

ALGORITMOS GENÉTICOS Y EVOLUTIVOS

PRÁCTICA 2

(Calibración de motores automática mediante
Estrategias Evolutivas)



Grado en Ingeniería Informática

Campus de Colmenarejo

Curso 2020/2021

Autor

Eduardo Ureña Toledano - 100329937@alumnos.uc3m.es

Índice

1. Introducción	3
2. Diagrama de la Estrategia Evolutiva	3
3. Parámetros propuestos y análisis comparativo de los resultados	4
3.1. Primera situación (4 rotores)	4
3.2. Segunda situación (10 rotores)	7
4. Mejorando la Estrategia Evolutiva	12
4.1. Primera situación (4 rotores)	12
4.2. Segunda situación (10 rotores)	14
4.3. Tercera situación (6 rotores)	15
5. Conclusiones y problemas encontrados	17

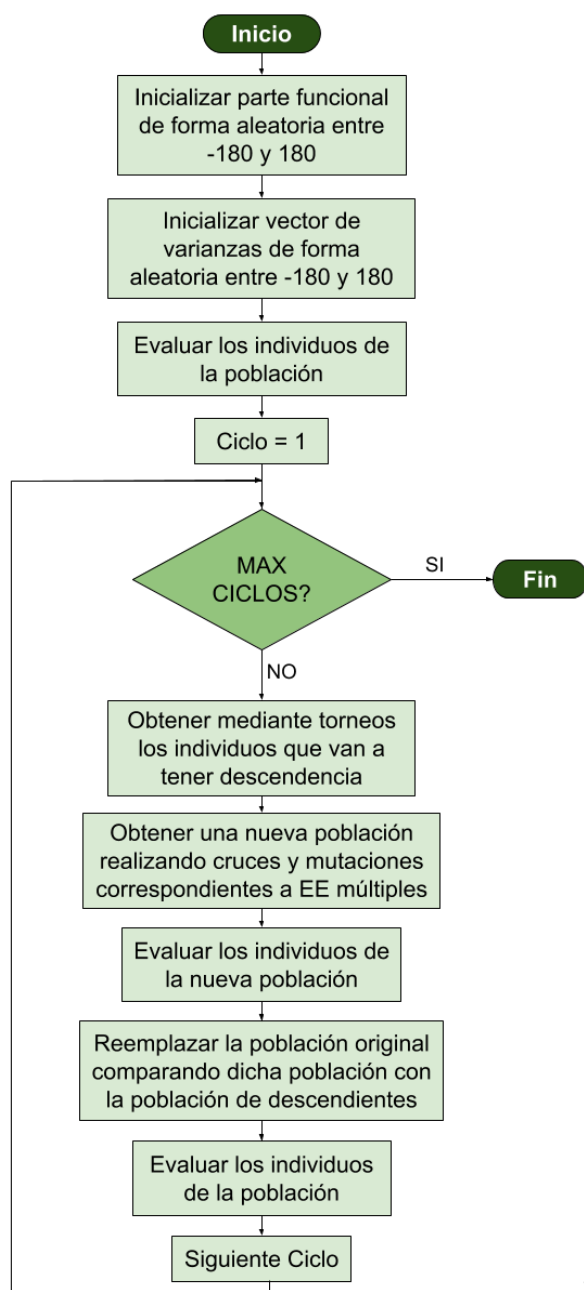
1. Introducción

En este documento se expondrán los resultados obtenidos mediante Estrategias Evolutivas en la tarea de buscar una combinación óptima de giros de los rotores de un brazo robot, para conseguir que suelde de la forma más precisa posible.

En mi caso, empecé desarrollado un modelo $(\mu + \lambda)$ -EE, pero más adelante se puede apreciar que hago un cambio en el reemplazo por “inclusión” para conseguir un mejor funcionamiento del algoritmo, convirtiéndolo en un “híbrido” entre $(\mu + \lambda)$ -EE y (μ, λ) -EE.

En el apartado llamado “Mejorando la Estrategia Evolutiva” hablaré de la causa por la que he decidido cambiar el reemplazo por inclusión por un “híbrido” entre inclusión e inserción.

2. Diagrama de la Estrategia Evolutiva



3. Parámetros propuestos y análisis comparativo de los resultados

Para probar la Estrategia Evolutiva es necesario tener en cuenta dos situaciones:

1) El brazo tiene 4 rotores, y para obtener el fitness de cada individuo es necesario llamar al siguiente servicio web:

<http://memento.evannai.inf.uc3m.es/age/robot4?c1=<x1>&c2=<x2>&c3=<x3>&c4=<x4>>.

2) El brazo tiene 10 rotores, y para obtener el fitness de cada individuo es necesario llamar al siguiente servicio web:

<http://memento.evannai.inf.uc3m.es/age/robot10?c1=<x1>&c2=<x2>&c3=<x3>&c4=<x4>&c5=<x5>&c6=<x6>&c7=<x7>&c8=<x8>&c9=<x9>&c10=<x10>>.

Cabe destacar que las pruebas de ambas situaciones están realizadas usando “inclusión” como estrategia de reemplazo, siguiendo de esa forma el procedimiento de $(\mu + \lambda)$ -EE.

3.1. Primera situación (4 rotores)

En esta primera situación, he decidido hacer pruebas con una poblaciones de 100, 80, 70 y 50, y además he decidido seleccionar en cada torneo a 3 individuos, por lo que para 100, 80 y 70 será necesario usar un 3% de porcentaje de torneo, mientras que para 50 un 4%.

La razón por la que he decidido seleccionar a 3 individuos es, que anteriormente a dicha decisión realicé unas cuantas pruebas con otras proporciones, y los resultados eran pésimos, encontrando fitness que nunca bajaban de las decenas.

En la siguiente tabla se pueden observar los resultados obtenidos al hacer cuatro pruebas con cada combinación de parámetros:

Tamaño de población	Porcentaje de torneo	Fitness conseguido	Valores de giro	Evaluaciones necesarias
100	3%	1.11867275347	c1 = 10.124994851716608 c2 = 6.009943576140707 c3 = 3.6745430538783364 c4 = 2.7683389759597876	679750
		0.999106039149	c1 = 10.12215163116594 c2 = 5.991016432745084 c3 = 3.6691827641522545 c4 = 2.7616400116886366	143050
		0.994959124749	c1 = 10.123456892332577 c2 = 5.987636419028674 c3 = 3.6666707195327883 c4 = 2.760393843892305	182800

100	3%	0.495122218797	c1 = 10.153688635595001 c2 = 5.960417708955077 c3 = 3.695548961503733 c4 = 1.761899941126129	1078450
80	3%	2.86853818096	c1 = 10.122020022986899 c2 = 5.931548481190543 c3 = 3.693537048619376 c4 = 2.684599608314147	6350
		1.98997054835	c1=10.123105506174074 c2=4.992804709044901 c3=4.661853423793054 c4=1.7651552829010493	23000
		1.98995230193	c1=11.118597684838868 c2=4.992397978774472 c3=3.666584240106503 c4=1.7652222357089282	32900
		1.98995756386	c1=10.123240472931295 c2=4.992477028478744 c3=2.6713905938811764 c4=1.7653679628321275	16850
70	3%	0.99497910774	c1=10.123610459997169 c2=5.987392351366129 c3=2.6717421744412215 c4=1.7655203027128734	17290
		0.995617798303	c1=11.117382725310279 c2=5.986171282555665 c3=3.666873294309828 c4=1.765317508932813	15505
		0.9949591266	c1=10.123451670709176 c2=5.9876556543737385 c3=4.661640245523627 c4=1.765440927931203	21175
		0.0	c1=10.123456785843166 c2=5.987653999994307 c3=3.666666003631859 c4=1.7654320972864852	38605
50	4%	0.995068558319	c1=10.123466313420467 c2=5.987162768622403 c3=3.66632922322594 c4=2.760834831053203	19160
		2.98488346945	c1=9.128488124560834 c2=5.987741599533371 c3=4.661470392002577 c4=2.7604046088799765	26600

50	4%	0.994959057116	c1=10.123456771041612 c2=5.987654026025418 c3=3.6666659930154553 c4=0.7704737988199277	27920
		1.98991811419	c1=10.123456758196417 c2=4.9926953105794425 c3=2.6717075182229046 c4=1.7654320549697131	30560

Como se puede observar en la tabla, con todas las combinaciones se encuentran mínimos locales, pero solo se encuentra una vez la solución óptima (0.0), con una población de 70 individuos y un 3% en los torneos, aunque con una población de 100 se podría haber encontrado también una vez, ya que el algoritmo llegó a una solución con un “0.495122218797” de fitness, y a partir de ahí solo podía seguir bajando, ya que había sorteado todos los mínimos locales.

Para determinar qué combinación es mejor en relación fitness/nº evaluaciones, he decidido hacer la media del fitness y las evaluaciones de las cuatro ejecuciones de cada combinación, obteniendo los siguientes resultados:

Tamaño de población	Porcentaje de torneo	Media de fitness	Media de evaluaciones
100	3%	0,9019650340	521012
80	3%	2,2096046488	19775
70	3%	0,7463890082	23144
50	4%	1,7412072998	26060

Se puede observar que en cuanto a la media de fitness el mejor resultado se obtiene con una población de 70 individuos, y en cuanto a la media de evaluaciones la mejor media se encuentra con 80 individuos, por lo que si se busca encontrar el menor fitness en el menor número de evaluaciones, ambas serían soluciones no dominadas, pero teniendo en cuenta que con 80 individuos se encuentra la peor solución en cuanto a la media de fitness, concluyo que la mejor solución se obtiene con 70 individuos.

3.2. Segunda situación (10 rotores)

En la segunda situación, he decidido hacer pruebas con una poblaciones de 80, 70, 60 y 50, y he decidido seleccionar en cada torneo a 3 individuos igual que en la primera situación, de forma que para 80 y 70 será necesario usar un 3% de porcentaje de torneo, mientras que para 60 y 50 será necesario usar un 4%.

En este caso he decidido hacer cinco pruebas para cada combinación de parámetros en vez de cuatro, debido a que los mejores fitness encontrados en cada ejecución eran más irregulares con 10 rotores que con 4.

En la siguiente tabla se pueden observar los resultados obtenidos al hacer cinco pruebas con cada combinación de parámetros:

Tamaño de población	Porcentaje de torneo	Fitness conseguido	Valores de giro	Evaluaciones necesarias
80	3%	6.96819499649	c1=2.228239292081948 c2=2.491747357888854 c3=1.3235763391837512 c4=1.7639079182685564 c5=1.4259474532210883 c6=0.0026559157834314495 c7=0.34359913859611024 c8=0.9333283980750738 c9=0.23469495123127093 c10=5.926642626283759	75920
		4.97683768054	c1=3.223759512443016 c2=2.4943415851101713 c3=1.324567006172623 c4=2.7595226956567234 c5=-0.5632700396169417 c6=-0.0009403227554096538 c7=1.338169737029055 c8=0.9344705262374859 c9=-0.7626059711246626 c10=3.937509081658782	59960
		9.94959456279	c1=3.2234068057891414 c2=3.4875989721518903 c3=-0.6652017496309449 c4=3.755219696048969 c5=-0.562465083807261 c6=-0.9949578051009803 c7=0.3432843387101923 c8=0.9344186472208147 c9=0.23465458478857418 c10=3.9381569743278235	54800

80	3%	1.98991825959	c1=3.223460103637863 c2=3.4876532614156677 c3=1.32465457338642 c4=2.760403133930968 c5=0.43244050188845473 c6=1.0116230176667688e-05 c7=0.3432006694512506 c8=0.9345527964435524 c9=0.23453926883781298 c10=2.943266760617711	69800
		2.98487717129	c1=3.2234568821558502 c2=3.487653910109382 c3=1.3246629418617968 c4=1.7654320715492993 c5=1.4274020994777468 c6=-1.1455565634472615e-07 c7=-0.6517586901032534 c8=0.9345435884786392 c9=0.23453008074139156 c10=4.93319874243042	80480
70	3%	56.7189045837	c1=4.218714408876345 c2=3.485888505833632 c3=1.3273015364814749 c4=0.7719015122290109 c5=-97.41940913808502 c6=100.9446847949526 c7=-27.43648188567309 c8=-117.37165994430897 c9=105.54758010874406 c10=-4.1936053866623695	37870
		9.94965630412	c1=5.213610238274539 c2=3.48791551026291 c3=2.3196281682294084 c4=2.7603532459947213 c5=-7.7434465316348735 c6=43.48814664510135 c7=-12.477338763552893 c8=-10.579283553141039 c9=113.67813042566996 c10=29.428675161880278	69055
		13.9935031759	c1=3.223441423460752 c2=2.4926479858787305 c3=2.319598909534842 c4=1.7474503843063678 c5=0.432528500523326 c6=-0.9948557956888966 c7=1.3381104968998294 c8=3.919383631774141 c9=-0.7607565885305287 c10=3.938226146544176	86590

70	3%	5.96997033357	c1=4.218904722788657 c2=3.4872251164410106 c3=2.3200347212367256 c4=2.760502374934988 c5=-0.5619951900173409 c6=-0.995026259748714 c7=0.34346316133380717 c8=0.9342025123970219 c9=0.2346725355631651 c10=2.9433234054663155	70630
		2.98989676037	c1=3.223388671084492 c2=2.491889829785249 c3=1.3242730296539464 c4=1.7703527811488766 c5=0.432293086979842 c6=0.9951024087550068 c7=-0.6515684346469766 c8=0.9343844446224729 c9=0.2345258378785377 c10=3.938664413626798	72835
60	4%	9.94989666128	c1=3.224071346574379 c2=3.48763252662592 c3=1.3246322674087252 c4=1.7645018000442676 c5=0.43212262300275467 c6=-0.9950359081163969 c7=0.3428891612618782 c8=-0.060484337355094904 c9=-1.755754906922494 c10=1.9483138228527124	51090
		9.94981832741	c1=4.218977709781412 c2=3.4880881567501727 c3=0.32972858913961905 c4=0.7704598477278257 c5=1.4278970136220654 c6=0.00023449036905553316 c7=-0.6520048300235601 c8=0.9343926305297319 c9=-0.7599881798786216 c10=5.927850198851127	56580
		60.6980588479	c1=2.2290570204819553 c2=1.4971653489650363 c3=1.324336687662315 c4=0.7701188610364006 c5=0.4357136499526267 c6=-0.9907623879409544 c7=1.3383530166934245 c8=0.935077901432054 c9=-6.730996991917356 c10=5.929711033489647	46320

60	4%	3.99476912073	c1=3.2225410705497066 c2=3.4882445938313165 c3=2.3209521903274215 c4=1.766339204762454 c5=-0.5638730191772041 c6=0.0010973207670851473 c7=0.35131609430366356 c8=-0.061358230539799184 c9=1.2307914731659164 c10=3.9382224916252833	99150
		6.97624802131	c1=3.2235887972035346 c2=5.48112952201042 c3=2.317924512713346 c4=1.764152567827974 c5=0.4317439812354547 c6=1.0009710616505199 c7=0.34374677415239263 c8=0.9360053110177317 c9=0.2346680003517037 c10=2.9419134595950807	65760
50	4%	5.97825640175	c1=4.21840688360702 c2=4.483675997007005 c3=2.3180871477506755 c4=1.7678889460554306 c5=0.432018227900562 c6=-0.9942128305475553 c7=0.33872406427773144 c8=0.9378912103349034 c9=-0.7599301090150595 c10=4.934252789522663	61475
		8.95488922091	c1=3.223820233340166 c2=2.4927708310488472 c3=3.3142755252066447 c4=1.766474789847486 c5=0.4325187178294908 c6=-0.9949426232799493 c7=0.3431802274865185 c8=-0.0604229430865126 c9=1.2294306460595879 c10=4.93319559578198	65525
		41.7973459553	c1=2.2337560279473134 c2=9.457233322451753 c3=0.32914702092545833 c4=2.7582563349989497 c5=-0.5620128223627878 c6=0.0021069575189225894 c7=0.3422846266772086 c8=-0.06199661342693022 c9=-0.7630836134673774 c10=3.9391504638327812	33425

50	4%	3.97995557481	c1=3.2236210627156177 c2=4.4826496523092985 c3=1.324658067471669 c4=2.7602539553706977 c5=0.4326798213405876 c6=0.9953142454652256 c7=-0.6520141562972636 c8=0.9340565030317876 c9=0.23463916344508776 c10=3.938480563275772	65750
		1.98999722791	c1=3.2228421053313645 c2=3.487678798247198 c3=2.319610782349675 c4=1.7654932682281097 c5=0.432353065570126 c6=-7.823543996701907e-05 c7=1.3381512572858165 c8=0.9345531024507512 c9=0.23454683959756098 c10=3.9381981906030057	59300

Como se puede observar en la tabla, con 10 rotores los mejores fitness encontrados son peores que con 4, y en ocasiones se obtienen fitness bastante altos, siendo los menores fitness encontrados 1.98991825959 con una población de 80 y 1.98999722791 con una población de 50, estando ambos cerca de un mínimo local, por lo que habrían llegado a ese mínimo en caso de haber realizado algunas evaluaciones más.

Para determinar qué combinación es mejor en relación fitness/nº evaluaciones, hago la media del fitness y las evaluaciones de las cinco ejecuciones de cada combinación, obteniendo los siguientes resultados:

Tamaño de población	Porcentaje de torneo	Media de fitness	Media de evaluaciones
80	3%	5,37388453414	68192
70	3%	17,92438623153	67396
60	4%	18,31375819573	63780
50	4%	12,54008887614	57095

Se observa que en cuanto a la media de fitness el mejor resultado se obtiene con 80 individuos, pero en cuanto a la media de evaluaciones la mejor media se encuentra con una población de 50 individuos, por lo que si se busca encontrar el menor fitness en el menor número de evaluaciones, ambas serían soluciones no dominadas, pero pienso que el fitness es más importante que el número de evaluaciones, y por tanto la mejor solución se encuentra con 80 individuos.

4. Mejorando la Estrategia Evolutiva

Cómo con la Estrategia Evolutiva $(\mu + \lambda)$ -EE no he conseguido que con 4 rotores se encuentre la solución óptima sistemáticamente (con 10 rotores considero que es más complicado llegar a dicha solución), necesito mejorar de alguna manera el algoritmo (o ver si tiene algún fallo de programación) para que sea capaz de sortear los mínimos locales más fácilmente, y por tanto llegar a 0.0 en casi todas las ejecuciones del programa.

Después de muchos intentos sin éxito de encontrar algún fallo en el algoritmo, me doy cuenta que dicho fallo no existe, y decido probar suerte programando una $(1+1)$ -EE.

Con dicha $(1+1)$ -EE obtengo resultados aún peores, ya que ejecuto el programa muchas veces, y tan sólo consigo llegar a un mínimo local una vez, obteniendo en las demás ejecuciones fitness del orden de las decenas y alguno algo más bajo pero bastante superior que con la Estrategia Evolutiva $(\mu + \lambda)$ -EE.

Intento encontrar algún error de programación en este algoritmo, pero dicho error tampoco existe, por lo que decido mejorar de alguna manera la Estrategia Evolutiva $(\mu + \lambda)$ -EE.

Después de mucho pensar, me doy cuenta que la única manera de mejorar el funcionamiento del programa es, llegado a la fase del reemplazo, comparar la población de descendientes con la población actual, en lugar de compararla con la población de progenitores, de manera que nunca se pierda la mejor solución encontrada hasta el momento, obteniendo un mejor funcionamiento de la Estrategia Evolutiva para este problema concreto, usando un híbrido de $(\mu + \lambda)$ -EE y (μ, λ) -EE.

Para demostrar que el algoritmo funciona de forma más satisfactoria después de realizar la modificación mencionada anteriormente, hago cinco pruebas para cada número diferente de rotores, con la población con la que se han obtenido mejores resultados en cada situación, y añadiendo en esta ocasión una tercera situación con 6 rotores, buscando llegar en esta última al menos una vez a un fitness de 0.0 para concluir que el funcionamiento del algoritmo es el esperado, además de llegar casi siempre a 0.0 en la situación con 4 rotores.

4.1. Primera situación (4 rotores)

En esta primera situación he utilizado una población de 70 individuos, ya que ha sido el número de individuos por población que mejores resultados ha dado en la Estrategia Evolutiva $(\mu + \lambda)$ -EE para el calibrado de los giros de 4 rotores.

En la siguiente tabla se muestran los resultados de las cinco pruebas realizadas para 4 rotores, pudiéndose observar que se consigue un fitness de 0.0 en cuatro de las cinco ejecuciones, obteniéndose en la quinta el mínimo local más bajo "0.994959057093".

Tamaño de población	Porcentaje de torneo	Fitness conseguido	Valores de giro	Evaluaciones necesarias
70	3%	0.0	c1=10.123456788828127 c2=5.98765399595069 c3=3.6666660015058734 c4=1.7654320984908094	31675
		0.0	c1=10.123456789303342 c2=5.987653998162166 c3=3.6666659992078756 c4=1.765432103182394	27895
		0.0	c1=10.123456788861592 c2=5.987653997309762 c3=3.6666659972750186 c4=1.765432097690059	24745
		0.0	c1=10.123456788916611 c2=5.9876539978534975 c3=3.6666660030509353 c4=1.7654320963935386	23380
		0.994959057093	c1=9.128498135294258 c2=5.987653994581602 c3=3.666666013075087 c4=1.7654320794311995	30205

En la situación con 4 rotores se puede apreciar que el algoritmo ha mejorado considerablemente, al menos para dicho número de rotores, ya que llega a un fitness de 0.0 en casi todas las ejecuciones.

En la siguiente tabla se puede observar la diferencia entre la Estrategia Evolutiva $(\mu + \lambda)$ -EE y la híbrida, en cuanto a la media de fitness y evaluaciones:

Estrategia Evolutiva	Media de fitness	Media de evaluaciones
$(\mu + \lambda)$ -EE	0,7463890082	23144
Híbrida	0,198991811419	27580

La media del mejor fitness encontrado es bastante inferior en la Estrategia Evolutiva híbrida que en la $(\mu + \lambda)$ -EE.

4.2. Segunda situación (10 rotores)

En esta segunda situación he utilizado una población de 80 individuos, ya que ha sido el número de individuos por población que mejores resultados ha dado en la Estrategia Evolutiva ($\mu + \lambda$)-EE para el calibrado de los giros de 10 rotores.

En la siguiente tabla se muestran los resultados de las cinco pruebas realizadas para 10 rotores, pudiéndose observar que no se obtiene un fitness de 0.0 en ninguna ejecución, como era de esperar, pero se queda en mínimos locales relativamente bajos en la mayoría de las ejecuciones, siendo el mínimo fitness obtenido “1.98991811419”.

Tamaño de población	Porcentaje de torneo	Fitness conseguido	Valores de giro	Evaluaciones necesarias
80	3%	4.97479528547	c1=3.2234568652930453 c2=3.487653929646456 c3=2.3196216574428568 c4=0.770473495069008 c5=0.43244336638804465 c6=-1.090504490109324e-07 c7=1.3381585349610647 c8=0.9345435284746034 c9=1.2294886644299485 c10=2.943281334447301	57920
		4.97479528547	c1=4.21841541396528 c2=4.482612758986305 c3=2.3196216803575527 c4=0.7704734360219413 c5=0.4324433328156958 c6=0.9949586581485128 c7=0.34320002191909854 c8=0.9345435046897972 c9=0.23453006895958237 c10=3.938239923536599	62720
		4.97479528547	c1=4.218415428821339 c2=3.4876540530410094 c3=2.3196217239522245 c4=2.760390642287867 c5=0.4324433485129968 c6=-4.824118161967568e-08 c7=1.3381585767519983 c8=0.9345435016304685 c9=0.23452990284897554 c10=2.9432813435989336	66080

80	3%	7.95966741893	c1=1.2335446242725652 c2=3.4876540120475856 c3=1.3246630109121573 c4=1.7654321276228373 c5=-0.5625152115834585 c6=3.1660880697097464e-08 c7=-0.6517585126460536 c8=1.9295020414692508 c9=0.23452998400529718, c10=2.9432813872290837	72560
		1.98991811419	c1=3.223456819206132 c2=3.487653927887532 c3=1.3246630223625477 c4=1.7654322416126915 c5=0.4324434414446551 c6=-1.2411172435380884e-08 c7=1.3381586699440557 c8=0.9345435093906352 c9=0.2345300000993948 c10=2.943281453628253	88280

En la situación con 10 rotores no se puede apreciar a simple vista si el algoritmo ha mejorado, por lo que comparo la Estrategia Evolutiva $(\mu + \lambda)$ -EE con la híbrida, mirando la media de fitness y evaluaciones para una población de 80 individuos.

Estrategia Evolutiva	Media de fitness	Media de evaluaciones
$(\mu + \lambda)$ -EE	5,37388453414	68192
Híbrida	4,97479427791	69512

Como se puede apreciar, la media del mejor fitness obtenido para cinco ejecuciones con una población de 80 individuos es inferior en la Estrategia Evolutiva híbrida que en la $(\mu + \lambda)$ -EE, por lo que no podemos negar que la híbrida sea mejor que la $(\mu + \lambda)$ -EE para este problema concreto.

4.3. Tercera situación (6 rotores)

Por último, y no pudiendo haber negado aún la hipótesis de que la Estrategia Evolutiva híbrida es mejor que la $(\mu + \lambda)$ -EE, procedo a realizar las pruebas para la tercera situación, en la que he utilizado una población de 100 individuos para las cinco ejecuciones, ya que es

la población con la que a he observado que se obtenían mejores resultados, habiendo puesto en ejecución varios programas en paralelo con diferentes poblaciones.

En la siguiente tabla se muestran los resultados de las cinco pruebas realizadas para 6 rotores, pudiéndose observar que no sólo se obtiene un fitness de 0.0 al menos en una ocasión, sino que se obtiene en tres ejecuciones de las cinco, obteniéndose en las otras dos un fitness de “0.994959057093”, siendo este el mínimo local más bajo, por lo que se puede concluir que la Estrategia Evolutiva híbrida es mejor que la $(\mu + \lambda)$ -EE.

Tamaño de población	Porcentaje de torneo	Fitness conseguido	Valores de giro	Evaluaciones necesarias
100	3%	0.0	c1=3.223456789064152 c2=2.487654000882589 c3=2.324663002875667 c4=1.7654320950713411 c5=1.4324434350072017 c6=3.543545654557589	47950
		0.0	c1=3.223456788344471 c2=2.487654003660923 c3=2.3246629997651516 c4=1.765432104877582 c5=1.4324434282144254 c6=3.5435456458588344	74650
		0.0	c1=3.2234567886732144 c2=2.487653998281701 c3=2.324663003200251 c4=1.7654320985010206 c5=1.4324434306418485 c6=3.5435456455921135	65200
		0.994959057093	c1=3.223456784368186 c2=2.4876539859854714 c3=2.3246629851139815 c4=1.7654321096381136 c5=1.4324434305402327 c6=4.538504310173375	44050
		0.994959057093	c1=2.2284981536952526 c2=2.4876540147660724 c3=2.3246629790309585 c4=1.765432103598559 c5=1.4324434401326307 c6=3.543545657619489	60550

5. Conclusiones y problemas encontrados

Esta práctica me ha resultado bastante enriquecedora al haber programado dos Estrategias Evolutivas, aunque la experiencia ha sido algo agri dulce, ya que por una parte no he podido llegar a la solución óptima con estas sin salirme del “guión” en una de ellas, pero por la otra este hecho me ha obligado a darle vueltas a la cabeza para encontrar una solución a dicho problema, haciendo que me de cuenta que ningún algoritmo resuelve todo tipo de problemas.

Aún así, me gustaría saber si hay alguna forma de obtener la solución óptima sin necesidad de salirse del guión.