

# Modelo de semáforo inteligente empleando un algoritmo genético

## Intelligent traffic light model using a Genetic Algorithm

Fernando Mejía<sup>a</sup>, German Torres<sup>b</sup>, Eduardo Villa<sup>c</sup>

<sup>a</sup> Facultad de Informática y Electrónica/Escuela de Ingeniería Electrónica en Control y Redes Industriales, Escuela Superior Politécnica de Chimborazo

<sup>b</sup> Unidad de Admisión y Nivelación, Escuela Superior Politécnica de Chimborazo

<sup>c</sup> Facultad de Informática y Electrónica/Escuela de Ingeniería en Sistemas, Escuela Superior Politécnica de Chimborazo

**Correspondencia Autores:** msmejiaedwinf@yahoo.com, gtorres@esepoch.edu.ec, evilla@esepoch.edu.ec

**Recibido:** agosto 2016, **Publicado:** diciembre 2016

**Resumen**— El control automático y las técnicas del Softcomputing han evolucionado en resolver los problemas de control óptimo que consiste en asignar tiempos a los semáforos de acuerdo a la llegada de vehículos en el cruce de calles donde debe fluir el tráfico de una manera rápida. En este artículo se ha desarrollado un algoritmo genético que controle los tiempos de cambio de luces entre verde, amarilla y roja con el objetivo de minimizar al máximo la congestión vehicular de la ciudad de Riobamba. Para esto se realiza un seguimiento de lo que está pasando en las calles con cada uno de los vehículos que llegan al semáforo. Se realiza también una revisión de que tiempos se adaptaron mejor, es decir, los que minimizaron mejor el tráfico, se hace a través de un cruce y mutación que nos da el método de aprendizaje de los algoritmos genéticos. Entonces de esta manera se controla el tráfico de Riobamba y se ayuda a minimizar en gran medida basado en sensores el flujo vehicular.

**Palabras Claves**— Inteligencia Artificial, Softcomputing, Algoritmos Genéticos, Semáforo Inteligente, Sensores.

**Abstract**— the automatic control and techniques of the Softcomputing have evolved in solving the problems of optimum control that consists of assigning times to the traffic lights according to the arrival of vehicles in the crossing of streets where the traffic must flow in a fast way. In this article, a genetic algorithm has been developed that controls the time of change of lights between seeing, yellow and red in order to minimize the traffic congestion of the city of Riobamba. For this, it is tracked what is happening on the streets with each of the vehicles that arrive at the traffic light. We also performed a review of which times were better adapted, that is, those that minimized the traffic better, is done through a cross and mutation that gives us the method of learning the genetic algorithms. This way the traffic of Riobamba is controlled and it is helped to greatly minimize sensor-based traffic flow.

**Keywords**— Artificial Intelligence, Softcomputing, Genetic Algorithms, Intelligent Semaphore, Sensors.

### I. INTRODUCCIÓN

Desde los años cincuenta, existen evidencias que se usaron ordenadores para realizar ciertos procesos naturales [Neum, 56], [Neum, 58]. Uno de los primeros trabajos que utilizan un proceso evolutivo para la resolución de problemas en computadoras es el de Friedberg [Frie, 58], [Frie, 59]. Aquí se manifiesta la utilización de algoritmos evolutivos para programación automática.

A mediados de los 60's ya se establecen las bases de lo que hoy es la computación evolutiva. Donde los trabajos de Fogel [Foge, 66] en San Diego de California, se desarrollaron el campo de la Programación Genética.

Se desarrollan también los algoritmos genéticos por Holland [Hol, 69] en la Universidad de Michigan, a la vez las estrategias evolutivas en la Universidad de Berlín [Rech, 65].

En el año 1991 se reúnen varios investigadores en el Workshop on Paralell Problem Solving from Nature en Dortmund, en donde en realidad nace la Computación Evolutiva como término [Schw, 91] para englobar las tres técnicas empleados que son: algoritmos genéticos, estrategias evolutivas y la programación genética [1].

La investigación en teorías sobre la evolución es el paradigma neo-Darwiniano. En esta teoría se da una afirmación de que la vida puede ser completamente justificada mediante procesos físicos, mediante poblaciones y especies [Hoff, 89].

Esta oportunidad se potencia con el auge del paradigma de la Teoría de la Complejidad, que privilegia la no linealidad, la irreversibilidad y el no determinismo, en esferas que recorren todo el universo científico [2].

Los Sistemas Expertos son pioneros en la idea de obtener modelos partiendo de expresiones verbales, de manera que los decisores humanos pueden aplicar su experiencia esencial en problemas concretos. La representación del conocimiento sobre la base de la Lógica es aquí protagónica. En los últimos tiempos se ha desarrollado una disciplina matemático-informática llamada Soft-Computing o Inteligencia Computacional, [3-4]. Entre los fundamentos de esta disciplina está también los algoritmos genéticos [4-6].

A partir de la década de los 80 del siglo anterior, los responsables de los departamentos de tráfico intuyeron que las soluciones tradicionales que se estaban aplicando no serían capaces de resolver esta problemática. Estas soluciones típicamente implicaban la construcción de nuevas infraestructuras o la ampliación de las ya existentes, pero en muchos casos no eran soluciones viables debido principalmente a los elevados costes, a la carencia de espacio disponible y al inevitable impacto medioambiental. Por tanto, los departamentos de tráfico de todo el mundo están muy interesados en la optimización de las

infraestructuras existentes, con el objeto de mejorar al máximo sus prestaciones [5]. La Unión Europea tiene diferentes medios, a través de los cuales pretende mitigar los indeseables impactos citados con anterioridad. Desde el establecimiento de normas de comportamiento para los turismos nuevos, con el fin de reducir las emisiones de CO<sub>2</sub> [3], hasta el uso de una normativa para disminuir el ruido de vehículos de motor, pasando por las medidas ordenadoras del tráfico, como pueden ser, entre otras, el funcionamiento de los semáforos que regulan una intersección vial.

La función principal de los semáforos en el control de una intersección es dar el paso alternativamente a los distintos grupos de vehículos, peatones, bicicletas y/u otro sistema de transporte, de tal manera que éstos pasen a través de la intersección con un mínimo de problemas, riesgos y demoras. Los objetivos del diseño de una intersección regulada por semáforos son:

- Reducir y prevenir cierto tipo de accidentes en la intersección y en las intersecciones aledañas.
- Reducir las demoras que experimentan los peatones y los vehículos al intentar cruzar la intersección, y al mismo tiempo evitar los obstáculos en las intersecciones más cercanas causados por colas largas.
- Reducir el consumo de combustible de los vehículos en la intersección.
- Reducir la emisión de contaminantes al aire (CO<sub>2</sub>), que afectará positivamente [3,5]

En este artículo se revisa una de las técnicas de inteligencia artificial que son los algoritmos genéticos, la importancia en el desarrollo de sistemas automáticos en el mundo de la semaforización inteligente, se describe cuáles son los dispositivos más comunes, así como la técnica para desarrollar el algoritmo genético.

Debido a los índices de demora de los tiempos en los cruces de semáforos y por ende la correspondiente molestia que los conductores han expresado, entonces se realiza una automatización de los semáforos para que exista una fluidez mayor respecto a los tiempos.

El objetivo es implementar y analizar un algoritmo genético para disminuir los tiempos y mejorar la movilidad de los vehículos en la ciudad de Riobamba en el Ecuador, dando prioridad a las vías con mayor tráfico. El algoritmo aprende de los cambios que se efectúan en los semáforos y de esta manera busca obtener los tiempos óptimos.

Este sistema inteligente captura a través de los sensores donde existe mayor cantidad de autos y de acuerdo a los tiempos seleccione los ciclos más óptimos para seguir actualizando su aprendizaje.

## II. GENERALIDADES

En este trabajo de investigación se desarrolla un análisis de cómo realizar la captura de los datos de la longitud de cola de los vehículos que llegan a los semáforos. Esto se realiza mediante sensores ubicados en cada calle de donde están los semáforos.

Este sistema inteligente captura la longitud de cola vehicular en cada semáforo, y da prioridad a las intersecciones con mayor afluencia vehicular. Luego

selecciona los ciclos más óptimos adaptados al tráfico, con el fin de realizar nuevas generaciones a partir del proceso de aprendizaje de los algoritmos genéticos. Al momento de finalizar el análisis de ciclos, son realizadas actualizaciones en cada semáforo para volver a ejecutar el proceso.

La longitud en tiempo del ciclo completo de los semáforos, es decir, el tiempo necesario para que se dé una sucesión completa de indicaciones en los semáforos conectados a un mismo regulador. Independientemente de lo que resulte de los cálculos, la duración del ciclo tiene que estar forzosamente comprendida entre los límites que fija la psicología del conductor. La práctica indica que ciclos menores de 35 segundos o mayores de 120, se acomodan difícilmente a la mentalidad del usuario de la vía pública [6].

Debido a que la buena coordinación de la red de semáforos de la ciudad y la optimización de los ciclos y fases de los mismos son herramientas fundamentales para tratar de resolver los problemas de congestión de tráfico y sobresaturación de vehículos, como se demuestra en algunos trabajos [4], es por lo que se ha diseñado un nuevo modelo, que es la principal aportación del presente artículo, para la resolución de dichos problemas

### A. Algoritmos Genéticos

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas. [7]

Un algoritmo genético consiste en una función matemática o una rutina de software que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación. [8]

Versiones más complejas de algoritmos genéticos generan un ciclo iterativo que directamente toma a la especie (el total de los ejemplares) y crea una nueva generación que reemplaza a la antigua una cantidad de veces determinada por su propio diseño. Una de sus características principales es la de ir perfeccionando su propia heurística en el proceso de ejecución, por lo que no requiere largos períodos de entrenamiento especializado por parte del ser humano, principal defecto de otros métodos para solucionar problemas, como los Sistemas Expertos [6].

### B. Metodología

Se realizó una revisión sistemática de la literatura científica con base en la adaptación de la metodología PRISMA [Preferred Reporting Items for Systematic

Reviews and Meta-Analyses][Urrutia y Bonfil, 2010]. La pregunta de investigación establecida para conducir el proceso metodológico fue el siguiente: ¿Cuáles son los enfoques y estrategias de implementación de los Algoritmos Genéticos para el uso de semáforos inteligentes? Los principios básicos de los Algoritmos Genéticos fueron establecidos por Holland, y se encuentran bien descritos en varios textos. Goldberg, Davis, Michalewicz, Reeves. [6]

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario, individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente [3]. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "superindividuos", cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los Algoritmos Genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos [5]. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos, descendientes de los anteriores, los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema. [7]

### C. Proceso de Control Inteligente

#### 1. Cantidad de Vehículos

Se captura la cantidad de vehículos que llegan al semáforo, es decir, la cantidad de vehículos que tiene un semáforo en cola. Esto se realiza por medio de una red de sensores ubicados a 80mts, 40mts y 2mts de cada semáforo. Actualmente se han implementado en Riobamba dos tipos de sensores: uno de presión, que es ubicado debajo del asfalto, y otro de presencia o proximidad (infrarrojos), que es ubicado en la parte izquierda y derecha de la calle (figura 1). Se propone utilizar en una futura implementación el sensor de cámara, no solo por ser el más económico, sino también porque tiene un rango de visión más amplio.

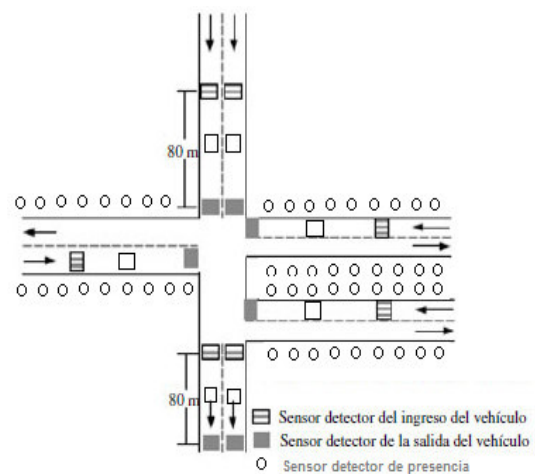


Fig. 1. Modelo Estándar de los sensores

El prototipo retorna por cada semáforo la cantidad real de vehículos que se encuentran en la carretera, de manera parecida a como si contara con un sensor real. Luego de recorrer las intersecciones y de capturar la cantidad de vehículos por cada semáforo, se organizan de mayor a menor, siendo las intersecciones prioritarias las que mayor cantidad de vehículos tengan. La figura 2 muestra la ubicación del sensor de impacto o presión. De esta manera, se crea la primera generación de tiempos de los ciclos verde y rojo, y el tiempo de ciclo amarillo se ha determinado como una constante con un valor de dos segundos. Esta primera generación creada se realiza a partir de la demanda de vehículos; se establecen unos límites para esta y evitar que un punto del sistema sea sobrecargado.

La función de adaptación con los tiempos de los ciclos que se crearon en la primera generación se calcula el impacto en el sistema, es decir, se evalúa cuáles de los tiempos generados se adaptaron al sistema de tráfico de manera positiva, cuáles ayudaron a minimizar el tráfico y cuáles contaron con el menor tiempo del ciclo. Esto se hace comparando el flujo vehicular y los tiempos de ciclo, y el resultado es guardado para las siguientes generaciones de ese semáforo.

2. Función de adaptación

Los tiempos que se generan en cada semáforo por la cola de autos que intervienen en el mismo se calcula el impacto en el sistema, esto es, evaluar cuales de esos tiempos puede adaptarse al sistema, en donde está el mejor tiempo para disminuir de manera positiva el tráfico en la ciudad y finalmente donde están los tiempos menores del ciclo.

Esto se hace comparando el flujo vehicular y los tiempos de ciclo, y el resultado es guardado para las siguientes generaciones de ese semáforo. Los tiempos que no se adaptaron al ciclo son desechados, siguiendo el esquema de evolución de Darwin [8], y así para cada semáforo.



Fig. 2. Sensor de impacto o presión

Cambio de rojo a verde y viceversa

Teniendo los diferentes niveles de pertenencia arrojados por la función de adaptación, los mismos deben ser procesados para generar una salida óptima. La tarea del sistema inteligente es tomar los niveles de pertenencia y apoyado en los tiempos mejores, se genera la salida del color rojo o verde.

Cuando existen una gran carga de vehículos en una parte de la calle y en la otra existen menos se da prioridad donde está la mayor carga, dando un 40% más de los tiempos de acuerdo al número de vehículos para que puedan pasar de un lado de la calle al otro.

Debe haber un tiempo de desfase, esto va a depender de la distancia que existe entre cada calle, para poder cruzar y que no existan choques entre vehículos.

3. Algoritmo Genético Aplicado

En la figura 4 se muestra el algoritmo genético aplicado bajo la metodología que se indica:

Es una de las mejores metodologías a seguir dado que permite tener una población coherente al inicio, luego seleccionamos los padres, realizamos el cruce y mutación, y obtenemos las descendencias de donde escogeremos el mejor individuo o mejores.

En la figura 5 se muestra el algoritmo que se debe seguir para poder seleccionar al mejor individuo.

A los individuos se los representa mediante Cromosomas, siguiendo con esta situación podemos decir que el proceso de reproducción o cruce no es más que crear un nuevo individuo a partir de los padres elegido, es decir combinando los dos vectores que representan a los mismos.

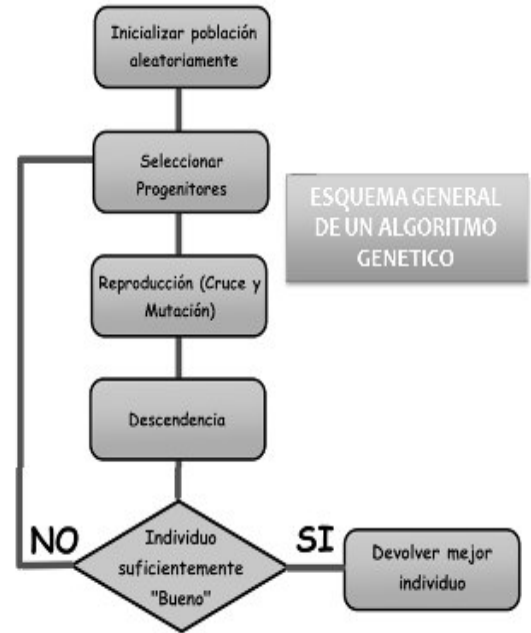


Fig. 4. Metodología Aplicada

```

BEGIN
INICIALIZAR de forma aleatoria una población con soluciones candidatas EVALUAR cada candidato
REPEAT UNTIL ( CONDICIÓN DE TERMINACIÓN == true )
1. SELECCIONAR progenitores
2. RECOMBINAR progenitores seleccionados obteniendo descendencia
3. MUTAR descendencia
4. EVALUAR nuevos candidatos
5. SELECCIONAR individuos para la próxima generación
END REPEAT
END
    
```

Fig. 5. Algoritmo de programación a seguir.

En la figura 6 se presenta un ejemplo de cómo se obtiene el nuevo individuo a partir de sus padres elegidos:

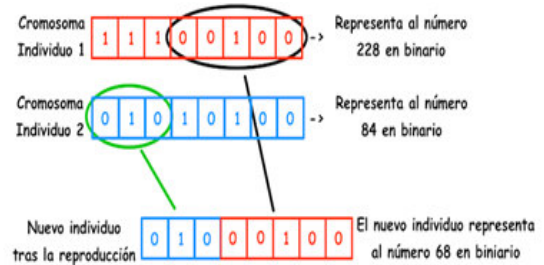


Fig. 6. Cruce de 2 individuos

En lo referente a la generación de nuevos individuos [6], existe también la posibilidad de que se realice la mutación de algún Gen, esto realiza al azar para que un

elemento del Cromosoma cambie de valor, por ejemplo, el valor complementario como se muestra en la figura 7.

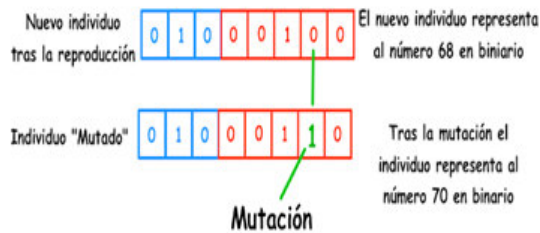


Fig. 7. Mutación del nuevo individuo

Como se ha comentado los Algoritmos Evolutivos sirven para resolver problemas de optimización combinatoria, pero también se puede decir que no siempre alcanzan la mejor solución, por lo tanto hay que definir un umbral a partir del cual consideramos que la solución que buscamos es lo suficientemente buena. Este umbral se obtiene a partir de una “función de evaluación” que se define siempre que se utilicen algoritmos evolutivos. Esta función de evaluación es conocida como “Función de Fitness” aunque también lo denominan función de calidad, función de aptitud o función objetivo.

#### D. Propuesta

En este artículo utilizamos la figura 1 y 2 en donde se muestra las formas como van a estar conectados cada uno de los diferentes sensores en las calles de la ciudad de Riobamba y como se recogerá la información.

La idea principal de este artículo es dar solución al problema de la congestión vehicular en los cruces de las vías y entonces de manera automática controlar el semáforo a través de un algoritmo genético el mismo que controla los tiempos de cambio de las luces de color roja, amarilla y verde, siguiendo los lineamientos de la figura 4, 5, 6 y 7.

	A	B	C	D	E	F	G	H	I	J
1										
2		via1								
3	Padre 1	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6	sensor7	sensor8	sensor9
4		1	0	1	1	0	0	1	1	0
5										
6	Padre 2	1	1	0	1	0	1	0	1	1
7										

Fig. 8. Datos recogidos de los sensores en Excel

El proceso de la toma de datos por parte de los sensores tanto de los que están en el piso como de los que están a los costados depende de la cantidad de autos que lleguen y especialmente de la manera como vaya aprendiendo el algoritmo. El algoritmo que controla esto se basa en el

principio de Selección, cruce, mutación y elección del mejor individuo que se adapte a esta circunstancia.

El algoritmo genético usado es el que se muestra en la figura 8, en donde se recoge los datos de los sensores en Excel para cada una de las vías en las cuales estamos realizando la investigación, es el siguiente:

```
function rtn = creator(genomeLength, fitnessFn, options)
temp = floor(5.*rand(5,1));
rtn = temp';
```

```
function rtn = objective(param)
seq = [1 2 3 4 5];
temp = 5;
for i = 1:length(param)
if eq(seq(i),floor(param(i)))
temp = temp - 1;
end
end
rtn = temp;
```

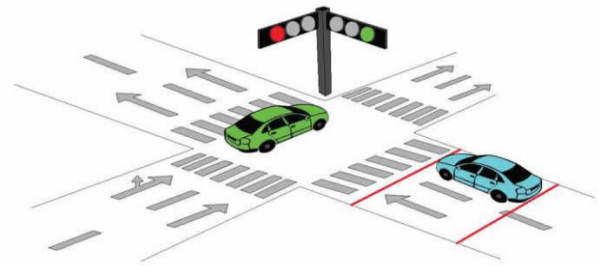


Fig.9. Cruce de vías con dos semáforos

Con estos datos se ingresa a la función que se crea en Matlab con los siguientes parámetros:

La función de Fitness tiene que llegar a lo que se muestra en la figura 9, donde en la una vía tiene que estar en rojo y en la otra tiene que estar en verde.

La mejor solución del algoritmo genético luego de realizar el proceso es la que se muestra en la figura 10.

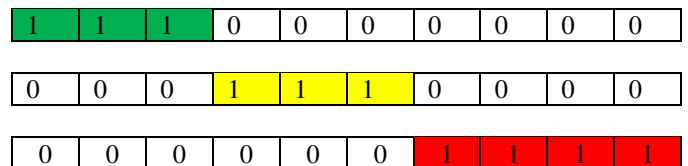


Fig.10. Cromosomas con sus 3 genes y colores de semáforos para poder encenderlos

### III. RESULTADOS

Para comprobar el correcto funcionamiento del algoritmo genético diseñado y programado en el arduino, todo esto se realizó y programó en MATLAB como se muestra en la figura.8. Para ejecutar el algoritmo se inicia el toolbox, ingresando en la línea de comandos optimtool. Luego que se abre la ventana en el campo Solver se selecciona “ga-Genetic Algorithm”. La función Rastrigin está incluida en Matlab, por lo tanto, no es necesario definirla en algún archivo “.m”, solo se tiene que ingresar como función de fitness “@rastriginsfcn” y en Number of variables 2.

## Modelo de semáforo inteligente empleando un algoritmo genético

En la figura 11 se observa la manera como el algoritmo genético para nuestra investigación de acuerdo a la metodología planteada en la figura 4, nos da 71 iteraciones para darnos una respuesta óptima.

Con esta información insertada en la figura 11 ya se puede ejecutar el algoritmo. En primer lugar, se resolverá con todas las opciones por default y luego se intentará mejorar los resultados mediante el uso de algunas opciones. Luego de ejecutar el algoritmo mediante el botón Start se observan los resultados en la ventana de ejecución que se muestra en la figura 12. Se puede observar que al cabo de 71 generaciones el algoritmo se detuvo porque “el cambio promedio en los valores de la función de fitness es menor al especificado por las opciones”. Y al hacerlo encontró un mínimo en el punto (0.874, 0.953) donde la función toma el valor de 5.099.

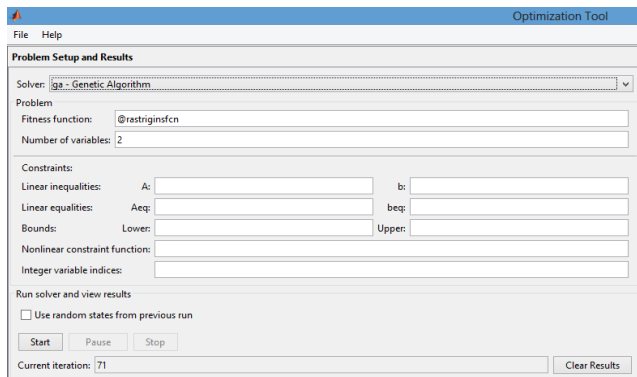


Fig.11. Algoritmo genético en Matlab.

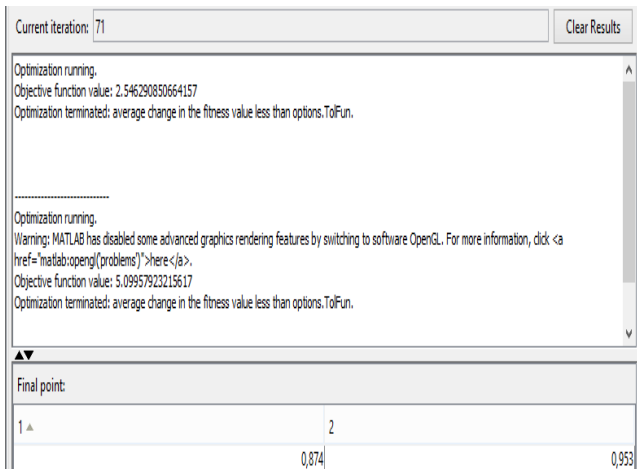


Fig.12. Venta de ejecución con los resultados de fitness

Este resultado que se muestra en la figura 12, siempre va a depender de la naturaleza del problema y puede conllevar a que tomemos el mismo como aceptable o no.

En este caso, que ya se conoce la ubicación del mínimo global, es claro que queda un margen para mejorarlo.

Para analizar de mejor manera el desempeño del algoritmo podemos analizar algunas graficas de los resultados del algoritmo. Para empezar, se mostrarán las gráficas de los valores con los mejores individuos y la distancia promedio entre ellos, las cuales se seleccionan en

las opciones de “funciones de plot” que se muestra en la figura 13.

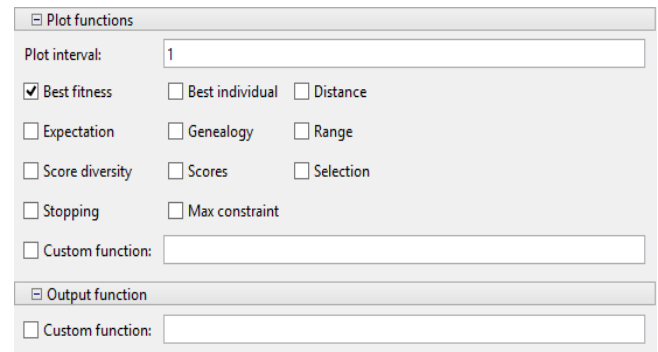


Fig.13. Funciones de plot en el intervalo 1 – Best fitness

Al hacer de nuevo clic en Start tenemos la figura 16 donde nos muestra que al pasar de generaciones el mejor valor alcanzado sigue mejorando, así como el valor promedio de los individuos (mean fitness). A su vez la distancia promedio entre individuos va disminuyendo progresivamente, aunque aún se tiene una población bastante dispersa. Esto puede indicar que el algoritmo se detuvo demasiado pronto.

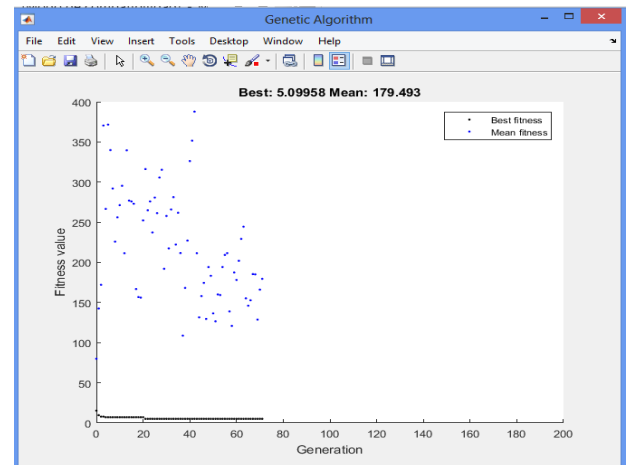


Fig.14. Plot Functions con plot interval de 1 – Best fitness

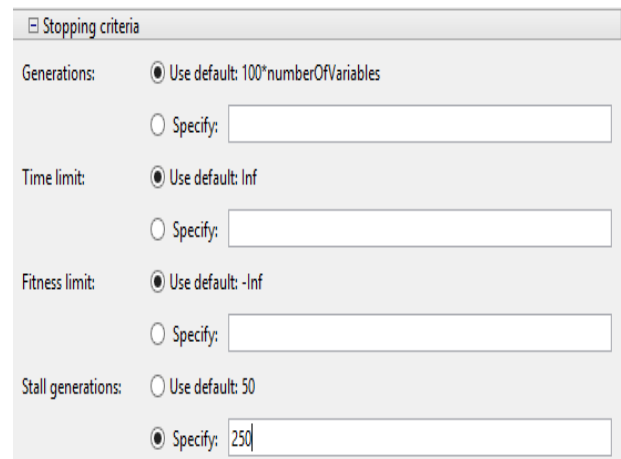


Fig.15. Opcion de Stopping criteria – Stall generations

En el caso anterior el motivo de la detención del algoritmo fue: “el cambio promedio en los valores de la función de fitness es menor al especificado por las opciones”. Esto se puede modificar mediante la opción de “Stall generations” en el menú de opciones de criterios de detención. Por defecto es 50, y para una nueva prueba se llevará a 250, tal como se muestra en la figura 15:

En este caso se obtiene la figura 16 luego de ejecutar el algoritmo.

En este caso se observa un resultado considerablemente mejor. La función alcanza un valor cercano a 0.0072906245126134195 (-0,006, 0,001), lo cual es un resultado más satisfactorio para cualquier aplicación ingenieril. Al observar las mismas graficas que en el caso anterior se puede ver como la población converge al cabo de las generaciones. En este caso el criterio de detención del algoritmo fue el de máxima cantidad de generaciones (valor por defecto: 200), tal como se muestra en la figura 17.

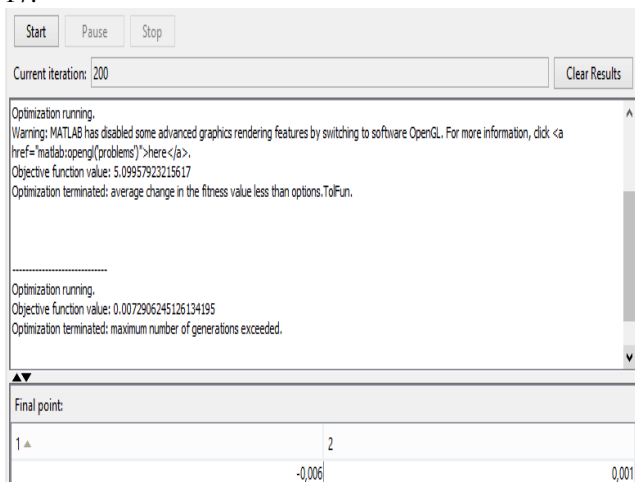


Fig.16. Generaciones igual a 200

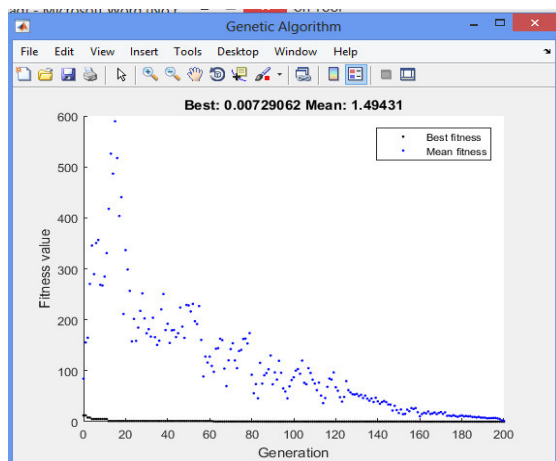


Fig.17. Mejor individuo en el intervalo

La verificación de los datos obtenidos, se ha utilizado la herramienta optintool - toolbox de Matlab, la cual sirve como referencia para corroborar que el algoritmo genético realiza la función de optimización y obtiene el mejor individuo obteniéndose un rango de -1 a 0 que es un valor que nos permite decir que nuestro algoritmo es el adecuado.

Para verificación de las placas realizadas se utilizó el multímetro digital Fluke, con el cual se hicieron las pruebas de continuidad, medición de voltajes, detección de falsos contactos entre terminales y algunos análisis en las salidas de los pines y las interfaces.

#### IV. DISCUSIÓN

La toma de decisiones la debe generar el algoritmo genético planteado [2] para presencia o ausencia de vehículos y el tiempo de demora entre los colores verde, amarillo y rojo, en el caso planteado que es la de contar cuantos vehículos están en un lado de la calle u otro y tomar las decisiones correctas para que el flujo de vehículos sea dinámico, resulta muy beneficioso utilizar esto.

La dimensión del tiempo en cuanto a respuesta del sistema esta medido en segundos. Los datos obtenidos se obtienen de los sensores que están en el asfalto y los que están a los costados, ellos permiten contar de manera adecuada los vehículos que ingresan y salen de la calle continua al semáforo.

Ahora también hay que decir que cada instante que se vaya ejecutando el algoritmo genético, el mismo irá aprendiendo de cada intervalo de tiempo en el día, se debe mencionar que el semáforo se utiliza de 6 de la mañana a 8 de la noche ininterrumpidamente.

Se debe decir que cuando se desarrollen este tipo de automatización en sentido industrial los nuevos modelos de procesamiento y control vistos, junto con otros de relativa novedad, se engloban en la denominada Soft Computing.

#### V. CONCLUSIONES

En este artículo se ha demostrado que con una herramienta de algoritmos genéticos se puede lograr el control de las vías de una ciudad y en este caso de Riobamba.

Sin dejar de lado la metodología para la construcción de los algoritmos genéticos que son la selección, cruce, mutación y selección del mejor individuo también hay que saber hacer el cruce y cuál es la mejor mutación en ese momento que el algoritmo se está ejecutando.

El uso de los sensores y del Arduino es de gran ayuda para este tipo de trabajos de investigación, pero hay que saber escogerlos, al momento de realizarlos en la realidad.

El número de individuos aumenta exponencialmente con el número de intervalos que nosotros le demos, cuanto más grande es se puede llegar a la mejor solución.

El número de sensores es de acuerdo a la practicidad y relación que puedan tener con la cola de vehículos en la calle donde vayan a coger los datos.

En un próximo trabajo se estará mostrando las simulaciones realizadas con ejemplos prácticos y los tiempos que se demoran cada uno de los semáforos en cada una de las intersecciones

#### RECONOCIMIENTOS

Agradezco a mi familia en especial a mi madre que está ahí para darme las fuerzas que todo hijo necesita. (Mejia, F.).

**REFERENCIAS**

- [1] Piedrafita R (2000) Ingeniería de la Automatización, Alfa omega.
- [2] Sánchez S. (2012) Optimización de estructuras mediante algoritmos genéticos, Valencia – España.
- [3] Gestal M. and Rivero D. (2010). Introducción a los algoritmos genéticos y la programación genética, Coruña – España.
- [4] Verdegay J.L. (2005). Una revisión de las metodologías que integran la Soft computing. Actas del Simposio sobre Lógica Fuzzy y Soft Computing LFSC 2005 (EUSFLAT).151-156.
- [5] Holland J. (2016). Algoritmos Genéticos Adaptativos, Portugal.
- [6] Araujo L. and Cervigon C. (2009). Algoritmos Evolutivos – Un enfoque práctico. Ra-Ma.
- [7] Gestal M. (2013). Introducción a los algoritmos genéticos. Coruña – España.
- [8] Berzal F. (2016). Algoritmos Genéticos. Departamento de Ciencias de la Computación e IA. España