Algorisme evolutiu per a la presa de decisions basat en NEAT

Arnau Mayoral Macau

Resum– En aquest projecte s'implementa un algorisme evolutiu de tipus NEAT per a ser utilitzat en els entorns que proporciona la llibreria OpenAI Gym i per a Unity, dues de les eines més utilitzades per a provar algorismes d'aprenentatge reforçat en petits videojocs. Els algorismes NEAT (Neuro evolution of augmenting topologies) es basen en generar xarxes neuronals simples i fer servir tècniques evolutives per a complexificar les xarxes i, eventualment, arribar a una xarxa que representi l'òptim global al problema minimitzant el seu cost computacional.

Paraules clau– Neuroevolució, Neuroevolution of augmenting topologies, NEAT, algorisme evolutiu, complexificació, aprenentatge reforçat, Unity, Open AI Gym

Resumen– En este proyecto se implementa un algoritmo evolutivo de tipo NEAT para ser utilizado en los entornos que proporciona OpenAI Gym y para Unity, dos de las herramientas más utilizadas para probar algoritmos de aprendizaje reforzado en pequeños videojuegos. Los algoritmos NEAT (Neuro evolution of augmenting topologies) se basan en generar redes neuronales simples y usar técnicas evolutivas para complejizar las redes y, eventualmente, llegar a una red que represente el óptimo global al problema minimizando el coste computacional.

Keywords– Neuroevolución, Neuroevolution of augmenting topologies, NEAT, Algoritmo evolutivo, complejificación, aprendizaje reforzado, Unity, Open AI Gym

Abstract– This project implements a NEAT evolutionary algorithm to be used in the environments provided by the OpenAI Gym library and for Unity, two of the most used tools to test reinforced learning algorithms in small video games. NEAT (Neuro evolution of augmenting topologies) algorithms generate simple neural networks and gradually complexificate them using evolutionary techniques and, eventually, reach a network that represents the global optimum to the problem while minimizing its computational cost.

Keywords– Neuroevolution, Neuroevolution of augmenting topologies, NEAT, evolutionary algorithms, complexification, reinforcement learning, Unity, Open AI Gym

1 Introducció - Context del treball

Els videojocs són una de les indústries més grans del planeta, arribant a generar més beneficis anuals que la indústria del cinema i de la música juntes l'any 2016. Això pot ser degut al fet que el catàleg de videojocs ofereix un immens nombre de situacions en les que col·locar al jugador, i incloure'l en el seu petit món. Potser és per aquesta gran varietat de situacions que els videojocs han sigut un espai de proves recurrents per a fer proves d'algorismes d'intel·ligència artificial. Ofereixen un entorn controlat en el qual fer benchmarking dels models de Machine Learning. També és habitual utilitzar tècniques de IA per a desenvolupar els comportaments de certs elements del joc, habitualment, el comportament dels enemics. Aconseguint un comportament intel·ligent dels elements del joc (agents) s'assoleix un alt nivell d'immersió i ajuda a fer que el joc no es faci repetitiu.

En aquest treball l'objectiu és implementar un algorisme

• E-mail de contacte: arnau.mayoral10@gmail.com
• Menció realitzada: Computació
• Treball tutoritzat per: Jordi Gonzalez Sabaté Departament de Ciències de la Computació
• Curs 2021/22

de tipus NEAT des de zero. Aquest serà posat a prova primer amb alguns problemes clàssics de la literatura d'aprenentatge reforçat i, finalment, serà implementat en C per a la seva utilització en l'entorn del motor de videojocs Unity. El joc emprat per a l'avaluació del model en Unity es tracta d'un joc d'estratègia tipus Tower Defense en el qual el model haurà de fer de l'enemic al qual s'enfronta el jugador. Considerarem un èxit el treball si s'aconsegueix que el model jugui fent ús d'estratègies que podríem considerar intel·ligents dins la partida, arribant fins i tot a guanyar a un humà.

2 ESTAT DE L'ART

Actualment, els treballs de referència sobre els algorismes NEAT són els desenvolupats per la Universitat de Texas. Des del seu departament de ciències de la computació, Kenneth O. Stanley ha desenvolupat diversos treballs sobre aquests algorismes d'ençà que en va escriure l'article fundacional el 2002[1]

Entre els treballs consta un article sobre la coevolució en un joc d'un contra un [2], on es presenten les bases de la coevolució, una estratègia d'entrenament dels NEAT en entorns on els individus s'enfronten entre ells per a sobreviure.

A la xarxa es poden trobar moltes implementacions dels algorismes NEAT aplicats a diversos entorns com problemes típics en la literatura de l'aprenentatge reforçat [3], o coneguts jocs com el Super Mario World[4].

En aquest treball es desenvoluparà una implementació genèrica d'aquest algorisme en Python i una versió coevolutiva en C# per a ser utilitzada en el motor de videojocs Unity.

3 NEUROEVOLUTION OF AUGMENTING TOPOLOGIES

Els algorismes Neuroevolutius han demostrat un gran rendiment en aprenentatge reforçat, sobretot en alguns tipus de problemes concrets [5]. És per això que d'ençà que van sorgir n'han sortit moltes variants com el seu germà gran HyperNEAT, una versió que ha demostrat ser més eficient en problemes que requereixen xarxes neuronals molt més grans, amb molts valors d'entrada, com podria ser per processament d'imatges [6].

Els NEAT són algorismes evolutius que es basen en la intel·ligència artificial subsimbòlica, concretament en les xarxes neuronals, l'algorisme més habitual i versàtil d'aquesta categoria [7]. Les xarxes neuronals són un model matemàtic que permet processar un vector d'input i obtenir-ne un d'output. Les xarxes estan formades per unitats de processament anomenades neurones. Aquestes neurones poden estar connectades entre si, de forma que els valors d'entrada van passant per diverses neurones, les quals alteren el valor. El nombre de neurones de la xarxa i les connexions entre elles formen el que s'anomena topologia. Una neurona processa els valors que li arriben i passa el resultat a les següents neurones a les que estigui connectada. Aquest processament és una suma ponderada dels seus inputs passada per una funció no lineal, la qual permetrà que la xarxa pugui desenvolupar un comportament no lineal. La topologia de la xarxa i els pesos atorgats a les connexions són el que fan que la xarxa assoleixi els resultats estimats. Per tant, trobar els pesos correctes és la part d'aprenentatge que ha de resoldre el nostre model, en el nostre cas la neuroevolució.

La "Neuroevolution of augmenting topologies"s'encarrega de trobar la topologia i la combinació de pesos que representen una millor solució al nostre problema. Ho farà començant a partir de xarxes molt petites, probablement incapaces de generar un resultat òptim, i les anirà complexificant (evolucionant) a poc a poc, fins a trobar un resultat òptim o proper al desitjat [1]. Aquest comportament ens assegura que la solució trobada és una xarxa tan senzilla com sigui possible i, per tant, la solució amb menys cost computacional.
Per a la complexificació parlarem de mutacions, ja que seran canvis aleatoris en les xarxes. Consistirà en: afegir neurones a la xarxa, afegir connexions entre neurones existents i en variar aleatòriament els pesos de la xarxa. D'aquesta manera la xarxa va obtenint a poc a poc la capacitat d'adoptar comportaments cada cop més complexos. Cada canvi en la topologia serà un gen. Per tant, cada xarxa tindrà el seu genoma, es a dir, el conjunt de canvis en la topologia que s'ha fet respecte a la xarxa buida. Aquest genoma ens permetrà saber si dues xarxes són properes genèticament o dit d'una altra manera, si comparteixen una part de la topologia.
El funcionament del model serà crear un grup de xarxes buides. En el símil de la teoria de l'evolució, les xarxes són diferents individus pertanyents a una població. Durant l'execució del model, es crearan generacions d'individus, encreuant i mutant els individus que mostrin un millor rendiment (selecció natural segons el valor d'adaptació). Aquest factor està basat en el fet que l'encreuament de dues xarxes genera un resultat igual o millor que el dels individus originals. D'aquesta manera al cap de les generacions el valor d'adaptació o rendiment dels individus anirà augmentant.

El pseudocodi de l'algorisme NEAT que s'implementarà serà el següent:

Algorithm 1 NEAT

```
1:  procedure MAINLOOP(n_gens)
2:      Population ← newPopulation()
3:      for k = 1, k++, while k < n_gens do
4:          Population ← NextGen(Population)
5:
6:  procedure NEXTGEN(Population)
7:      species ← Speciate(Population)
8:      Mutate(species)
9:      Fitness(species)
10:     adjustFitness(species)
11:     return OffSpring(species)
```

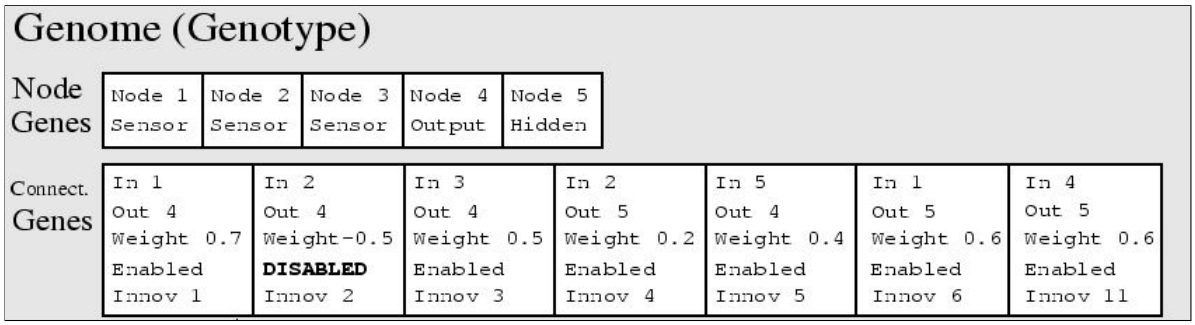

Fig. 1: Codificació del genoma d'un individu [1, 8]

3.1 Codificació de les xarxes

Per a poder desenvolupar un algorisme evolutiu és indispensable construir les xarxes d'una manera que ens permeti operar amb gens, per exemple, per fer recombinació genètica (reproducció). Per això cal codificar la topologia de la xarxa d'una manera concreta, que a més ens permeti donar un context d'història evolutiva a la xarxa. D'aquesta manera podrem determinar la familiaritat entre xarxes (veure Fig.1). Per fer-ho s'assignarà un nombre d'innovació a cada gen, és a dir, a cada canvi que es faci en una xarxa. Aquest nombre serà el mateix pels mateixos canvis que es produeixin, independentment de l'individu. D'aquesta manera si dos individus comparteixen una connexió entre dues neurones, aquesta serà simbolitzada pel mateix gen, amb el mateix nombre d'innovació. Amb això cada gen contindrà: neurona d'entrada, neurona de sortida, pes, si el gen està actiu o no (més endavant es veurà perquè), i el nombre d'innovació.
Una agrupació de gens es pot considerar una unitat funcional. Dit d'una altra manera, una unitat funcional és un conjunt de neurones i connexions concretes que poden efectuar una operació lògica respecte els paràmetres d'entrada. Per exemple, en un agent, poden representar la lògica de: si aquests dos paràmetres es compleixen, efectua aquesta acció. Gràcies a la codificació única dels gens la unitat funcional, si té un impacte positiu en el valor d'adaptació, serà heretada pels descendents dels individus que la tinguin, conservant així la funcionalitat.

3.2 Processament de la xarxa

El processament de la xarxa és el procés que generarà una acció segons els valors d'entrada de la xarxa. Generalment com les xarxes s'organitza en capes que s'envien valors entre elles, aquest processament es duu a terme des de la capa d'entrada fins la de sortida. En canvi, en el cas dels NEAT, no ho podem fer així, ja que les neurones no estan organitzades en capes. Això fa que s'hagi de calcular el valor de final a inici, utilitzant una funció recursiva que recorrerà tota la xarxa fent un recorregut DFS. D'aquesta manera, començant per una neurona d'output, baixarà en profunditat fins a arribar a les neurones de input i anirà pujant els valors. Això ho farem construint una matriu d'adjacència a partir del genoma de l'individu, i ens ajudarem també d'una estructura hashtable per a emmagatzemar els resultats de les neurones que anem podent calcular de manera que, si una neurona ja s'ha calculat, no la tornem a repetir. Això reduirà el cost computacional del procés.

3.3 Mutació de les xarxes

El procés de mutació de les xarxes és el procés que afegeix complexitat als individus. És un component clau dels NEAT, ja que aquests es basen a complexificar a poc a poc models simples per a obtenir la solució més simple a un problema. Amb el procés de mutació les xarxes aconsegueixen dues coses: trobar una combinació de pesos adequada i afegir elements a la topologia per tal que la xarxa sigui capaç de resoldre problemes cada vegada més complexos.

La mutació és un element que, com en l'evolució natural, té un component aleatori. Per a cada generació d'individus, aquests tenen una probabilitat de tenir algun tipus de mutació, la qual afegirà un gen al seu genoma i potencialment, millorarà el seu rendiment.

Les mutacions es duran a terme segons un percentatge de probabilitat que s'ha d'establir des de l'etapa de disseny. Per tant, hem d'escollir unes probabilitats γ_1, γ_2, γ_3 per a les mutacions de pesos, nova neurona i nova connexió respectivament. Els valors d'aquestes constants seran discutits més endavant, generalment seran valors petits per tal que les xarxes evolucionin a poc a poc i les xarxes simples tinguin temps de trobar una combinació de pesos adequada al problema.

Mutació de pesos La mutació de pesos consisteix a alterar tots els pesos de la xarxa aleatòriament. Aquesta pertorbació estarà escalada segons un paràmetre *step* que ens permetrà controlar si les mutacions són més brusques o més subtils.

Nova neurona En el cas de la mutació de nova neurona, el que es farà serà escollir una connexió de la xarxa aleatòriament, i es dividirà la connexió en dos introduint-hi una neurona al mig. D'aquesta manera convertirem una connexió existent en una nova neurona i dues noves connexions que la uniran a la xarxa, afegint dos gens nous. Per tal de fer aquesta mutació menys agressiva els pesos que es posaran a les dues noves connexions seran d'1 per a una de les dues, i a l'altra se li assignarà el pes que tenia la connexió original. D'aquesta manera l'addició d'un nou node no canvia el comportament d'una xarxa en un principi, però permet que en futures mutacions es desenvolupi un comportament diferent ja que permetrà que sorgeixin noves connexions amb aquest nou node. Per motius de deixar enregistrat el passat genètic dels individus, els gens que representen la connexió que està sent dividida no s'elimina, es desactiva. D'aquesta manera la xarxa no tindrà en compte aquesta connexió, però quan comparem dos genomes podrem veure el seu passat

genètic i podrem determinar si dos individus tenen un passat comú.

Nova connexió Per la mutació de nova connexió, s'escollirà una parella de nodes que no tingui una connexió, i es crearà un nou gen que codifiqui la nova connexió amb un pes aleatori. Cal tenir en compte que s'ha de restringir quines neurones poden fer d'entrada i sortida de la connexió. D'aquesta manera una neurona d'entrada mai podrà ser la sortida d'una connexió, i el mateix passa amb les neurones d'output però a l'invers.

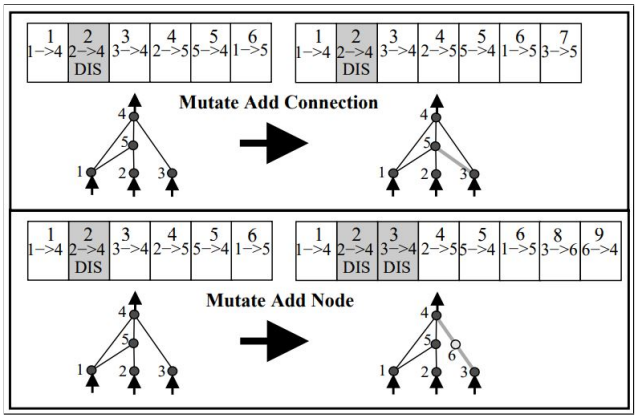

Fig. 2: Mutació genètica d'un individu segons la codificació establerta

3.4 Recombinació de les xarxes

L'operador d'encreuament és un dels processos més importants dels algorismes NEAT, i alhora un dels més complicats de dissenyar. Aquesta funció s'encarrega d'encreuar dos individus i generar-ne una descendència, la qual, hauria de ser igual o millor que els seus progenitors. La manera més habitual de fer-ho és fixant-se amb els genomes dels progenitors i els identificadors dels seus gens (nombre d'innovació) [1]. Per entendre la recombinació genètica cal entendre vàries coses. Els nombres d'innovació dels gens marquen l'antiguitat de la primera aparició d'aquell gen. Per tant, a més petit el nombre més antic. Per a recombinar, el primer que farem serà quedar-nos amb els nombres d'innovació més grans dels dos genomes dels progenitors. Entre aquests dos nombres ens quedarem amb el més petit. Aquest punt l'anomenarem frontera. Un cop tenim aquest valor considerarem els gens amb nombre més alt com a gens d'excés. Entre els gens més antics que el valor frontera, considerarem els gens no compartits com a gens disjunt.

Per a crear el genoma de la descendència, aquest heretarà els gens compartits aleatòriament d'un dels progenitors. I heretarà els gens disjunts i d'excés del progenitor amb més valor d'adaptació. D'aquesta manera el descendent heretarà les unitats funcionals del progenitor més adaptat, amb la possible variació d'algun gen compartit. Així s'assegura que el descendent serà o igual o millor que els progenitors.

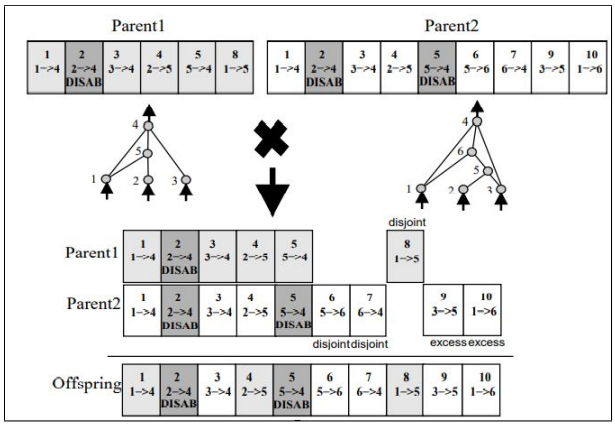

Fig. 3: Exemple d'aplicació del operador de recombinació genètica

3.5 Control de varietat en el genoma: especiació

Un dels problemes dels algorismes genètics és que pot convergir en un màxim local molt ràpidament. Això en termes evolutius es tradueix en un individu amb un valor d'adaptació superior a la mitjana, acaba imposant els seus gens a la resta de xarxes, sense donar oportunitat que aquestes es desenvolupin. Aquest fet fa que no es pugui explorar tot el ventall de solucions possibles, i l'algorisme es quedi amb el primer individu que representa una millora significativa del que es tenia. El factor d'exploració de l'espai de solucions es diu variabilitat genètica, i cal que sigui tan elevada com sigui possible durant tota l'execució de l'algorisme si no volem una convergència prematura. A més a més, es considera que els resultats d'aplicar el operador d'encreuament entre genomes propers solen ser millors que quan s'encreuen individus amb un genoma molt diferent.

Una de les possibles solucions a aquest problema és utilitzar l'agrupament d'individus en espècies, per a protegir unes espècies del progrés de les altres. D'aquesta manera es duria a terme l'evolució per generacions però només entre els individus de la mateixa espècie.

Per a poder dividir els individus en espècies, com en l'evolució natural, ens cal definir un operador de familiaritat genètica. Això ho farem establint un operador de distància entre genomes. Definint un llindar de distància δ que permetrà determinar si dos individus són prou propers per a ser considerats de la mateixa espècie.

$$\delta = \frac{\lambda_1 * E}{N} + \frac{\lambda_2 * D}{N} \lambda_3 * \omega$$

Fig. 4: Operador de distància entre genomes. On E són el nombre de gens d'excés, D el nombre de gens disjunts i ω la diferencia mitjana dels pesos entre els gens compartits. Les constants λ_1, λ_2, λ_3 permeten canviar la importància que es dona a cada tipus de diferència entre genomes, i N és el nombre de gens que tenen els genomes (ens quedem amb el petit dels dos) per tal de normalitzar la distància independentment de la longitud dels genomes.

A la Fig.4 podem veure el operador de distància més utilitzat [8] per a aquest tipus de codificació.

Les espècies tindran un representant, escollit aleatòriament entre els individus que la formen. D'aquesta manera quan s'introdueixi un individu a la població s'inclourà en la primera espècie que tingui un representant amb una distància inferior a δ amb el nouvingut.

3.6 Selecció de pares

El pas a la següent generació d'individus va estrictament lligar al procés de selecció natural que s'aplica a la generació actual. Aquest procés de selecció ha de penalitzar a les espècies que no estan progressant i afavorir a les que sí, però sense eliminar espècies emergents que puguin aportar una solució vàlida més endavant. La selecció es fa atorgant un major nombre de descendents a les espècies més ben adaptades respecte a les menys adaptades, així com funcionaria en l'evolució natural de les espècies. D'aquesta manera les espècies que presenten un bon rendiment tenen l'opció de tenir més individus i, per tant, d'evolucionar més ràpidament.

Un dels problemes que causa aquest tipus de selecció natural és que les espècies adaptades cada cop obtindrien més individus fins a un punt que el model no seria sostenible. És per això que s'ha d'afegir un sistema de penalització a les espècies sobrepoblades. Un sistema molt utilitzat per a penalitzar aquests casos és el *Explicit fitness sharing* [9], un sistema que ajusta el valor d'adaptació dels individus de manera que rebin una penalització en el cas de hi hagi molts individus semblants dins la mateixa espècie. D'aquesta manera les espècies s'autoregulen per a no créixer massa i mantenir només els individus que representen una solució que val la pena conservar. Com veiem en la Fig.**??**, per a cada individu i, es divideix el seu valor d'adaptació entre el nombre d'individus que estiguin a una distància prou propera.

$$f'_i = \frac{f_i}{\Sigma_j^n sh(dist(i,j))} \tag{1}$$

$$sh(x) = x < \delta ? 1 : 0 \tag{2}$$

$$n_k = \frac{F_k}{F_t}|P| \tag{3}$$

Fig. 5: (1)(2)Aquesta és la manera d'ajustar el valor d'adaptació dels individus segons la distància amb els individus de la seva espècie. (3) Nombre n_k de descendència atorgat a l'espècie k, on F_k és la mitjana del valor d'adaptació ajustat dels individus de l'espècie k, F_t és la mitjana del valor d'adaptació ajustat de tota la població i $|P|$ és el nombre d'individus a la població[10]

Un cop tenim el valor d'adaptació ajustat al context de l'espècie ja podem calcular el nombre de descendència que s'atorga a cada espècie tenint en compte el valor d'adaptació que té respecte a la resta d'espècies com podem veure a la Fig.5(3)

Quan tenim el nombre d'individus que ha de tenir cada espècie a la següent generació s'ha de generar aquests individus mitjançant l'operador d'encreuament. D'aquesta manera, per a cada espècie, agafarem un primer progenitor aleatòriament entre el 20% amb més valor d'adaptació ajustat i l'encreuarem amb un segon progenitor agafat aleatòriament d'entre tots els individus de l'espècie. D'aquesta manera un progenitor sempre serà dels més adaptats, i s'assegura la continuïtat de l'espècie. Una vegada s'han generat els n_k nous individus de l'espècie, substituïm els individus anteriors per la seva descendència. En algunes implementacions NEAT el campió de l'espècie (el que té major valor d'adaptació) passa directament a la nova generació, per evitar que es perdi un individu amb bon rendiment a causa de la probabilitat d'encreuaments i la recombinació genètica.

3.7 Restriccions

Una de les coses que podem tenir en compte, és aplicar restriccions explícites als individus. En alguns casos algunes decisions que l'algorisme pot prendre no són adequades, ja que pot no reunir les condicions per a prendre la decisió. En el joc en el qual s'ha treballat, encara que l'algorisme ho decideixi, la decisió no es durà a terme si l'individu no reuneix les condicions adequades. De totes maneres, i per a explotar al màxim la capacitat dels NEAT, es pot fer que els individus aprenguin a respectar aquestes restriccions. Això es pot dur a terme mitjançant penalitzacions als individus que intentin saltar-se la restricció. D'aquesta manera en lloc de fer el ranking segons el nombre de victòries, ho canviarem perquè sigui un sumatori de totes les fitness aconseguides en les partides guanyades. Aquesta fitness ve donada per la fórmula Fig.6. Així, els individus primer hauran d'aprendre a complir les normes, i després, a maximitzar les victòries.

$$fitness+ = victòria?(200 - penalització(violacions)) : 0 \tag{4}$$

Fig. 6: On violacions es el nombre de cops que l'individu ha intentat llençar una unitat sense l'or corresponent

3.8 Adequació dels paràmetres

Durant l'explicació de l'algorisme s'han anat trobant diferents paràmetres que cal ajustar per a trobar un bon funcionament de l'algorisme. Els paràmetres són de gran importància, ja que evitaran mal-funcionaments de l'algorisme, mantindran la variabilitat genètica i evitaran la convergència prematura [11]. Aquests paràmetres poden canviar segons el tipus de problema que s'hagi de resoldre, però existeixen configuracions que s'han demostrat sòlides i robustes.

Paràmetres:

new_node_mutation_rate: Probabilitat d'una mutació de nova neurona: 0.05

new_link_mutation_rate: Probabilitat d'una mutació de nova connexió: 0.08

weight_mutation_rate: Probabilitat de mutació dels pesos: 0.9
step: Proporció de la mutació de pesos: 2
λ_1: Pes dels gens d'excés en la mesura de la distància genètica : 1
λ_2: Pes dels gens disjunt en la mesura de la distància genètica : 1
λ_3: Pes de la diferència mitjana de pesos en la mesura de la distància genètica: 0.3
δ: Valor llindar per a determinar l'especiació: 3.0 i augmenta en 0.3 cada generació , ja que els genomes es fan més distants amb el pas de les generacions.
DropOffAge: Les espècies que portin més de 15 generacions sense mostrar una millora, seran eliminades.

4 ENTORN

L'entrenament dels algorismes NEAT depèn molt del tipus fitness function que apliquem, ja que cada entorn on es poden utilitzar té una manera diferent de considerar d'èxit o el fracàs d'un agent que pren decisions. Depenent d'aquest factor l'estratègia d'entrenament pot canviar. Per a veure quina és la millor manera de dur a terme l'entrenament cal tenir clar les bases de l'entorn.

En el cas que s'ha tractat, l'entorn es tracta d'un videojoc. El joc es senzill i consta d'una única mecànica. La mecànica és un carril, on els dos jugadors col·loquen unitats pagant-ne un cost en or. S'obté una moneda d'or cada segon fins a acumular un màxim de 15. Aquestes unitats, un cop col·locades, caminen cap a la base del jugador contrari per intentar destruir-la. Pel camí, es poden topar amb les unitats que ha llençat l'altre jugador i s'enfronten. En cas d'arribar a la base del contrari, la unitat ataca a la base i li va traient punts de vida fins que la destrueix, guanyant així la partida. Hi ha 4 unitats amb diferents característiques, de manera que combinar-les de forma adequada (de forma estratègica) junt amb una bona gestió de l'or, fa la diferència.

5 COEVOLUCIÓ

El sistema d'entrenament d'un NEAT, depèn de la tipologia de l'entorn. En joc on s'ha treballat, els individus s'han d'enfrontar entre si. De manera que la supervivència d'un individu depèn directament dels enfrontaments amb els altres individus. Aquest concepte es diu coevolució [12]. Això fa que determinar el millor individu requereixi més d'una partida per individu.

La manera ideal i exhaustiva de fer-ho seria enfrontant cada individu contra la resta d'individus de la població. Si féssim un recompte del nombre de victòries de cada individu podríem elaborar un ranking fàcilment. Aquest sistema, però requereix un nombre molt elevat de partides. Per a N individus ens deixaria amb n*(n-1)/2 partides, que per a N=150 ens deixaria amb 11.175 partides. Aquest és un nombre massa elevat de partides i comporta un cost computacional i temporal massa alt.

El sistema més utilitzat en el cas de l'entrenament de les NEAT, és el de l'enfrontament basat en especiació. Aprofitant el sistema de divisió en espècies de la metodologia NEAT. Les espècies garanteixen que els individus que les componen tenen una topologia semblant, i, per tant, amb el mateix input generarien una resposta semblant. D'aquesta manera podem considerar que les espècies representen estratègies de joc. Així, podem passar de trobar el millor individu, a trobar la millor estratègia. Per a això dividirem les partides en dues fases. Primer trobarem el millor individu de cada espècie, que farà de representant, i després enfrontarem els representants en un torneig. Segons el resultat del representant en el torneig s'atorgarà un nombre de descendències a cada espècie per acabar generant la següent generació. El resultat pot ser, per exemple, el nombre de victòries en el torneig.

Hall of Fame Una altra pràctica recurrent és afegir al torneig els campions d'altres generacions. D'aquesta manera els nous individus es veuen obligats a vèncer estratègies que han guanyat en el passat. Així garantim que la progressió està orientada cap a la millora continua del rendiment.

Host-Parasite Una tècnica habitual en el cas de l'enfrontament d'individus és la separació de la població en dues subpoblacions, les quals funcionen per separat en el procés d'especiació, però que competiran en el torneig. D'aquesta manera ens permet reduir encara més el nombre de partides a jugar (ja que les espècies tindran menys individus) i a més a més, les poblacions no es molestaran a l'hora de generar estratègies semblants però no iguals. [13]

6 AVALUACIÓ

6.1 Mètriques

Per a comprovar que l'algorisme funciona correctament ho podem fer de diverses maneres. Ho podem fer en l'àmbit subjectiu: es pot jugar contra els individus i veure quin és que té un comportament més intel·ligent, o podem utilitzar les mateixes mesures de l'algorisme per a trobar aquells individus que resulten més prometedors.
Com els NEAT generen molts individus, comprovar manualment els individus és una tasca feixuga i pesada, és per això que ens cal recollir les dades de tots els individus que anem generant, d'aquesta manera podem establir mètriques per a trobar aquells individus que tenen potencial per ser els millors.

Victòries contra *hall of fame* Si mantenim un *hall of fame* amb el campió de cada generació, podem enfrontar els campions entre si i veure si el pas de les generacions genera individus capaços de guanyar o com a mínim empatar amb els anteriors campions.

Fitness per partida En aquest cas la mètrica que utilitzarem serà la mitjana de fitness per partida. Això vol dir que

els individus han d'estar prou adaptats per a tenir una bona fitness contra totes les altres estratègies que es puguin trobar en el torneig. Si agafem un llistat dels campions dels tornejos de cada generació hauríem de veure com la tendència és creixent, doncs el procés evolutiu farà que a mesura que avancin les generacions els individus estiguin més adaptats.

Longitud del genoma per generació El genoma és el conjunt de gens que determinen la topologia de la xarxa neuronal. Un genoma més gran, codifica una xarxa neuronal més complexa que un genoma més petit. De manera inversa i a causa del mateix motiu, els genomes petits tenen menys cost computacional que els grans. Si analitzem la longitud del genoma durant els campions de totes les generacions podem veure si el problema requereix unes xarxes molt poblades i connexes o pot ser resolt de manera eficient amb una xarxa més petita.

Il·legalitats per partida Una altra manera d'avaluar l'algorisme es comprovar si la política de penalització de moviments il·legals està funcionant correctament. Per a fer-ho haurem de mostrar el nombre d'il·legalitats del campió de cada generació. Hauríem d'observar que el nombre va decreixent amb el nombre de generacions i s'hauria de mantenir baix.

6.2 Anàlisi de resultats

Per a un bon anàlisi de resultats, ens cal executar l'algorisme vàries vegades canviant certs paràmetres o estratègies. D'aquesta manera podem veure a la pràctica quin efecte tenen diferents entrenaments o polítiques.

Abans d'analitzar els resultats, cal aclarir diferents conceptes per tal de poder ser capaços d'analitzar correctament les mètriques.

- La mesura de fitness és relativa a la mateixa generació. És a dir, la fitness d'un individu no depèn només del seu rendiment, sinó que depèn del rendiment amb relació als altres. Per exemple, dos individus idèntics podrien rebre una fitness molt diferent en diferents generacions, ja que els enfrontaments serien contra individus diferents, que poden estar més avançats que ells o menys. És per això que és normal veure pujades i baixades de la fitness al llarg de les generacions, perquè molts cops apareix un individu que despunta respecte a la resta, però a la següent generació la resta d'individus milloren i, en relació, la fitness del primer individu baixa.

- El procés de mutació i recombinació genètica són erràtics. No sempre milloren als individus. Això fa que els valors de les mètriques tinguin molt soroll, canvien molt de generació en generació, tant cap a bé com a mal. És per això que l'algorisme es va regulant per anar eliminant a poc a poc aquells individus que no han presentat una millora respecte als antecessors. Això fa que sí que es pugui deduir una tendència general de les mètriques.

Reintroducció d'individus En les figures Fig.10 i Fig.11 de l'apèndix podem veure com canvien les mètriques si utilitzem l'estratègia de reintroducció d'individus. Aquesta estratègia consisteix a afegir a cada generació un campió aleatori de generacions passades. S'utilitza per no perdre massa qualitat genètica degut a la mutació o a males recombinacions. És útil , ja que molts cops la mutació esguerra als descendents dels individus més preparats, de manera que es perd l'estratègia a la qual representava l'individu. Reintroduint a la població a individus que ja han demostrat tenir un comportament adequat ens assegura que, com a mínim la qualitat dels individus es mantengui durant el temps. Com es pot apreciar la reintroducció d'individus té beneficis clar en la fitness i en el nombre d'il·legalitats (violations) que cometen els individus. En la figura Fig.11 podem veure com la tendència a l'alça de la fitness es manté de la mateixa forma que ho fa la tendència a la baixa de les il·legalitats. Aquest comportament pot ser associat al fet que tot i els processos erràtics de la mutació i la combinació de genomes (erràtics perquè no sempre suposen un pas endavant en el procés d'adaptació) sempre queden individus adaptats, gràcies a la reintroducció. En el cas de la Fig.10 podem veure que la causa de la davallada final en la fitness és deguda que els individus han perdut aquella part del genoma que els feia respectar les normes i evitar cometre il·legalitats, de manera que les infraccions castiguen greument la fitness dels individus. Amb la reintroducció d'un individu que contingui aquests gens, podríem recuperar aquesta capacitat en els individus.

Valor d'adaptació global El valor d'adaptació global o fitness global és el resultat de comparar els resultats de cada generació. De la mateixa manera que la fitness és una mesura dels individus respecte a la seva generació, la fitness global és la mesura del rendiment entre generacions. Aquesta mètrica ha de ser necessàriament positiva, doncs la millora del rendiment amb el pas de les generacions és una propietat inherent de l'evolució.

En la Fig.7 podem veure com el model genera individus millor o iguals d'adaptats que l'anterior en la gran majoria dels casos. També es pot observar com, a mesura que el nombre de generacions augmenta, el model té dificultats per mantenir el rendiment.
En l'annex la figura 9 mostra aquesta mètrica (victòries+empats) per a tres execucions diferents. Podem veure que es mantenen els resultats esperats en totes elles.

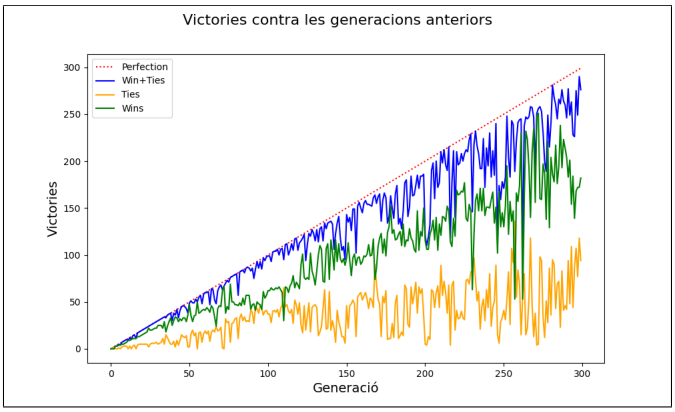

Fig. 7: Resultats de fer combatre el nou campió contra tots els anteriors durant 300 generacions. En vermell tenim la progressió ideal que pot assolir el model. Aquest ideal correspon a guanyar a tots els campions de les generacions anteriors. En blau tenim el total de victòries i empats assolits al enfrontar a tots els campions anteriors. En verd i groc tenim el nombre de victòries i empats respectivament.

Restricció d'il·legalitats Les il·legalitats, com s'ha explicat en el punt anterior, no són massa rellevants en el cas d'un problema com el que es tracta, ja que es poden impedir de forma explícita. Tot i així, si volem que els individus també desenvolupin aquesta lògica en el seu raonament, podem aplicar la política de penalització que farà que aquells individus que es saltin les restriccions, vegin davallades les seves possibilitats de perpetuar els seus gens a les següents generacions. En la Fig.8 podem veure com seria la mètrica d'il·legalitats sense restriccions en contra posició de la mètrica que s'obté en una execució amb restriccions com podrien ser les de les figures Fig.10 i Fig.11.

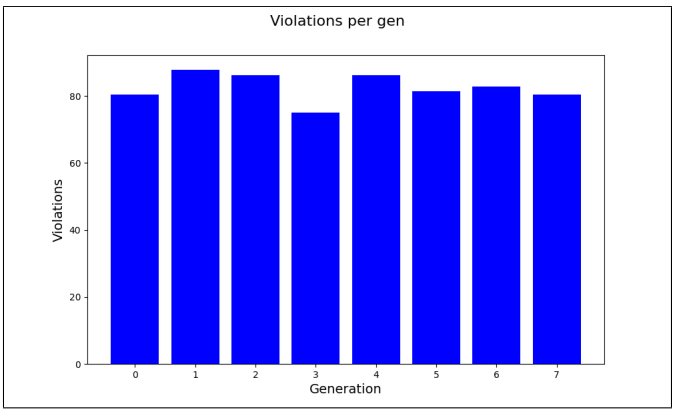

Fig. 8: Mètrica de il·legalitats per a una execució sense restriccions a il·legalitats

Cost computacional El cost computacional és una mètrica important en el cas de les NEAT, ja que la reducció d'aquest és un dels objectius principals que va impulsar el naixement d'aquest algorisme. En el cas que del disseny de comportaments per a videojocs, aquest aspecte no és tan rellevant. És així, perquè normalment hi ha prou temps per a prendre la decisió. El mateix passa en el joc que ens ocupa, doncs l'estat de la partida canvia cada segon. Així doncs hi ha un segon per a prendre la decisió. L'algorisme desenvolupat pren les decisions molt ràpidament doncs les xarxes dels individus no arriben a tenir una mida prou gran per a arribar a tenir un temps d'execució d'un segon.
En aquest apartat també cal mencionar el cost de l'entrenament. En el cas dels videojocs el temps d'execució de les partides normalment està acotat, doncs les partides poden tenir una duració concreta. En aquest cas les partides es duen a terme, com a màxim, durant 12 segons, que realment són partides de 120 segons a velocitat x10. Com la limitació de la computació no és el hardware, sinó que és la duració de les partides, la mètrica de cost computacional perd valor, ja que la duració de l'entrenament depèn exclusivament del nombre de generacions que es vulguin fer, del nombre d'individus a les poblacions i del nombre d'espècies que es creïn (ja que determinarà si es juguen més o menys partides). Per a un total de 400 generacions de 64 individus, l'algorisme s'ha executat durant 5 hores. Això ens deixa amb 400 campions, individus considerats possibles solucions.

Valoració subjectiva Les mètriques que podem extreure no ens permeten detectar si les estratègies desenvolupades poden ser considerades ïntel·ligents". És per això que cal agafar els individus amb més fitness (els que tenien un avantatge clara respecte als contrincants) i comprovar manualment el seu funcionament. Jugant partides contra aquests individus, podem afirmar que alguns individus sí que desenvolupen lògiques avançades, i n'hi ha d'altres que només van desenvolupar fitness perquè no s'havien trobat cap estratègia que els pogués guanyar. En aquest últim cas, un humà pot perfectament trobar la manera de guanyar l'estratègia que presenta l'individu. Una de les lògiques més avançades que s'han pogut trobar en alguns individus són:

- Contraatac: Hi ha una unitat molt ràpida que arribava a la base enemiga i tenia temps de fer mal a la torre enemiga abans que aquesta la destruís. Alguns individus aprofitaven que l'enemic no tenia or per a tirar una unitat d'aquest tipus i aconseguir baixar una mica la vida de la torre de l'enemic.
- DPS: Alguns individus s'adonen que la unitat "arquera"és la que més mal per segon produeixen, a canvi de la poca vida que tenen. Si acumules prou arqueres ets capaç de destruir casi qualsevol unitat del joc abans que aquesta et pugui arribar a tocar les arqueres (a causa del rang). D'aquesta manera aconseguir acumular-ne fa que atacar la torre rival sigui molt senzill, dons les unitats que crea el contrari no poden parar el teu grup d'arqueres.
- Explosió: Hi ha una habilitat amb un cost d'or elevat que fa mal a totes les unitats del terreny de joc, com una explosió. Concretament, fa prou mal per a destruir a la unitat "arquera". Alguns individus han desenvolupat la capacitat de detectar quan el rival ha acumulat arqueres (estratègia anterior) i en contraatac tiren aquesta explosió que permet destruir-les instantàniament a totes a la vegada i per un cost inferior al que ha gastat el contrari per crear les arqueres.
- Distracció: Alguns individus guanyaven a base de llençar una arquera i anar tirant unitats barates davant

seu. Com les arqueres caminen lent, les unitats barates l'avancen de seguida, de manera que les unitats enemigues es "distreien"pegant a les unitats barates mentre l'arquera podia anar atacant des de lluny i obrir pas a poc a poc.

Malgrat això no s'ha trobat cap individu que combini totes les estratègies i sàpiga quan utilitzar una o altre depenent del context. Tot i així, aquestes estratègies són suficients per a vèncer a humans que no tinguin massa experiència en aquest joc i per a empatar amb els que en saben més.

7 Conclusions

Per concloure cal comentar que l'avaluació i resultats d'un algorisme NEAT és molt dependent del tipus d'entorn al que el sotmets. Fins al punt que si el joc no està ben dissenyat, l'algorisme aprèn a guanyar les partides utilitzant aquells elements mal dissenyats, com podria ser una tropa que és prou forta i barata com perquè si la tires tota l'estona l'enemic no té temps a guanyar-te i acaba la partida en empat. Això demostra que els algorismes NEAT poden ser utilitzats per a tasques de testeig de jocs, on es comprovi que no hi ha cap estratègia "poc elaborada"de guanyar.

Agraïments

Referències

[1] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, Jun. 2002. [Online]. Available: https://direct.mit.edu/evco/article/10/2/99-127/1123

[2] "NEAT Robot Competitive Coevolution Demo." [Online]. Available: http://nn.cs.utexas.edu/pages/research/neatdemo.html

[3] "Gym Documentation." [Online]. Available: https://www.gymlibrary.ml/

[4] "MarI/O - Machine Learning for Video Games." [Online]. Available: https://www.youtube.com/watch?v=qv6UVOQ0F44

[5] F. J. Gomez and R. Miikkulainen, "Solving Non-Markovian Control Tasks with Neuroevolution."

[6] J. Lowell, K. Birger, and S. Grabkovsky, "Comparison of NEAT and HyperNEAT on a Strategic Decision-Making Problem."

[7] T. Bolander, "What is AI – and where is it heading? Part II: Symbolic and subsymbolic AI."

[8] "3 NeuroEvolution of Augmenting Topologies (NEAT)." [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume21/stanley04a-html/node3.html

[9] E. J. Carmona Suarez and S. Fernandez Galan, *Fundamentos de la computación evolutiva*, 2019th ed. S.A. MARCOMBO.

[10] K. O. Stanley, "Efficient Evolution of Neural Networks through Complexification," p. 180.

[11] "NeuroEvolution of Augmenting Topologies." [Online]. Available: https://www.cs.ucf.edu/ kstanley/neat.html

[12] "Coevolution - Understanding Evolution," Mar. 2021. [Online]. Available: https://evolution.berkeley.edu/evolution-101/mechanisms-the-processes-of-evolution/coevolution/

[13] K. O. Stanley and R. Miikkulainen, "Competitive Coevolution through Evolutionary Complexification," *Journal of Artificial Intelligence Research*, vol. 21, pp. 63–100, Feb. 2004. [Online]. Available: https://jair.org/index.php/jair/article/view/10367

APÈNDIX

A.1 Gràfiques d'execucions

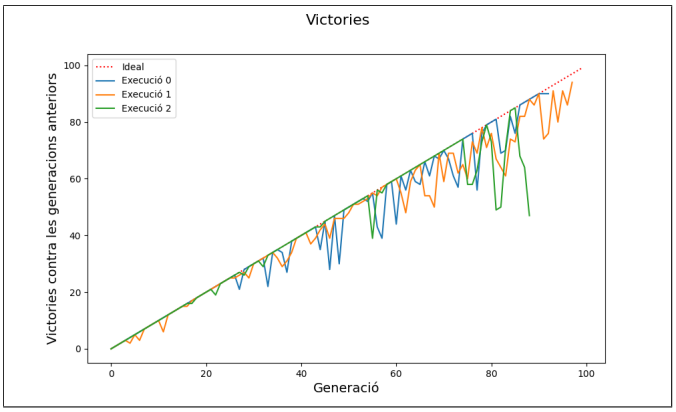

Fig. 9: Mètrica de Global Fitness per a 3 execucions diferents

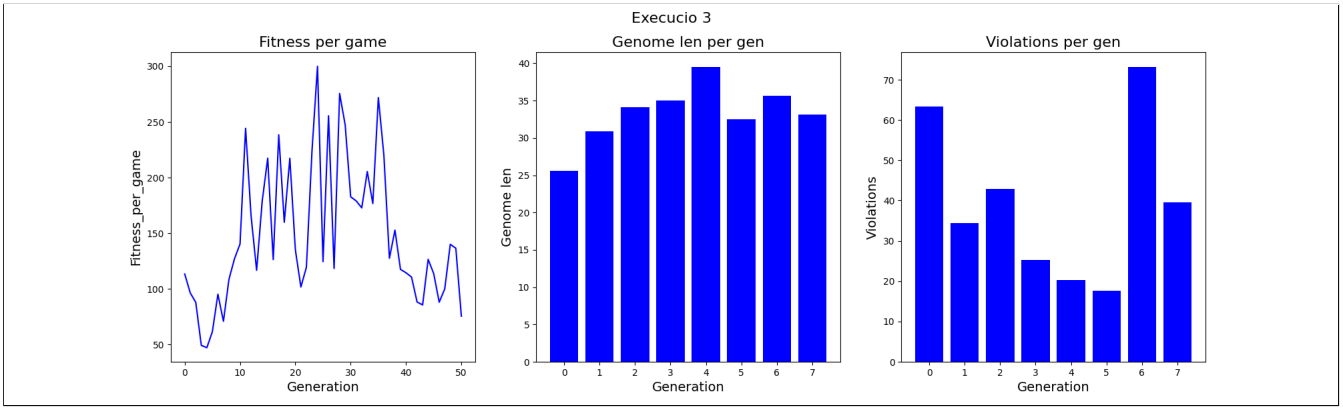

Fig. 10: Gràfica que exposa la fitness per partida, la longitud del genoma i el nombre d'il·legalitats dels campions de cada generació. El eix X representa el nombre de generacions agrupades, es a dir, s'agrupen les generacions en grups i es fa una mitjana de la mètrica per evitar soroll

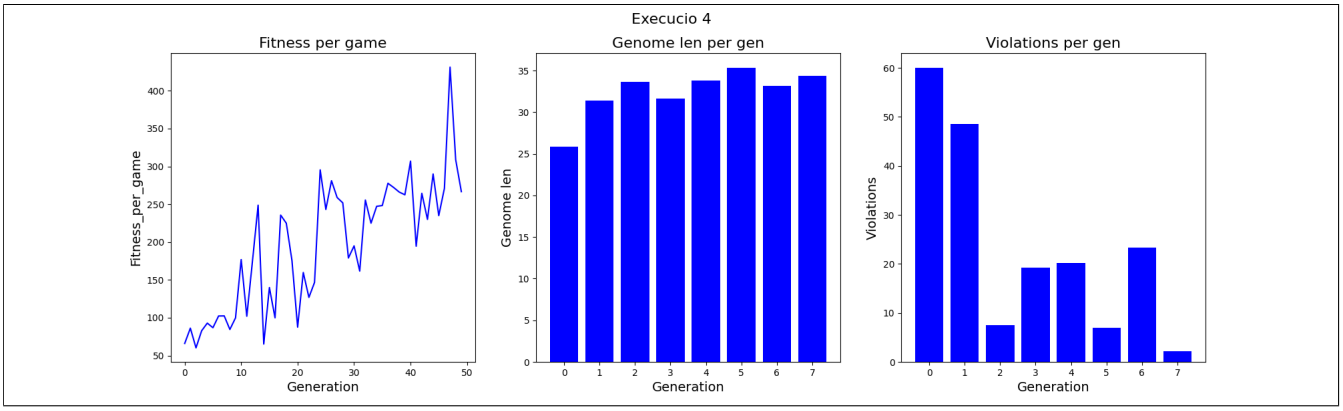

Fig. 11: Aquesta figura presenta les mètriques obtingues al executar l'algorisme amb reintroducció d'individus.