Resolviendo un problema criptoaritmético con algoritmo firefly

María Laura Acuña, Marcelo Espinoza, Cecilia María Luciana Gómez

Alumnos de 5to año de la carrera ingeniería en sistemas de información, Facultad Regional Resistencia, Universidad Tecnológica Nacional – French 414 – Resistencia, Chaco, Argentina. CP 3500.

{Marilau\_ml, marceloespinoza00, [Cecilia.mlgz}@gmail.com](mailto:Cecilia.mlgz%7d@gmail.com)

**Abstract.** En el siguiente paper se demuestra una forma de resolver un problema criptoaritmeticos de suma o restas, utilizando para ello algoritmo de enjambre, específicamente el algoritmo firefly desarrollado por [1], tal algoritmo fue adaptado para que se ajuste al problema planteado.

**Keywords:** Algoritmos genéticos, criptoarimética, algoritmos de enjambre, firefly, inteligencia artificial

1. Introducción

Los problemas criptoaritmeticos son puzles donde las letras son reemplazadas por número, son problemas de restricciones [2], que si se intenta resolver manualmente genera un gran espacio de búsqueda, dificultando así la generación de la solución. En [2] se plantea una solución utilizando algoritmos genéticos con operaciones de cruza y mutación, en [3] se plantea una solución utilizando también algoritmos genéticos, con operaciones de mutación, y se demuestra la mejora en performance respecto a la búsqueda heurística En [6] se describe cómo solucionar el problema del viajantes aplicando el algoritmo propuesto por [1]. Por tales motivos se plantea una solución implementando, con algunas adaptaciones el algoritmo planteado por [1], redefiniendo una función distancia y la función acercamiento.

1. ¿Qué son los algoritmos de enjambre?

Estos algoritmos simulan el comportamiento que tienen ciertas especies que se organizan grupalmente para subsistir. Algunos de los algoritmos de enjambres más conocidos son “La optimización de colonias de hormigas”, “La optimización de colonias de abejas”, y entre estos tipos de algoritmos se encuentra el desarrollado por [1]. La inteligencia de enjambre estudia el comportamiento colectivo compuestos por muchos individuos interactuando localmente y con su entorno [5].

1. Problema Planteado

El problema consiste en resolver puzles criptoaritmeticos, de sumas o restas, con la implementación del algoritmo firefly [1].

Este problema presenta las restricciones siguientes:

* + La cantidad distintas de letras de los operandos a sumar (o restar) no debe ser mayor a 10.
  + Cada letra se identifica con un único número y ese número representa a una única letra.
  + El resultado deber ser acorde a la suma algebraica de los operandos.

1. Solución propuesta

En la fig. 2 se muestran las principales funciones de la implementación propuesta. En la fig. 4 se detalle la aplicación del algoritmo en sí.

Los valores de entrada se colocan en un vector que sirve como base para el desarrollo de todo el algoritmo. Por ejemplo si se ingresa APPLE + LEMON = BANANA, el vector conformado, luego de acomodarlo y quitarle los elementos repetidos, se completa con “-“ hasta completar las diez posición. El vector formado queda:

enalopnb--

Para la implementación se definió la función objetivo como “obtenerBrillo” la que recibe como parámetro la suma algebraica de los valores que corresponden a los operadores de entrada, y el resultado que corresponden a los valores del operando ingresado como resultado.

public function obtenerBrillo( $suma, $resultado)

{ $res=$this->intToArray($resultado);

$sum=0;

$i=(sizeof($res)-1);

$j=(sizeof($suma)-1);

$counter=0;

if(sizeof($res)<sizeof($suma)){

$counter=sizeof($res);

}else{

$counter=sizeof($suma);

}

for ($k = ($counter-1); $k >=0; $k--) {

if($res[($i)]==$suma[($j)]){

$i--;

$j--;

$sum++;

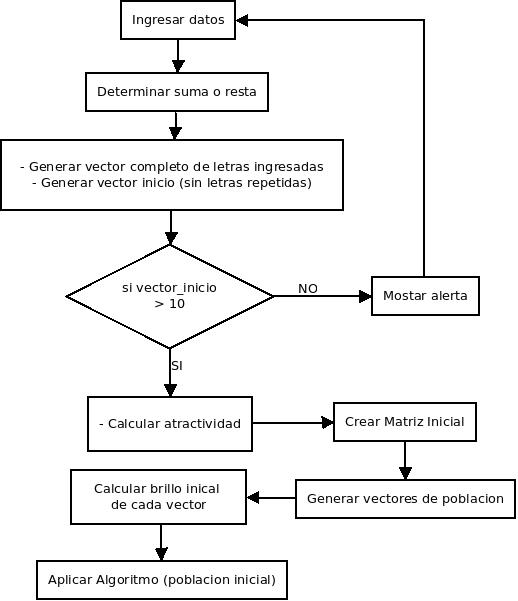
}

}

return $sum;

**Fig. .** Definición de la función objetivo.

Su funcionamiento consiste en devolver un valor de brillo que depende de la cantidad de elementos que tenga el vector de resultado ingresado, y las coincidencias que tenga con el vector suma, comparando posición a posición..



**Fig. .** Inicio del algoritmo planteado

La función distancia definida en la figura 3, recibe como parámetros dos vectores que corresponden a la suma algebraica de dos luciérnagas diferentes y devuelve un valor entre uno y diez.

public function distancia($X1, $X2) {

$i = count($X1);

$d = 0;

for ($j=0; $j < $i; $j++) {

if ($X1[$j] = $X2[$j]) {

$d = $d + 10\*$j;

}

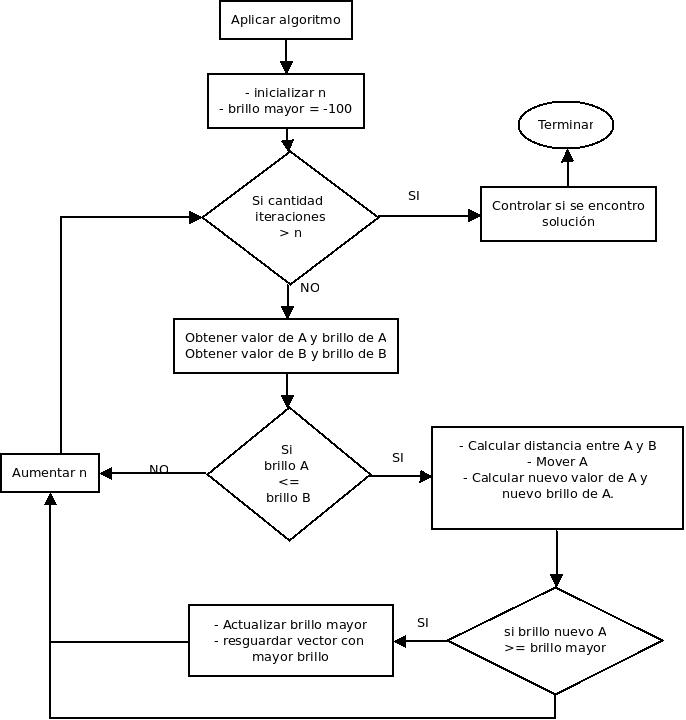
}

$d = fmod($d, 10);

return $d;

}

**Fig. .** Función distancia



**Fig. .** Desarrollo de la solución

1. Conclusión
2. Referencias
3. X.-S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178 - SAGA 2009
4. Resolución de Problemas Criptoaritméticos Utilizando Algoritmos Genéticos 43 JAIIO - EST 2014 - ISSN: 1850-2946 17º Concurso de Trabajos Estudiantiles. - 2014
5. Evolutionary Algorithm to Solve Cryptarithmetic Problem Transactions on engineering, computing and technology VI – World Enformatika Society Decembre 2004
6. Algoritmos genéticos – John H. Holland – Investigación y Ciencia – Septiembre 1992
7. Algoritmos de inteligencia de enjambres sobre GPU: una revisión exhaustiva. XX Congreso Argentino de Ciencias de la Computación - Buenos Aires, 2014
8. Solving of Travelling Salesman Problem using Firefly Algorithm with Greedy Approach Special Issue: International Conference on Non - Linear System & Optimization in Computer & Electrical Engineering - 2015