

Trabajo Práctico 3

Algoritmos Genéticos

Autores

Nagelberg, Martin (56698) mnagelberg@itba.edu.ar

Grethe, Juan (57370) jgrethe@itba.edu.ar

Grabina, Martin (57360) mgrabina@itba.edu.ar

72.27 - Sistemas de Inteligencia Artificial

Prof. María Cristina Parpaglione

9 de octubre de 2019

Contenido	
Introducción	2
Problema	2
Implementación	2
Métricas Clave	2
Casos estudiados y resultados	3
Variación de % de selección	4
Variación de métodos de selección	4
Boltzmann	4
Variación métodos de reemplazo	5
Variación métodos de mutación	5
Variación métodos de cruce	6
Variación tamaño inicial de población	6
Conclusiones	7
Anexo - Gráficos	8
Variacion % de selección	8
100% Elite - 0% Ruleta	8
85% Élite - 15% Ruleta	8
50% Elite - 50% Ruleta	8
Variación de métodos de selección	9
50% Elite - 50% Tournaments Deterministic	9
50% Roulette - 50% Ranking	9
50 % Elite - 50% Ranking	9
50 Roulette - 50% Tournaments Deterministic	10
Boltzmann	10
Sin Boltzmann	10
Con Boltzmann	10
Variación métodos de reemplazo	11
Método 3	11
Método 2	11
Método 1	11
Variación métodos de mutación	12
GenUniform	12
MultiGenUniform	12
MultiGenNonUniform	12
GenNonUniform	13
Variación métodos de cruce	13
Cruce de a dos puntos	13
Cruce uniforme	14
Cruce Anular	14
Variación tamaño inicial de población	14
10	14
100	14
1000	15

Introducción

El objetivo de este trabajo práctico fue implementar un algoritmo genético capaz de ser utilizado para obtener la mejor configuración para un jugador de tal forma que optimice su performance.

Problema

Una empresa de videojuegos precisa un sistema para obtener cuales son las mejores variaciones de los jugadores de tal forma que se optimice su performance en el juego.

Hay 5 tipos de ítems:

- Casco
- Chaleco
- Guantes
- Botas
- Arma

Además, se analiza la altura del mismo, ya que características como ataque y defensa dependen de ello también.

Implementación

La implementación del sistema demandó dos semanas y fue hecha en Java 8 para aprovechar el procesamiento paralelo que proveen los streams. Se hizo foco en respetar programación orientada a objetos para facilitar su mantenimiento y agregado de nuevos métodos.

Dada la gran cantidad de métodos y valores parametrizables del sistema, se hizo un archivo de configuración, del cual se obtienen todas las configuraciones con las que se ejecutara el algoritmo. Además, los ítems disponibles para los personajes no están predeterminados, más bien, se leen de archivos cuyos paths deben estar en este archivo de configuración. Como condición de corte se implementó una función que centraliza las 4 formas de condición de corte (parametrizables también) donde si se cumple al menos una sola ya suficiente para cortar.

Métricas Clave

Algunas de las métricas clave analizadas:

- Generaciones
Define la cantidad de iteraciones necesarias para obtener un resultado. A menor cantidad de generaciones, más rápido se obtuvo el mismo.
- Fitness Máxima
Define la mejor performance presente en la población actual.
- Fitness Promedio
Define el promedio de las performances en la población actual, muy útil para interpretar mejoras en la población generación tras generación.

Casos estudiados y resultados

Dada la gran cantidad de metodos y parametros que posee, se hace imposible probar todas las combinaciones. Es por esto, que al igual que en el trabajo práctico anterior, se procedió a setear una configuración base, de la cual se fueron variando los parámetros uno a uno, con el fin de sacar conclusiones de estas ejecuciones.

La configuración base fue:

- replacementMethodA=Elite
- replacementMethodB=Ranking
- selectionMethodA=Elite
- selectionMethodB=Ranking
- isBolztmann=false
- crossOverMethod=UniformCross
- mutationMethod=GenUniform
- replacementMethod=3
- maxGenerations=5000
- optimalFitness=40.0
- fitnessEpsilon=0.2
- maxGenerationFitnessUnchanged = 200
- maxGenerationPopulationUnchanged = 100
- populationEpsilon=0.2
- mutationUniformProbability=0.1
- selectionMethodAPercentage=0.5
- replacementMethodAPercentage=0.5
- initialSize=1000
- nextGenerationPercentage=0.8
- crossOverProbability = 0.6
- type=Warrior
- tournamentsM=3

Variación de % de selección

En este caso se probó utilizando el método de reemplazo 1, el cual la generación siguiente son todos los hijos de la generación actual y usa sólo dos métodos de selección, variando el % del método 1. Los casos analizados fueron:

- 100% Elite - 0% Ruleta
- 85% Elite - 0% Ruleta
- 50% Elite - 50% Ruleta

En este caso pudimos apreciar que para 100% de un método obtuvimos resultados muy malos (fitness cercano a 20, convergencia prematura), y es entendible teniendo en cuenta que uno solo significa que no estaríamos haciendo selección en sí mismo.

A diferencia de esto, para el caso 85% ya se nota una mejora, ya que converge de forma más suave y llega a mejores resultados (toca picos de 24).

Y finalmente para 50-50, es donde obtuvimos los mejores resultados, dado que utilizamos por igual prioridad ambos métodos y obtuvimos una curva suave, creciente estricta, y el mejor resultado en cuanto a fitness de todos. Además, obtuvimos una mejoría en la población total con respecto a los casos anteriores, dado que ahora el promedio representa un 65% del mejor y antes era cercano al 50%.

Variación de métodos de selección

En este apartado volvemos a utilizar el método de selección 1, utilizando dos métodos al 50% cada uno y los fuimos variando. Los casos analizados fueron:

- 50% Elite - 50% Tournaments Deterministic
- 50% Roulette - 50% Ranking
- 50 % Elite - 50% Ranking
- 50 Roulette - 50% Tournaments Deterministic

En este caso pudimos observar que cuando se eligen dos métodos determinísticos, como elite y torneos, el sistema converge a un máximo local. En el caso de poner dos métodos que tenían más una parte de random se lograban mejores resultados, pero no pasaba siempre. Algunas veces convergía a un personaje con un fitness no tan alto como los mejores encontrados. El mejor resultado lo obtuvimos utilizando un porcentaje de un método determinístico y un porcentaje de un método random. Esto le daba la fuerza al método determinístico para “saltarse” el máximo local y llegar a resultados óptimos.

Boltzmann

En este apartado se probó el funcionamiento del algoritmo para los casos:

- Con Boltzmann
- Sin Boltzmann

Es importante aclarar que se utilizó Ruleta como método de selección secundario.

En este caso los resultados fueron los esperados, con Boltzmann, el sistema converge de manera levemente más rápida (en términos de generaciones) hacia un mejor fitness, dado que por ejemplo sin boltzmann se llega a 30 de fitness luego de 60 generaciones y con boltzmann luego de 50.

Vale aclarar también, que la utilización de Boltzmann no es gratuita, ya que demanda más procesamiento y eso se ve claramente en el tiempo de ejecución extra que demora un poco más. Pero el resultado más sobresaliente es que con Boltzmann, toda la población obtuvo una fitness muy cercana a la máxima (99%) mientras que sin Boltzmann el promedio de la población estaba en un 85% del máximo.

Variación métodos de reemplazo

En este apartado se probó el funcionamiento del algoritmo variando métodos de reemplazo para los casos:

- Método 1
- Método 2
- Método 3

Con los resultados obtenidos, pudimos observar que el método de selección 1 obtenía buenos resultados en cuanto a el fitness máximo pero la población normalmente no mejoraba. Esto quiere decir que tenias algunos individuos muy buenos y otros con un fitness muy bajo.

Por otra parte el método de selección 2 obtienen también buenos resultados en cuanto al fitness máximo y mejoraba el fitness de la población con respecto al 1.

Por último, con el método 3 obtuvimos los mejores resultados tanto en cuanto a fitness máximo como promedio. Hay que tener en cuenta que no siempre el fitness máximo era superior a los otros 2 métodos. Llegamos a la conclusión de que como el fitness poblacional aumentaba casi tanto como el máximo, la población converge a un set de individuos muy parecidos. Esto hacía que se llegara a un máximo local del cual no podía salir y muchas veces no era éste fitness el óptimo esperado.

Variación métodos de mutación

En este apartado se probó el funcionamiento del algoritmo variando métodos de mutación para los casos:

- GenUniforme
- MultiGenUniforme
- GenNoUniforme
- MultiGenNoUniforme

Se ve claramente en los resultados que obtuvimos, que la utilización del método MultiGen, provoca una convergencia muy turbulenta, mientras que Gen (simple) converge de manera más suave y directa.

En cuanto al fitness, en el caso de Gen siempre converge más rápidamente a un fitness mejor, mientras que el MultiGen nunca llega a este, variando entre valores menores.

En cuanto a Uniforme o No Uniforme, concluimos que el caso uniforme converge casi de manera estricta y continua, mientras que No Uniforme genera “saltos” en el camino.

(ver gráficos en anexo)

Variación métodos de cruce

En este apartado se probó el funcionamiento del algoritmo variando métodos de cruce para los casos:

- Cruce de a un punto
- Cruce de a dos puntos
- Cruce uniforme
- Cruce Anular

Pudimos observar que método de cruce incidía en cuanto tardaba la población en llegar a un máximo, siendo cruce a un punto el más lento. En cuanto al fitness, tanto máximo como promedio, los resultados obtenidos fueron similares en todos los casos.

(ver gráficos en anexo)

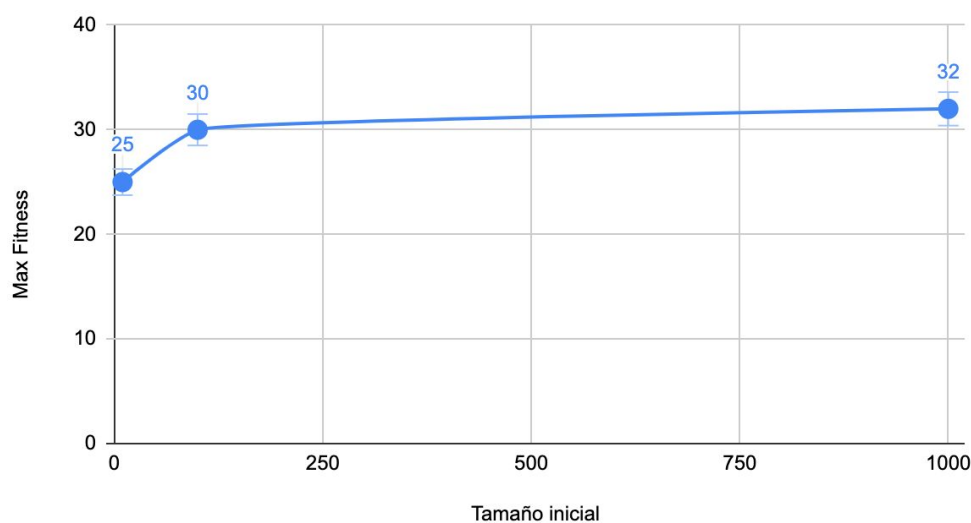
Variación tamaño inicial de población

En este apartado se probó el funcionamiento del algoritmo variando el tamaño inicial de la población para los casos:

- 10
- 100
- 1000

Con los resultados obtenidos, pudimos probar la hipótesis que dada una mala parametrización del tamaño inicial (tamaño chico = 10) puede llevar a no obtener los resultados óptimos esperados. En nuestro caso, con tamaño 10, la convergencia fue muy inestable, con altas y bajas, pero nunca superando el fitness 25, cosa si se superó apenas a 50 generaciones de tamaño inicial 100 y apenas a 15 generaciones para tamaño 1000.

Max Fitness por Tamaño inicial



Entonces podemos ver que a mayor tamaño inicial, el algoritmo performa mejor, la convergencia es más estable y puede llegar a óptimos más altos.

(ver gráficos correspondientes en el anexo)

Conclusiones

El warrior con la mejor fitness obtenida es:

- height=1.6701615311624989
 - VEST id=480374
 - GLOVES id=340516
 - HELMET id=356049
 - BOOTS id=924388
 - WEAPON id=84630,

Generation 1536

Best. Fitness: 34.339525324307395

Avg. Fitness: 24.86920120976502

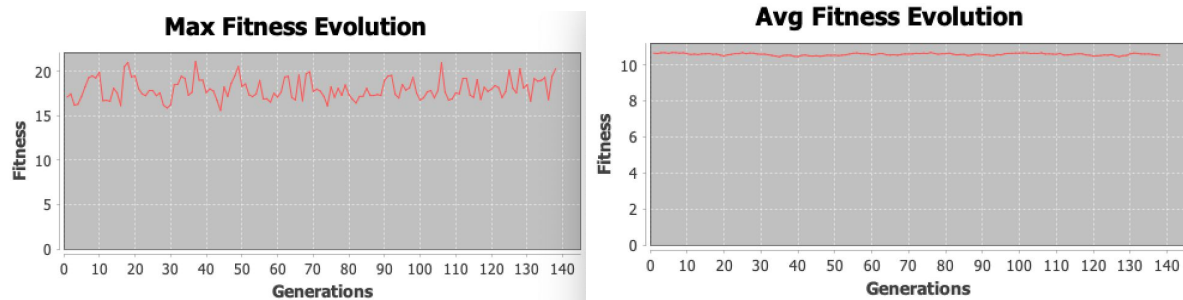
En base a los casos estudiados y sus resultados obtenidos concluimos entonces:

- Es fundamental poseer variedad poblacional para no caer en máximos locales.
- Una mala parametrización puede conllevar a convergencia prematura.
- A mayor tamaño inicial, convergencia más estable y mejores resultados (evita óptimos locales) pero tiene un tiempo de procesamiento mayor.
- Es importante agregar cierta aleatoriedad para evitar caer en máximos locales.
- El método de reemplazo 3 obtuvo mejores promedios de fitness en la población, pero los máximos nos encontramos con el método 2.
- Los métodos de cruza incidían en la velocidad de convergencia.

Anexo - Gráficos

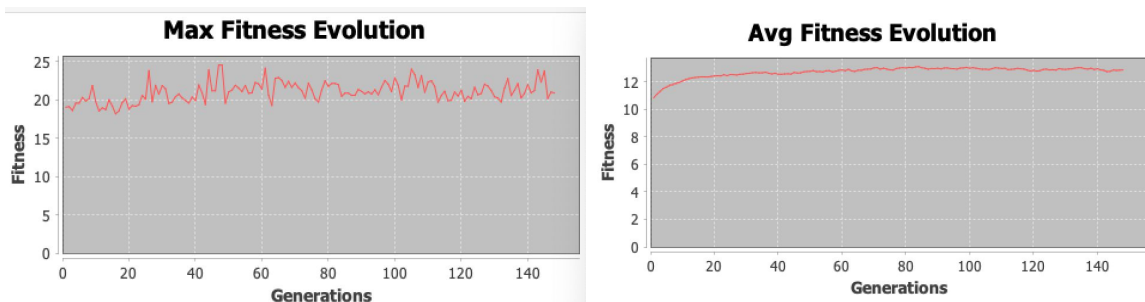
Variacion % de selección

100% Elite - 0% Ruleta



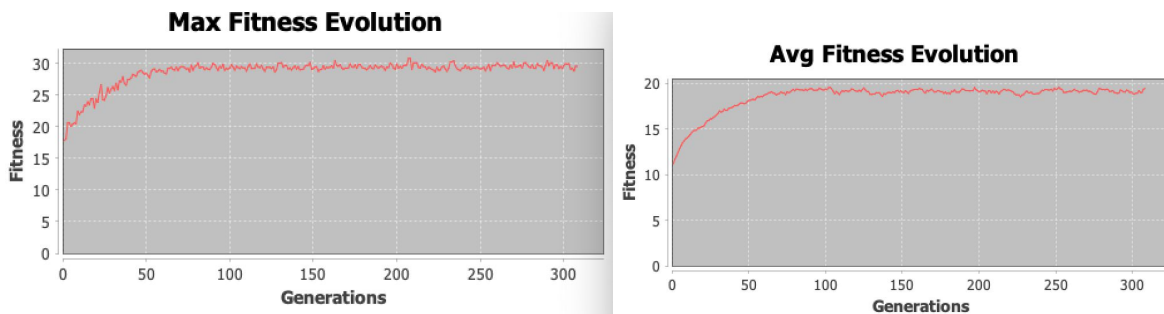
Generation 138 | Max Fitness: 20.29636225151729 | PromFitness: 10.52627314960226

85% Élite - 15% Ruleta



Generation 148 | Max Fitness: 20.899403315224504 | PromFitness: 12.853926320653517
Generation 61 | Max Fitness: 24.1347961406448 | PromFitness: 12.795712123507151

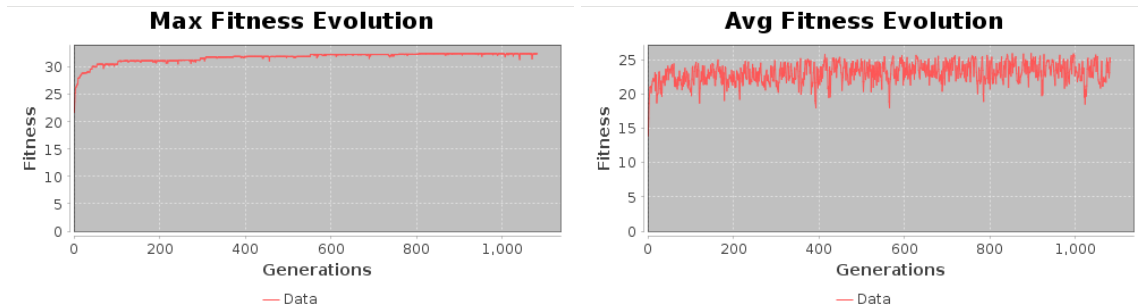
50% Elite - 50% Ruleta



Generation 308 | Max Fitness: 29.579471966690846 | PromFitness: 19.42336081442559

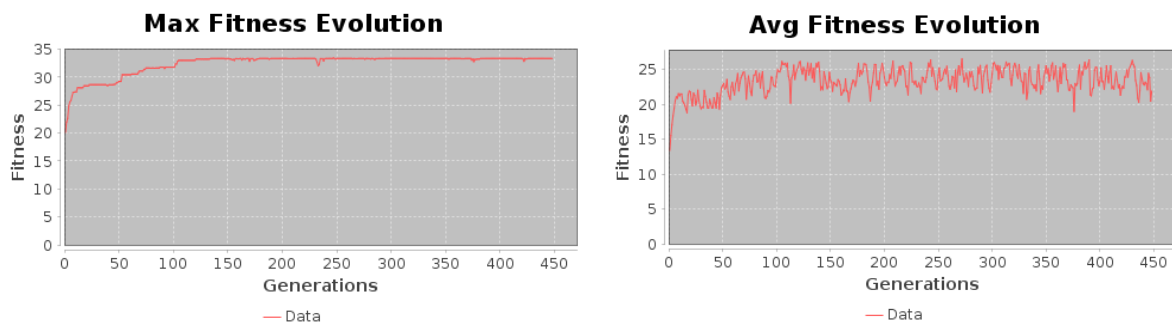
Variación de métodos de selección

50% Elite - 50% Tournaments Deterministic



Generation 1081 | Max Fitness: 32.445251165558595 | PromFitness: 24.67800084553079

50% Roulette - 50% Ranking

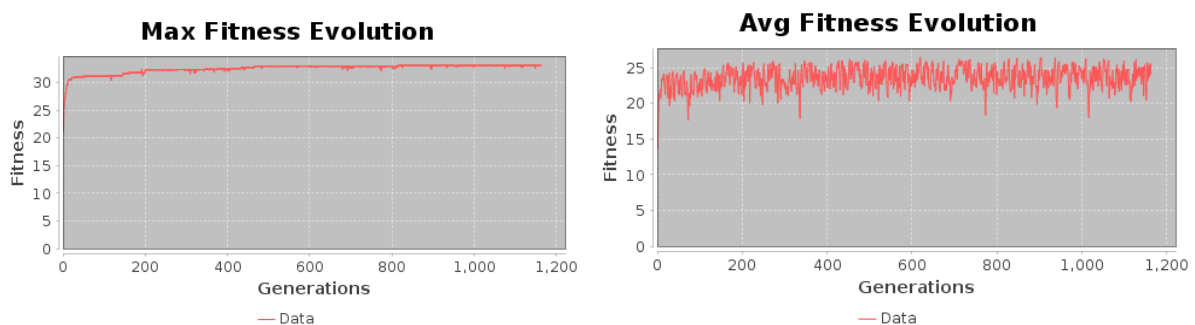


Generation 448 | Max Fitness: 33.33669484165846 | PromFitness: 21.90132680500888

Generation 515 | Max Fitness: 34.003930279826015 | PromFitness: 24.118202967921764

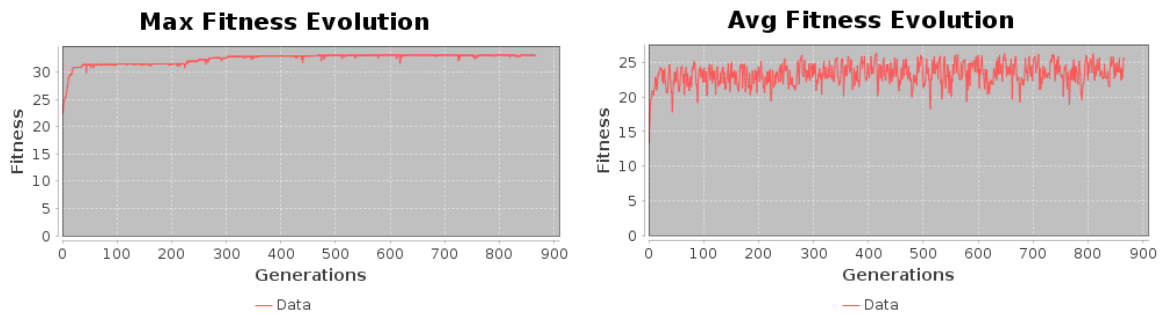
Generation 1307 | Max Fitness: 32.68664111649985 | PromFitness: 23.360035163233952

50 % Elite - 50% Ranking



Generation 1162 | Max Fitness: 33.12965655190983 | PromFitness: 25.545213355674978

50 Roulette - 50% Tournaments Deterministic



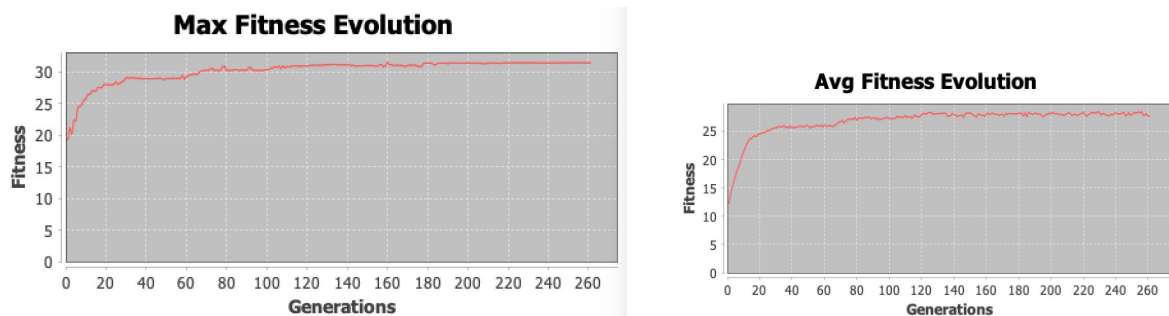
Generation 866 | Max Fitness: 33.05678747676323 | PromFitness: 25.5176908155406

Otros resultados:

Generation 1304 | Max Fitness: 34.69983281794151 | PromFitness: 24.330002413543557

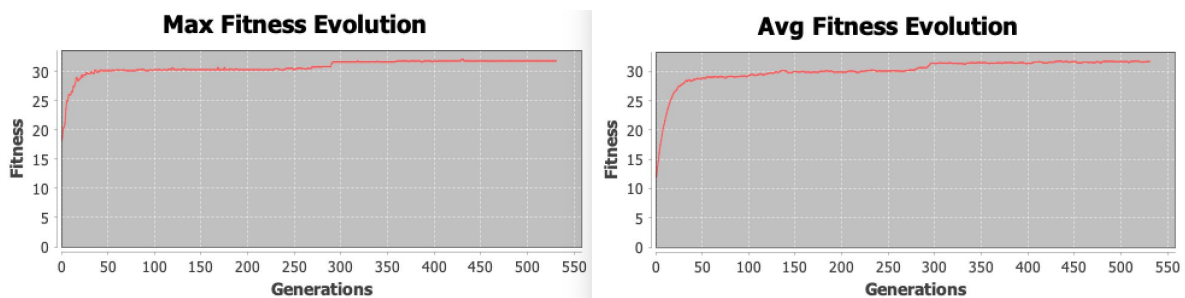
Boltzmann

Sin Boltzmann



Generation 261 | Max Fitness: 31.431176053191987 | PromFitness: 27.65914641194458

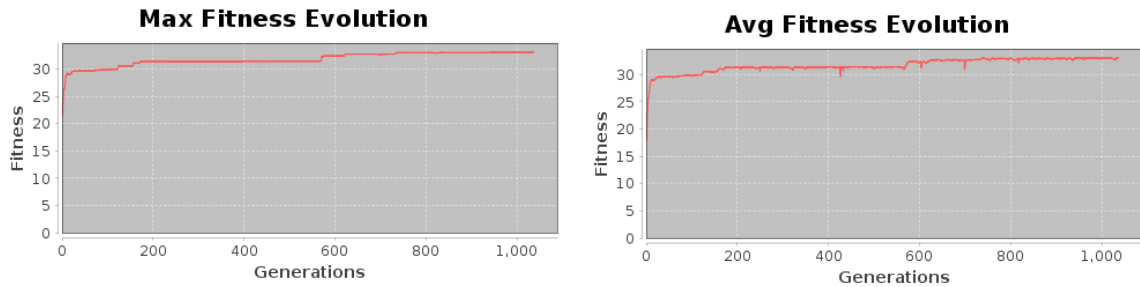
Con Boltzmann



Generation 531 | Max Fitness: 31.833919203963706 | PromFitness: 31.743221895878253

Variación métodos de reemplazo

Método 3



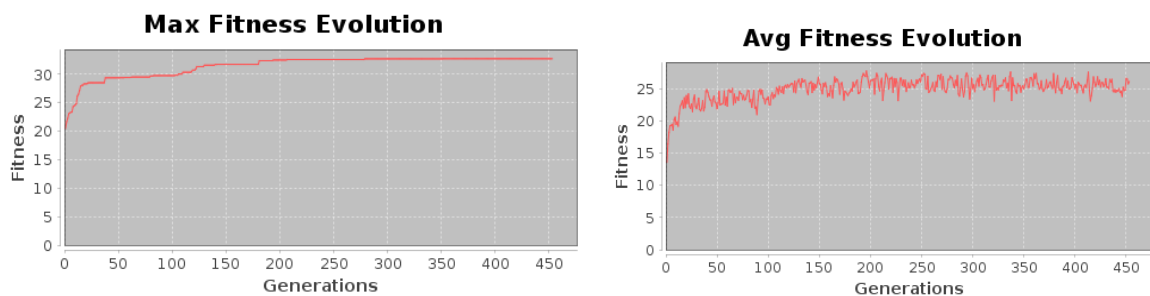
Generation 1037 | Max Fitness: 33.08998385741288 | PromFitness: 33.089974795706205

Otros resultados:

Generation 1636 | Max Fitness: 34.066706238061734 | PromFitness: 34.06365763164089

Generation 924 | Max Fitness: 33.60336782346512 | PromFitness: 33.50157972304013

Método 2



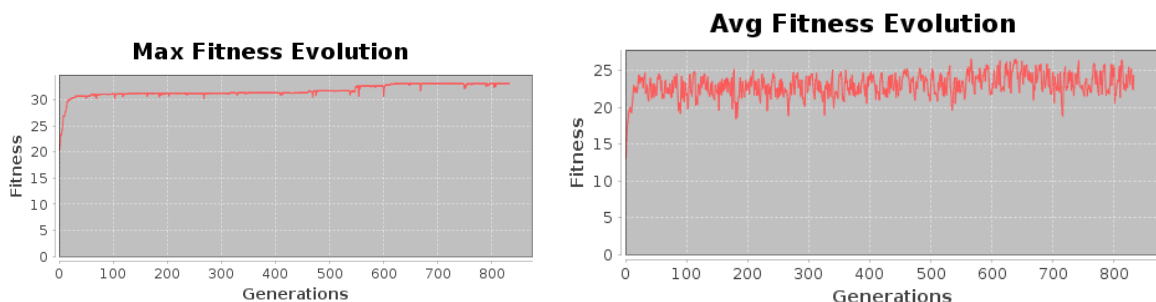
Generation 453 | Max Fitness: 32.741325915123994 | PromFitness: 26.126712095337027

Otros resultados:

Generation 712 | Max Fitness: 33.392871832084644 | PromFitness: 27.906488896695837

Generation 631 | Max Fitness: 33.1643335251 | PromFitness: 25.17786581956563

Método 1



Generation 832 | Max Fitness: 33.12419245779161 | PromFitness: 24.29285518867817

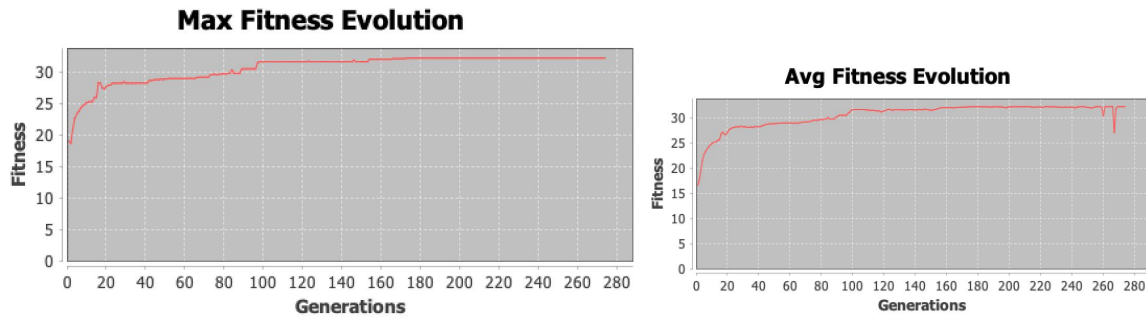
Otros resultados:

Generation 236 | Max Fitness: 32.363915157226124 | PromFitness: 22.880513102709422

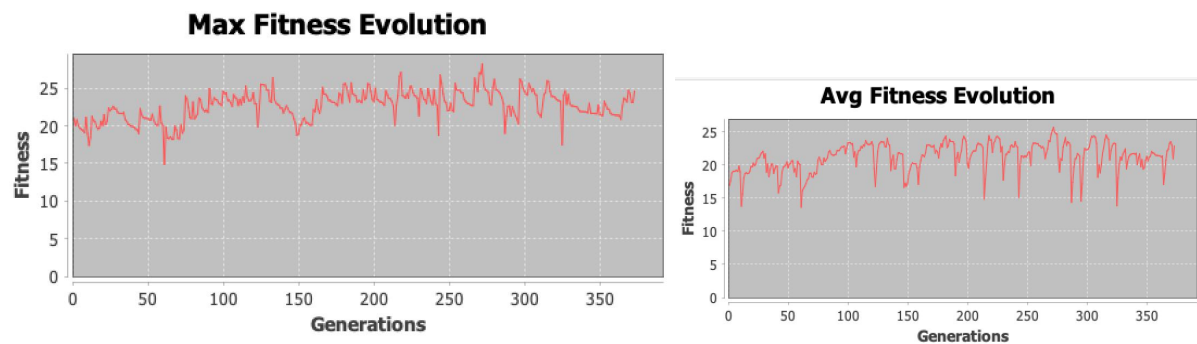
Generation 607 | Max Fitness: 31.9013938306675 | PromFitness: 24.588125507907613

Variación métodos de mutación

GenUniform



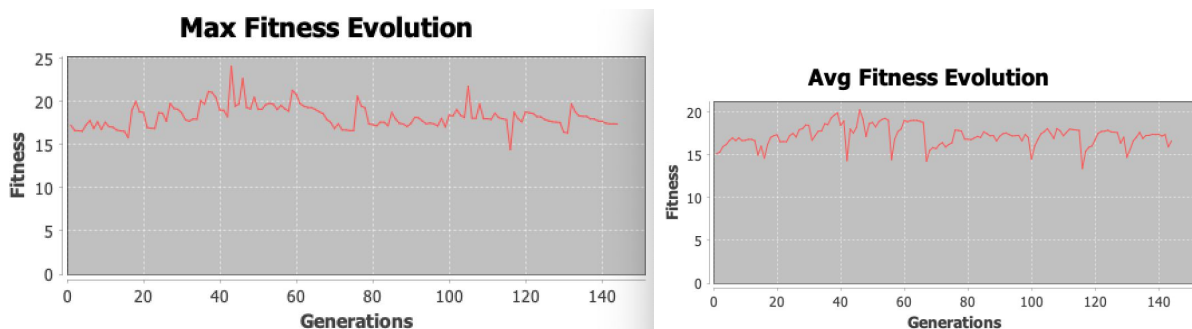
MultiGenUniform



Generation 373 | Max Fitness: 24.601279170779218 | PromFitness: 22.861833420029846

Generation 280 | Max Fitness: 26.169530220674286 | PromFitness: 22.710904046757566

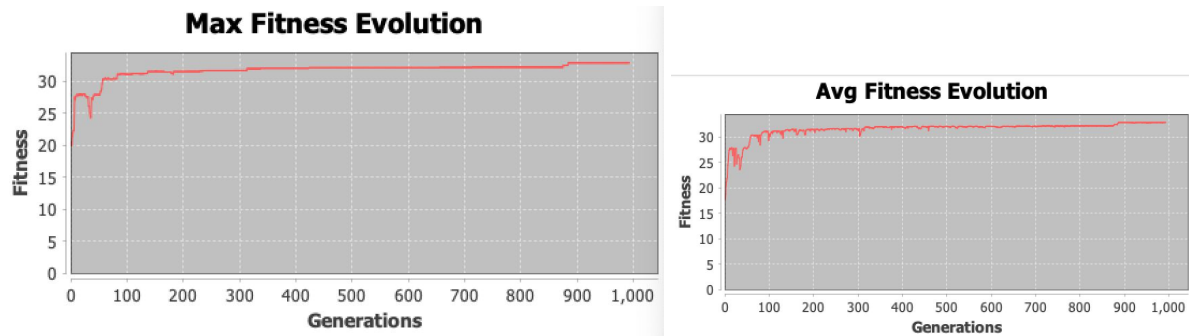
MultiGenNonUniform



Generation 144 | Max Fitness: 17.389175208227975 | PromFitness: 16.64635029562173

Generation 43 | Max Fitness: 24.04480426506928 | PromFitness: 18.038495385159578

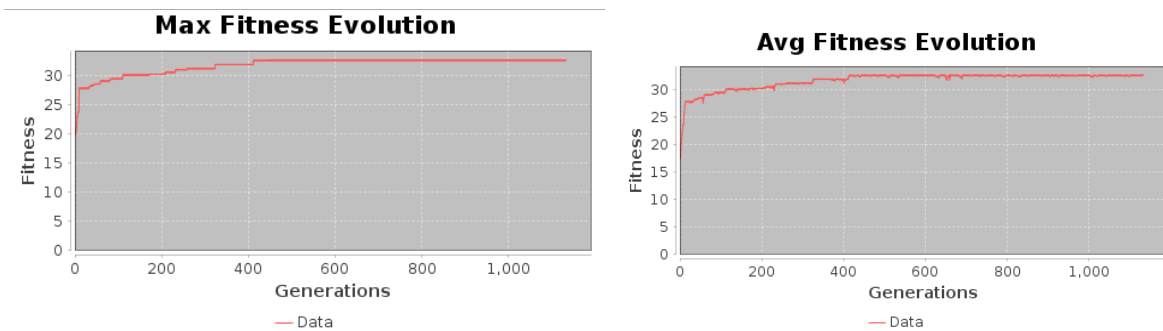
GenNonUniform



Generation 993 | Max Fitness: 32.8756242745248 | PromFitness: 32.86734321679335

Variación métodos de cruce

Cruce de a un punto



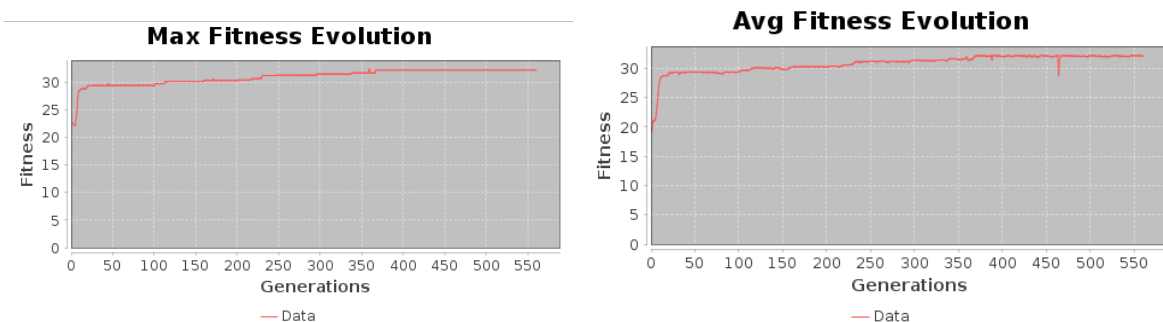
Generation 1133 | Max Fitness: 32.602340294627304 | PromFitness: 32.5987909171261

Otros resultados:

Generation 882 | Max Fitness: 32.49499468797042 | PromFitness: 32.332198359825426

Generation 1666 | Max Fitness: 33.50903208411508 | PromFitness: 33.415100618491444

Cruce de a dos puntos



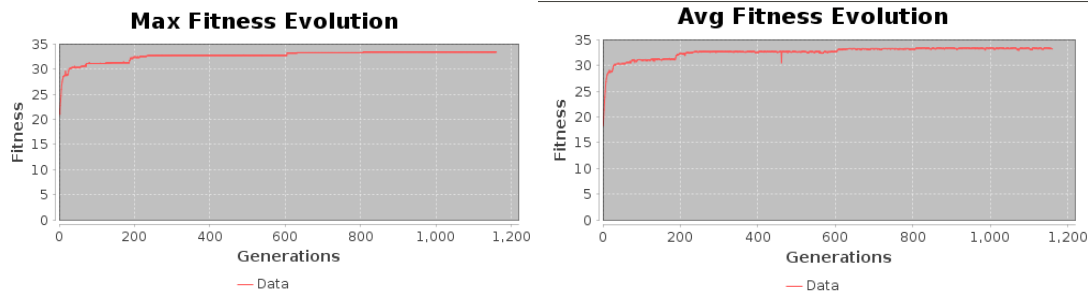
Generation 560 | Max Fitness: 32.199820584357596 | PromFitness: 32.17326104941303

Otros resultados:

Generation 589 | Max Fitness: 32.165067842072126 | PromFitness: 32.039537679242784

Generation 498 | Max Fitness: 31.500534063262187 | PromFitness: 31.406475163500726

Cruce uniforme

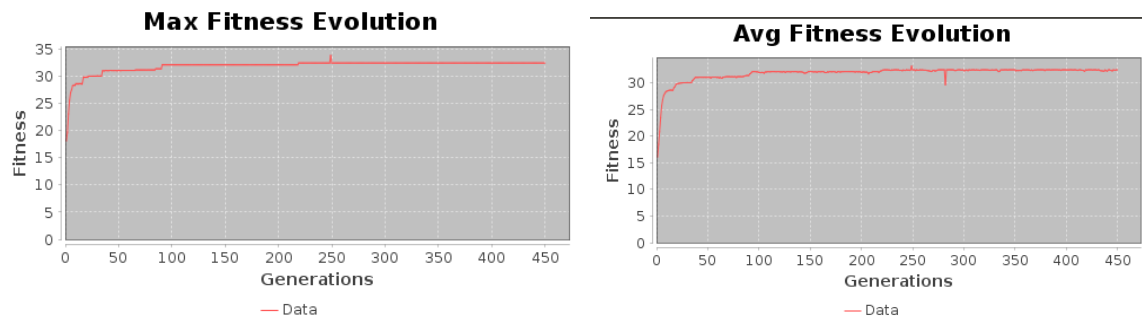


Generation 1160 | Max Fitness: 33.38378043995476 | PromFitness: 33.22014648831003

Otros resultado:

Generation 610 | Max Fitness: 32.054120837800646 | PromFitness: 31.97445597341291

Cruce Anular



Generation 450 | Max Fitness: 32.404965083030035 | PromFitness: 32.40470486712924

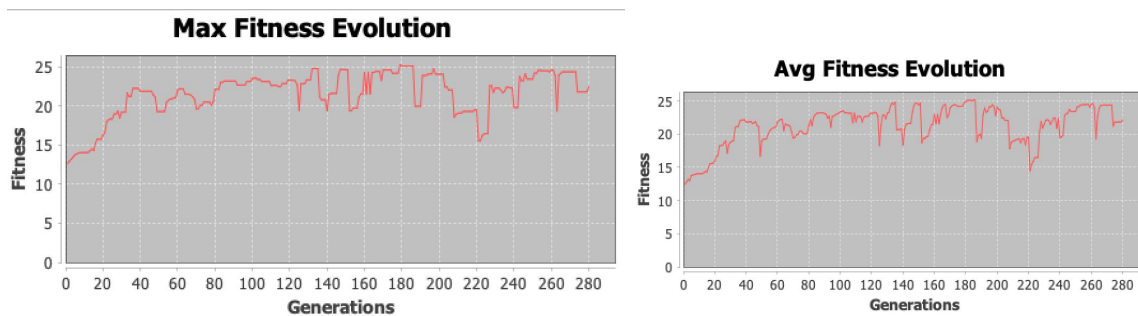
Otros resultado:

Generation 621 | Max Fitness: 33.62624572630153 | PromFitness: 33.479696626799644

Generation 611 | Max Fitness: 33.05067626217607 | PromFitness: 32.94370115178347

Variación tamaño inicial de población

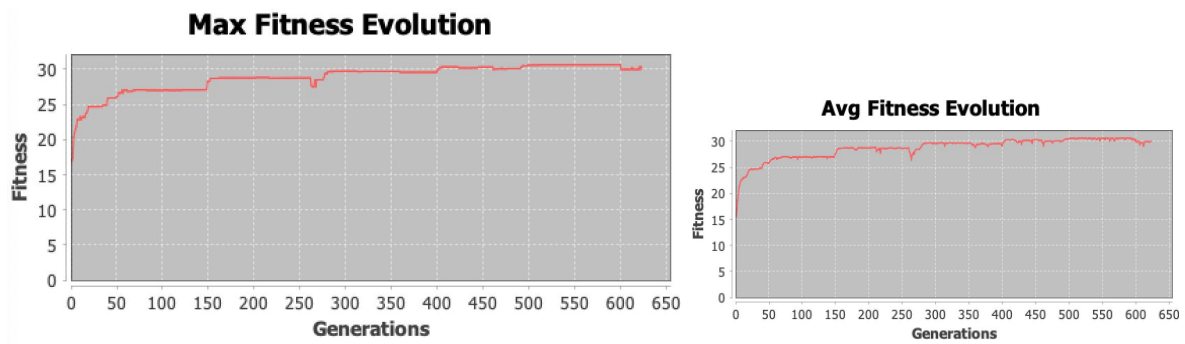
10



Generation 181 | Max Fitness: 25.134608330143468 | PromFitness: 25.06652993933733

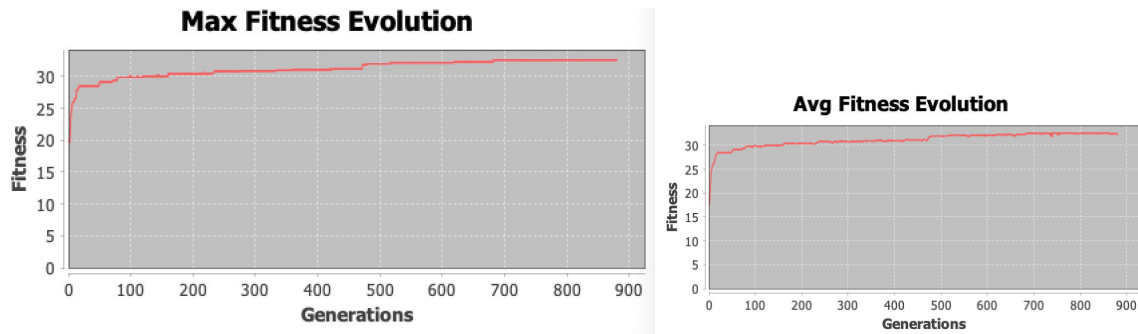
Generation 280 | Max Fitness: 22.457999532335947 | PromFitness: 22.140531491392725

100



Generation 622 | Max Fitness: 30.384406623449706 | PromFitness: 30.067653243556197

1000



Max Fitness: 32.582158030953934 | Avg. Fitness: 32.41822538065371