

Algunas Mejoras en la Evaluación de los Individuos de la Programación Genética para la Regresión Simbólica

Joan Camps, Josep Maria Garrell

Resumen— Uno de los usos directos de la Programación Genética es la obtención de modelos que rigen un sistema a partir de un conjunto de datos representativos que ejemplifican su funcionamiento. Para la modelización, o regresión simbólica, hay distintos métodos según se tenga algún conocimiento del modelo o no. La Programación Genética puede ser usada directamente para la regresión simbólica sin necesidad de tener ninguna heurística que permita orientar la obtención del modelo matemático.

La idea principal de este trabajo se centra en la reducción del coste temporal del proceso. Para conseguirlo, se trata de reducir el espacio de todas las posibles funciones a generar, considerando que muchas de ellas serán parecidas pero con algún factor de desplazamiento o escalamiento respecto a la función buscada. Debido a estos factores es fácil que en el proceso de selección estas funciones se pierdan y no aparezcan en la siguiente generación cuando, con un pequeño ajuste, serían las mejores candidatas.

Se propone una nueva forma de evaluar las funciones de la población al realizar estos ajustes. Usando las medianas de los puntos obtenidos y sus desviaciones se llega a centrar la nueva función respecto a la esperada y, con ello, se favorece a las más parecidas.

Palabras clave— programación genética, regresión simbólica, computación evolutiva, inteligencia artificial

I. INTRODUCCIÓN

LA regresión simbólica es el proceso por el cual a partir de la experimentación sobre un sistema se consigue obtener el modelo matemático al cual responde ese sistema. Se necesita una muestra amplia de distintos estados en los que se puede encontrar el sistema. Se parte de un buen número de datos de entrada al sistema y se obtienen los correspondientes datos de salida del sistema. Con este conjunto de ejemplos de datos de entrada y salida, supuestamente representativos, se busca la manera de obtener el modelo matemático que los rige.

Uno de estos métodos con que se puede obtener un modelo matemático partiendo de diversos ejemplos es la Programación Genética (GP, *Genetic Programming*) [11], [12]. Con ésta se

consigue evolucionar una población de diversas fórmulas matemáticas hasta encontrar la que rige al sistema. Como en casi todos los métodos evolutivos, y más en éste, el gran obstáculo que tiene la Programación Genética es su alto coste computacional. Además de todo el proceso evolutivo, los individuos en la Programación Genética se representan en forma de árboles con lo que el coste de su evaluación es aún mayor respecto a las representaciones lineales. Debe añadirse también el alto coste que supone la obtención de las constantes adecuadas en los modelos, clásicamente ajustadas por el método del *constant perturbation*.

En este trabajo se propone un método para reducir de manera considerable el espacio de las posibles fórmulas a obtener. Como consecuencia, se observará que se contribuye a una convergencia hacia la solución más rápida a la vez que se reduce el coste del proceso de manera apreciable.

Después de considerar los distintos métodos de regresión simbólica entre los que se halla la Programación Genética, en el presente artículo se detallan las aportaciones realizadas para el ajuste de las funciones de la población. En el apartado 4 se presenta la experimentación realizada con el fin de verificar las nuevas aportaciones. Para finalizar, se presentan las conclusiones y las posibles líneas de trabajo futuro.

II. REGRESIÓN SIMBÓLICA USANDO PROGRAMACIÓN GENÉTICA

La Programación Genética (*Genetic Programming*, GP) [11], [12] es una técnica de computación evolutiva parecida a los Algoritmos Genéticos (*Genetic Algorithms*, GA) [9] pero usando una representación distinta de los individuos de su población.

El funcionamiento de esta técnica se basa en la existencia de una población de individuos (programas) que van evolucionando con el paso del tiempo debido al cruce y a la mutación, básicamente. Usando una medida de *fitness* específica, se obtiene un valor objetivo para cada uno de estos individuos, posibilitando que éstos perduren en la nueva generación o, por lo contrario, tiendan a desaparecer de la población. En este caso,

los individuos son programas y el *fitness* es una función que permite medir en qué grado el problema que se trata es solucionado.

La GP tiene diversas utilidades como la clasificación [6], o la predicción [2], [3], [7], [8], pero cabe destacar la regresión simbólica o modelado [1], [13] como una de las principales. Mediante la regresión simbólica se intenta encontrar la fórmula que modela el funcionamiento de un sistema. En este caso, los individuos de la población son ecuaciones que modelan este sistema. El cálculo del *fitness* se reduce a valorar en qué medida se aciertan los resultados esperados a partir de las muestras usadas para el entrenamiento del sistema.

Debe destacarse el alto coste temporal que supone cualquier proceso evolutivo, pero cabe añadir que la obtención de las constantes adecuadas en los modelos, supone aún un coste mayor. El método clásico de ajuste de constantes es el *constant perturbation* aunque hay otros mejores pero más costosos como el *numeric mutation* [5] y otros operadores [4] o híbridos realizados entre la Programación Genética y los Algoritmos Genéticos como el GA-P [10].

III. AJUSTANDO LAS FUNCIONES OBTENIDAS

Como acabamos de mencionar, una de las aplicaciones más directas y más extendidas de la Programación Genética es la regresión simbólica. En este proceso, se aprecia que el ajuste de las funciones supone una carga importantísima para el proceso evolutivo. Se observa que las distintas funciones que aparecen en la población genética se acercan mucho a la función exacta del modelo pero por algún factor de ajuste de ciertas constantes no obtienen la probabilidad que deberían para garantizar su selección en el proceso evolutivo.

En esta sección se describen las mejoras que se proponen con la finalidad de ajustar estos valores constantes de forma que el espacio de las funciones candidatas se reduzca considerablemente. Se propone dos aportaciones a esta mejora y otra contribución resultado de la combinación de las dos aportaciones anteriores.

A. Ajustando el desplazamiento

Durante el proceso evolutivo, los individuos de la población parten de un conjunto de funciones más o menos aleatorias que se van reemplazando por aquellas que responden con menos error a los datos que se usan en el proceso de búsqueda de la función ideal. En un principio, la población es como un caldo de cultivo de funciones totalmente distintas, con unos *fitness* muy diferencia-

dos entre ellas. Unas evaluaciones pueden resultar muy buenas y otras muy malas. Durante el proceso evolutivo se irá homogeneizando la población obteniendo un conjunto de funciones bastante parecidas aunque con pequeñas variantes entre ellas. Dependiendo del ajuste de los parámetros iniciales, este proceso puede encontrar la respuesta esperada o puede acercarse a ella pero también se puede centrar en una función aparentemente bastante buena pero que no responde a la que se esperaba.

Lo que sería de esperar y positivo para la correcta convergencia hacia una buena solución en el proceso evolutivo es que el *fitness* fuera proporcional a lo que llegaría a ser la función ideal del modelo, pero esta medida ideal no existe. Observando como evoluciona la población, se puede ver como en algunos casos se obtiene una función muy parecida o igual a la esperada pero con algún valor constante desajustado que conlleva una evaluación mala de la función. Este problema de ajuste de los parámetros constantes de la función tiene soluciones muy costosas [10].

Una posibilidad que se observa en algunas funciones es que se haya obtenido la función esperada, o alguna muy parecida, pero que esté desplazada verticalmente respecto lo que debería ser con la función ideal esperada. En estos casos, sobre todo si el desplazamiento es notable, se observa que la función puede llegar a ser eliminada de la población. La primera mejora que se propone incide en el ajuste de este desplazamiento. Se trata de encontrar un cierto valor constante que permita ajustar la función que se trate de manera que se aproxime más a la función esperada.

Supongamos que la función esperada en el proceso es la siguiente:

$$f(x) = 2x^2 - 3x + 1 \quad (1)$$

Supongamos también que en la población se encuentran las siguientes funciones:

$$f_d(x) = 2x^2 - 3x - 9 \quad (2)$$

$$f_n(x) = 3x^2 - \frac{4}{x-1} + 3 \quad (3)$$

El conjunto de datos usados en el proceso de la regresión simbólica es muy reducido [-3, -1, 0, 2 y 4].

En la Fig. 1, se puede observar la representación de las 3 funciones planteadas. Como se puede apreciar, la función esperada $f(x)$ se parece mucho a la función $f_d(x)$ mientras que es muy distinta de la función $f_n(x)$. Sería de esperar

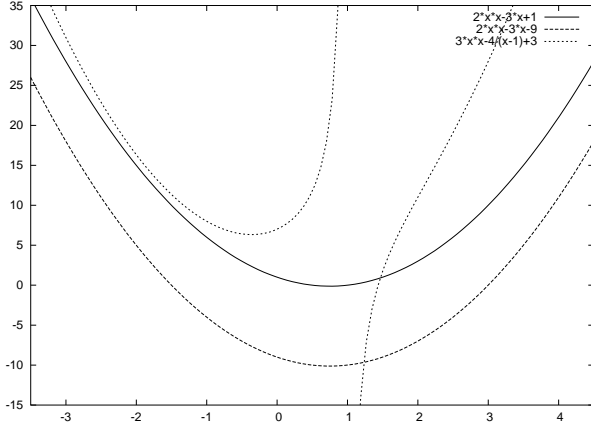


Fig. 1. Ejemplo del desplazamiento.

que en el proceso de selección la función $f_d(x)$ tuviera una probabilidad más alta de ser seleccionada que la función $f_n(x)$. Calculando sus *raw fitness* usando los 5 puntos como datos en el proceso evolutivo, observamos que el *fitness* de $f_d(x)$ es de 50 mientras que el *fitness* de $f_n(x)$ es de 47.6. Esto supone que la probabilidad de que una función sea seleccionada para la siguiente generación es ligeramente más alta en el caso de $f_n(x)$ que en $f_d(x)$, apreciándose claramente que lo correcto sería justamente lo contrario.

Para poder solucionar este problema sería interesante poder desplazar la función $f_d(x)$ el justo valor constante necesario para hacerla coincidir exactamente con la función esperada. La forma que se propone para encontrar este valor constante de desplazamiento se basa en las medias aritméticas de todos los puntos de entrenamiento usados. Se trata de calcular la media de los puntos esperados así como la media de los puntos obtenidos para la función que se intenta ajustar. Calculando la diferencia entre estos valores medios se obtiene un valor que, en el caso ideal que hemos propuesto, coincide con el desplazamiento necesario para ajustar la función obtenida a la ideal.

Si suponemos que $X_{f(x)}$ es la media de los puntos obtenidos de la función esperada $f(x)$ y que $X_{f_d(x)}$ es la media de los puntos obtenidos de la función $f_d(x)$, el desplazamiento D que deberíamos aplicar a la función $f_d(x)$ sería de $D = X_{f(x)} - X_{f_d(x)}$.

Si ahora sumamos D a la función $f_d(x)$, observamos que ésta coincide perfectamente con la función esperada $f(x)$, siendo su *raw fitness* de 0. Aplicando la misma transformación a $f_n(x)$, el *fitness* decremента ligeramente hasta 38.27.

B. Ajustando el escalado

Otra posibilidad que se observa en algunas funciones es que se haya obtenido la función esperada, o alguna muy parecida, pero que esté escalada respecto la que debería ser. Como en el anterior proceso de ajuste, se observa que la función puede también llegar a ser eliminada de la población. La segunda mejora que se propone incide en el ajuste de este escalado. Como en el caso anterior, se trata de encontrar un cierto valor constante que permita ajustar la función que se trate, de manera que se aproxime más a la función esperada.

Supongamos que tenemos la misma función esperada $f(x)$ y que en la población se encuentra también $f_n(x)$ y la siguiente función:

$$f_e(x) = 4x^2 - 6x + 2 \quad (4)$$

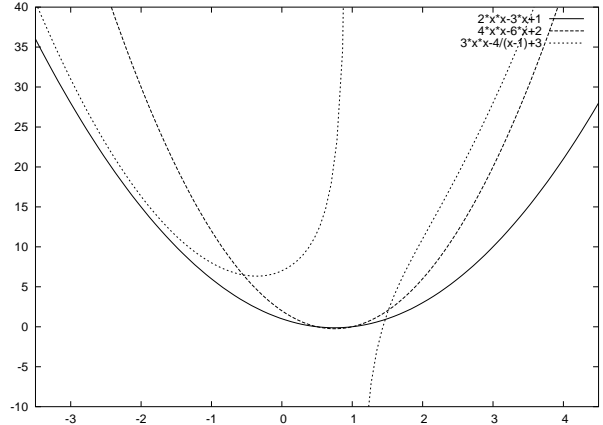


Fig. 2. Ejemplo del escalado.

En la Fig. 2, se puede observar la representación de las 3 funciones planteadas.

Como se puede apreciar, la función esperada $f(x)$ se parece bastante a la función $f_e(x)$ pero con cierta deformación. Al igual que antes, sería de esperar que en el proceso de selección la función $f_e(x)$ tuviera una probabilidad más alta de ser seleccionada que la función $f_n(x)$. Calculando sus *raw fitness* usando los 5 puntos como datos en el proceso evolutivo, observamos que el *fitness* de $f_e(x)$ es de 59 mientras que el *fitness* de $f_n(x)$ es de 47.6. Esto supone que la probabilidad de que una función sea seleccionada para la siguiente generación es más alta en el caso de $f_n(x)$ que en $f_e(x)$ cuando sería correcto lo contrario.

En este caso, para solventar el problema, sería interesante si pudiéramos escalar la función $f_e(x)$ el justo valor constante necesario para hacerla coincidir exactamente con la función esperada. La forma que se propone para encontrar este valor constante de escalado se basa en partir de las

desviaciones de todos los puntos de entrenamiento usados. Se trata de calcular la desviación estándar de los puntos esperados así como la desviación estándar de los puntos obtenidos para la función que se intenta ajustar. Calculando el cociente entre estas desviaciones se obtiene un valor que, en el caso ideal que hemos propuesto, coincide con el factor de escalado necesario para ajustar la función obtenida a la ideal.

Si suponemos que $\sigma_{f(x)}$ es la desviación estándar de los puntos obtenidos de la función esperada $f(x)$ y que $\sigma_{f_e(x)}$ es la desviación estándar de los puntos obtenidos de la función $f_e(x)$, el factor de escalado E que deberíamos aplicar a la función $f_e(x)$ sería de $E = \frac{\sigma_{f(x)}}{\sigma_{f_e(x)}}$.

Si ahora multiplicamos E a la función $f_e(x)$, observamos que ésta coincide perfectamente con la función esperada $f(x)$, siendo su *raw fitness* de 0. Aplicando la misma transformación a $f_n(x)$, el *fitness* decrementa hasta 27.46.

C. Ajustando el desplazamiento y el escalado

La última posibilidad que planteamos es la combinación de las dos que se acaban de exponer. Es el caso en que aparece una función muy parecida a la esperada pero que está escalada y desplazada a la vez. Igual que en los casos anteriores, esta función también puede correr el riesgo de ser eliminada de la población de una generación a la otra aun siendo bastante buena. En este caso el ajuste que se debería realizar sería doble. Se trataría de encontrar dos valores constantes que nos permitieran realizar el doble ajuste.

Supongamos que tenemos la misma función esperada $f(x)$ y que, como antes, en la población se encuentra también $f_n(x)$ y la siguiente función:

$$f_{de}(x) = 4x^2 - 6x - 3 \quad (5)$$

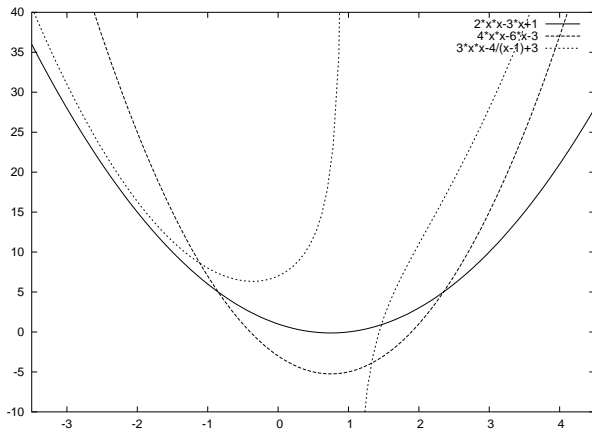


Fig. 3. Ejemplo del desplazamiento y escalado.

En la Fig. 3, se puede observar la representación de las 3 funciones planteadas.

Como se puede apreciar, la función esperada $f(x)$ sigue pareciéndose bastante a la función $f_{de}(x)$ pero con cierta deformación y cierto desplazamiento. Al igual que antes, sería de esperar que en el proceso de selección la función $f_{de}(x)$ tuviera una probabilidad más alta de ser seleccionada que la función $f_n(x)$. Calculando sus *raw fitness* usando los 5 puntos como datos en el proceso evolutivo, observamos que el *fitness* de $f_{de}(x)$ es de 46 mientras que el *fitness* de $f_n(x)$ es de 47.6. Esto supone que la probabilidad de que una función sea seleccionada para la siguiente generación es muy parecida en ambos casos.

En este caso, para solventar el problema, sería interesante si pudiéramos escalar y desplazar la función $f_{de}(x)$ lo justo para hacerla coincidir exactamente con la función esperada. La forma que se propone para encontrar este valor constante de escalado y de desplazamiento es la combinación de las dos soluciones anteriores; a partir de las medias y de las desviaciones estándares. Se trata de calcular la media de $f(x)$ ($X_{f(x)}$) y de $f_{de}(x)$ ($X_{f_{de}(x)}$) así como el factor E , usado en el ejemplo anterior, con el fin de ser aplicados a la función f_{de} de forma apropiada.

Si la función, como en el ejemplo, es justamente la esperada con un desplazamiento y un escalado, se puede apreciar que f_{de} se transforma perfectamente en $f(x)$ aplicando el siguiente cálculo:

$$[f_{de}(x) - X_{f_{de}(x)}] * E + X_{f(x)} \quad (6)$$

Si ahora aplicamos este cálculo a la función $f_{de}(x)$, observamos que ésta coincide perfectamente con la función esperada $f(x)$, siendo su *raw fitness* de 0. Aplicando la misma transformación a $f_n(x)$, el *fitness* se sitúa en 25.52.

Partiendo de que la transformación propuesta de escalado y desplazamiento responde a la fórmula básica $g(x) = af(x) + b$, donde a y b son dos valores que se desean calcular, aplicamos el cálculo de probabilidad con el fin de encontrar la forma en que se obtienen.

Aplicando el cálculo probabilístico sobre la media de la función transformada, $\mu[g(x)] = \mu[af(x) + b]$, obtenemos la siguiente ecuación: $\mu_{g(x)} = a\mu_{f(x)} + b$

Haciendo el mismo proceso con la desviación estándar de la función transformada, $\tau_{g(x)}^2 = E[(af(x) + b)^2] - E[af(x) + b]^2$, se llega a la ecuación: $\tau_{g(x)}^2 = a^2\tau_{f(x)}^2$

Con estas ecuaciones obtenemos el valor de a y b :

$$a = \frac{\tau_{g(x)}}{\tau_{f(x)}} \quad (7)$$

$$b = \mu_g(x) - a\mu_f(x) \quad (8)$$

Aplicando estas equivalencias sobre la fórmula básica de $g(x) = af(x) + b$, obtenemos que,

$$g(x) = [f(x) - \mu_f(x)] \frac{\tau_g(x)}{\tau_f(x)} + \mu_g(x) \quad (9)$$

IV. EXPERIMENTACIÓN

Con el fin de verificar las suposiciones presentadas en el apartado anterior, se ha llevado a cabo un conjunto de pruebas para comprobar experimentalmente si el cambio en la evaluación de la población supone realmente una ventaja. En primera instancia, cabe esperar un aumento en el porcentaje del tiempo de evaluación de la población. De la experimentación debe concluirse si la mejora en la convergencia de la población es provechosa frente a la sobrecarga que conlleva la evaluación de ésta o, por el contrario, no supone una mejora considerable.

Para realizar este estudio se ha usado 3 problemas distintos que se pasan a describir seguidamente. A posteriori se pasa a explicar el juego de pruebas realizadas con los resultados obtenidos.

A. Descripción de los problemas

Se ha experimentado a partir de 3 distintas funciones sobre las que se ha generado aleatoriamente 25 puntos (en el dominio $[-12..12]$ de las abscisas) para poder evaluar a los individuos en el proceso evolutivo.

Las funciones utilizadas han sido:

$$f_1(x) = 2x^2 - 3x + 1 \quad (10)$$

$$f_2(x) = x^4 + x^3 + x^2 + x \quad (11)$$

$$f_3(x) = x^2 + \frac{5}{x} + 3x - 1 \quad (12)$$

B. Pruebas realizadas

Con los 3 problemas planteados, se han llevado a cabo un total de 1440 ejecuciones. En cada una de las ejecuciones se ha utilizado una población de 200 individuos. Se ha fijado el fin del proceso evolutivo cuando el error sea menor de 10^{-2} aunque en la mayoría de las ejecuciones realizadas se llega a un error de 10^{-5} . Se ha fijado una profundidad de árbol de 5 con una inicialización tipo *Ramped Half-And-Half*. El conjunto de los operadores considerados en el proceso son: suma, resta, producto, división y valor absoluto. El método evolutivo usado es el generacional con elitismo en grado uno, con una evaluación de *raw fitness* y con el método de selección de la ruleta.

Los parámetros que se han variado en la configuración para testear el método han sido las

probabilidades de cruce (0.6 y 0.8) y de mutación (0.2, 0.4, 0.6 y 0.8). Para cada una de estas configuraciones se ha realizado 15 ejecuciones con distintas semillas aleatorias. Otro parámetro que se ha modificado es el uso de constantes o no en el proceso. En síntesis, para cada función, se ha realizado un conjunto de 480 ejecuciones; 240 con el método tradicional de evaluación y 240 con la mejora propuesta en este artículo. Las ejecuciones se han realizado sobre un Pentium III a 700 MHz.

C. Resultados obtenidos

De los datos obtenidos en el conjunto de ejecuciones se han tratado 2 aspectos importantes: el promedio de generaciones que ha supuesto encontrar la función exacta o una muy parecida y el promedio en coste temporal que se incrementa debido a la sobrecarga en el método de evaluación de cada individuo.

En las tablas I y II se muestra un resumen de los promedios y desviaciones de los resultados del número de iteraciones realizadas para llegar a la función deseada en cada uno de los problemas y según se haya usado o no las constantes en el proceso. Se presentan los promedios y las desviaciones estándar para cada conjunto de ejecuciones, tanto para el proceso habitual de evaluación de la función como para la mejora propuesta en este artículo.

TABLA I
PROMEDIO DEL NÚMERO DE ITERACIONES REALIZADAS EN
EL CONJUNTO DE EJECUCIONES PARA CADA
CONFIGURACIÓN.

Evaluación	Normal	Mejorada
$f_1(x)$ sin cnt.	100.88	39.88
$f_1(x)$ con cnt.	384.19	66.28
$f_2(x)$ sin cnt.	75.50	61.76
$f_2(x)$ con cnt.	221.10	128.20
$f_3(x)$ sin cnt.	1710.40	243.64
$f_3(x)$ con cnt.	221.10	168.73

Como se puede apreciar, en todos los casos se llega a un número inferior de generaciones necesarias para llegar a los mismos resultados. No sólo el promedio de las iteraciones disminuye sino que las desviaciones obtenidas del conjunto de las ejecuciones también disminuyen de forma considerable. Cabe destacar la mejora relativa que se da en algunos casos como en la función $f_3(x)$ sin el uso de constantes. En este caso tenemos un 85.76 % de mejora respecto al método clásico. En la misma función, pero con el uso de las constantes en el proceso evolutivo, se aprecia que la

TABLA II
DESVIACIÓN ESTÁNDAR DEL NÚMERO DE ITERACIONES
REALIZADAS EN EL CONJUNTO DE EJECUCIONES PARA
CADA CONFIGURACIÓN.

Evaluación	Normal	Mejorada
$f_1(x)$ sin cnt.	93.38	61.30
$f_1(x)$ con cnt.	840.55	68.69
$f_2(x)$ sin cnt.	61.96	67.94
$f_2(x)$ con cnt.	639.30	206.04
$f_3(x)$ sin cnt.	4211.11	192.63
$f_3(x)$ con cnt.	639.30	161.13

mejora es muy inferior, de un 23.69 % solamente. Por el contrario, tanto en la $f_1(x)$ como en la $f_2(x)$ la mejora es mucho mayor en el caso del uso de constantes.

En las tablas III y IV se muestra un resumen de los promedios y desviaciones de los resultados del tiempo empleado para llegar a la función deseada en cada uno de los problemas y, al igual que en las tablas anteriores, según se haya usado o no las constantes en el proceso. Se presentan los promedios y las desviaciones estándar para cada conjunto de ejecuciones, tanto para el proceso habitual de evaluación de la función como para la mejora propuesta en este artículo.

TABLA III
PROMEDIO DEL TIEMPO EMPLEADO (EN SEGUNDOS) EN EL
CONJUNTO DE EJECUCIONES PARA CADA CONFIGURACIÓN.

Evaluación	Normal	Mejorada
$f_1(x)$ sin cnt.	2.71	1.20
$f_1(x)$ con cnt.	10.21	1.93
$f_2(x)$ sin cnt.	2.19	1.92
$f_2(x)$ con cnt.	5.61	3.79
$f_3(x)$ sin cnt.	187.76	28.09
$f_3(x)$ con cnt.	1883.39	248.68

TABLA IV
DESVIACIÓN ESTÁNDAR DEL TIEMPO EMPLEADO EN EL
CONJUNTO DE EJECUCIONES PARA CADA CONFIGURACIÓN.

Evaluación	Normal	Mejorada
$f_1(x)$ sin cnt.	2.28	1.79
$f_1(x)$ con cnt.	23.25	2.04
$f_2(x)$ sin cnt.	1.79	2.00
$f_2(x)$ con cnt.	14.61	5.66
$f_3(x)$ sin cnt.	499.83	22.57
$f_3(x)$ con cnt.	2501.82	606.37

Tal y como se observa, el coste temporal en todos los casos se reduce de manera importante.

Como la mejora en la evaluación de los individuos a partir del cálculo de sus medias y sus desviaciones estándar supone un sobrecoste temporal, pudiera haber sucedido que los resultados globales fueran peores aunque, como se ha podido comprobar mejoran en su totalidad. En las desviaciones obtenidas sobre los promedios del coste temporal, se observa que en todos los casos éstas disminuyen a excepción de $f_2(x)$ sin constantes que, prácticamente, se mantiene.

Al igual que con las iteraciones, la mejora relativa temporal que se observa en algunos casos es muy buena, como en la función $f_3(x)$. En este caso tenemos un 85.10% sin constantes y un 86.80% con constantes de mejora. Al igual que sucedía con las iteraciones, en las funciones $f_1(x)$ y $f_2(x)$ la mejora es mucho mayor en el caso del uso de constantes que sin ellas.

V. CONCLUSIONES Y TRABAJO FUTURO

El objetivo del trabajo era comprobar si se podía mejorar la evaluación de los individuos en la Programación Genética aplicada a la regresión simbólica con el fin de reducir el coste computacional del proceso. Para ese fin se había propuesto una modificación en el cálculo del error de las funciones respecto los datos de entrenamiento. Con el cálculo de las medias de los puntos tratados y las desviaciones estándar respecto a estos puntos, nos proponíamos reducir el espacio de las posibles funciones objetivo de forma considerable.

Con la experimentación realizada se ha podido comprobar que en todos los casos llevados a cabo se mejoran los resultados. En la mayoría de los problemas la reducción ha sido muy considerable, aun observando el total de los promedios globales. Por lo visto, se habría podido obtener alguna configuración excepcional en cuanto a mejora pero es en los valores promedios donde se aprecia la utilidad del nuevo proceso de cálculo en la evaluación. Esta reducción en iteraciones y en coste temporal se ratifica por la reducción en la desviación tanto para el promedio de las iteraciones como para el promedio del tiempo utilizado.

En trabajos posteriores nos proponemos, en primer lugar, realizar una experimentación más extensa y con funciones mucho más dispares y complejas. También puede ser interesante comprobar este sistema con algún tipo de núcleo genético más complejo y evolucionado como pueden ser los GA-P [10]. Así como con el proceso clásico de evaluación de los individuos se iba de una primera generación prácticamente aleatoria hacia una generación bastante más homogénea, sería interesante también comprobar si el grado de diversidad de la población se mantiene a lo

largo del proceso evolutivo. De no ser así, podría ocurrir que el proceso evolutivo quedara fijado en algún mínimo local y se dificultara la obtención de la función esperada. Es interesante observar cómo las funciones que, respecto la ideal, están escaladas y desplazadas llegan a ajustarse perfectamente a la solución. En este sentido, una posible línea de futuro a tener en cuenta es el análisis de cómo afecta la nueva evaluación de los individuos en las demás funciones, ya sea incrementando o decrementando su *fitness*.

AGRADECIMIENTOS

Agradecemos a *Enginyeria i Arquitectura La Salle*, de la *Universitat Ramon Llull*, por su apoyo a nuestro grupo de investigación en sistemas inteligentes, así como el soporte del *Ministerio de Ciencia y Tecnología* bajo el proyecto TIC2002-04036-C05-03 con participación de fondos FEDER y el soporte del *Departament d'Universitats, Recerca i Societat de la Informació* para grupos de investigación consolidados de Catalunya 2002-SGR-00/55.

REFERENCIAS

- [1] G. Calzada, J.M. Garrell, R. Barti, *ModelGP: A GP-based platform for predictive model construction*, Proceedings of the third International ICSC Symposia on Intelligent Industrial Automation IIA'99 and Soft Computing SOCO'99, 1999.
- [2] J. Camps, J.M. Garrell, E. Golobardes, D. Vernet, *Diseño de Funciones de Similitud para el Razonamiento Basado en Casos usando Programación Genética: estudio con problemas sintéticos*, Actas del Segundo Congreso español sobre Metaheurísticas, Algoritmos evolutivos y bioinspirados, 2003.
- [3] G. Corral, A. Zaballos, J. Camps, J.M. Garrell, *Prediction and control of short-term congestion in ATM Networks using Artificial Intelligence techniques*, Proceedings of the IEEE International Conference on Networking, 2001.
- [4] J.W. Davidson, D.A. Savic, G.A. Walters, *Method for the identification of explicit polynomial formulae for the friction in turbulent pipe flow*, Journal of Hydroinformatics, 1, 2, 1999.
- [5] M. Evett, T. Fernandez, *Numeric Mutation Improves the Discovery of Numeric Constants in Genetic Programming*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'97), 1997.
- [6] J.M. Garrell, E. Golobardes, E. Bernadó, X. Llorà, M. Salamó, J. Camps, D. Vernet, J. Bacardit, C. Vallesplá, G. Corral, A. Zaballos, J. Rios, *Aprendizaje Artificial en Sistemas de Clasificación, Predicción y Diagnóstico Automático*, Actas del Workshop de Minería de Datos y Aprendizaje, IBERAMIA, 2002.
- [7] E. Golobardes, X. Llorà, M. Salamó, J. Martí, *Computer Aided Diagnosis with Case-Based Reasoning and Genetic Algorithms*, Journal of Knowledge Based Systems, 2001.
- [8] E. Golobardes, M. Nieto, M. Salamó, J. Camps, G. Calzada, J. Martí, D. Vernet, *Generació de funcions de similitud mitjançant la Programació Genètica pel Raonament Basat en Casos*, 4t Congrés Català d'Intel·ligència Artificial, Butlletí de l'ACIA, 25, 2001.
- [9] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [10] L.M. Howard, D.J.D'Angelo, *The GA-P: A Genetic Algorithm and Genetic Programming Hybrid*, IEEE Intelligent Systems, Vol.10, 3, 1995.
- [11] J.R. Koza, *Genetic Programming. On the programming of computers by means of natural selection*, Massachusetts Institute of Technology Press, 1992.
- [12] J.R. Koza, *Genetic Programming II. Automatic Discovery of Reusable Programs*, Massachusetts Institute of Technology Press, 1994.
- [13] M. Ribó, G. Calzada, M. Castaño, J.R. Regué, J.M. Garrell, *Modelado de Dispositivos de Microondas Mediante Programación Genética*, Libro de Actas de la VII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA), 1999.