12	12798	6,25	1118
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	12034	4,37	704	839	12043	12549,53	12642	6,23	1135
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11997	0,03	3	121	12049	12510,25	12316	6,33	1138
200	10	64	1000	0,75	0,050	OX	Inter	Azar	M+L c/E.	11986	0,01	1	64	11986	12502,52	12575	6,21	1143
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11933	4,57	644	800	12140	12503,72	12660	7,08	1156
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11809	4,13	664	803	12173	12538,67	12321	6,2	1139
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11970	0,09	12	146	12029	12543,42	12401	5,66	1134
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11984	0,42	75	203	12144	12568,77	12519	5,8	1128
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	12002	5,44	993	1143	12091	12541,94	12447	5,48	1150
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11923	4	729	876	12072	12519,39	12967	5,47	1147
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	12078	0,51	94	253	12195	12556,12	12422	5,7	1159
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11976	0,58	91	225	12125	12506,06	12668	6,28	1134
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11927	5,48	876	1015	12014	12541,38	12661	6,25	1139
200	10	64	1000	0,75	0,100	OX	Inter	Azar	M+L c/E.	11989	0,02	1	92	12035	12503,36	12825	5,89	1134
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	11989	0,01	1	103	12053	12501,47	12176	5,82	1139
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	11971	0,04	6	150	12004	12540,95	12111	6,14	1166
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	12063	0,91	133	252	12096	12561,03	12386	6,82	1119

200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	12046	0,02	1	93	12179	12586,08	12329	6,73	1135
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	12010	0,03	3	130	12075	12472,11	12576	6,74	1162
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	12008	1,72	278	404	12090	12528,33	12470	6,29	1126
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	11959	0,07	8	164	12101	12498,23	12577	6,64	1162
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	11948	0,02	2	109	12076	12582,53	12669	6,2	1145
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	11854	0,03	3	125	12013	12520,72	12948	6,88	1161
200	10	64	1000	0,8	0,001	OX	Inter	Azar	M+L c/E.	12065	0,01	1	64	12065	12495,62	12380	6,73	1154
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11938	0,04	4	136	12199	12534,67	12514	6,71	1151
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	12047	0,02	1	96	12127	12525,73	12213	6,6	1134
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11875	0,01	1	64	11875	12533,5	12470	7,22	1184
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11921	0,04	4	125	12004	12521,56	12452	6,75	1126
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11901	0,02	1	99	11958	12529,75	12627	6,47	1156
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11804	0,53	89	251	12156	12517,02	12594	5,84	1162
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11969	0,02	2	106	12083	12516,16	12594	5,85	1115
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	12089	0,01	1	97	12147	12556,86	12553	6,13	1128
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	12024	0,05	5	140	12126	12576,41	12536	5,91	1144
200	10	64	1000	0,8	0,005	OX	Inter	Azar	M+L c/E.	11908	0	1	64	11908	12498,77	12673	5,85	1140
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	11843	0,01	1	99	11974	12569,52	12552	5,81	1140
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	11895	0,05	7	155	12136	12525,61	12549	5,85	1157
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	12054	0,01	1	98	12151	12559,45	12517	5,85	1152
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	11932	0,02	2	110	12071	12551,75	12420	6,15	1134
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	12055	0,01	1	64	12055	12563,39	12660	6,64	1146
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	12067	0,03	3	129	12080	12528,27	12653	6,53	1154
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	12014	3,89	588	729	12173	12531,84	12756	6,64	1141
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	11968	0,02	1	99	12087	12551,12	12349	6,68	1165
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	11786	0,03	3	121	12071	12495,09	12439	6,64	1129
200	10	64	1000	0,8	0,010	OX	Inter	Azar	M+L c/E.	12004	0,04	3	128	12014	12516,42	12356	6,59	1149
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11902	0,01	1	64	11902	12554,73	12551	6,63	1140
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11941	0,13	17	163	12110	12505,69	12884	6,83	1146
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11948	5,03	734	868	12064	12521,83	12683	6,79	1134
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11901	2,07	312	456	12089	12500,34	12203	6,66	1144
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11942	0	1	64	11942	12511,72	12473	6,62	1160
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11936	0,08	10	146	12099	12458,59	12130	6,73	1137
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	12036	3,69	544	675	12065	12553,05	12675	6,74	1131
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11906	2,45	361	509	12003	12544,16	12399	6,73	1148
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11945	0,02	1	93	12131	12521,64	12400	7,65	1146
200	10	64	1000	0,8	0,050	OX	Inter	Azar	M+L c/E.	11946	3,04	450	605	12115	12542,73	12667	6,73	1155
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11911	2,19	322	476	12068	12517,48	12592	6,75	1154
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11994	0,01	1	64	11994	12525,06	12551	6,75	1147

200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11867	0,02	1	96	11900	12495,2	12263	6,73	1154
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11889	0,07	8	170	12118	12515,81	12610	6,74	1164
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11834	0,08	10	152	12035	12504,39	12620	6,72	1145
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11942	0,02	1	95	12133	12541,59	12225	6,5	1145
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11996	1,36	201	366	12110	12531,08	12592	6,73	1165
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11976	4,84	717	855	12134	12519,56	12262	6,73	1138
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11917	3,86	576	713	12030	12550,75	12635	6,73	1137
200	10	64	1000	0,8	0,100	OX	Inter	Azar	M+L c/E.	11887	0,05	5	134	12175	12545,03	12448	6,73	1136
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11978	0,08	8	160	12146	12537,28	12447	7,22	1154
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	12109	0,02	1	91	12112	12516,03	12575	7,13	1146
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	12069	0,02	1	99	12184	12584,75	12291	7,15	1139
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	12048	0,03	3	128	12054	12548,81	12417	7,12	1150
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11954	3,05	426	563	12090	12529,91	12437	6,96	1137
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11950	0,03	2	119	12096	12536,33	12802	7,12	1148
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11929	0,03	2	118	12065	12521,98	12187	7,13	1150
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	12006	0,07	7	143	12007	12506,05	12397	6,9	1142
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11940	0,01	1	64	11940	12594,2	12532	7,13	1166
200	10	64	1000	0,85	0,001	OX	Inter	Azar	M+L c/E.	11982	0,04	4	133	12125	12566,8	12536	7,09	1146
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	11889	0,05	5	157	12087	12533,86	12303	7,06	1160
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	11984	0,03	2	119	12075	12552,3	12862	7,18	1155
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12001	0,01	1	64	12001	12534,02	12921	7,05	1172
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12019	0,03	2	113	12037	12559,27	12489	7,08	1157
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12077	0,01	1	64	12077	12488,75	12457	6,82	1140
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12071	0,03	3	123	12190	12505,94	12598	7,19	1151
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12062	0,03	3	123	12100	12535,69	12396	7,2	1140
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12049	0,06	6	144	12089	12549,16	12458	7,14	1140
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	11891	3,87	535	666	11996	12529,97	12728	7,12	1131
200	10	64	1000	0,85	0,005	OX	Inter	Azar	M+L c/E.	12061	0	1	64	12061	12528,89	12667	6,23	1138
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11885	0,03	3	121	12074	12497,61	12435	6,2	1145
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11891	0	1	64	11891	12559,92	12550	6,31	1140
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	12036	0,07	8	158	12080	12489,02	12864	7,04	1154
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11940	2,68	379	537	12096	12567,52	12384	6,87	1158
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	12069	6,52	917	1058	12085	12568,11	12593	7,11	1141
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11955	0,04	3	124	12100	12547,2	12473	7,17	1131
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11988	0,03	2	113	12112	12552,25	12264	7,23	1163
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	12088	0,04	4	141	12110	12539,48	12459	7,13	1179
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	11925	0,05	5	152	12140	12534,03	12360	7,11	1158
200	10	64	1000	0,85	0,010	OX	Inter	Azar	M+L c/E.	12013	0,02	1	99	12080	12514,12	12465	7,18	1164
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11888	2,54	354	498	12079	12551,42	12708	7,18	1144

200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11996	0,05	5	145	12182	12510,48	12595	7,14	1159
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11928	0,06	6	155	12087	12531	12892	8,16	1172
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11824	0,03	2	120	11979	12513,08	12446	7,19	1170
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11939	5,52	800	952	12170	12588,34	12514	6,84	1152
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	12056	2,62	362	511	12222	12545,64	12512	7,19	1149
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11938	0,01	1	64	11938	12543,83	13083	7,18	1147
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	12020	0,03	2	112	12106	12536,98	12341	7,17	1136
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	11944	0,02	1	98	12076	12514,2	12655	7,15	1149
200	10	64	1000	0,85	0,050	OX	Inter	Azar	M+L c/E.	12090	4,29	612	760	12198	12605,55	12505	6,8	1148
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11918	0,39	53	186	11976	12548,27	13182	7,13	1133
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11920	5,25	732	876	12068	12509,75	12631	7,15	1144
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11901	0,11	13	173	12083	12512,8	12569	7,13	1161
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11873	2,86	402	554	12025	12519,61	12040	7,11	1152
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11910	5,4	752	904	12025	12503,22	12661	7,19	1152
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11942	0,04	3	121	12084	12546,92	12483	7,16	1148
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11925	0,02	1	98	12047	12523,2	12446	7,25	1150
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11892	0,02	1	98	12011	12496,53	12246	7,18	1158
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11953	0,03	2	122	12062	12505,36	12237	7,14	1163
200	10	64	1000	0,85	0,100	OX	Inter	Azar	M+L c/E.	11963	5,98	840	1006	11985	12557,25	12636	7,11	1166
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12020	0,03	2	117	12079	12505,78	12467	7,53	1152
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	11885	0,03	4	136	12217	12534,08	12377	6,54	1149
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12094	0,01	1	95	12095	12552,47	12513	7,25	1168
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12022	0,01	1	64	12022	12557,47	12322	7,53	1145
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	11905	0,01	1	64	11905	12527,31	12439	7,58	1148
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	11894	0,08	8	144	12017	12552,98	12454	7,56	1140
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12029	0,04	3	122	12118	12521	12397	7,51	1135
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12015	0,05	5	147	12071	12530,22	12615	7,56	1154
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	12094	0,04	3	127	12143	12566,27	12365	7,89	1133
200	10	64	1000	0,9	0,001	OX	Inter	Azar	M+L c/E.	11895	0,01	1	64	11895	12491,08	12562	7,45	1174
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12060	0,03	2	119	12156	12538,2	12921	7,46	1136
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	11986	0,04	3	126	12120	12514,36	12304	7,42	1161
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12079	7,07	944	1115	12124	12528	12379	7,48	1171
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	11935	0,02	1	101	12091	12566,77	12155	7,46	1140
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12038	0,11	12	174	12080	12559,33	12708	7,52	1165
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12019	0,03	2	117	12152	12543,38	12647	7,64	1155
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12047	0,03	2	115	12052	12552,84	12638	7,45	1149
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12073	0,04	3	130	12105	12578,55	12400	7,41	1151
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	11998	1,28	168	301	12047	12511,78	12398	7,47	1133
200	10	64	1000	0,9	0,005	OX	Inter	Azar	M+L c/E.	12027	0,03	2	119	12084	12526,12	12618	7,49	1166

200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11837	0,99	131	286	12078	12506,11	12524	6,78	1155
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11911	0	1	64	11911	12553,7	13180	6,55	1162
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11797	0,09	12	159	12091	12525,05	12718	6,57	1148
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	12065	1,15	171	337	12201	12571,27	12463	7,32	1166
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11972	0,01	1	64	11972	12493,08	12912	7,47	1162
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11883	0,04	4	126	12028	12486,36	12497	7,45	1142
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	12043	0,02	1	98	12147	12572,56	12597	7,57	1145
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	12031	0,02	1	98	12143	12512,64	12897	7,52	1147
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11757	0,06	6	154	12107	12520,58	12454	7,26	1154
200	10	64	1000	0,9	0,010	OX	Inter	Azar	M+L c/E.	11964	0,04	3	131	12101	12547,61	12412	7,54	1164
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11946	0,02	1	100	12093	12526,38	12675	7,12	1157
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11955	4,14	553	697	12076	12597,31	12449	7,55	1144
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11876	0,02	1	100	11902	12530,08	12408	7,61	1167
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11993	0,92	119	256	12025	12566,08	12426	7,58	1137
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11967	0,09	9	170	12132	12532,52	12431	7,46	1163
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11936	0,03	2	116	12041	12510,45	12683	7,5	1152
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11942	7,02	920	1076	12069	12544	12388	7,63	1156
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11876	0,05	5	148	11961	12498,84	12394	7,57	1149
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11974	0,02	1	101	12205	12554,14	12862	7,53	1155
200	10	64	1000	0,9	0,050	OX	Inter	Azar	M+L c/E.	11891	0,04	3	131	12172	12578,53	13173	7,58	1146
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11953	0,03	2	112	12033	12535,47	12585	7,67	1144
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	12020	4,58	607	754	12093	12542,86	12230	7,54	1147
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11995	0,03	2	120	12054	12515,42	12340	7,57	1165
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11946	0,2	24	166	12115	12572,86	12226	7,61	1142
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11943	0,03	2	122	12134	12527,39	12641	7,58	1160
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	12000	4,7	621	771	12080	12540,05	12229	7,57	1150
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11974	3,23	425	588	12151	12557,17	12487	7,55	1163
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	12003	0,8	104	277	12107	12523,17	13082	7,57	1173
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11908	0,04	4	132	12190	12580,81	12580	7,55	1138
200	10	64	1000	0,9	0,100	OX	Inter	Azar	M+L c/E.	11797	0,07	7	170	12023	12452,42	12544	7,58	1180

Anexo 2: Resultados del análisis del diagrama de cajas

Instancia 200 x 10															
Op. X	Op. Mut	P0	Recambio	Prueba Mejor Fitness	PM Mejor Fitness	PC Mejor Fitness	Fitness	Op. X	Op. Mut	P0	Recambio	Prueba Fitness Óptimo	PM Fitness Óptimo	PC Fitness Óptimo	Mediana Fitness
OX	INS	NEHH	Gen.	3	0,01	0,5	11753	CX	INS	Azar	M+L-Elit.	4	0,05	0,5	11887,3
OX	INS	NEHH+Azar	M+L-Elit.	3	0,01	0,5	11601	OX	INT	NEHH+Azar	M+L-Elit. Puro	4	0,05	0,5	11886,9
CX	INT	NEHH	M+L-Elit. Puro	4	0,05	0,5	11751	PMX	INS	NEHH+Azar	M+L-Elit. Puro	4	0,05	0,5	11894,4
OX	INS	Azar	M+L-Elit. Puro	4	0,05	0,5	11723	PMX	INT	Azar	M+L-Elit.	4	0,05	0,5	11905,4
PMX	INS	NEHH	Gen.	4	0,05	0,5	11740	CX	INS	NEHH	M+L-Elit.	5	0,1	0,5	11912,2
PMX	INT	Azar	M+L-Elit.	4	0,05	0,5	11730	OX	INS	NEHH+Azar	M+L-Elit.	5	0,1	0,5	11889,9
2PX	INT	NEHH	Gen.	5	0,1	0,5	11730	CX	INT	Azar	M+L-Elit.	8	0,01	0,55	11868,1
PMX	INT	Azar	Gen.	5	0,1	0,5	11763	2PX	INT	NEHH+Azar	M+L-Elit.	9	0,05	0,55	11942,2
PMX	INT	NEHH	Gen.	5	0,1	0,5	11756	OX	INS	Azar	Gen.	9	0,05	0,55	11912,9
CX	INT	Azar	M+L-Elit.	6	0	0,55	11668	OX	INT	NEHH+Azar	Gen.	9	0,05	0,55	11905,9
PMX	INS	NEHH+Azar	M+L-Elit.	6	0	0,55	11789	2PX	INT	NEHH	M+L-Elit.	10	0,1	0,55	11915,4
PMX	INT	Azar	M+L-Elit. Puro	7	0,01	0,55	11661	CX	INS	Azar	M+L-Elit. Puro	10	0,1	0,55	11856,7
2PX	INS	NEHH+Azar	Gen.	8	0,01	0,55	11616	CX	INT	NEHH+Azar	Gen.	10	0,1	0,55	11903,1
OX	INS	NEHH+Azar	M+L-Elit. Puro	8	0,01	0,55	11772	2PX	INT	Azar	M+L-Elit.	11	0	0,6	11871,1
2PX	INS	NEHH	Gen.	9	0,05	0,55	11774	2PX	INT	Azar	Gen.	14	0,05	0,6	11847,1
CX	INS	Azar	M+L-Elit. Puro	9	0,05	0,55	11647	CX	INT	NEHH+Azar	M+L-Elit. Puro	14	0,05	0,6	11925,9
CX	INS	NEHH	M+L-Elit.	9	0,05	0,55	11765	PMX	INT	NEHH+Azar	Gen.	14	0,05	0,6	11934,1
OX	INS	NEHH	M+L-Elit.	9	0,05	0,55	11729	2PX	INT	NEHH	Gen.	15	0,1	0,6	11922,3
OX	INS	NEHH+Azar	Gen.	9	0,05	0,55	11682	CX	INS	NEHH+Azar	M+L-Elit.	15	0,1	0,6	11897,7
PMX	INS	NEHH	M+L-Elit. Puro	9	0,05	0,55	11764	CX	INT	NEHH	M+L-Elit. Puro	15	0,1	0,6	11936,9
PMX	INT	NEHH+Azar	M+L-Elit.	9	0,05	0,55	11766	OX	INS	NEHH	Gen.	15	0,1	0,6	11917
2PX	INT	Azar	M+L-Elit.	10	0,1	0,55	11641	PMX	INS	Azar	M+L-Elit. Puro	15	0,1	0,6	11896,9
2PX	INT	NEHH	M+L-Elit.	10	0,1	0,55	11733	PMX	INS	NEHH	M+L-Elit.	15	0,1	0,6	11919,6
CX	INT	Azar	Gen.	10	0,1	0,55	11713	PMX	INS	NEHH	M+L-Elit. Puro	15	0,1	0,6	11945,3
2PX	INS	NEHH+Azar	M+L-Elit. Puro	11	0	0,6	11709	PMX	INT	NEHH	M+L-Elit. Puro	15	0,1	0,6	11952
2PX	INS	Azar	Gen.	12	0,01	0,6	11685	PMX	INT	NEHH+Azar	M+L-Elit. Puro	15	0,1	0,6	11929,7
CX	INS	NEHH+Azar	M+L-Elit.	14	0,05	0,6	11739	OX	INT	NEHH+Azar	M+L-Elit.	19	0,05	0,65	11939
OX	INT	NEHH+Azar	Gen.	14	0,05	0,6	11696	PMX	INT	NEHH+Azar	M+L-Elit.	19	0,05	0,65	11933,1
OX	INT	NEHH+Azar	M+L-Elit.	15	0,1	0,6	11705	2PX	INS	NEHH	Gen.	20	0,1	0,65	11932,3
PMX	INT	NEHH	M+L-Elit.	15	0,1	0,6	11741	2PX	INS	NEHH	M+L-Elit.	20	0,1	0,65	11941
2PX	INS	Azar	M+L-Elit.	16	0	0,65	11677	2PX	INT	NEHH+Azar	Gen.	20	0,1	0,65	11921,4
2PX	INT	NEHH+Azar	M+L-Elit. Puro	17	0,01	0,65	11651	CX	INS	NEHH+Azar	M+L-Elit. Puro	20	0,1	0,65	11912,3
OX	INT	NEHH+Azar	M+L-Elit. Puro	18	0,01	0,65	11760	OX	INS	Azar	M+L-Elit.	20	0,1	0,65	11889,9
PMX	INT	NEHH	M+L-Elit. Puro	19	0,05	0,65	11754	OX	INT	NEHH	M+L-Elit. Puro	20	0,1	0,65	11934,7

CX	INS	NEHH	Gen.	20	0,1	0,65	11808		PMX	INS	Azar	M+L-Elit.	20	0,1	0,65	11904
CX	INT	NEHH+Azar	Gen.	20	0,1	0,65	11699		CX	INT	Azar	Gen.	22	0,01	0,7	11885,1
PMX	INT	NEHH+Azar	Gen.	21	0	0,7	11723		2PX	INS	Azar	M+L-Elit.	25	0,1	0,7	11879,6
2PX	INT	Azar	M+L-Elit. Puro	22	0,01	0,7	11618		2PX	INS	NEHH+Azar	Gen.	25	0,1	0,7	11913
CX	INS	Azar	Gen.	23	0,01	0,7	11646		2PX	INT	NEHH	M+L-Elit. Puro	25	0,1	0,7	11964,8
CX	INS	NEHH	M+L-Elit. Puro	28	0,01	0,75	11648		CX	INS	Azar	Gen.	25	0,1	0,7	11886,9
OX	INT	Azar	M+L-Elit.	28	0,01	0,75	11755		OX	INS	NEHH	M+L-Elit.	25	0,1	0,7	11921,7
PMX	INS	Azar	Gen.	28	0,01	0,75	11736		OX	INS	NEHH	M+L-Elit. Puro	25	0,1	0,7	11911,4
CX	INS	NEHH+Azar	Gen.	29	0,05	0,75	11741		OX	INS	NEHH+Azar	M+L-Elit. Puro	25	0,1	0,7	11935,6
OX	INT	Azar	M+L-Elit. Puro	30	0,1	0,75	11732		OX	INT	NEHH	Gen.	25	0,1	0,7	11925,9
2PX	INT	NEHH+Azar	Gen.	31	0	0,8	11719		PMX	INS	NEHH+Azar	M+L-Elit.	25	0,1	0,7	11912,1
OX	INS	Azar	Gen.	31	0	0,8	11644		PMX	INT	NEHH	Gen.	25	0,1	0,7	11944,9
CX	INT	Azar	M+L-Elit. Puro	33	0,01	0,8	11718		PMX	INT	NEHH	M+L-Elit.	25	0,1	0,7	11927,2
OX	INS	NEHH	M+L-Elit. Puro	34	0,05	0,8	11744		OX	INT	Azar	Gen.	29	0,05	0,75	11909,2
2PX	INT	NEHH	M+L-Elit. Puro	35	0,1	0,8	11757		PMX	INS	NEHH+Azar	Gen.	29	0,05	0,75	11914,8
2PX	INT	Azar	Gen.	36	0	0,85	11570		CX	INS	NEHH	M+L-Elit. Puro	30	0,1	0,75	11907,6
2PX	INT	NEHH+Azar	M+L-Elit.	36	0	0,85	11851		CX	INS	NEHH+Azar	Gen.	30	0,1	0,75	11906,3
2PX	INS	Azar	M+L-Elit. Puro	37	0,01	0,85	11660		2PX	INS	Azar	Gen.	31	0	0,8	11850,2
CX	INT	NEHH	Gen.	38	0,01	0,85	11750		CX	INT	Azar	M+L-Elit. Puro	32	0,01	0,8	11866,2
CX	INT	NEHH	M+L-Elit.	39	0,05	0,85	11761		2PX	INT	Azar	M+L-Elit. Puro	33	0,01	0,8	11891,5
CX	INT	NEHH+Azar	M+L-Elit. Puro	39	0,05	0,85	11686		CX	INT	NEHH+Azar	M+L-Elit.	33	0,01	0,8	11905,3
2PX	INS	NEHH	M+L-Elit.	40	0,1	0,85	11770		2PX	INT	NEHH+Azar	M+L-Elit. Puro	34	0,05	0,8	11894
CX	INS	NEHH+Azar	M+L-Elit. Puro	40	0,1	0,85	11696		OX	INS	NEHH+Azar	Gen.	34	0,05	0,8	11904
OX	INT	NEHH	Gen.	40	0,1	0,85	11682		OX	INT	NEHH	M+L-Elit.	35	0,1	0,8	11932,2
PMX	INS	Azar	M+L-Elit.	40	0,1	0,85	11745		PMX	INS	Azar	Gen.	35	0,1	0,8	11920,7
CX	INT	NEHH+Azar	M+L-Elit.	41	0	0,9	11722		PMX	INT	Azar	M+L-Elit. Puro	35	0,1	0,8	11927,5
PMX	INT	NEHH+Azar	M+L-Elit. Puro	41	0	0,9	11783		PMX	INT	Azar	Gen.	39	0,05	0,85	11939,7
OX	INT	Azar	Gen.	42	0,01	0,9	11772		2PX	INS	NEHH+Azar	M+L-Elit.	40	0,1	0,85	11888,3
2PX	INS	NEHH+Azar	M+L-Elit.	43	0,01	0,9	11737		CX	INT	NEHH	M+L-Elit.	40	0,1	0,85	11939,8
OX	INT	NEHH	M+L-Elit. Puro	43	0,01	0,9	11726		OX	INT	Azar	M+L-Elit.	40	0,1	0,85	11919,7
PMX	INS	NEHH	M+L-Elit.	43	0,01	0,9	11763		PMX	INS	NEHH	Gen.	40	0,1	0,85	11925,9
2PX	INS	NEHH	M+L-Elit. Puro	44	0,05	0,9	11692		2PX	INS	Azar	M+L-Elit. Puro	43	0,01	0,9	11875,5
OX	INS	Azar	M+L-Elit.	44	0,05	0,9	11680		2PX	INS	NEHH	M+L-Elit. Puro	45	0,1	0,9	11908,7
PMX	INS	Azar	M+L-Elit. Puro	44	0,05	0,9	11707		2PX	INS	NEHH+Azar	M+L-Elit. Puro	45	0,1	0,9	11902,7
PMX	INS	NEHH+Azar	Gen.	44	0,05	0,9	11769		CX	INS	NEHH	Gen.	45	0,1	0,9	11928,2
CX	INS	Azar	M+L-Elit.	45	0,1	0,9	11736		CX	INT	NEHH	Gen.	45	0,1	0,9	11940,3
OX	INT	NEHH	M+L-Elit.	45	0,1	0,9	11799		OX	INS	Azar	M+L-Elit. Puro	45	0,1	0,9	11897,7
PMX	INS	NEHH+Azar	M+L-Elit. Puro	45	0,1	0,9	11765		OX	INT	Azar	M+L-Elit. Puro	45	0,1	0,9	11907,9

Tabla 2: Resultados obtenidos de analizar los diagramas de caja

Anexo 3: Función principal del AG

```

#include "global.h"

#include <iostream>
#include <iomanip>
#include <algorithm>
#include <fstream>
#include <cstdlib>
#include <sstream>
#include <vector>
#include <time.h>
#include <string.h>
#include <stdlib.h>

#include "cromosoma.cpp"
#include "intercambio.cpp"
#include "mutacion.cpp"
#include "insercion.cpp"
#include "cruceDeOrden.cpp"
#include "cruceDeCiclo.cpp"
#include "cruceDosPuntos.cpp"
#include "cruceParcialmenteMapeado.cpp"
#include "torneoBinario.cpp"
#include "reemplazo.cpp"

using namespace std;

void menu(int, int);
void cargarMatriz();

int main(int argc, char** argv)
{
    if(argc < 11)
    {
        std::cout << "ag <cant_Tareas> <cant_Maq> <op_Cruce> <op_Mut>
<tam_Poblacion> <cant_Iteraciones> <PC> <PM> <Heurística> <Recambio>" <<
std::endl;
        std::cout << "Operadores Cruce: 1-OX, 2-2PX, 3-PMX, 4-CX" <<
std::endl;
        std::cout << "Operadores Mutacion: 1-Insercion, 2-Intercambio" <<
std::endl;
        std::cout << "Heurística: 1-Generación al azar. 2- NEHH Puro. 3-
NEHH + azar." << std::endl;
        std::cout << "Recambio: 1-Generacional con elitismo, 2- Mu + Lambda
Torneo Binario, 3- Mu + Lambda Elitismo" << std::endl;

        exit(0);
    }
    cantidadTareas = atoi(argv[1]);
    cantidadMaquinas = atoi(argv[2]);

    int operadorCruce = atoi(argv[3]);
    int operadorMutacion = atoi(argv[4]);

    tamañoPoblacion = atoi(argv[5]);
    cantidadIteraciones = atoi(argv[6]);

```

```

probabilidadCruce = atof(argv[7]);
probabilidadMutacion = atof(argv[8]);

heuristic = atoi(argv[9]);
recambio = atoi(argv[10]);

//Estadisticas
int mejorFitnessPoblacionInicial = 0;
float promedioFitnessPoblacionInicial = 0;

float promedioFitnessPoblacionFinal = 0;

float tiempoMejorSolucion = 0;
int cantidadEvaluacionesMejorSolucion = 0;
int generacionMejorSolucion = 0;
int mejorFitness = 0;

float tiempoTotalEjecucion = 0;
int cantidadTotalGeneraciones = 0;
int cantidadTotalEvaluaciones = 0;
//Fin Estadisticas

clock_t totalTime;

TorneoBinario torneo;
CruceDeOrden ox;
CruceDosPuntos px;
CruceParcialmenteMapeado pmx;
CruceDeCiclo cx;

Insercion inser;
Intercambio inter;

Reemplazo reemplazo;

cargarMatriz();

srand( time(NULL) );

Cromosoma *poblacion, *hijos;
//poblacion de la cual se seleccionaran los padres a cruzar
poblacion = (Cromosoma *)malloc (tamanoPoblacion * sizeof(Cromosoma));
//hijos resultantes del cruce de 2 padres
hijos = (Cromosoma *)malloc (tamanoPoblacion * sizeof(Cromosoma));

totalTime = clock();
cantidadEvaluaciones = 0;
promedioFitnessPoblacionInicial = 0;
cantidadTotalGeneraciones = 0;

//generacion de la poblacion inicial (P0)
for(int i = 0; i < tamanoPoblacion; i++)
{
    if(heuristic == 1)
    {
        poblacion[i] = Cromosoma(0);
    }
}

```

```

    }
    else if(heuristic == 2)
    {
        poblacion[i] = Cromosoma(1);
    }
    else if(heuristic == 3)
    {
        if(i < (tamanoPoblacion/2)) //la primera mitad al azar
        {
            poblacion[i] = Cromosoma(0);
        }
        else //la segunda mitad con NEHH
        {
            poblacion[i] = Cromosoma(1);
        }
    }

    poblacion[i].calcularFenotipo();
    poblacion[i].setReEvaluar(0);

    if(i == 0)
    {
        mejorFitnessPoblacionInicial = poblacion[0].getFenotipo();
        cantidadEvaluacionesMejorSolucion = cantidadEvaluaciones;
        tiempoMejorSolucion = (((double)clock() - totalTime) /
CLOCKS_PER_SEC);
    }
    if(mejorFitnessPoblacionInicial > poblacion[i].getFenotipo())
    {
        mejorFitnessPoblacionInicial = poblacion[i].getFenotipo();
        cantidadEvaluacionesMejorSolucion = cantidadEvaluaciones;
        tiempoMejorSolucion = (((double)clock() - totalTime) /
CLOCKS_PER_SEC);
    }
    promedioFitnessPoblacionInicial += poblacion[i].getFenotipo();
}

mejorFitness = mejorFitnessPoblacionInicial;
generacionMejorSolucion = 1;
promedioFitnessPoblacionInicial = promedioFitnessPoblacionInicial /
tamanoPoblacion;

//ciclo iterativo, condicion de corte cantidad de iteraciones
for(int itera = 0; itera < cantidadIteraciones; itera++)
{
    for(int m = 0; m < (tamanoPoblacion-1); m+=2)
    {
        //seleccion de padres a cruzar
        hijos[m] = torneo.seleccionar(poblacion, tamanoPoblacion);
        hijos[m+1] = torneo.seleccionar(poblacion, tamanoPoblacion);

        //cruce de 2 padres, obtenido 2 hijos. Se utiliza la misma
variable como entrada y salida.
        if(operadorCruce == 1)
        {
            //Cruce de Orden
            ox.cruce(hijos[m], hijos[m+1]);
        }
        else if(operadorCruce == 2)

```

```

    {
        //Cruce en 2 Puntos
        px.cruce(hijos[m], hijos[m+1]);
    }
    else if(operadorCruce == 3)
    {
        //Cruce Parcialmente Mapeado
        pmx.cruce(hijos[m], hijos[m+1]);
    }
    else
    {
        //Cruce de Ciclo
        cx.cruce(hijos[m], hijos[m+1]);
    }

    //mutacion de 2 hijos. Se utiliza la misma variable como
    entrada y salida.
    if(operadorMutacion == 1)
    {
        //Mucacion por Insercion
        inser.mutar(hijos[m]);
        inser.mutar(hijos[m+1]);
    }
    else
    {
        //Mucacion por Intercambio
        inter.mutar(hijos[m]);
        inter.mutar(hijos[m+1]);
    }
}

//re-evaluo el cromosoma solo si fue modificado a traves de un
operador de cruce o mutacion
for(int i = 0; i < tamañoPoblacion; i++)
{
    if(hijos[i].reEvaluar() == 1)
    {
        hijos[i].calcularFenotipo();
        hijos[i].setReEvaluar(0);
    }
}

//recambio de la poblacion
if(recambio == 1)
{
    //Reemplazo Generacional con elitismo
    reemplazo.reemplazoGeneracional(poblacion, hijos);
}
else if(recambio == 2)
{
    reemplazo.muMasLambdaElitismo(poblacion, hijos);
}
else if(recambio == 3)
{
    reemplazo.muMasLambdaElistimoPuro(poblacion, hijos);
}

//seleccion del mejor individuo

```

```

        for(int i = 0; i < tamañoPoblacion; i++)
        {
            if((int)mejorFitness > (int)poblacion[i].getFenotipo())
            {
                mejorFitness = poblacion[i].getFenotipo();
                tiempoMejorSolucion = (((double)clock() - totalTime) /
CLOCKS_PER_SEC);
                generacionMejorSolucion = itera+1;
                cantidadEvaluacionesMejorSolucion =
cantidadEvaluaciones;
            }

            cantidadTotalGeneraciones++;
        }

//salida de resultados
cantidadTotalEvaluaciones = cantidadEvaluaciones;
tiempoTotalEjecucion = (((double)clock() - totalTime) / CLOCKS_PER_SEC);

promedioFitnessPoblacionFinal = 0;
for(int i = 0; i < tamañoPoblacion; i++)
{
    promedioFitnessPoblacionFinal += poblacion[i].getFenotipo();
}
promedioFitnessPoblacionFinal = promedioFitnessPoblacionFinal /
tamañoPoblacion;

std::cout << cantidadTareas;
std::cout << "\t" << cantidadMaquinas;
std::cout << "\t" << tamañoPoblacion;
std::cout << "\t" << cantidadIteraciones;
cout.setf(ios::fixed|ios::showpoint);
//std::cout << setprecision(2);
std::cout << "\t" << probabilidadCruce;
std::cout << "\t" << probabilidadMutacion;
//cout.setf(ios::fixed|ios::showpoint);
cout.setf(ios::fixed);
std::cout << setprecision(0);
if(operadorCruce == 1)
    std::cout << "\tOX";
else if(operadorCruce == 2)
    std::cout << "\t2PX";
else if(operadorCruce == 3)
    std::cout << "\tPMX";
else
    std::cout << "\tCX";

if(operadorMutacion == 1)
    std::cout << "\tInser";
else
    std::cout << "\tInter";

if(heuristic == 1)
    std::cout << "\tAzar";
else if(heuristic == 2)
    std::cout << "\tNEHH Puro";
else
    std::cout << "\tNEHH + Azar";

```

```

if(recambio == 1)
    std::cout << "\tGen. c/E.";
else if(recambio == 2)
    std::cout << "\tM+L c/E.";
else
    std::cout << "\tM+L c/E. Puro";

std::cout << "\t" << mejorFitness;
cout.setf(ios::fixed|ios::showpoint);
std::cout << setprecision(2);
std::cout << "\t" << tiempoMejorSolucion;
cout.setf(ios::fixed);
std::cout << setprecision(0);
std::cout << "\t" << generacionMejorSolucion;
std::cout << "\t" << cantidadEvaluacionesMejorSolucion;

std::cout << "\t" << mejorFitnessPoblacionInicial;

cout.setf(ios::fixed|ios::showpoint);
std::cout << setprecision(2);
std::cout << "\t" << promedioFitnessPoblacionInicial;
std::cout << "\t" << promedioFitnessPoblacionFinal;

std::cout << "\t" << tiempoTotalEjecucion;
cout.setf(ios::fixed);
std::cout << setprecision(0);
std::cout << "\t" << cantidadTotalGeneraciones;
std::cout << "\t" << cantidadTotalEvaluaciones << "\n";

//Liberar recursos del Sistema
free(poblacion);
free(hijos);

return 0;
}

void cargarMatriz()
{
    char lectura[200];
    std::stringstream ss;
    ss << "../..instancias/tai_" << cantidadTareas << "_" << cantidadMaquinas;
    std::string filename = ss.str();
    ifstream archivo(filename.c_str());

    matriz = (int **) malloc(cantidadTareas*sizeof(int*));

    for(int j=0; j<cantidadTareas; j++)
    {
        matriz[j] = (int *) malloc(cantidadMaquinas*sizeof(int));
    }

    for(int i = 0; i < 3; i++)
        archivo.getline(lectura, 200, '\n');
}

```

```
for(int i = 0; i < cantidadMaquinas; i++)
{
    for(int j = 0; j < cantidadTareas; j++)
    {
        archivo >> matriz[i][j];
    }
}
archivo.close();
}
```


Anexo 4: Script de ejecución del AG

```
#!/bin/bash

if (($# < 8))
then
    echo "ERROR: se requieren los siguientes parametros"
    echo "./script <tareas> <maquinas> <Op. X> <Op. M> <tam. Pob.>
<iteraciones> <heuristica> <recambio>"
    echo "<Op. X>: 1-OX, 2-PX, 3-PMX, 4-CX"
    echo "<Op. M>: 1-INS, 2-INT"
    echo "<heuristica>: 1-Azar, 2-NEHH, 3-Azar+NEHH"
    echo "<recambio>: 1-Generacional, 2-M+L-TorneoBinario, 3-M+L-Elitismo"
    exit 0
fi
jobs=$1
machines=$2
opX=$3
opM=$4
tamPob=$5
iteraciones=$6
heuristica=$7
recambio=$8

if (($3 == 1))
then
    opXL="OX"
fi

if (($3 == 2))
then
    opXL="2PX"
fi
if (($3 == 3))
then
    opXL="PMX"
fi
if (($3 == 4))
then
    opXL="CX"
fi

if (($4 == 1))
then
    opML="INS"
fi
if (($4 == 2))
then
    opML="INT"
fi

if (($7 == 1))
then
    hL="Azar"
fi
if (($7 == 2))
then
```

```

        hL="NEHH"
    fi
    if (($7 == 3))
    then
        hL="NEHH+Azar"
    fi

    if (($8 == 1))
    then
        rL="Generacional"
    fi
    if (($8 == 2))
    then
        rL="M+L-TBinario"
    fi
    if (($8 == 3))
    then
        rL="M+L-Elit"
    fi

    file="../../../pruebas/output_"$jobs_"$machines_"$opXL_"$opML_"$hL_"$rL".csv"

    echo -e "Cant. Tareas\tCant. Maq.\tPoblación\tIteraciones\tPc\t\tPm\tOp. X\tOp.
    Mut.\tH° P0\tH° Reemplazo\tM° Fitness\tT° M° Fitness\tGeneración M°\tCant.
    Eval. M°\tM° PI\tPromedio PI\tPromedio PF\tT° Total\tCant. Total
    Generaciones\tCant. Total Eval" >> $file

    for pc in 0.5 0.55 0.6 0.65 0.7 0.75 0.80 0.85 0.9
    do
        for pm in 0.001 0.005 0.01 0.05 0.1
        do
            for ejecucion in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
            25 26 27 28 29 30
            do
                `../ag $jobs $machines $opX $opM $tamPob $iteraciones $pc
                $pm $heuristica $recambio >> $file `
                wait ${!}
            done
            echo -e " " >> $file
        done
    done
done

```