Sistemas de Inteligencia Artificial - 2019 1C

TP Nº3: Algoritmos Genéticos

Integrantes

Nombre	*Correo*	*Legajo*
Dammiano, Agustín	adammiano@itba.edu.ar	57702
Donoso Naumczuk, Alan	adonoso@itba.edu.ar	57583
Sanz Gorostiaga, Lucas	lsanz@itba.edu.ar	56312
Torreguitar, José	jtorreguitar@itba.edu.ar	57519

Fecha: 05/06/2019

1. Introducción

Este informe describe y analiza la implementación de un programa que busca obtener el mejor personaje de un videojuego mediante el uso de algoritmos genéticos, donde el criterio para saber cual es mejor se basa en la maximización de valores basados en ciertas características. Se exploran distintas variaciones sobre el algoritmo genético y mediante pruebas empíricas se desea sacar conclusiones sobre las mismas. La implementación del trabajo se realizó en Java 8.

2. Representación de un personaje

Se almacenan las características de un personajes en un arreglo, con el fin de simplificar la cruza y la mutación de los mismos. Las características de dicho personaje consiste de la altura, botas, pechera, guantes, casco y arma. Se utilizó como función de fitness a la función de desempeño especificada en la consigna del trabajo práctico.

3. Implementación del algoritmo genético

3.1. Algoritmos de reemplazo

Se implementaron tres algoritmos de reemplazo para generar las siguiente generación, los cuales se denominaron uno, dos y tres. El primero consta de elegir N padres de una población de N, los cuales pueden ser repetidos y luego son cruzados de a dos. Finalmente los N hijos se mutan y pasan a la siguiente generación. Los métodos de reemplazo dos y tres se implementaron como se vio en la clase.

3.2. Selección

Todos los métodos de selección fueron implementados como se vieron en clase. Estos se dividieron en tres tipos: ruleta, torneo y elite.

3.2.1. Ruleta

Los métodos de ruleta son aquellos que heredan de la clase `RouletteSelection` dado que todos comparten el mismo flujo principal para seleccionar cromosomas. Estos son ruleta mismo, ranking, universal y Boltzmann. Su principal diferencia está en cómo se calculan las aptitudes relativas para cada cromosoma y, en el caso de la selección universal sólo difiere en cómo se eligen los randoms. A pesar de esta diferencia, todos los métodos anteriores calculan las aptitudes acumuladas como está especificado en las filminas vistas en clase. Luego se encargan de seleccionar a aquellos cromosomas que cumplan la condición de que algún random se encuentre en el intervalo entre su aptitud acumulada y la aptitud acumulada anterior.

Una peculiaridad entre los métodos de selección del tipo ruleta se encuentra en el método de Boltzmann. El mismo depende de una temperatura T que a su vez depende de la generación en la que se está aplicando el método. Inicialmente se había propuesto utilizar $T(i) = e^{-0.0001i}$ para la i-ésima generación, con el conjunto de datos de prueba y una configuración de una cota de 1.000 generaciones máximo. Esto se vio obsoleto cuando se empezó a utilizar el conjunto de datos totales y al aumentar la cantidad máxima de generaciones a 10.000. Es gracias a esto que la fórmula para la temperatura quedó como $T(i) = e^{-\frac{i}{10G}}$ con G siendo la máxima cantidad de generaciones.

3.2.2. Torneo

Los métodos de torneo implementados son dos: el torneo determinístico y el probabilístico. Inicialmente se había decidido que en los m cromosomas que se seleccionan para competir, no habrían

repetidos ya que no tendría, suponiendo m=3, agarrar dos veces el mismo cromosoma y hacerlo competir consigo mismo.

Si bien esto es correcto para una población diversa, puede llevar a problemas si la población consiste de individuos iguales o casi iguales ya que seleccionar m cromosomas aleatorios y distintos sería imposible con una diversidad menor a m. Es por esto que al momento de elegir los m cromosomas aleatorios que van a competir entre sí, se dividen la selección en dos casos: elegir únicamente cromosomas distintos si la diversidad es mayor o igual a m, y elegir una combinación de los cromosomas existentes (duplicados) y que se peleen a sí mismos.

Al decir que hay una diversidad menor a m lo que ocurre en la implementación es que la lista de cromosomas se filtra mediante la creación de un `HashSet` basado en esa lista para que no queden cromosomas iguales. Una vez hecho esto, se vuelve a crear una lista de cromosomas, pero esta vez conteniendo únicamente los distinguidos. De esta forma sólo quedan cromosomas representativos de sus clases de equivalencia para que, al momento de elegir los m cromosomas que vayan a competir (con la diversidad mayor o igual a m, pero sin ser demasiado mayor) no tarde tanto la ejecución del programa tomando aleatoriamente cromosomas duplicados.

3.2.3. Elite

La selección por elitismo es probablemente la más simple de todas, y al mismo tiempo la que más rápidamente pueda llevar a una convergencia prematura dado que siempre elige los cromosomas más aptos de la población sin permitir la intromisión de otros cromosomas.

3.3. Cruza

Para la implementación de los métodos de cruza se decidió que los alelos de un cromosoma se vieran representados como un `ArrayList<Object>` y de esa manera realizar la cruza de dos cromosomas como se muestra en las filminas vistas en clase, con la diferencia de que en vez de tener un arreglo de bits se tiene un arreglo de `Object`. Así como en el ejemplo de las filminas, en el caso del cruce anular en el que el *locus+segmento* sea mayor al tamaño del arreglo, se prosigue intercambiando los alelos del principio del arreglo, mediante la operación de módulo (%).

3.4. Mutación

Las mutaciones pueden ser de un gen o multigen y también uniforme o no uniforme. La característica de ser de un gen o multigen consta de sí por cada mutación se puede mutar un gen (en el primer caso) o varios (en el segundo caso). En cuanto a si es uniforme o no uniforme, habla de si la probabilidad de mutar cambia de generacion a generacion. En el caso de que cambie la misma sigue la siguiente fórmula, donde *initProbability* es un parámetro configurable y se cumple que F(0) = *initProbability*:

$$F(X) = initProbability\left[1 - 2\left(\frac{1}{1+e^{-0.01X}} - 0.5\right)\right]$$

4. Procedimiento

Se acordó una configuración base, para la cual se realizó una corrida como referencia. Luego, manteniendo la misma semilla para la generación de números aleatorios, se fueron variando los distintos tipos de parámetros para poder realizar un análisis comparativo de cada uno de ellos.

Dado que el personaje asignado fue Guerrero 3, se seleccionaron los coeficientes de ataque (0.6), defensa (0.4), fuerza (0.8), agilidad (0.9), pericia (0.9), resistencia (1.2) y vida (1.1) acorde a ello.

Para la cantidad de la población se eligió 1000 cromosomas. Como criterio de corte se eligió alcanzar un número máximo de 10000 generaciones.

Para el método de cruza se utilizó cruce de dos puntos con probabilidad 0.6. Para mutación se optó por la uniforme, multigen, con probabilidad inicial 0.1.

Para el método de reemplazo, se escogió el tipo 2 con k igual a 500; y en cuanto a métodos de selección, se eligió ruleta con proporción en 1.

La semilla para generación de números aleatorios quedó con valor 1559702205841.

La configuración base mencionada puede verse en el archivo de configuración de la Configuración 1 del Anexo.

5. Análisis del resultado

5.1. Métodos de selección

5.1.1. Ruleta (base)

El método de ruleta se conforma con lo esperado al mantener una diversidad relativamente alta mientras aún tiende hacia la mejora de aptitud aunque esta sea lenta y resulte inferior comparada con los mejores métodos. Se puede observar que a medida que se aumenta la aptitud esta lo hace oscilando de a grandes saltos, a diferencia de otros métodos, como ranking o tournament.

5.1.2. Elite

Como se esperaba, este método pierde diversidad muy rápidamente, estabilizandose alrededor de una diversidad de aproximadamente 250 de la población. También se observa un incremento muy rápido en la aptitud siendo este el método de selección que consigue la mayor aptitud de todos.

5.1.3. Torneo

La selección por torneo determinístico logra mantener una diversidad relativamente alta y aún así alcanzar rápidamente una aptitud alta.

5.1.4. Torneo Probabilístico

La selección por torneo probabilístico permite mantener una de las diversidad mayor a la del torneo normal, esto se debe a que en esta además hay un factor de azar que interviene en la selección. Es el método que llegó a la peor aptitud de todos.

5.1.5. Boltzmann

Se puede apreciar que la diversidad cae muy abruptamente, tan abruptamente como la aptitud incrementa. Esto es contrario a lo que suele suceder con este método de selección el cual se supone incentiva la exploración. Esto puede deberse a que la función de temperatura utilizada comienza de por sí con una temperatura muy baja, haciendo que la presión de selección sea muy elevada al comienzo en lugar de solo al final.

5.1.6. Universal

En comparación a otros métodos de selección este presenta una mayor diversidad. La cual oscila entre un rango grande. Lo cual puede atribuirse a cómo se generan los random para hacer la ruleta de este método.

5.1.7. Ranking

La acelerada baja de diversidad en este método puede deberse a la gran diferencia de aptitud entre los individuos menos aptos y los más aptos, teniendo los primeros muy baja probabilidad de ser seleccionados en comparación a los otros. El mismo razonamiento explica el abrupto crecimiento en aptitud

en las primeras generaciones, seguido de incrementos leves en las siguientes generaciones donde solo compiten los mejores especímenes.

5.2. Métodos de mutación

5.2.1. Uniforme, multigen (base)

Como en esta configuración existe la posibilidad de que cada gen mute, es difícil que la diversidad baje. Uno se vería tentado a pensar que al tener mucha mutación lo que pasaría es que los individuos cambian constantemente y no puedan tender hacia un mejor aptitud, pero esto no pasa porque el método de selección otorga mejores posibilidades de pasar de generación y reproducirse a los individuos con mayor aptitud.

5.2.2. Uniforme, gen

En este método se observa que la diversidad cae abruptamente a alrededor del 250 de la población. Comparando esto con la diversidad obtenida con los mismos parámetros pero con mutación multigen, podemos afirmar que la mutación multigen es mucho más efectiva a la hora de generar diversidad que la mutación de un solo gen.

5.2.3. No uniforme, multigen

En ámbos métodos con mutación no uniforme puede observarse que la elección correcta de la función de decrecimiento de la probabilidad de mutación es esencial para mantener una diversidad saludable. Con el decrecimiento elegido la diversidad decae muy rápido, haciendo que la evolución está basada casi completamente en la cruza luego de las primeras centenas de generaciones, se puede observar que esta evolución no es apropiada dados los resultados de aptitud.

5.2.4. No uniforme, gen

En esta configuración se puede notar que cae rápidamente la diversidad porque la probabilidad de mutación cae. La misma decrece más rápido que la configuración no uniforme y multigen, esto puede atribuirse a que el factor multigen genera que un individuo mute mas. Cuando la probabilidad de mutación se acerca al 0 se puede notar que la aptitud se estanca en un valor.

5.3. Método de cruza

Se observa que todos los métodos de cruce tienen un impacto similar sobre la diversidad, puesto que para todos esta oscila alrededor del 750. En cuanto a la aptitud vemos que el método de dos puntos fue el que mayor aptitud logró, mientras que la de un punto se vio estancada alrededor de 24 puntos de aptitud y, finalmente, la cruza anular y uniforme se comportaron similarmente en este aspecto.

5.4. Mejor personaje encontrado

Después de realizar varias corridas de distintos algoritmos genéticos llegamos a obtener un fitness de 27.252888895386345 que surge de la configuración de élite del anexo. Dicho personaje posee las siguientes características:

Altura: 1.9151881288187305

Botas

Id: 402379
Fuerza: 0.8952108875870094

Agilidad: 7.890225895339538
Pericia: 0.8021923156307567
Resistencia: 0.14496950312703064
Vida: 0.22178588621404832

Pechera

Id: 86387
Fuerza: 17.97629844914879
Agilidad: 20.790450304062432
Pericia: 0.5853237575607095
Resistencia: 0.17671736672624094
Vida: 0.3865136669794049

Guantes

Id: 812209
Fuerza: 5.033878120265168
Agilidad: 3.3124066510411554
Pericia: 1.5200365575114227
Resistencia: 0.025508747961115383
Vida: 0.0736365140351266

Casco

Id: 581621
Fuerza: 9.998504033523353
Agilidad: 14.715558949543565
Pericia: 4.820360157516302
Resistencia: 0.2578647503443574
Vida: 0.117903278737343

Arma

Id: 988980
Fuerza: 34.25524636296503
Agilidad: 10.289241771566909
Pericia: 4.073521269198334
Resistencia: 0.8849680404548659
Vida: 0.02946352273026486

6. Conclusión

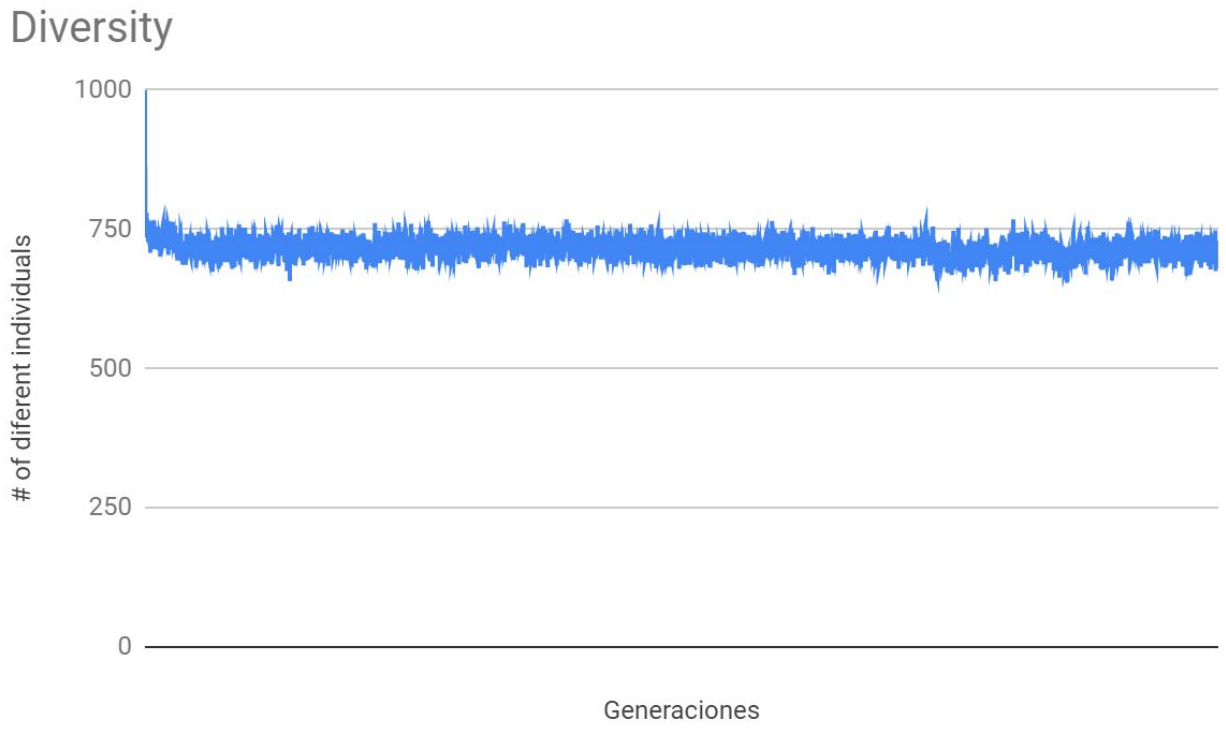
Luego de haber variado parámetros dentro del algoritmos genético es posible concluir que no todos influyen de la misma forma para encontrar un cromosoma ideal, con una convergencia no demasiado brusca como puede ser la del método de selección elitista. Se pudo percibir el efecto de variar los métodos de cruza no altera demasiado los resultados. Por otro lado, determinar si las mutaciones serán o no uniformes tiene una gran influencia a la hora de considerar qué parámetros elegir para mantener la diversidad.

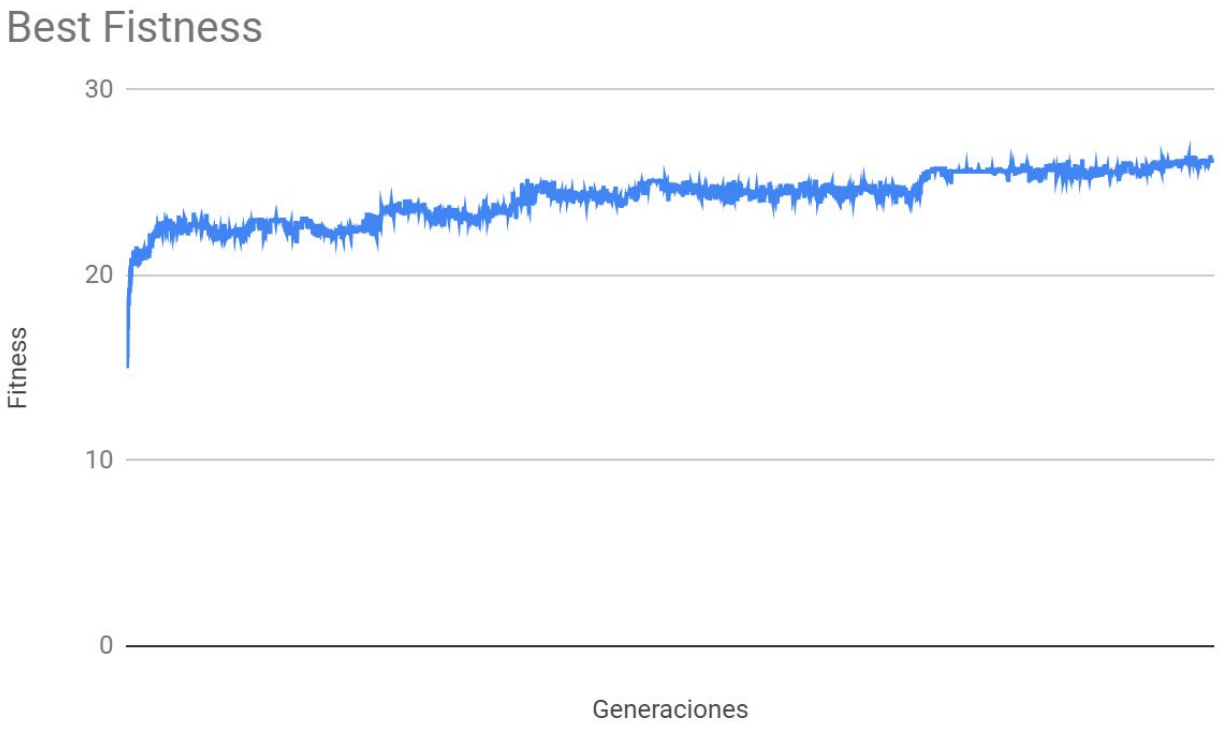
Dependerá del objetivo establecido lo que se considere o no como un parámetro *importante*. Si se prioriza obtener al individuo más apto, uno debería buscar tener una población que mantenga su diversidad, o mínimamente su capacidad de mutar. Pero esto tal vez podría tener como consecuencia que llevase más tiempo llegar a un mismo nivel de aptitud que con una configuración más restrictiva con respecto a la diversidad de la población (ej: selección elitista, sin multigen ni mutación uniforme). En este último caso el tiempo podría transformarse en un recurso escaso (con una cantidad máxima de generaciones relativamente grande), aunque uno se asegura que el algoritmo pondrá a prueba a una mayor variedad de individuos para poder llegar al resultado óptimo.

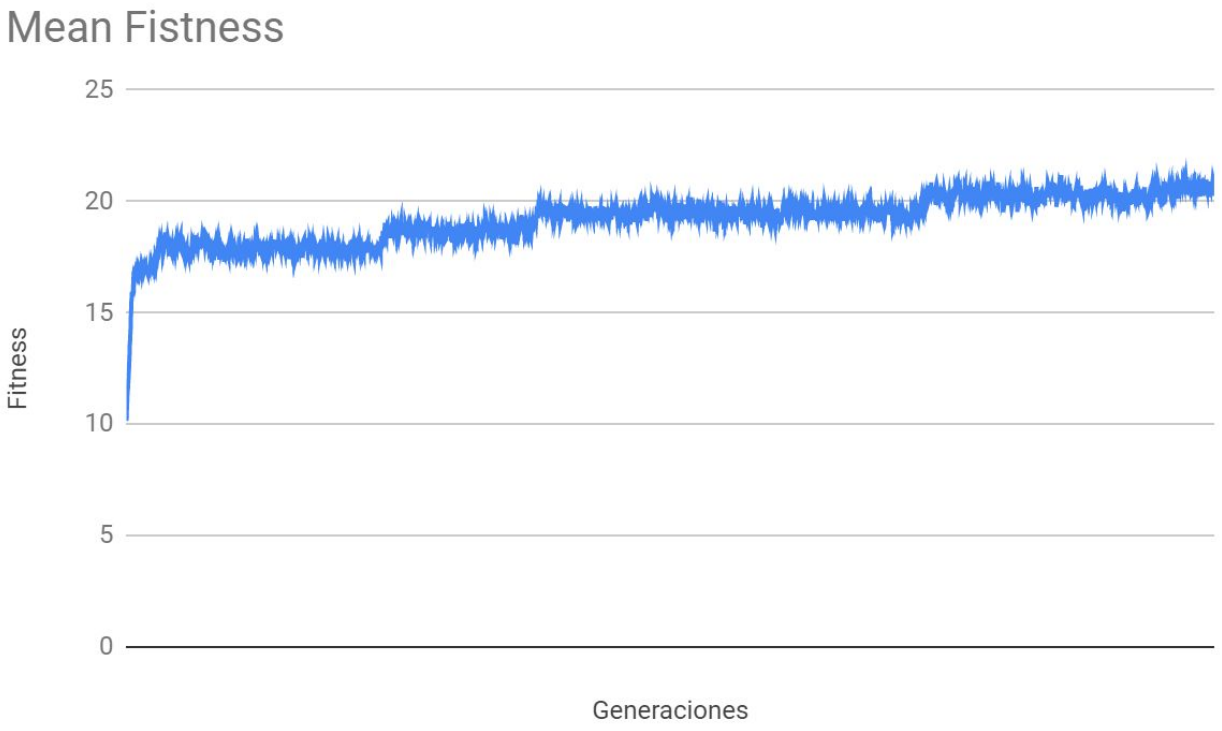
7. Anexo

Configuración 1: (configuración base)


```
{
  "classType": "warrior",
  "multipliers":
  {
    "strength": 0.8,
    "agility": 0.9,
    "expertise": 0.9,
    "resistance": 1.2,
    "life": 1.1
  },
  "populationQuantity": 1000,
  "cutCriteria":
  {
    "maxGeneration": 10000,
    "quantityOfGenerationToPerformChecks": 10000000,
    "fitness": 100000
  },
  "crossover": "twoPoint",
  "crossProbability": 0.6,
  "uniformProbability": 0.5,
  "mutation":
  {
    "isMultiGen": true,
    "isUniform": true,
    "initProb": 0.1
  },
  "replaceMethod":
  {
    "type": "second",
    "k": 500
  },
  "firstSelectionMethod":
  [
    {
      "selectionType": "roulette",
      "proportion": 1
    }
  ],
  "secondSelectionMethod":
  [
    {
      "selectionType": "roulette",
      "proportion": 1
    }
  ],
  "randomSeed": 1559702205841
}
```

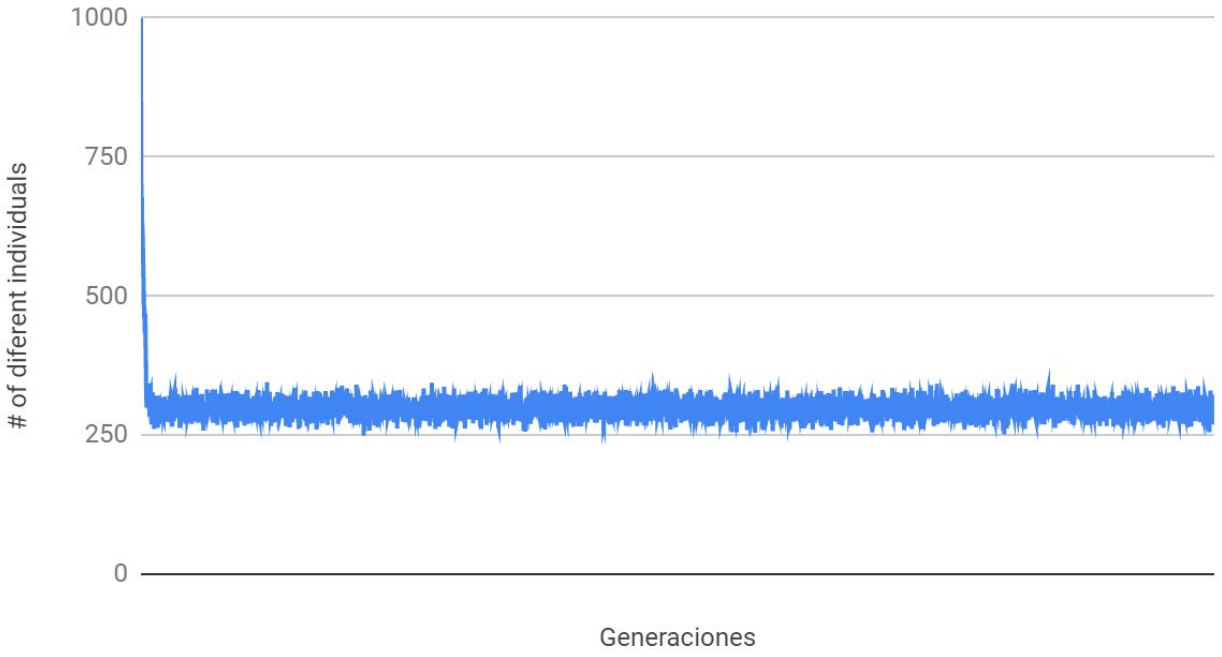





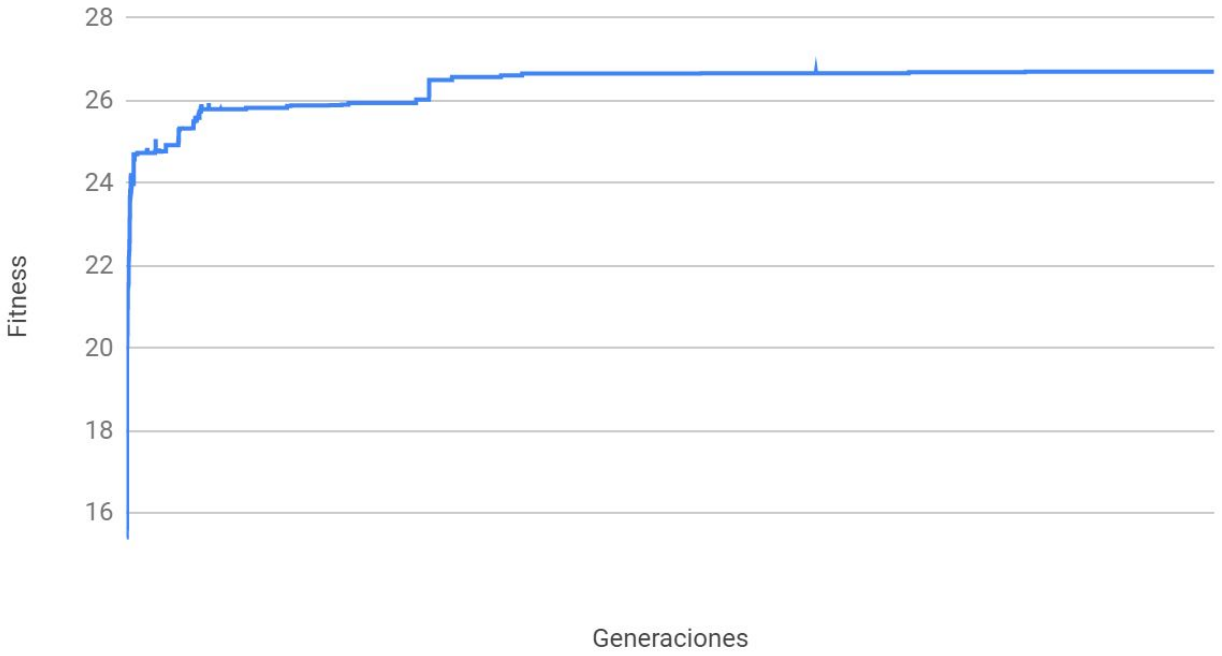


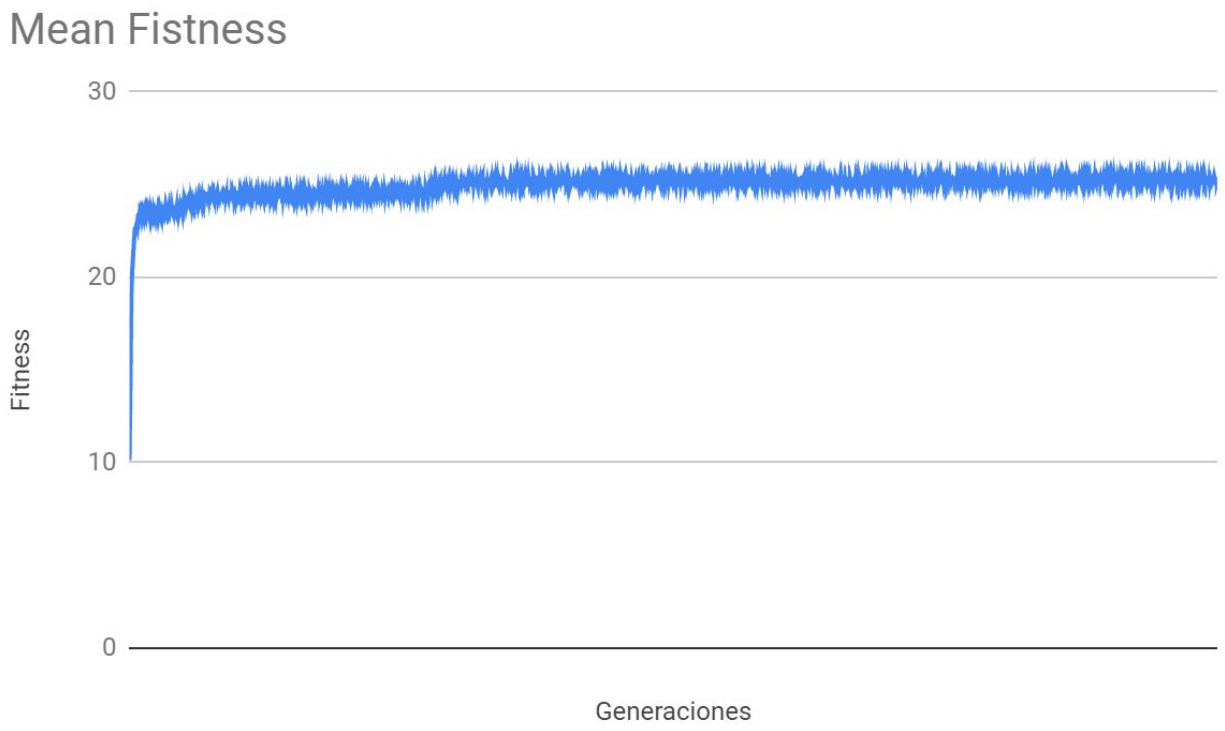
Configuración 2: Ranking → igual a la configuración base pero con *"ranking"* para ambos métodos de selección

Diversity



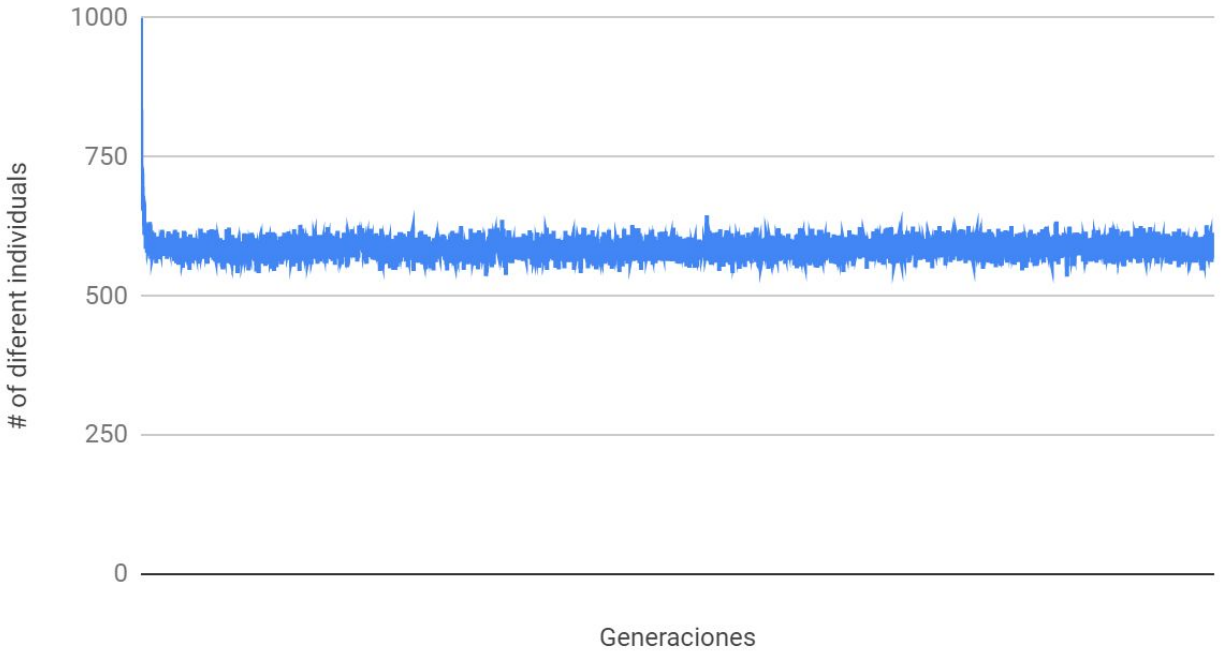
Best Fistness



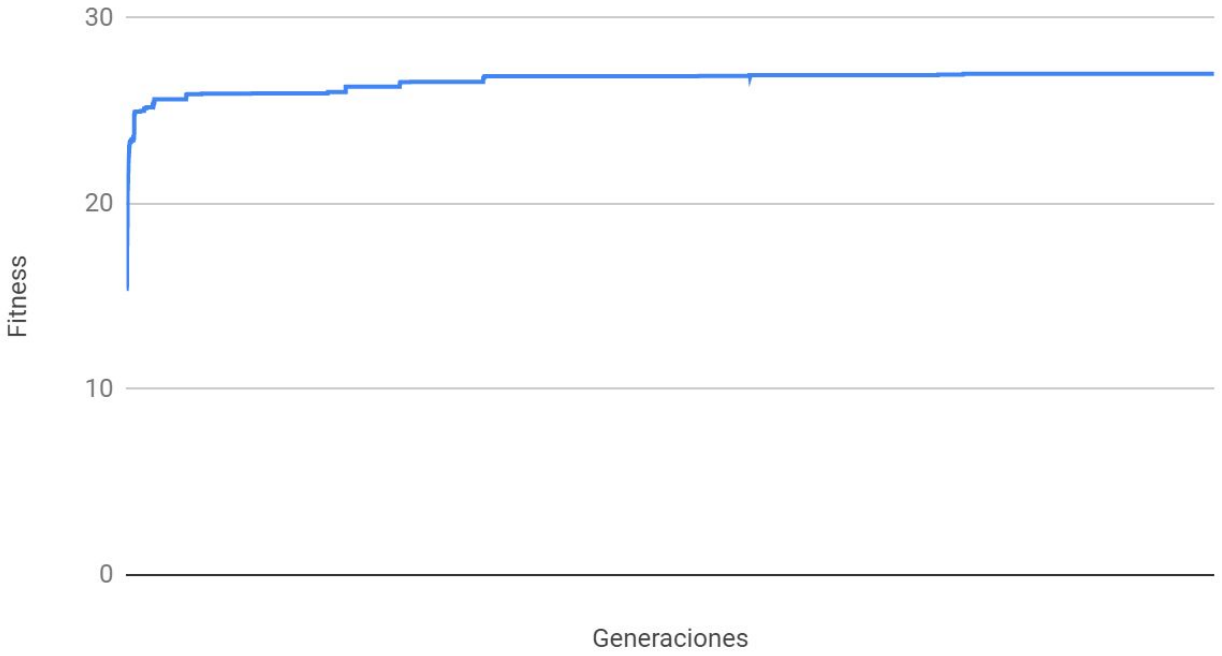


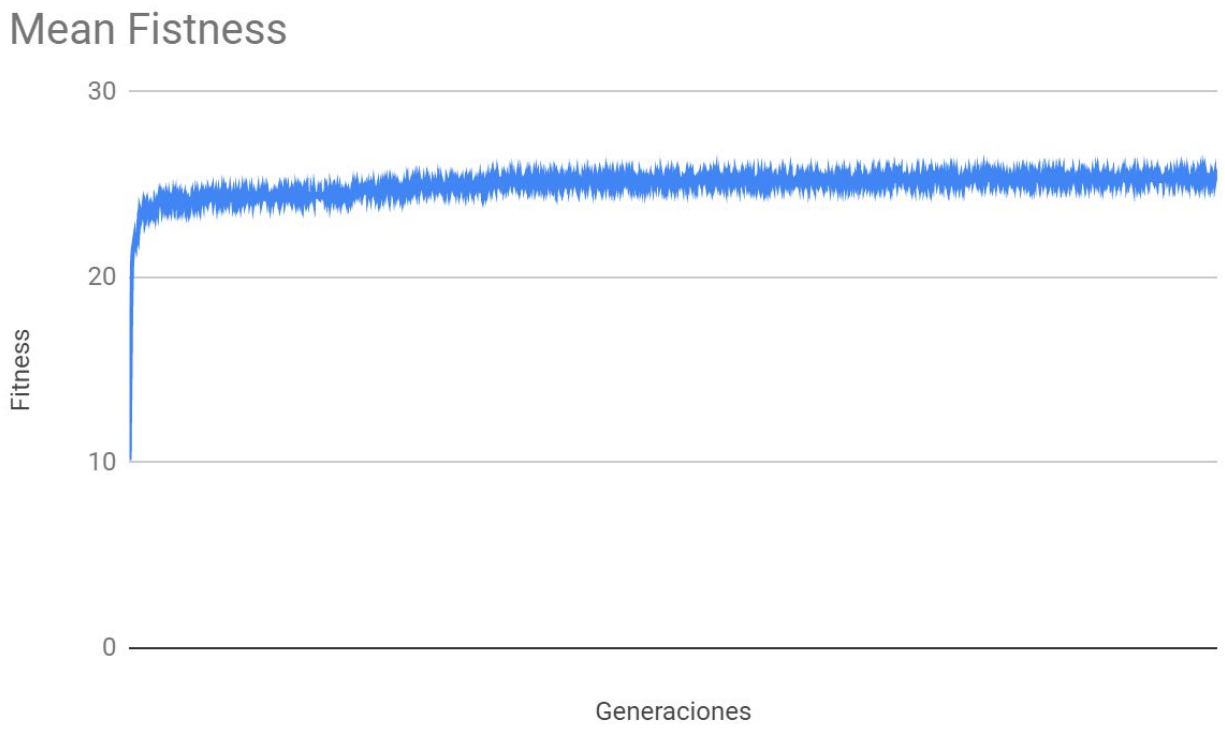
Configuración 3: Torneo → igual a la configuración base pero con *"tournament"* para ambos métodos de selección

Diversity



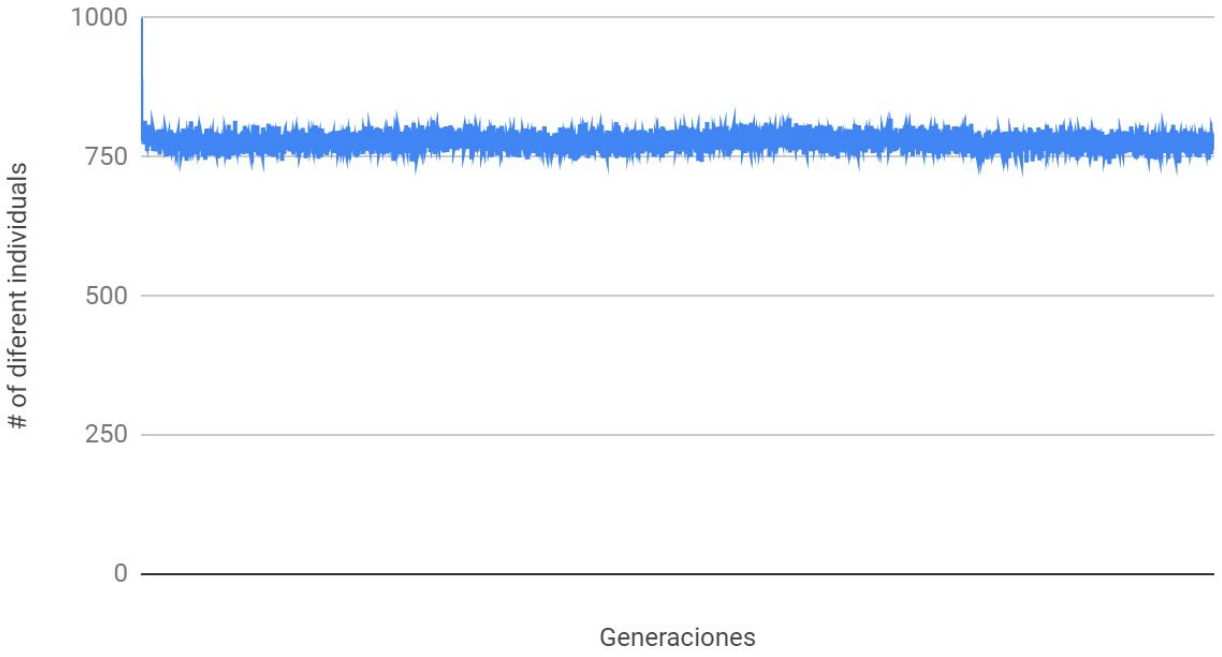
Best Fistness



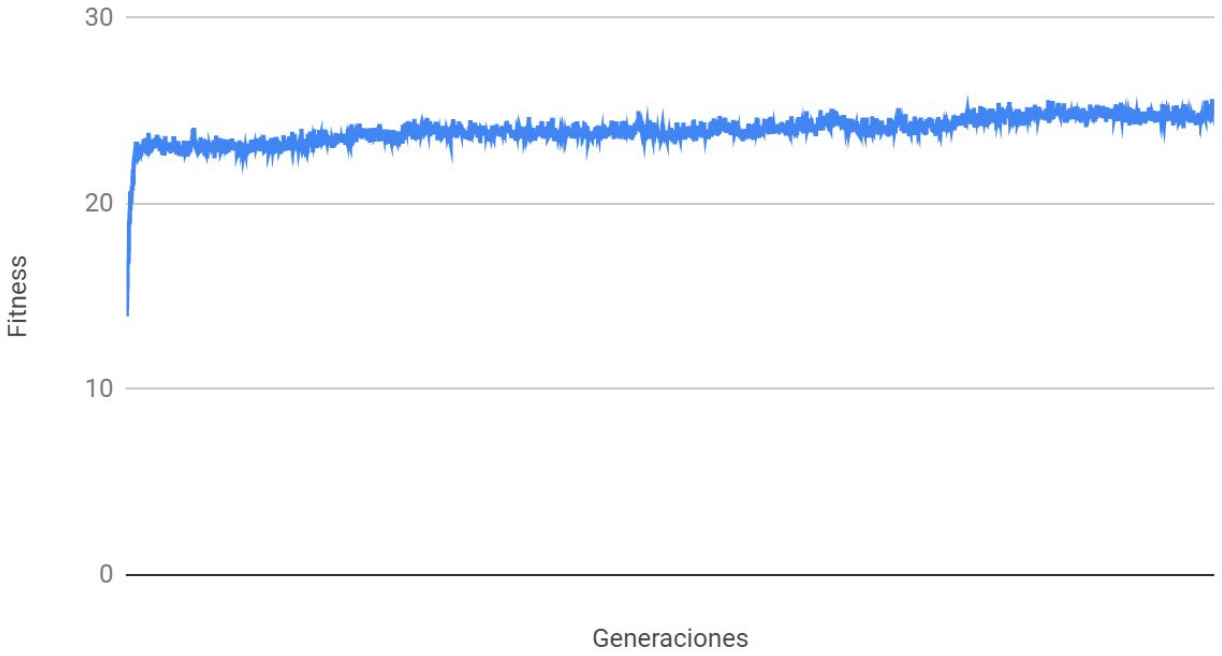


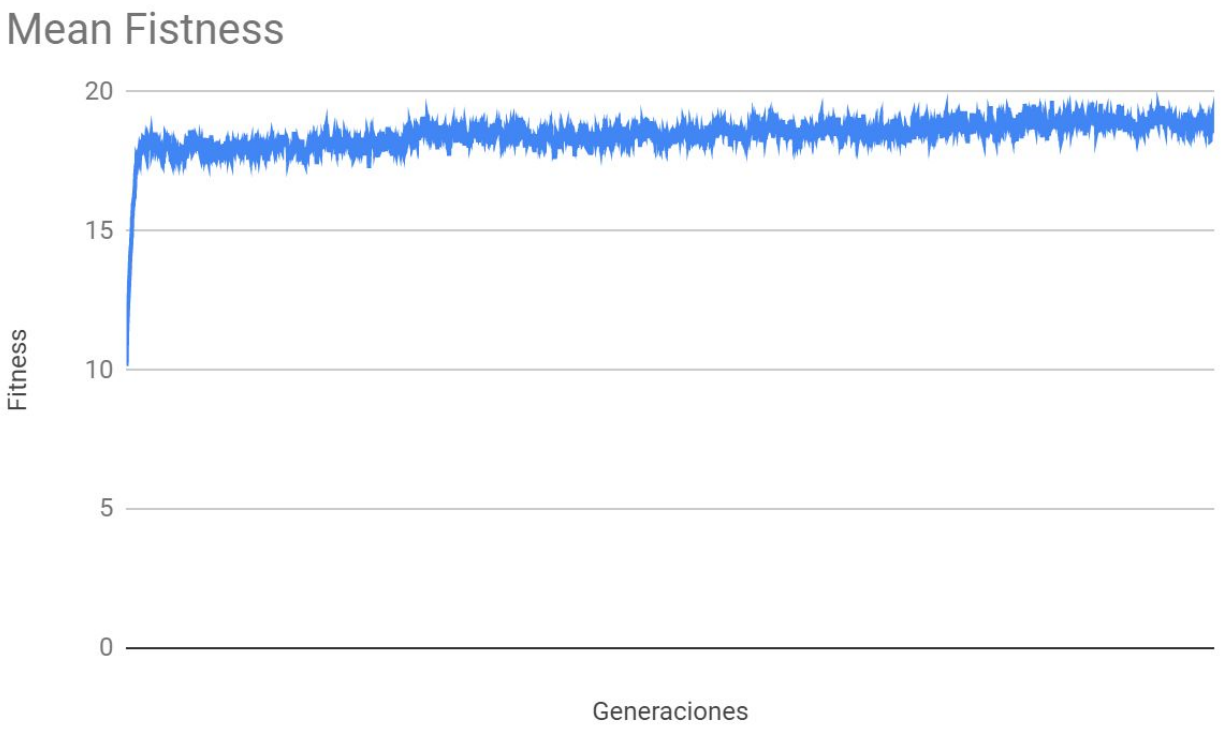
Configuración 4: Torneo probabilístico → igual a la configuración base pero con *"tournamentProb"* para ambos métodos de selección

Diversity



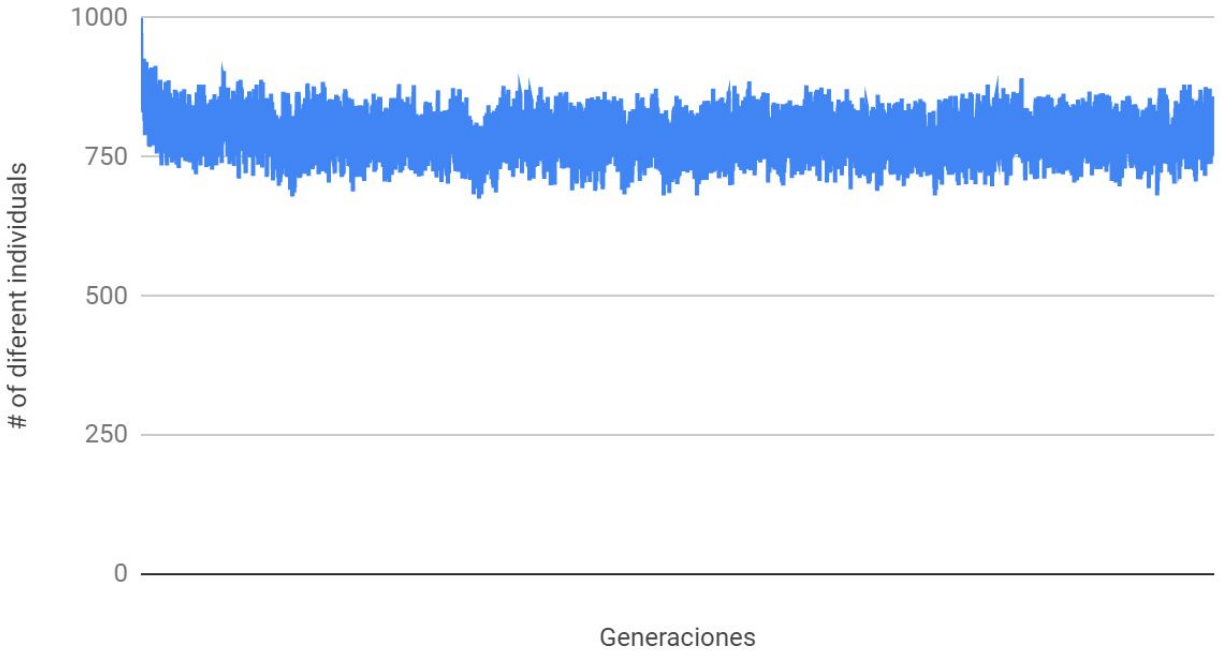
Best Fistness



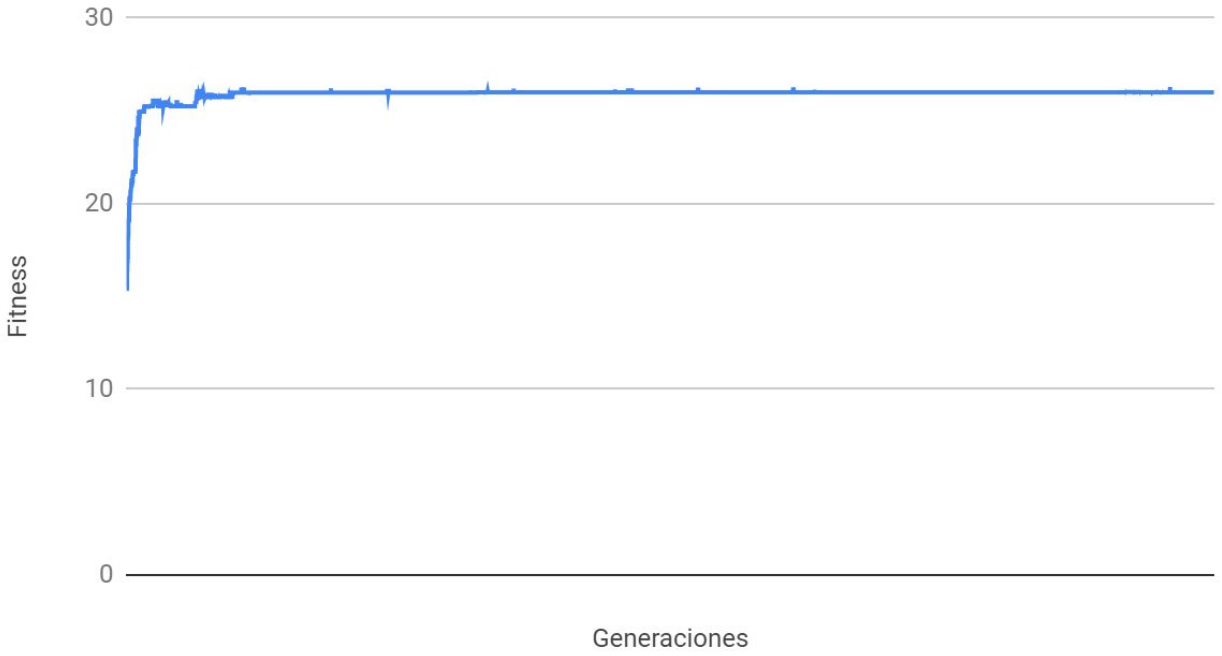


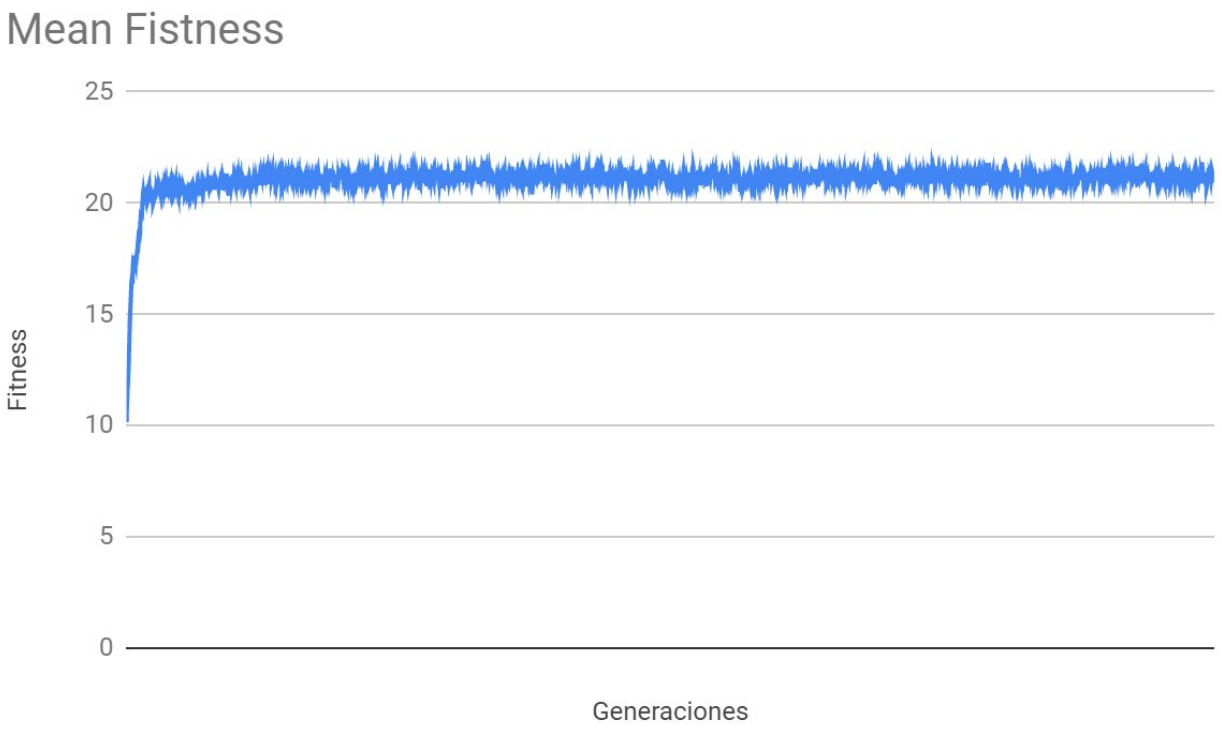
Configuración 5: Universal → igual a la configuración base pero con *"universal"* para ambos métodos de selección

Diversity

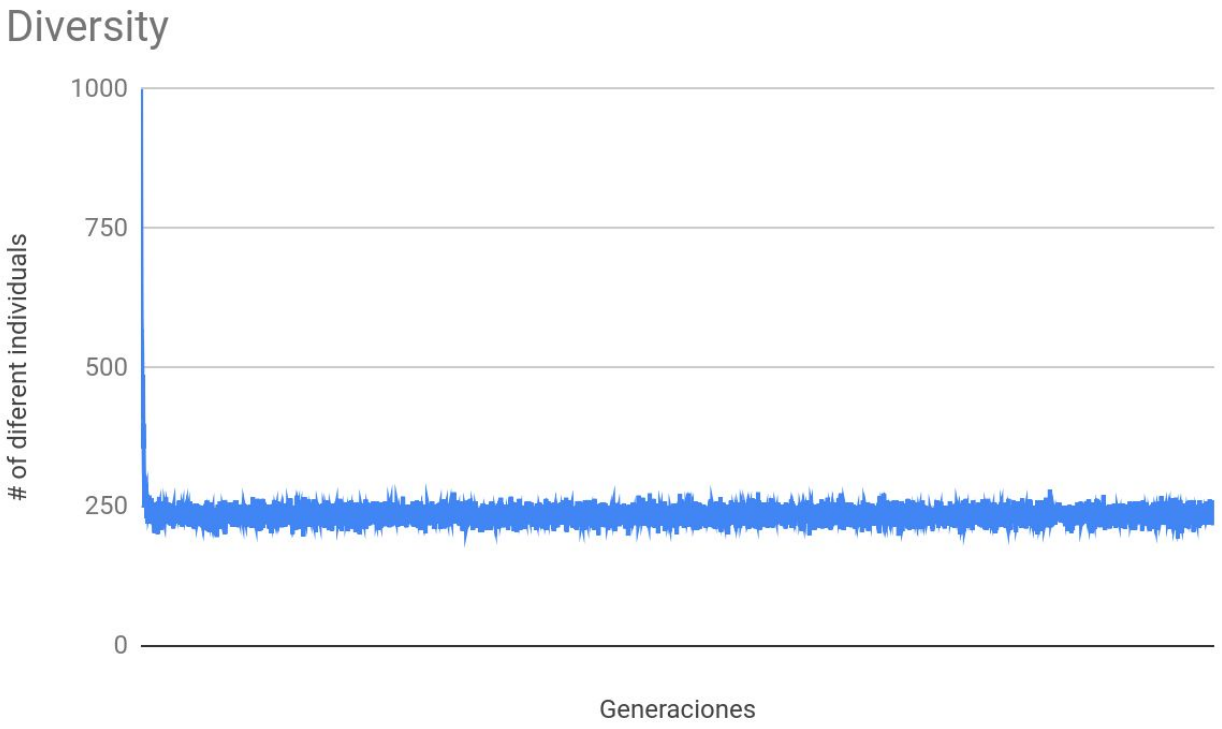


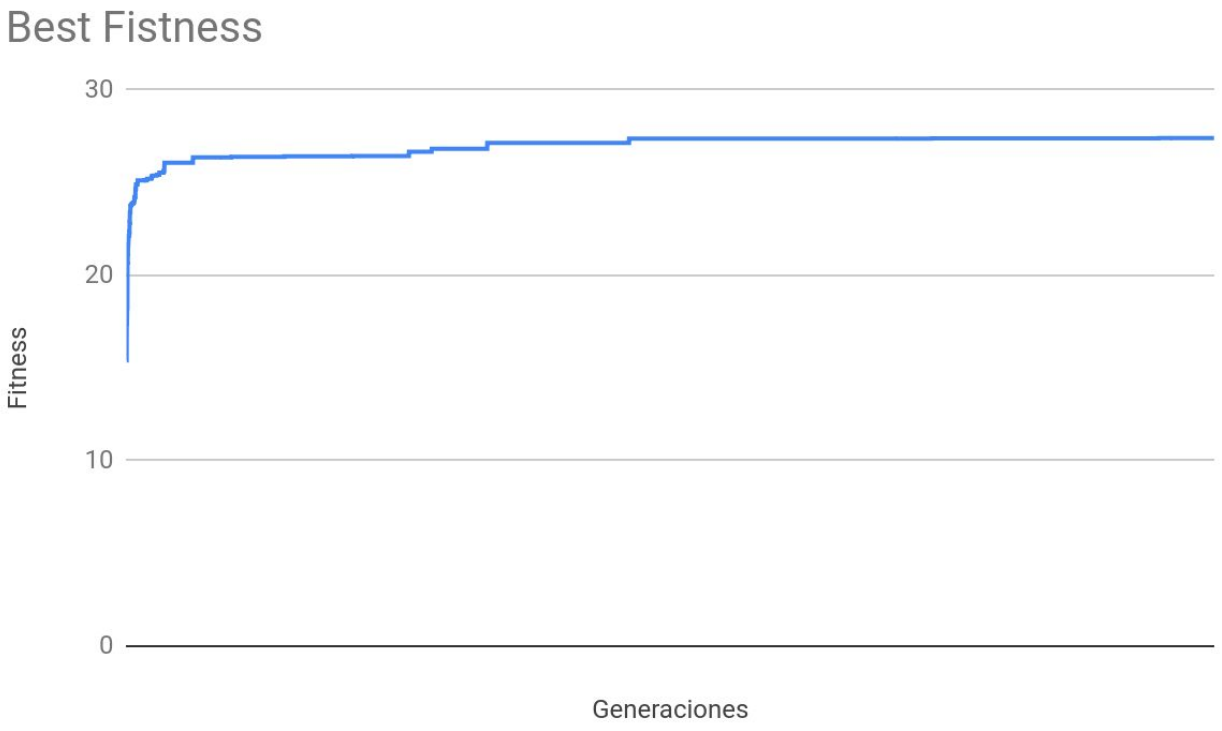
Best Fistness

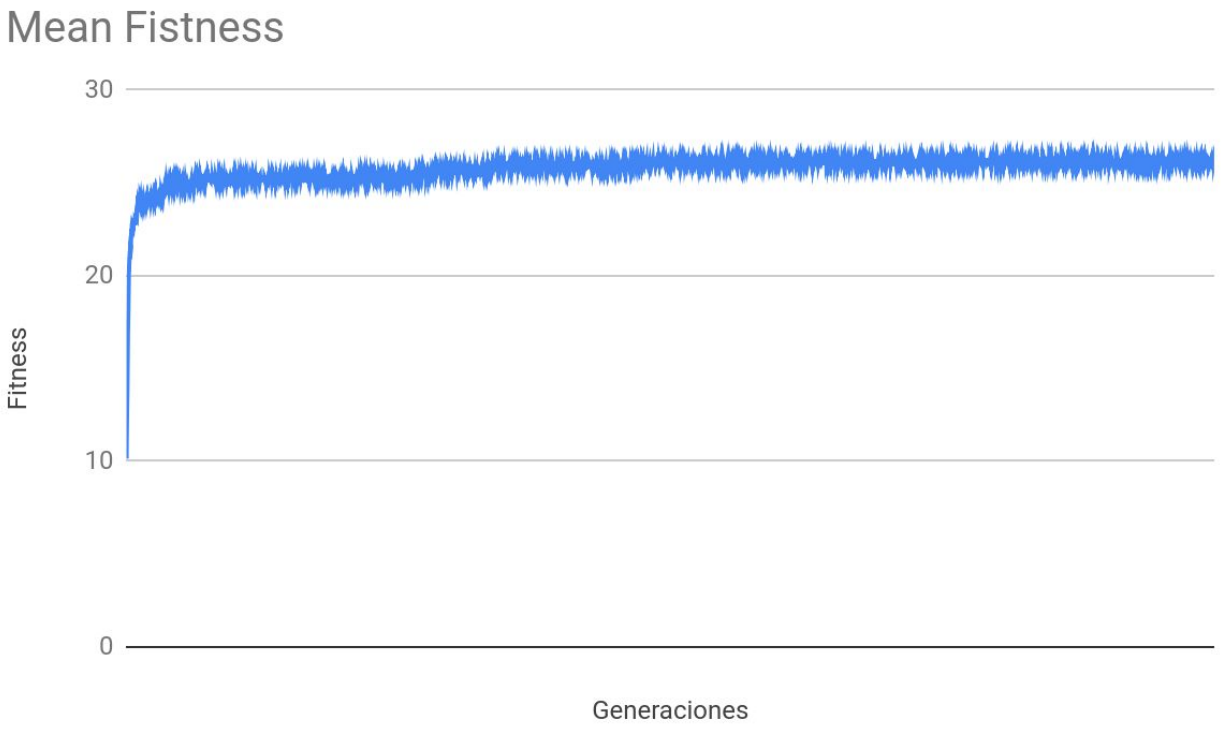




Configuración 6: Elite → igual a la configuración base pero con *"elite"* para ambos métodos de selección

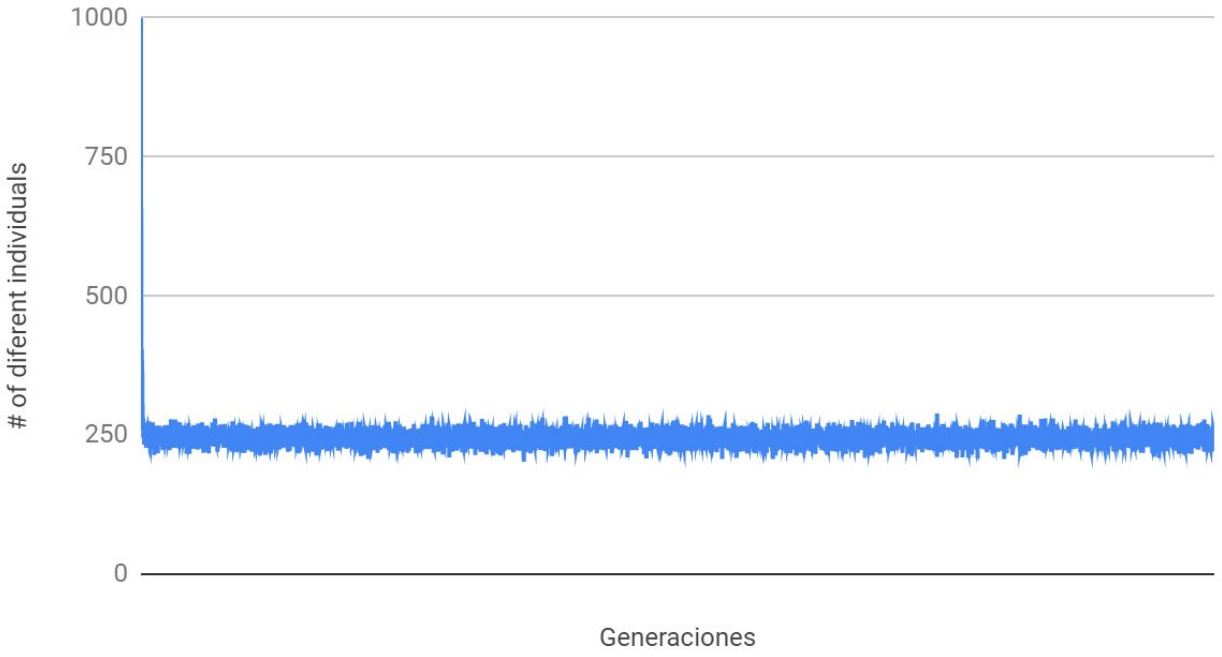




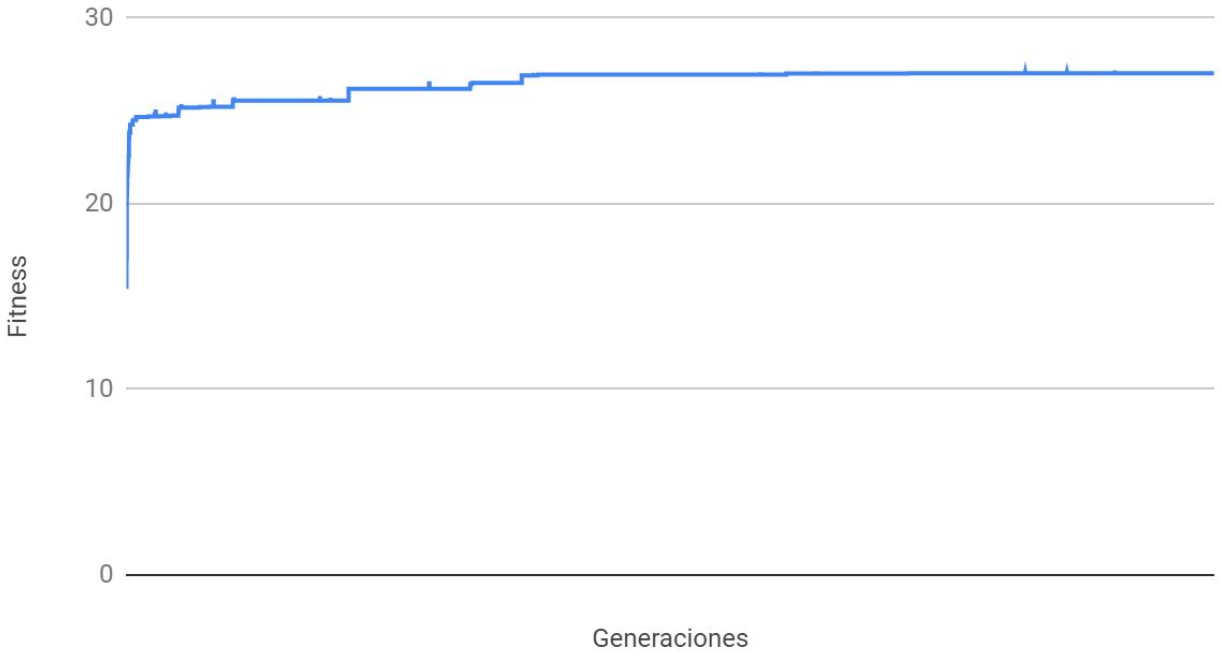


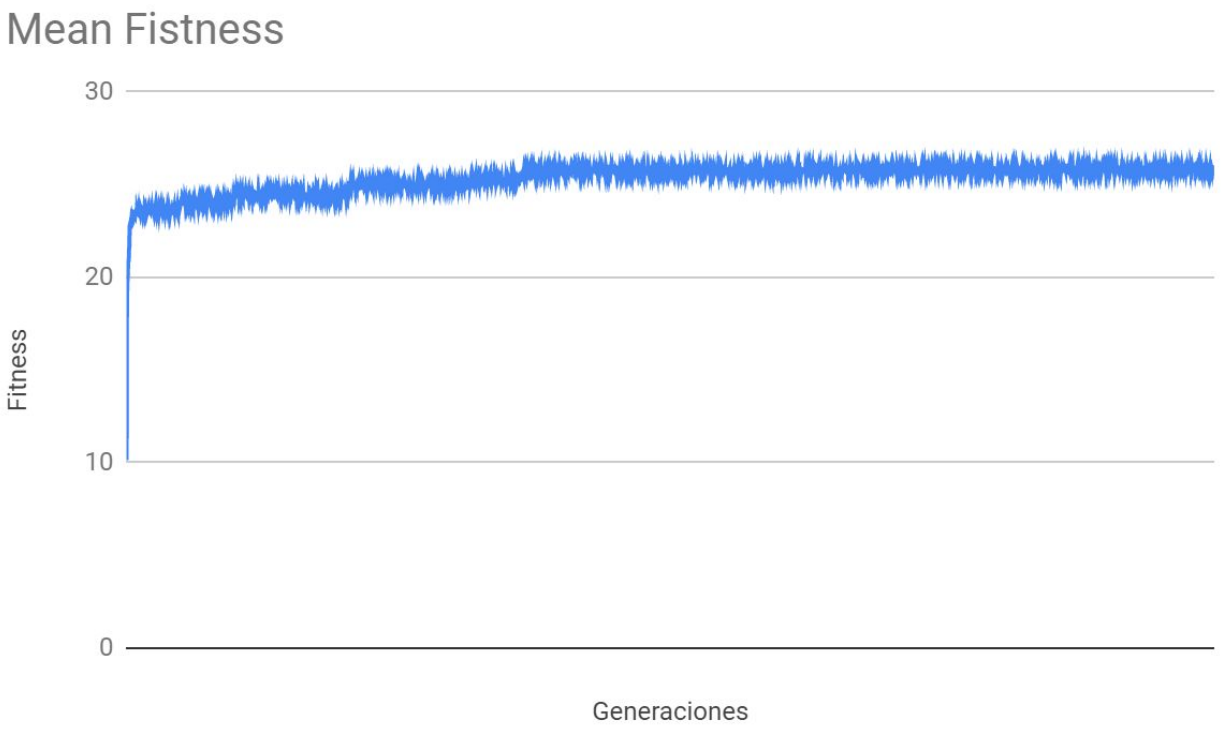
Configuración 7: Boltzmann → igual a la configuración base pero con *"boltzmann"* para ambos métodos de selección

Diversity



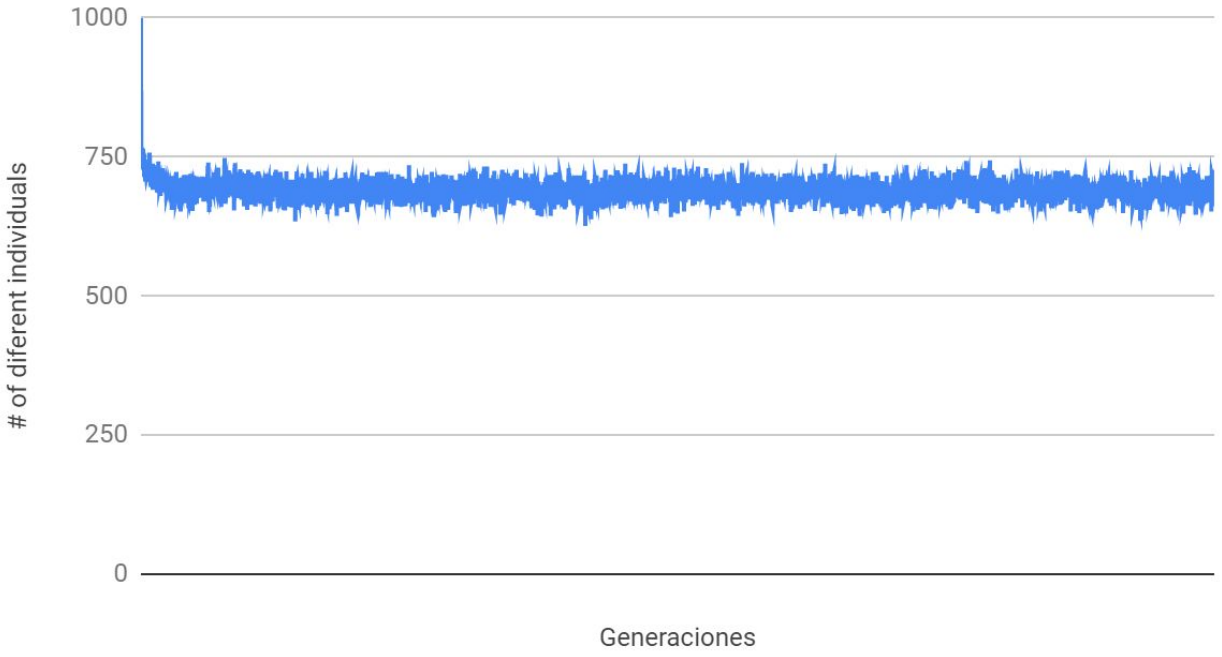
Best Fistness



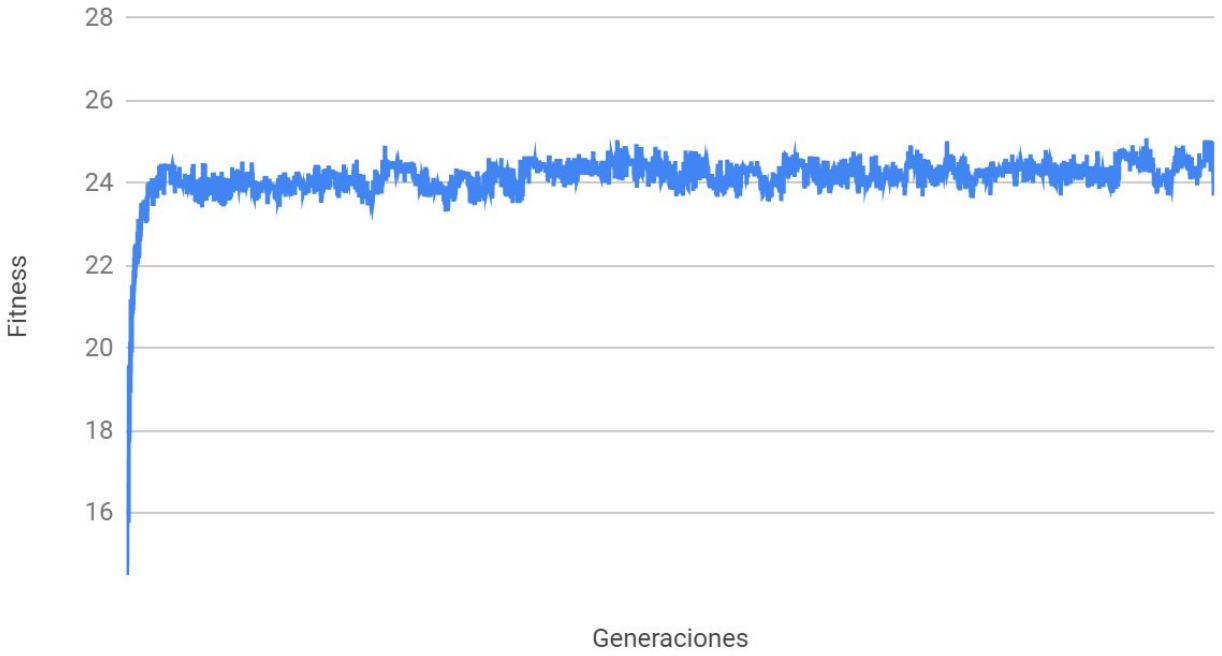


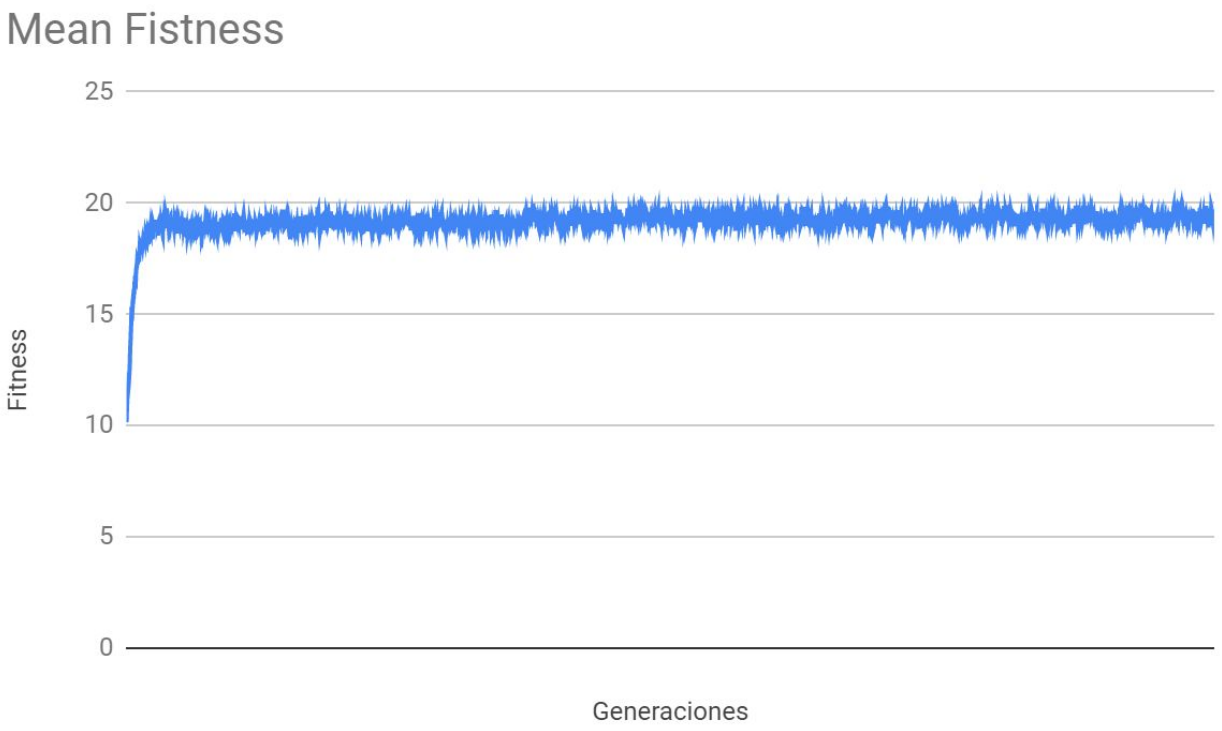
Configuración 8: Cruce de un punto → igual a la configuración base pero con *"onePoint"* para el parámetro *"crossover"*

Diversity



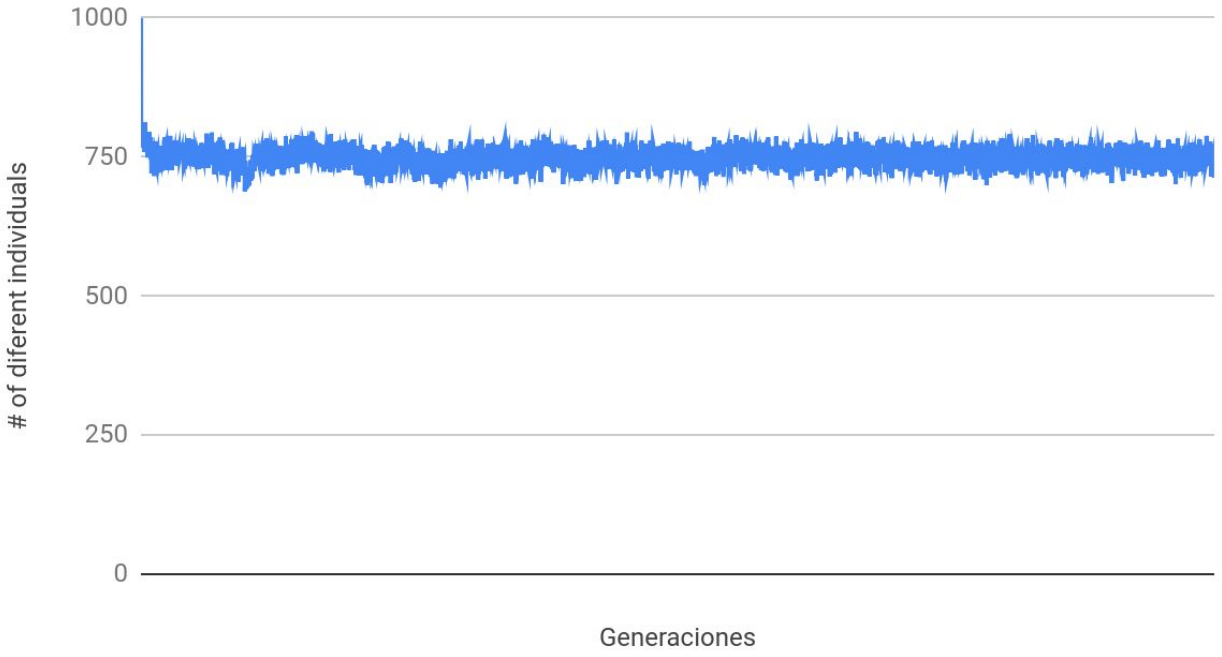
Best Fistness



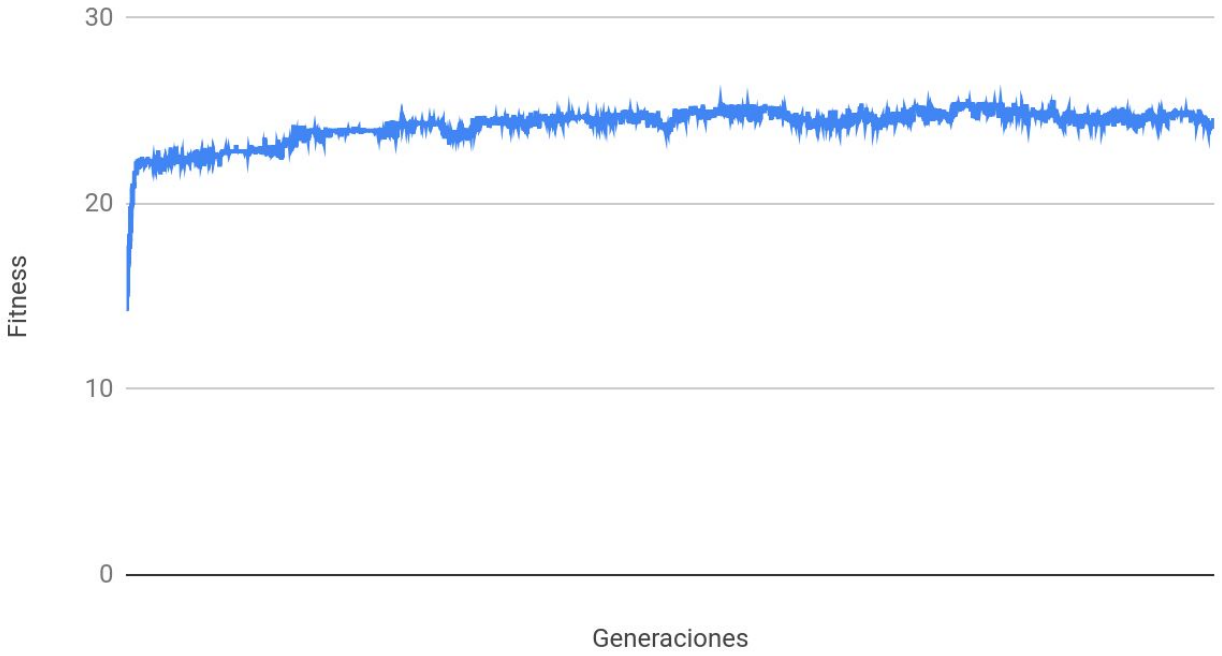


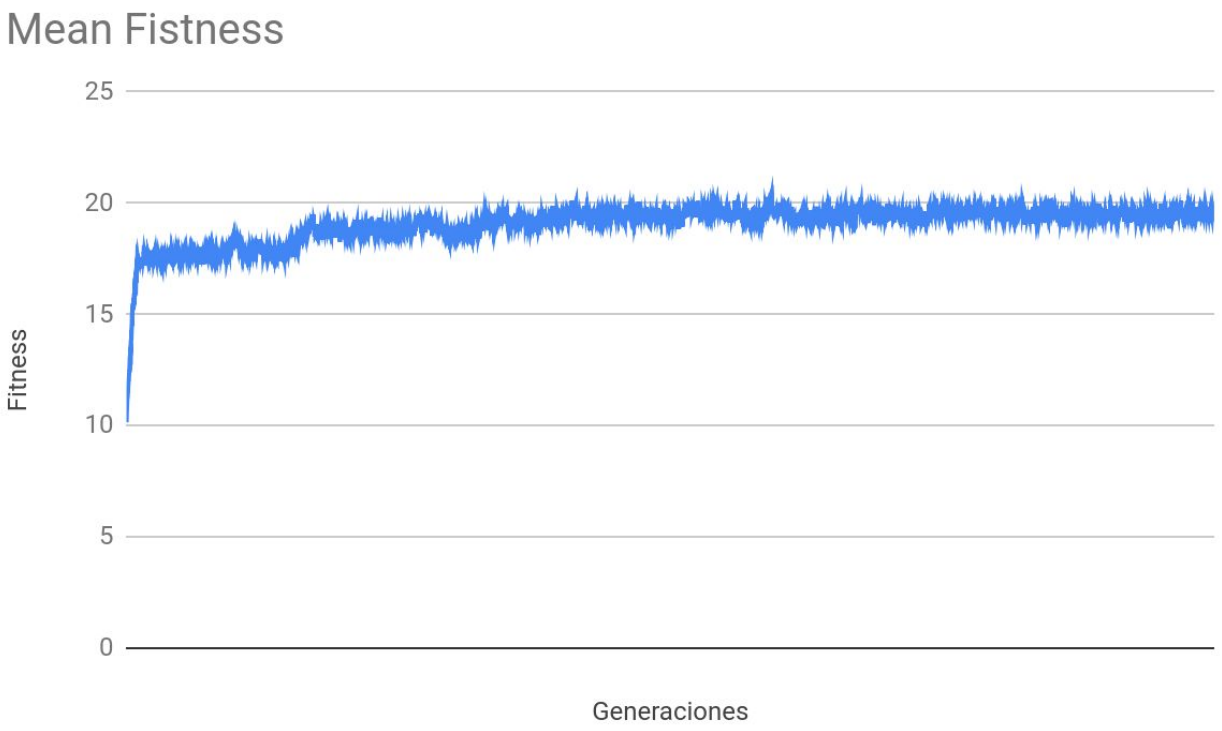
Configuración 9: Annular → igual a la configuración base pero con *"annular"* para el parámetro *"crossover"*

Diversity



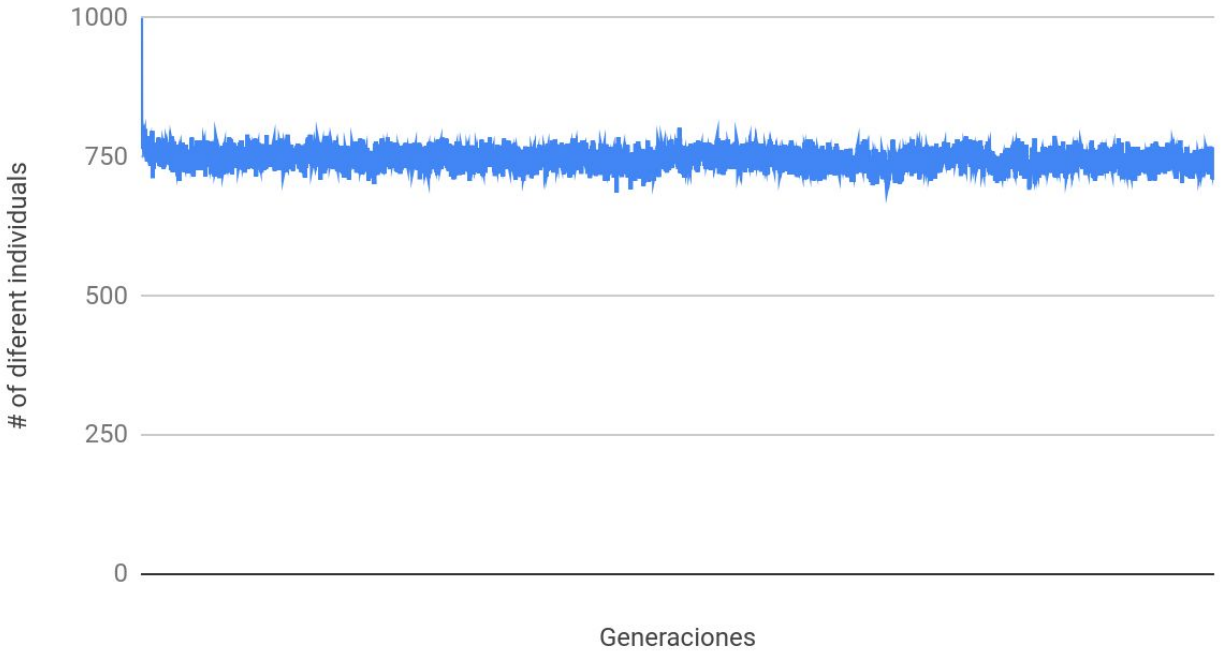
Best Fistness



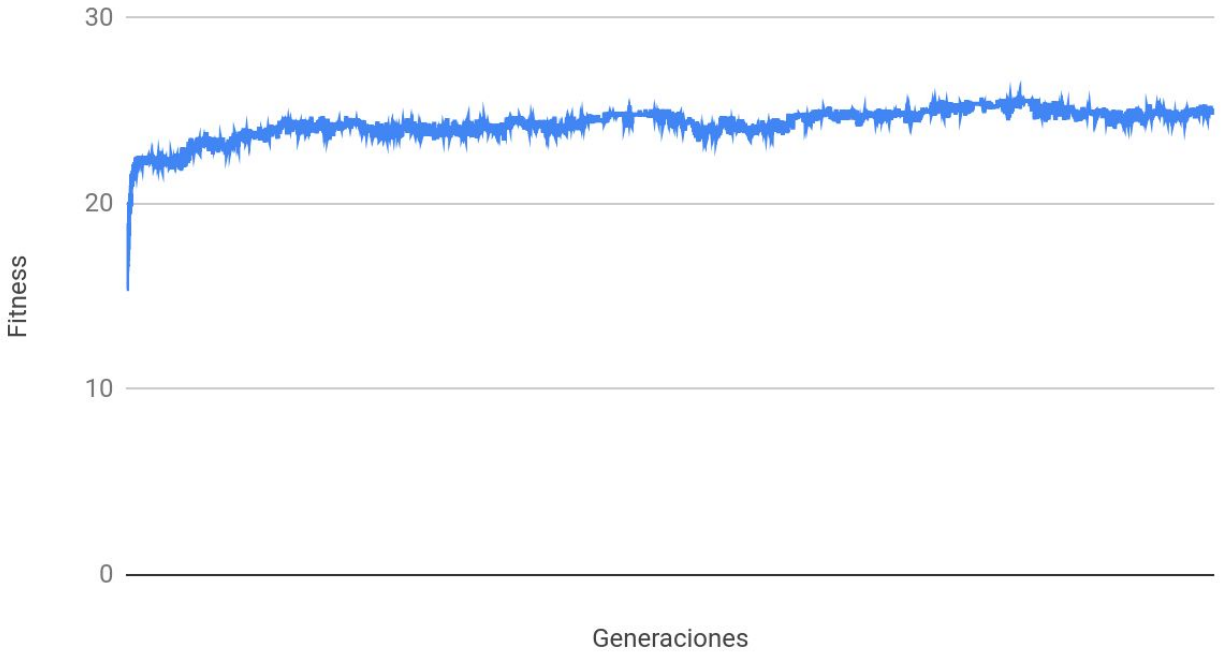


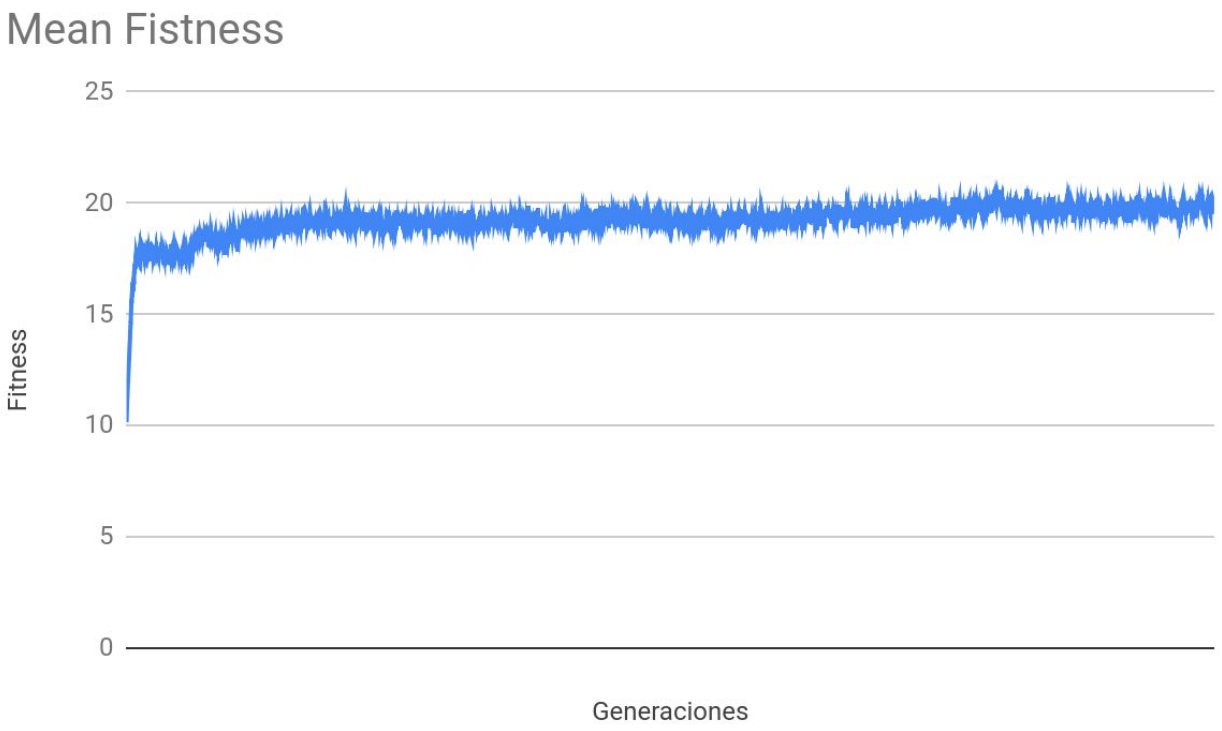
Configuración 10: Uniforme parametrizado → igual a la configuración base pero con *"twoPoint"* para el parámetro *"crossover"*

Diversity



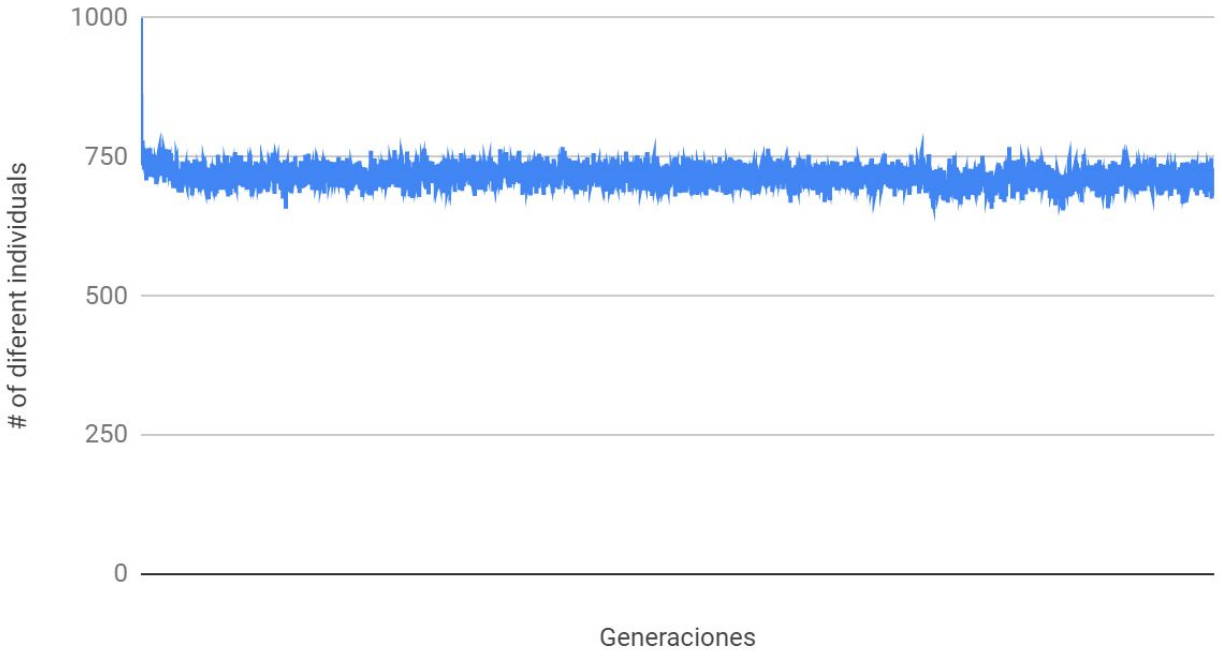
Best Fistness



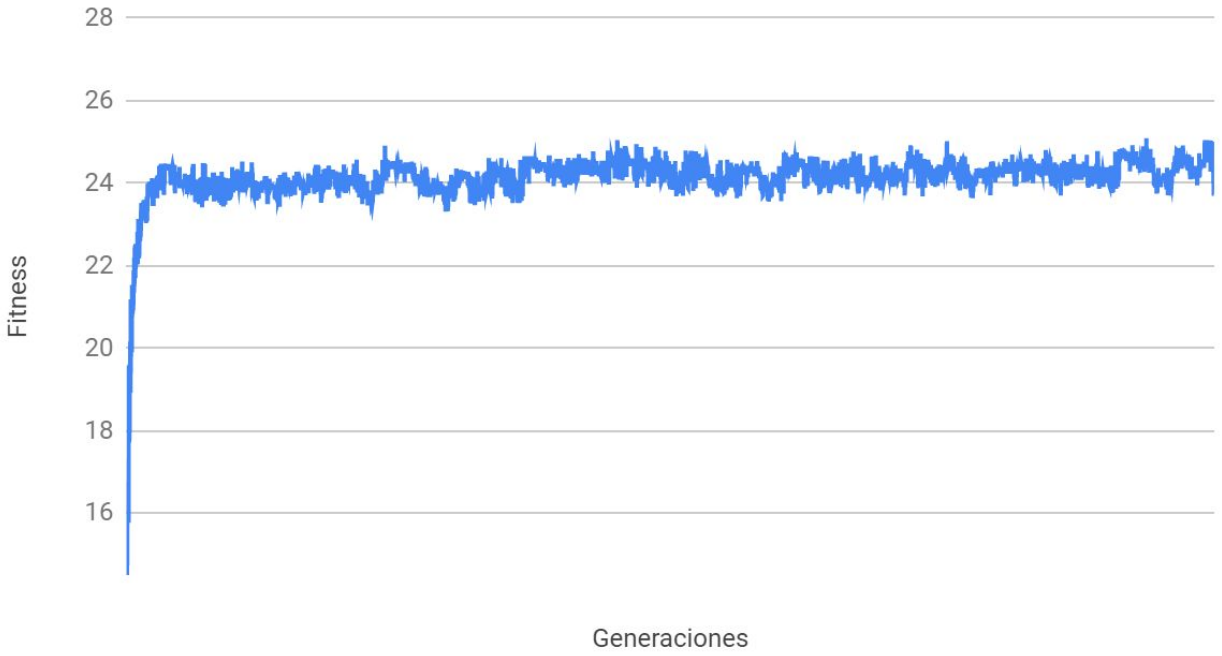


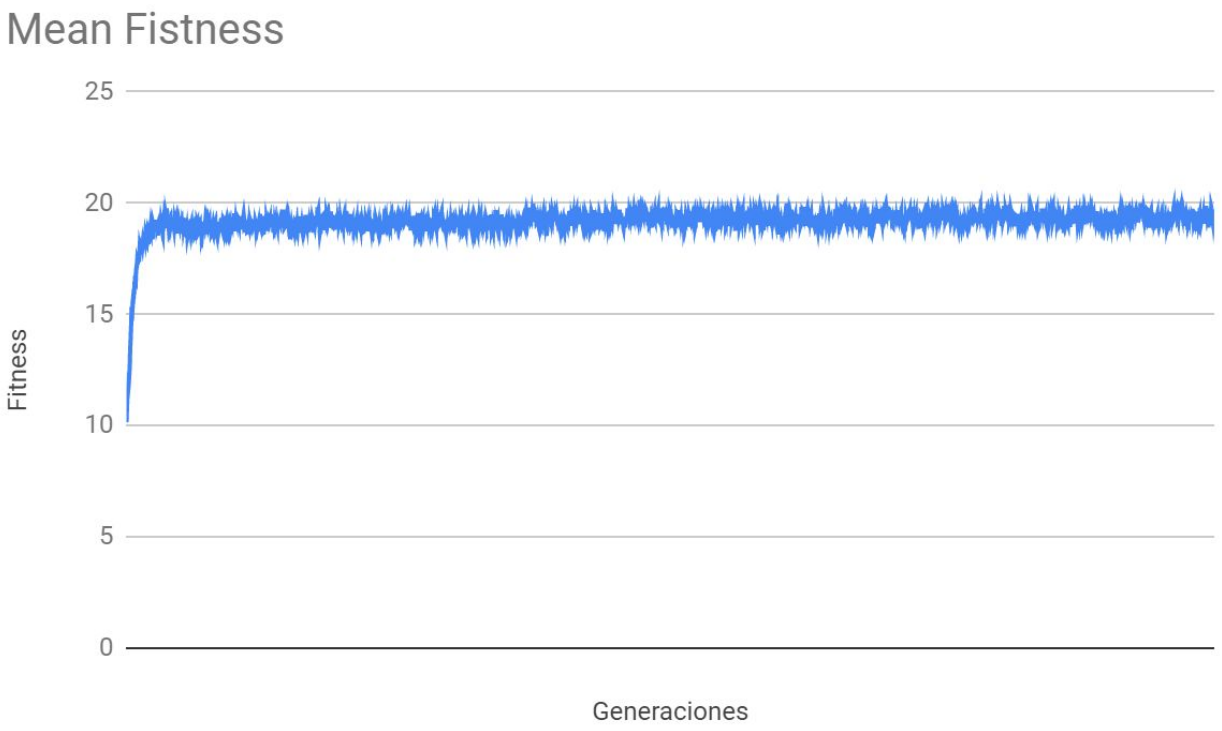
Configuración 11: Uniforme multigen → igual a la configuración base

Diversity



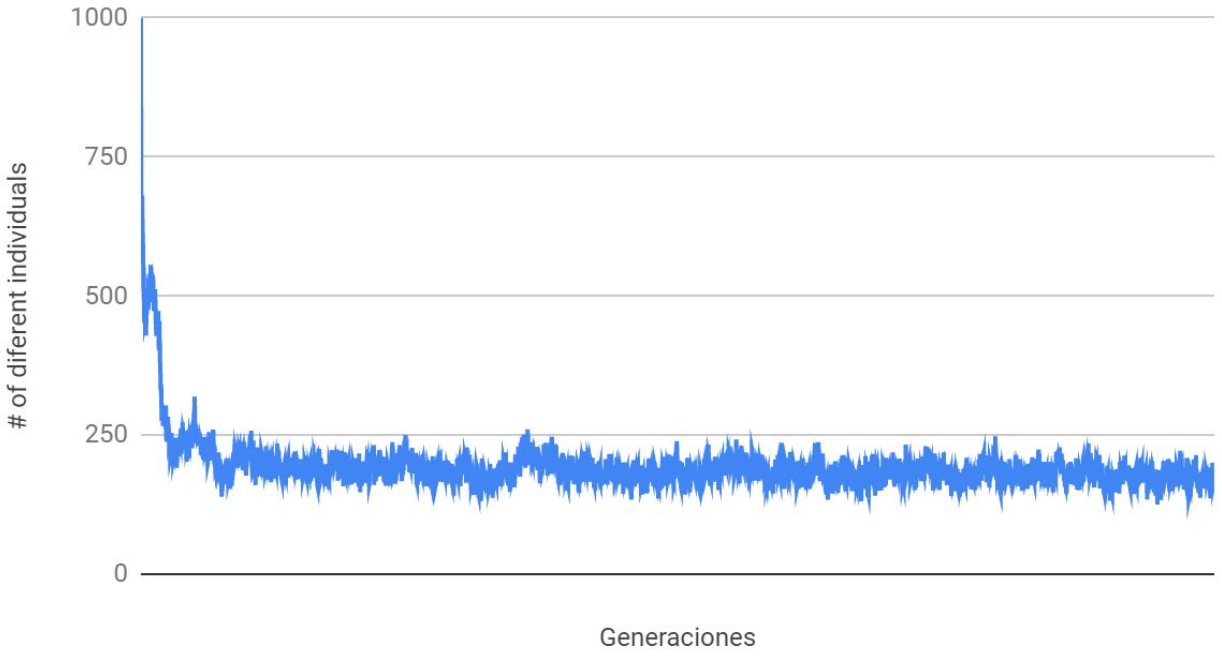
Best Fistness



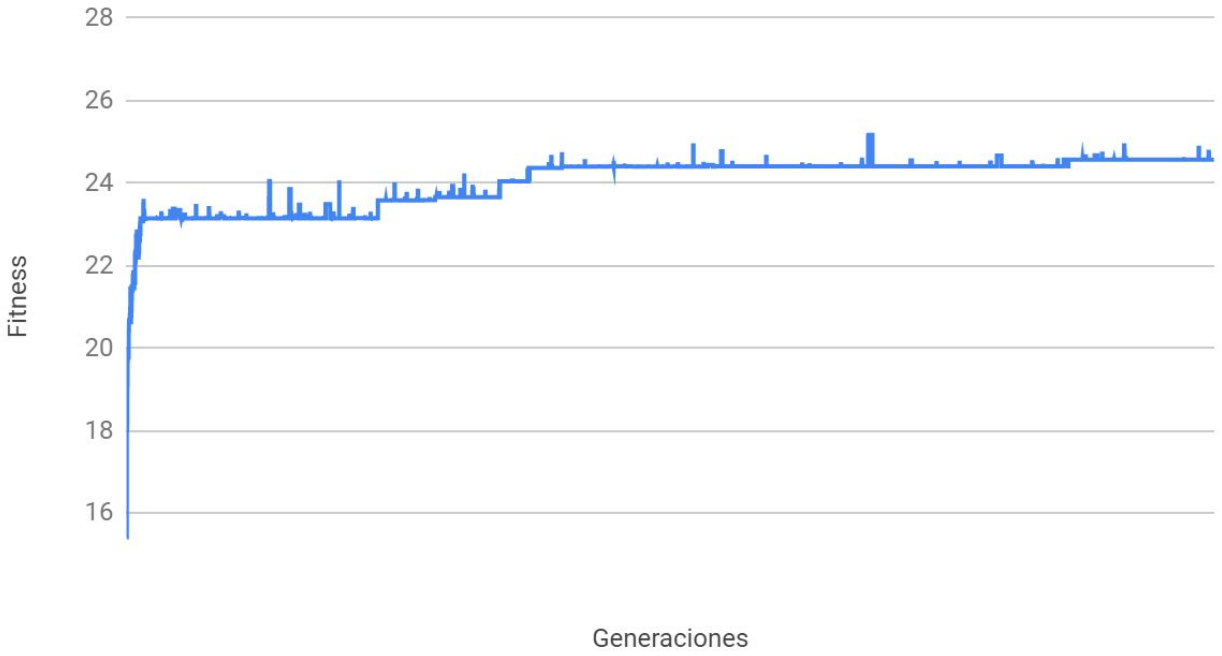


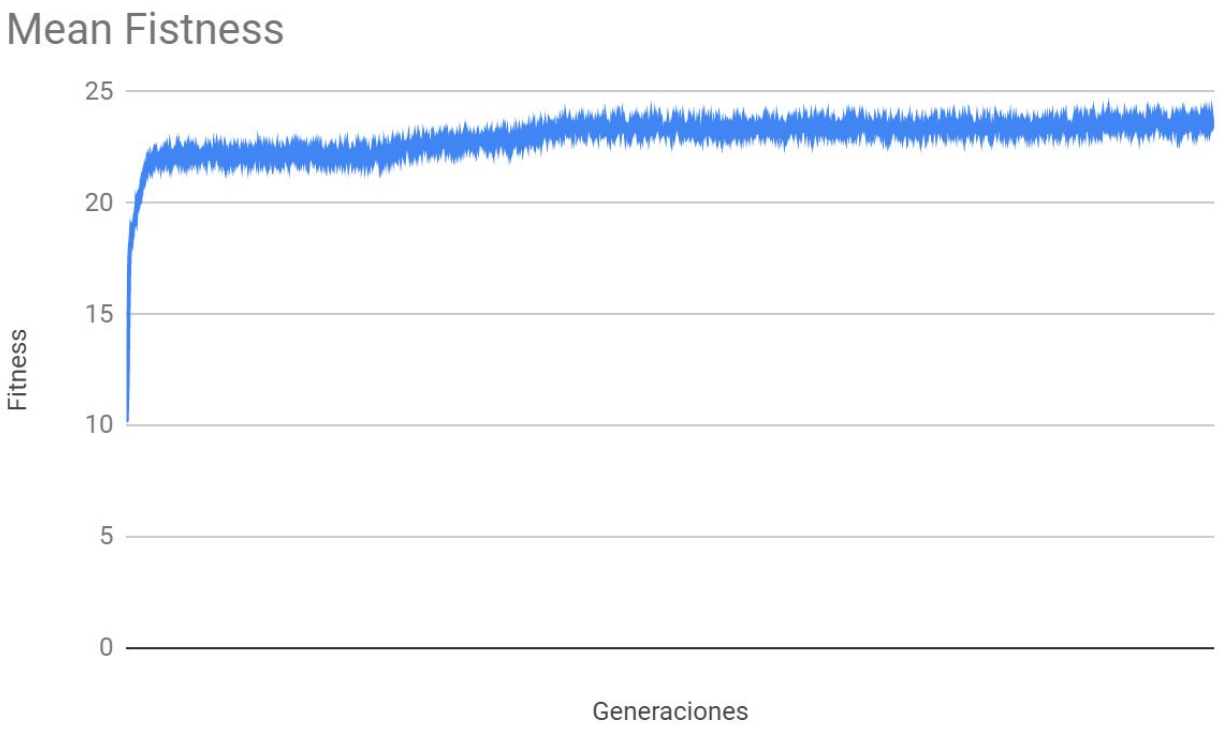
Configuración 12: Uniforme ungen → igual a la configuración base pero con *"false"* para el parámetro *"isMultiGen"*

Diversity



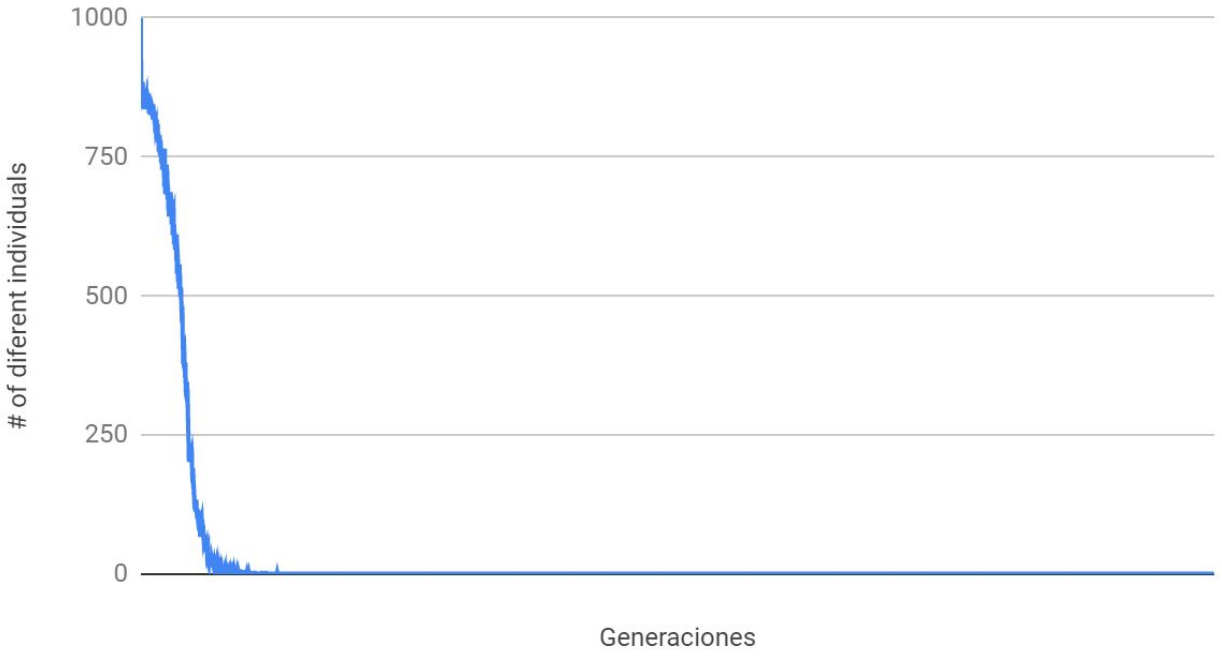
Best Fistness



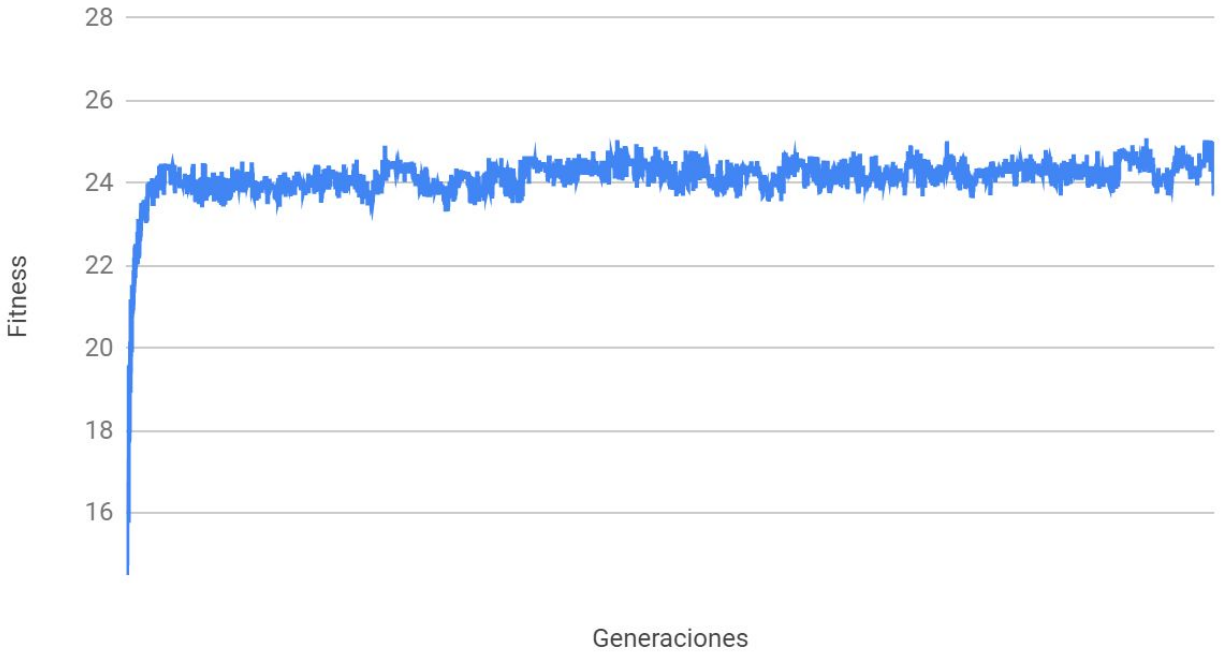


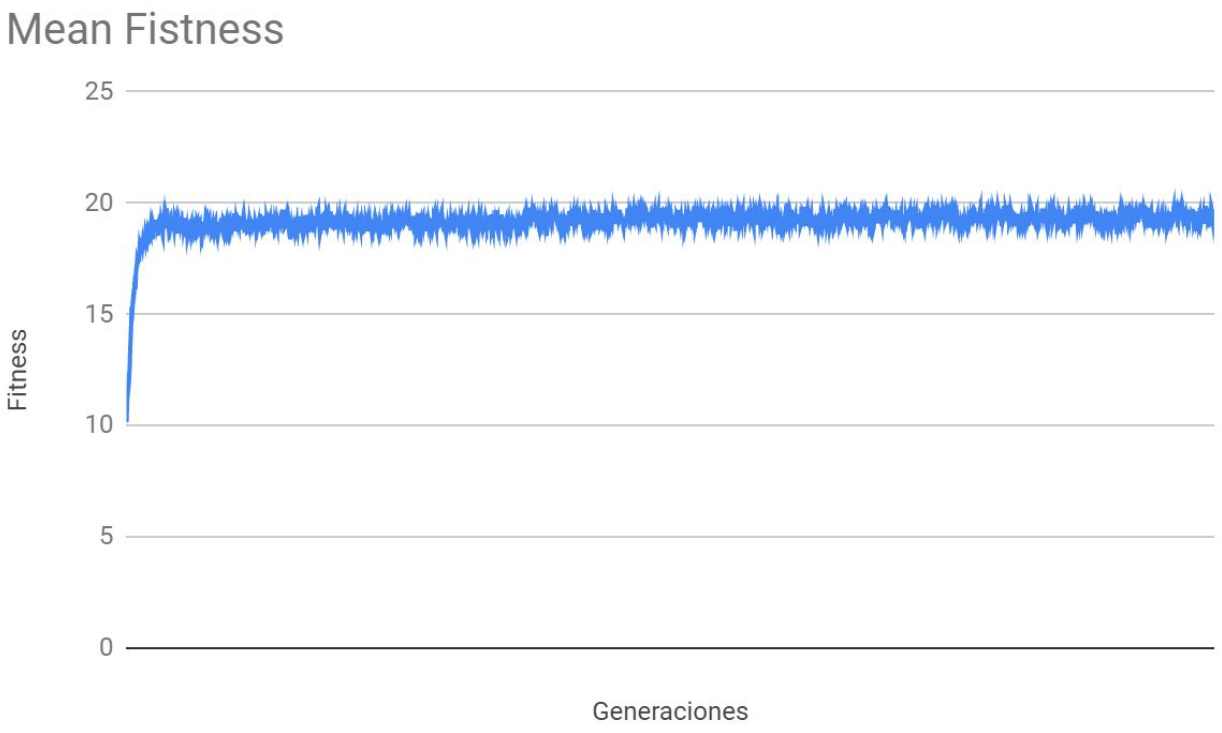
Configuración 13: No uniforme multigen → igual a la configuración base pero con *"false"* para el parámetro *"isUniform"* y con 0.5 en *"initProb"*

Diversity



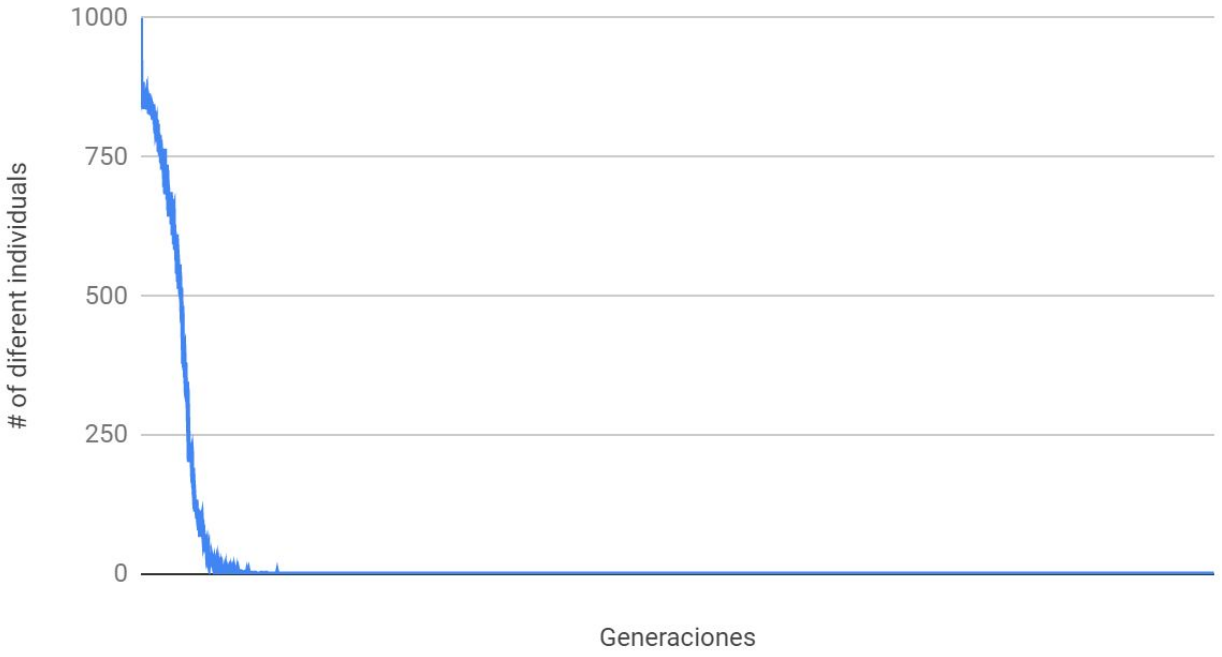
Best Fistness



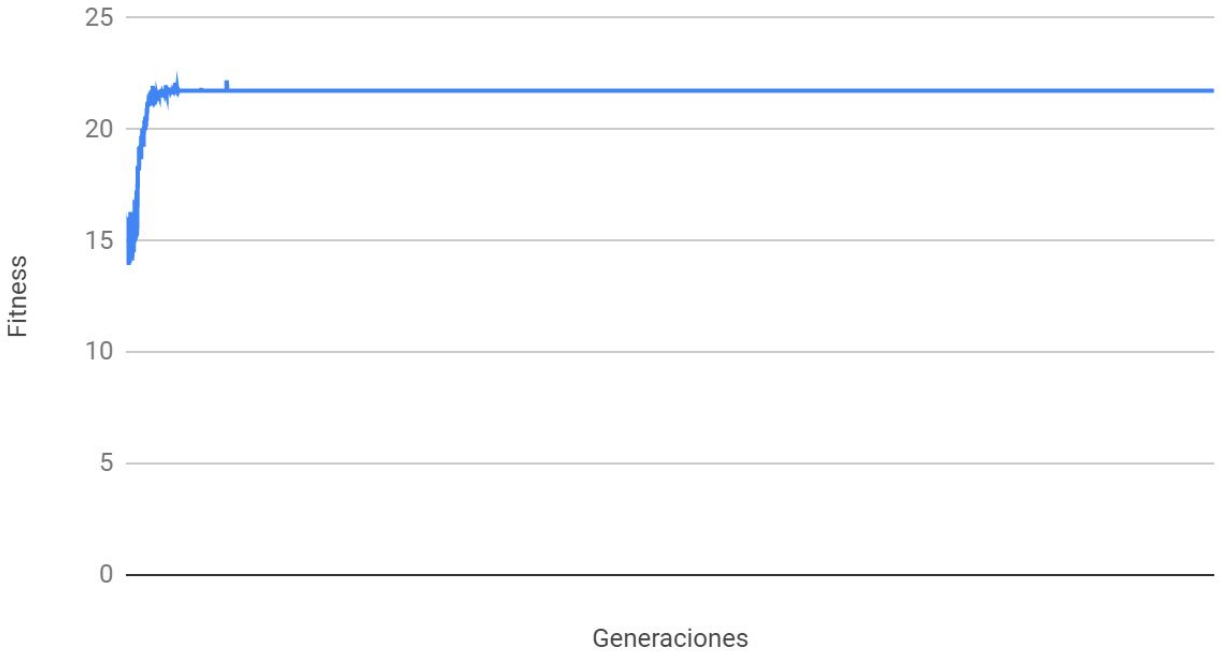


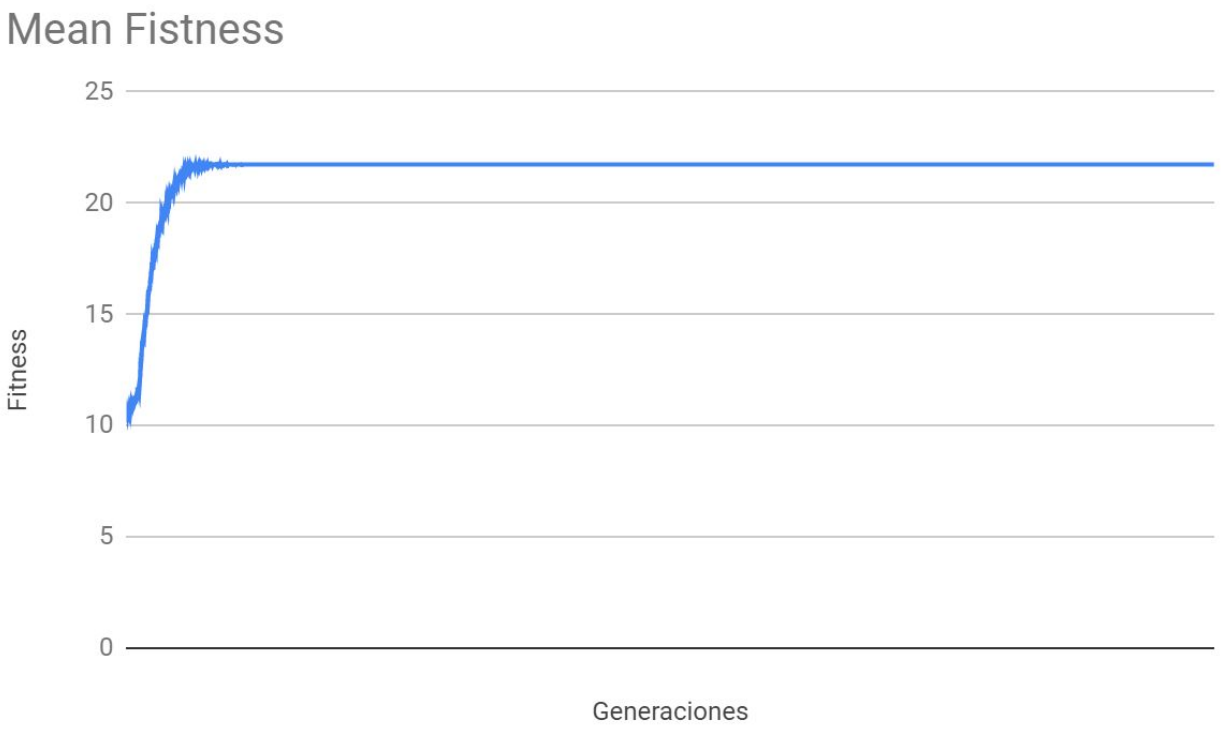
Configuración 14: No uniforme ungen → igual a la configuración base pero con *"false"* para el parámetro *"isMultiGen"*, *"false"* para *"isUniform"* y con 0.5 en *"initProb"*

Diversity



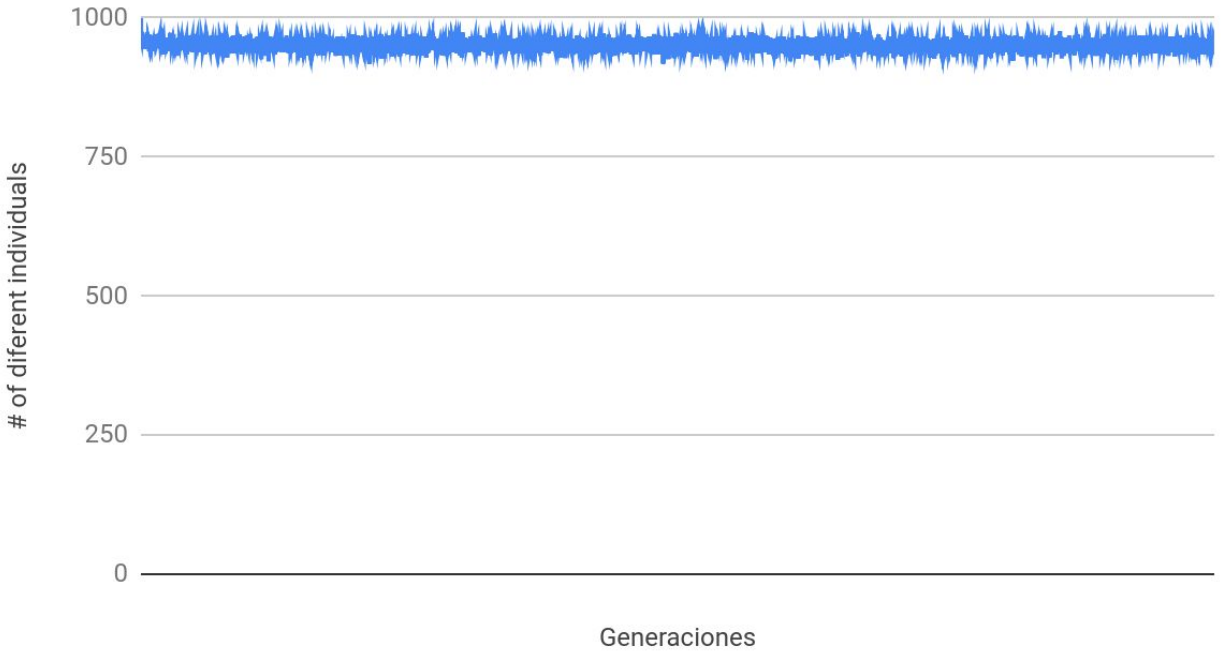
Best Fistness



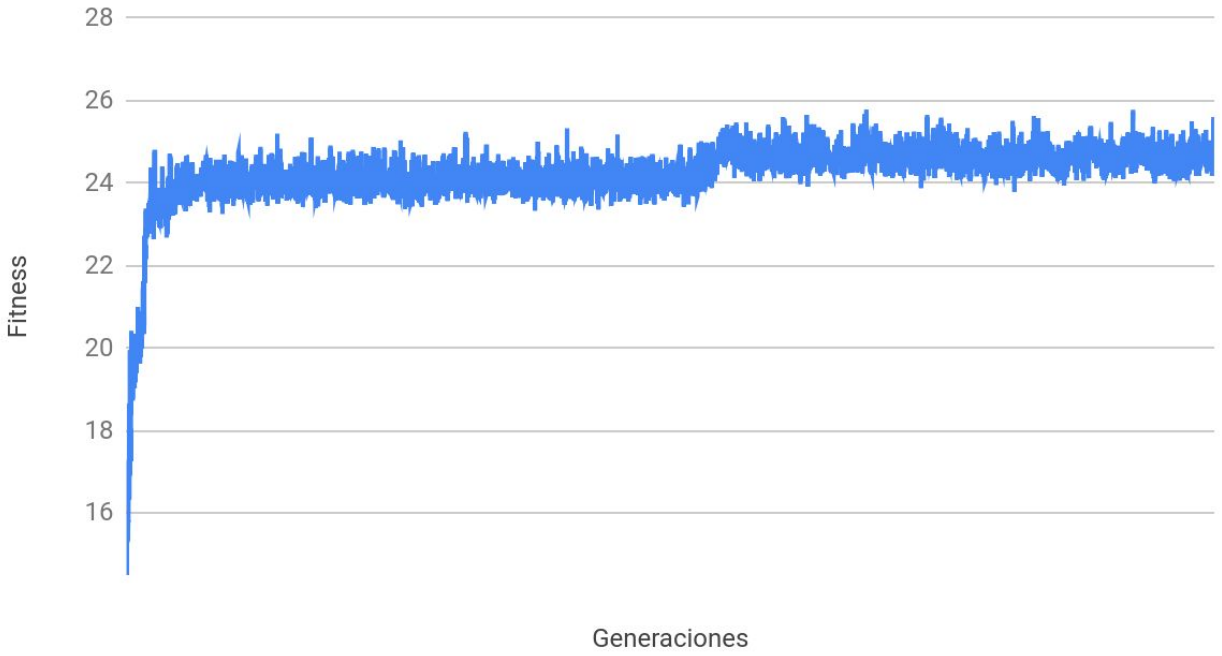


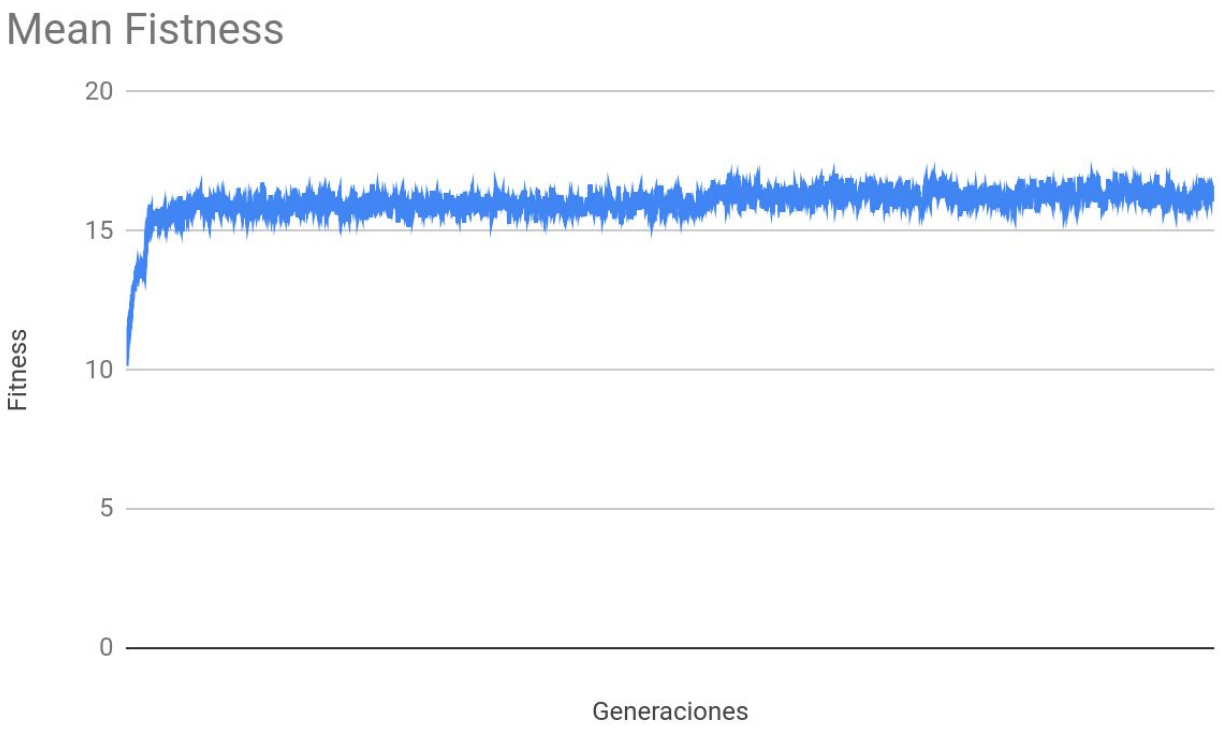
Configuración 15: Reemplazo 1 → igual a la configuración base pero con *"first"* para el parámetro *"type"*

Diversity



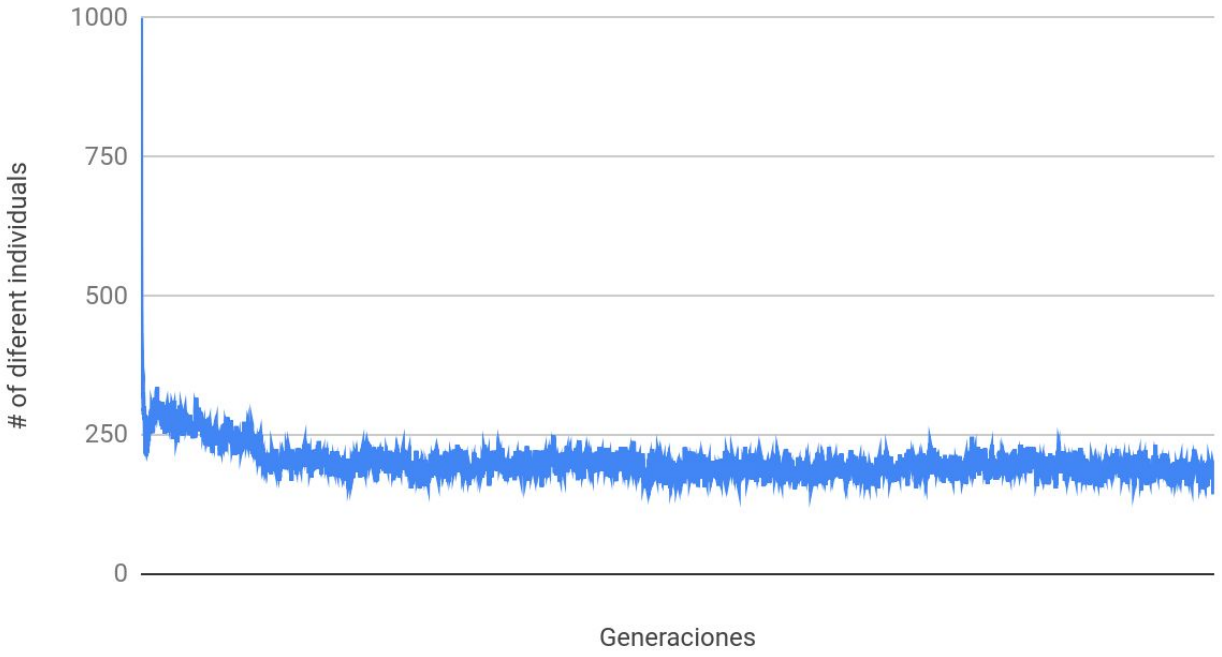
Best Fistness



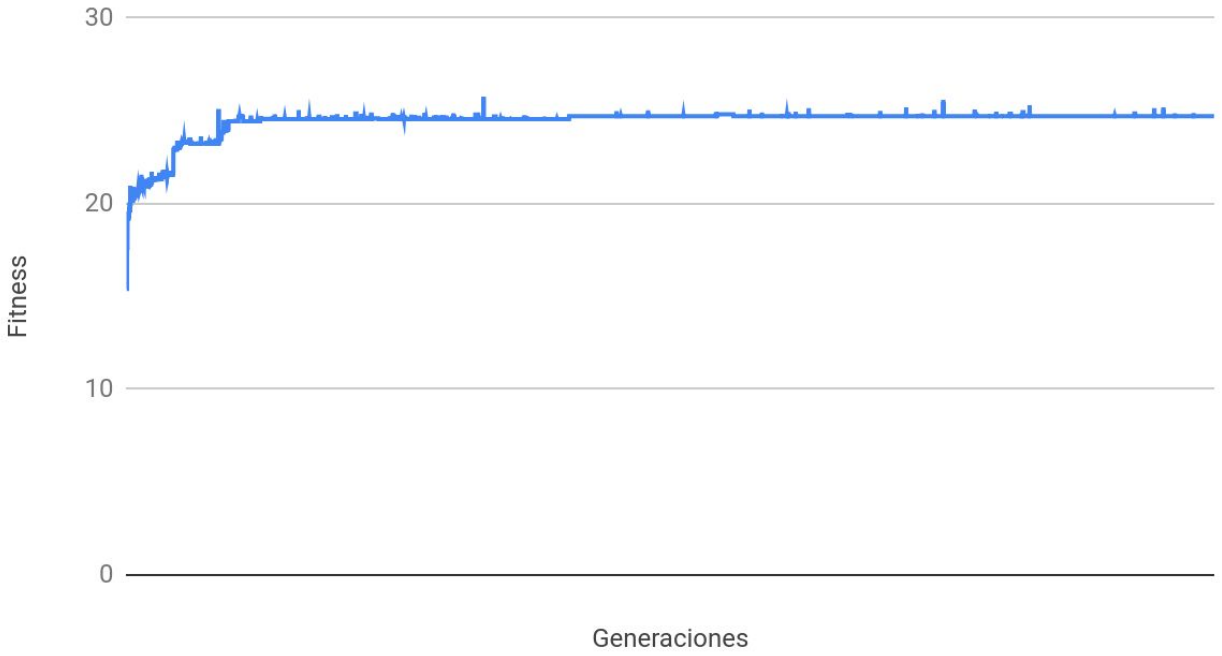


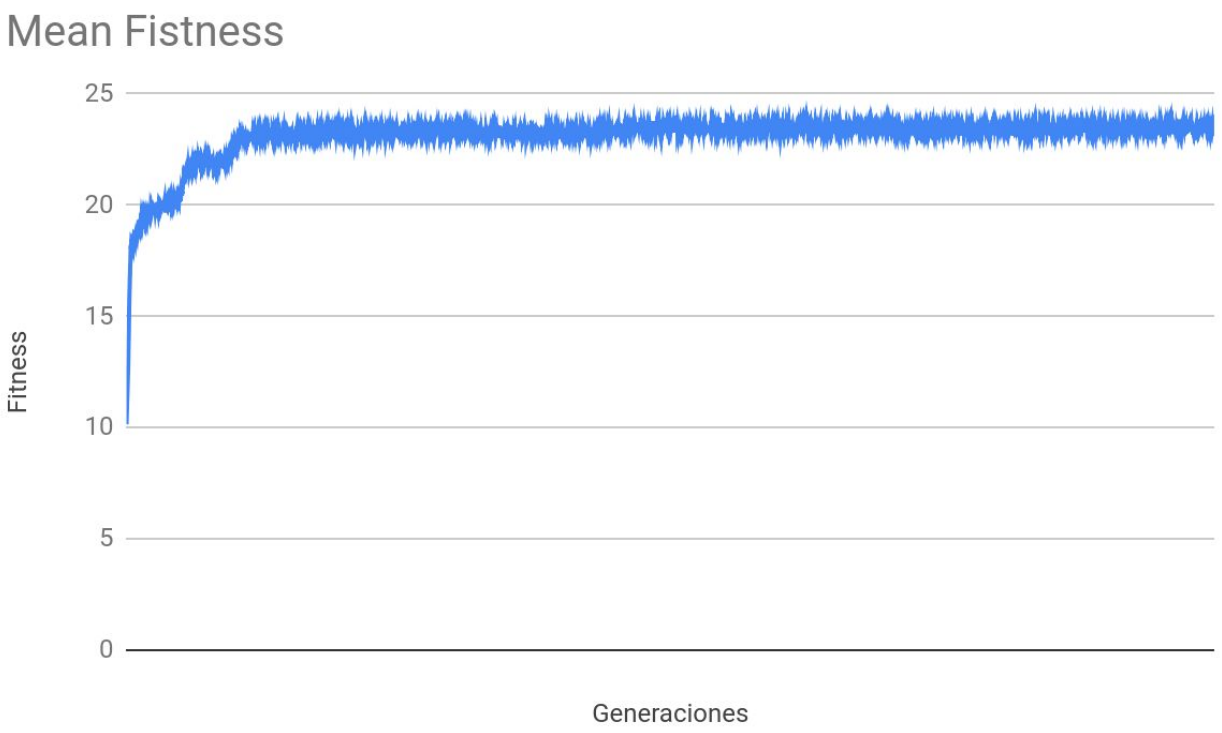
Configuración 16: reemplazo 3 → igual a la configuración base pero con *"third"* para el parámetro *"type"*

Diversity



Best Fistness




Mean Fistness
25
20
15
10
5
0
Fitness
Generaciones