

Memoria de la práctica 2: Jardín



Grupo 06

Javier García Viana

Ibon Malles Altolaquirre

Aspectos generales

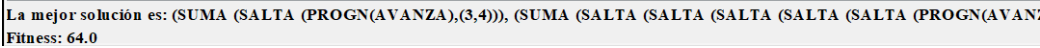
Se ha implementado un árbol n-ario de altura variable. Se han implementado tres tipos de inicializaciones para los árboles, un método de cruce, y cuatro métodos de mutación. El usuario puede elegir el control del bloating, la probabilidad de selección funcional, la altura máxima en la generación de los árboles, y las dimensiones del jardín. Para controlar el bloating se ha usado el método tarpeian con $n = 6$.

Análisis:

Jardín 8x8:

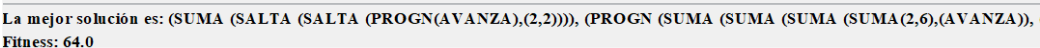
A continuación presentamos un número considerable de ejecuciones con distintas características.

Num Generaciones	Tam pob	Selección	Prob selección func	Mutación	Elitismo	resultado obtenido	Bloating
50	50	Montecarlo	0.9	Funcional	No	64	controlado
50	50	Montecarlo	0.9	Terminal	No	64	controlado
50	50	Montecarlo	0.9	Permutación	No	64	controlado
50	50	Montecarlo	0.9	Subárbol	No	64	controlado
50	50	Montecarlo	0.9	Funcional	No	62	no controlado
50	50	Montecarlo	0.9	Terminal	No	64	no controlado
50	50	Montecarlo	0.9	Permutación	No	64	no controlado
50	50	Montecarlo	0.9	Subárbol	No	64	no controlado
100	100	Torneo probabilística con $k = 3$	0.9	Terminal	No	64	controlado
100	100	ranking con $\beta = 2$	0.9	Subárbol	Si	64	controlado
120	200	Montecarlo	0.9	Funcional	No	144 (12*12)	controlado



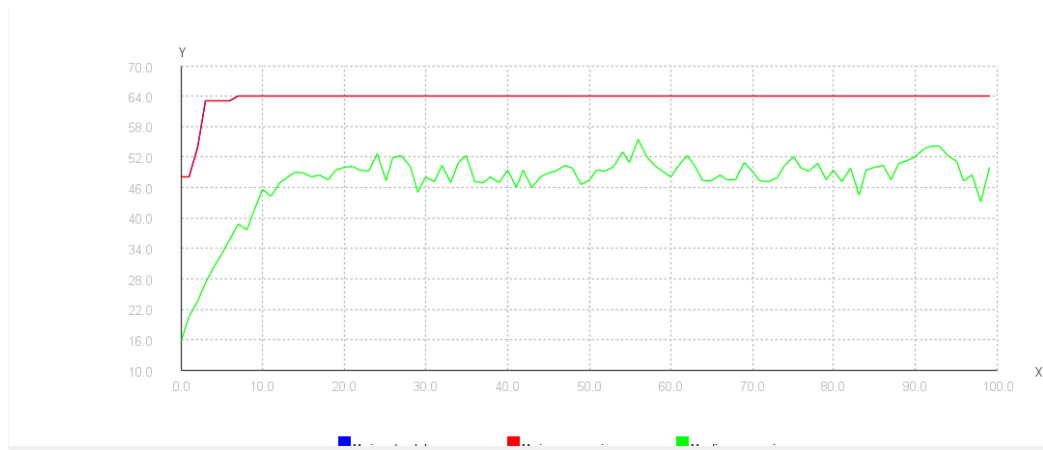
121,441	871	421,741	291	721	271	701	251,571
21,581	451,771	321	191,751	621	171	601	151
481	31,351	221,781	651	201,521	71	501	51
381	811	361,681	231,551	101	531,851	401	831
281	711	261	131,691	1,561,88	431	301,861	731
181	611	161	591	141,461	11,331	761	311,631
81,641	511	61	491	41	471,791	341,661	211
541	91,411	841	391	821	371	241,801	111,671

Montecarlo, mutación terminal, inicialización ramped and half con altura máxima = 5, y sin control de bloating:



163→,13→,177,321,59→,170→,1791,1311,811,1381	189→	1,861,1
871,881,111,1641,18→,681,1,601,101	421	→,1331,101,104→,1,1171
1,1191,114,891,9061,621,69→,1781	1101	431
1411	101,121→,10→,1791→,122→,791,111	23→,15315→,1671211,124
3→,1011,6→,116126→,921	1471	101,661,6
107→,11021	1,931,1127→,821→,1461,111,521,112→,175101,157	
03→,158	1,108→,56→,1291351,19555→,83171,1851,	1271,153→,17
2→,73→,158→,911,941,14→,761,98,1361,13	291	1,1871,18,601,16

Montecarlo, mutación subárbol, inicialización creciente con altura máxima = 5, y control de bloating:

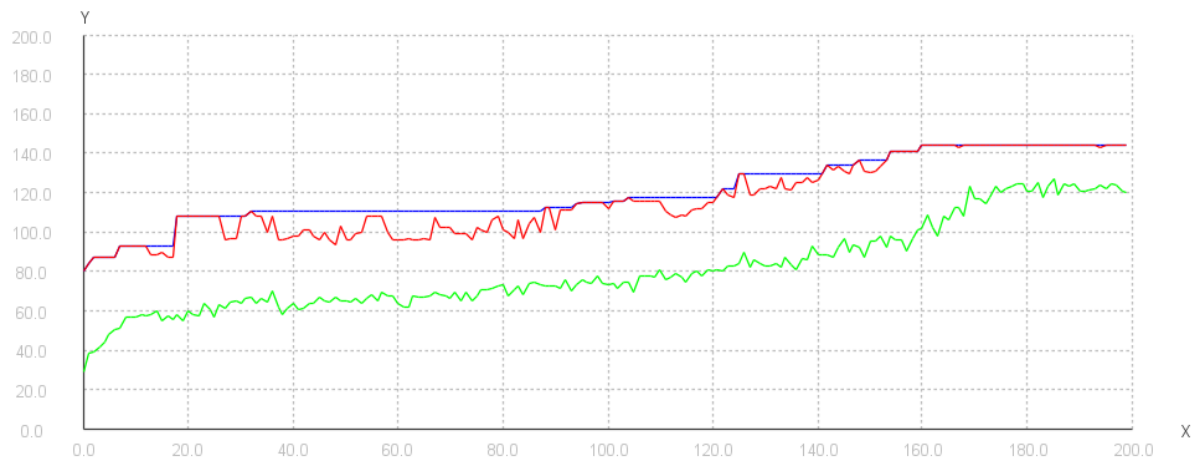


La mejor solución es: (SUMA (SUMA (SUMA (AVANZA), (SUMA (PROGN(7,2), (SALTA (PROGN (PROGN(AVANZA),(IZQUIERDA))),

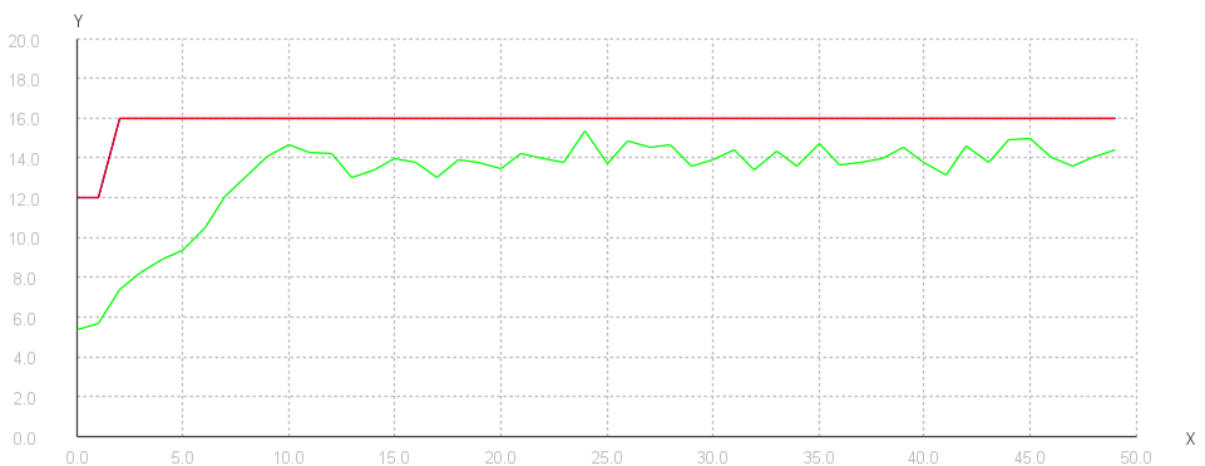
Fitness: 64.0

601	231	501	5→	321	591	141	411
421	61→	241	511	61	331	521	151
161	431	621	251	441	71	341	53→
541	171	361	631	261	45→	81	351
281	551	181	37→	01,641	271	461	91
101	29→	561	191	381	11	201	471
481	111	301	571	121	391	21	21→
221	491	41	311	581	13→	401	31

12x12: 120 individuos, 200 generaciones, inicialización creciente, selección montecarlo, mutación funcional y control de bloating



4x4: 50 generaciones, 50 individuos, inicialización creciente con altura máxima 3, selección truncamiento al 50%, mutación permutación y control de bloating:



Conclusiones

Se puede observar que el algoritmo alcanza el valor óptimo con muy pocas generaciones e individuos (alrededor de 50 y 50) para el problema original de 8x8, independientemente de la inicialización del árbol y su altura máxima, selección y mutación. Muy pocas veces no consigue llegar al óptimo (64) Para el problema de 12x12, el número de generaciones y el tamaño de la población tienen que aumentar bastante para que pueda encontrar el óptimo. Introduciendo un porcentaje bajo de elitismo, converge muy rápidamente a la solución.

Si no se controla el bloating, se generan programas mucho más grandes, y tarda más en ejecutar, pero también consigue encontrar el óptimo para el problema original. Si se controla el bloating, se encuentran soluciones óptimas más pequeñas en menor tiempo.

Con toda esta información, llegamos a la conclusión de que este problema es fácilmente resoluble.