

Algoritmo Genético para el TSP (Traveling Salesman Problem) con Monodevelop utilizando C#

La computación evolutiva es una parte de la inteligencia artificial donde se agrupan diferentes técnicas adaptativas enfocadas a encontrar las mejores soluciones para resolver problemas de optimización, estas técnicas parten de una de las principales ideas que existen en la teoría de la evolución biológica propuesta por Charles Darwin, la selección natural es el proceso histórico de transformación de unas especies en otras especies descendientes e incluye la extinción de la gran mayoría de las especies que han existido, lo que es sin duda un problema de optimización además de que muchos de los términos que utiliza la computación evolutiva son tomados de la biología, con la advertencia de que se son usados para representar más o menos la misma idea biológica lo que no equivale a que sean iguales.

La computación evolutiva se compone por las siguientes técnicas:

1. **Programación evolutiva:** Son una estrategia de optimización estocástica hacen un énfasis específico en los operadores genéticos tal y como se observan en la naturaleza.
2. **Estrategias evolutivas:** Esta técnica esta básicamente enfocada hacia la optimización. En esencia son métodos estocásticos con paso adaptativo, que permiten resolver problemas. A este método se le han agregado procedimientos propios de la computación evolutiva, que lo han convertido en un paradigma más de dicha metodología.
3. **Algoritmos genéticos:** Usan métodos que tienen analogía en la selección natural y la evolución.

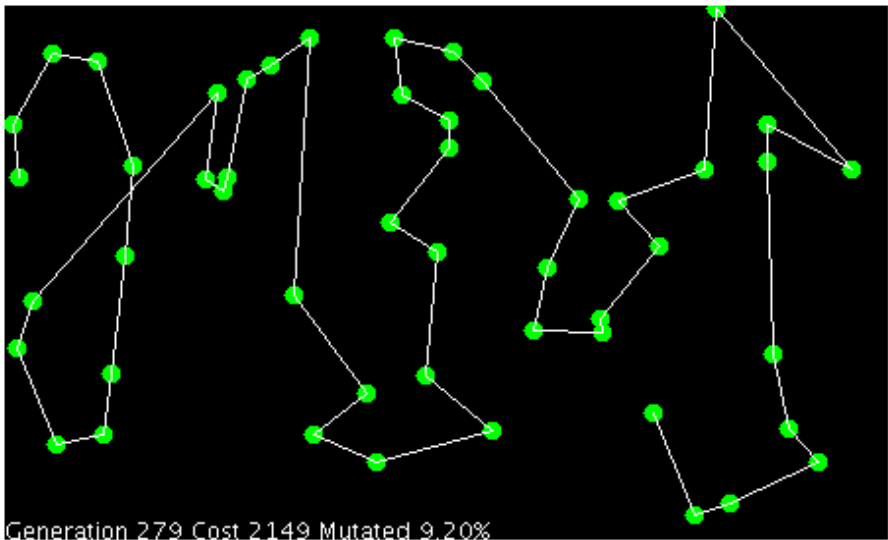
Entre los usos de la computación evolutiva se encuentran la planificación, el diseño de sistemas, la simulación, el reconocimiento de patrones, el control y la clasificación de datos.

La alternativa para una solución de problemas de planificación y de optimización donde la mayoría están se ubican dentro de la categoría de los problemas NP los cuales no pueden ser resueltos usando las técnicas convencionales y solo pueden ser atacados desde las técnicas de computación evolutiva donde el dominio de esos problemas le corresponden a los algoritmos genéticos (*John H. Holland, 1975*)

Como ejemplo de este tipo de aplicaciones me encontré en este [web site](http://www.heatonresearch.com) (<http://www.heatonresearch.com>) un [Applet Java](#), que utiliza algoritmos genéticos para obtener una solución optima al problema del agente viajero (Traveling Salesman Problem), en el mismo web site el autor nos hace un resumen del problema.

Fig 1 El applet del programa original

Genetic Algorithm



Generation 279 Cost 2149 Mutated 9.20%

Cities: ,Pop Size: ,Mutate%:

Instructions

1. Enter the number of cities the salesman much cross through.
2. Enter the population size, 1000 is good for 50 cities. Enter too small a population and a solution will not be found.
3. Enter the mutation percent, this is what percent of the population will be "mutated", which is a random element introduced into the genetic makeup.
4. Click Start, and watch the genetic algorithm try and find the optimal path.

Pues bien como un ejercicio de aprendizaje, traduje el código del applet de Java a C# como un proyecto de Monodevelop utilizando Winforms, a continuación algunas screenshots y el código fuente del proyecto. He de aclarar que debido a la falta de tiempo hice casi copia tal cuál del código, modificando las partes de Java que el compilador me marcaba como errores.

Fig 2 La solución en Monodevelop

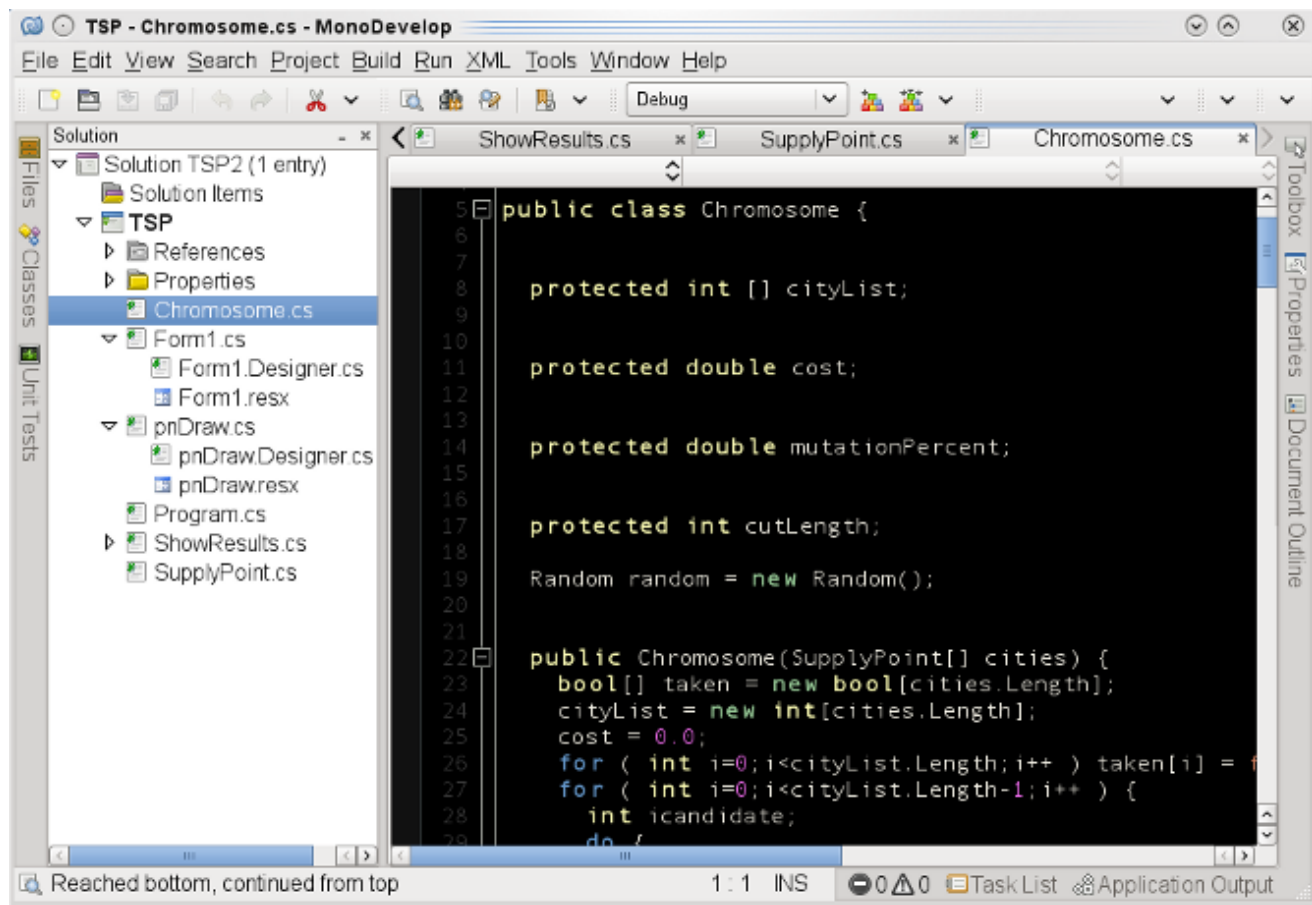


Fig 3 Inicio de la aplicación

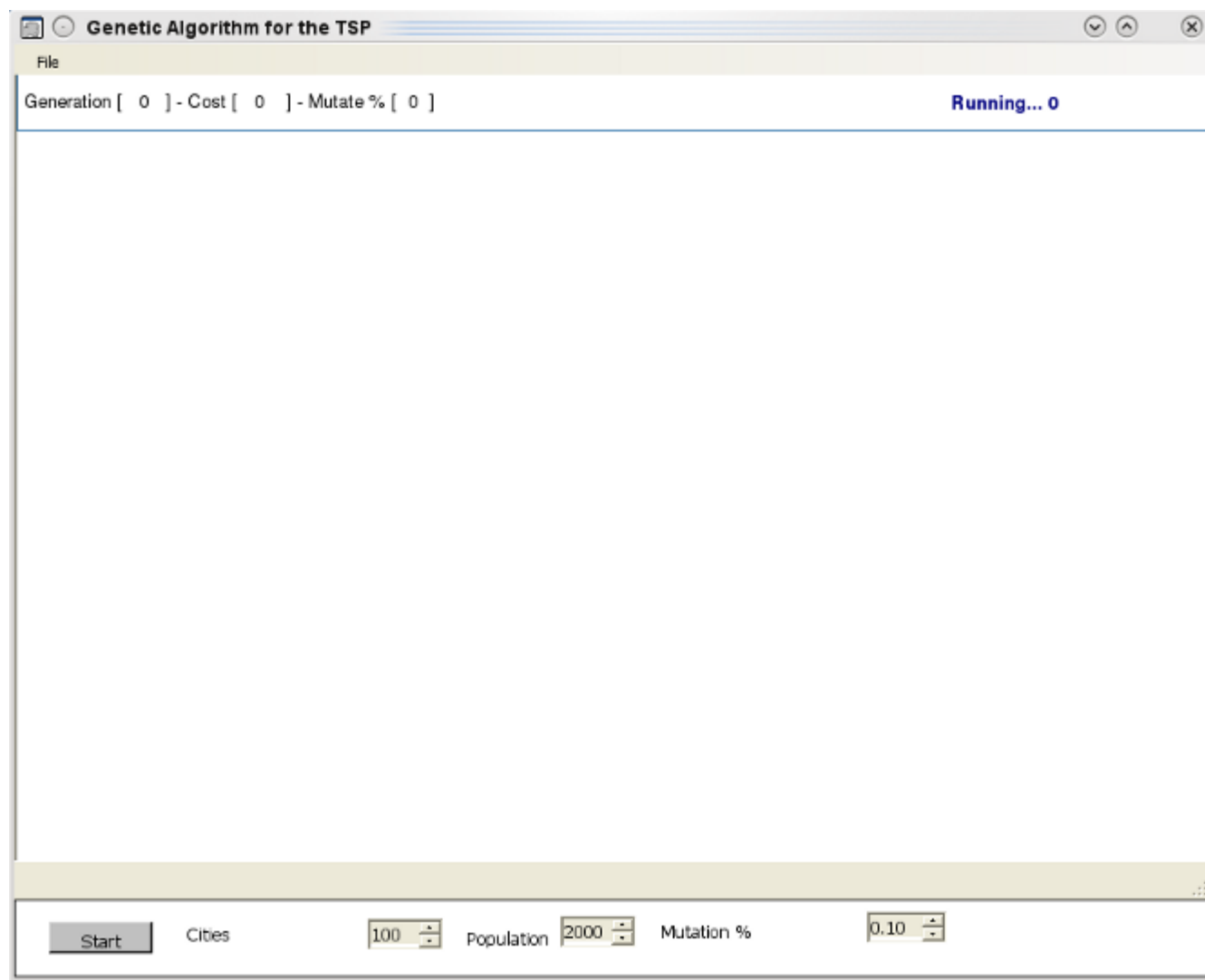


Fig 4 Aplicación en ejecución

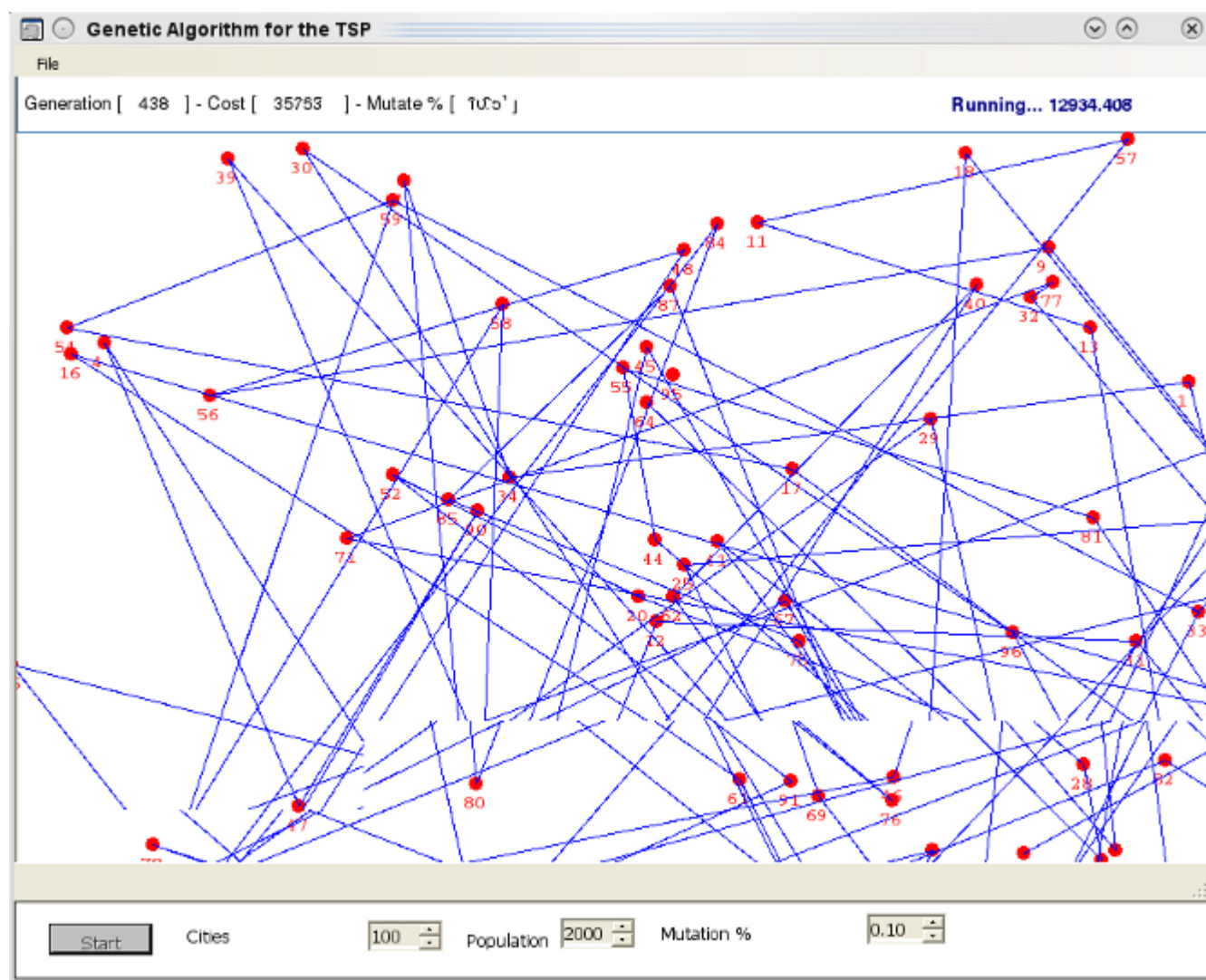
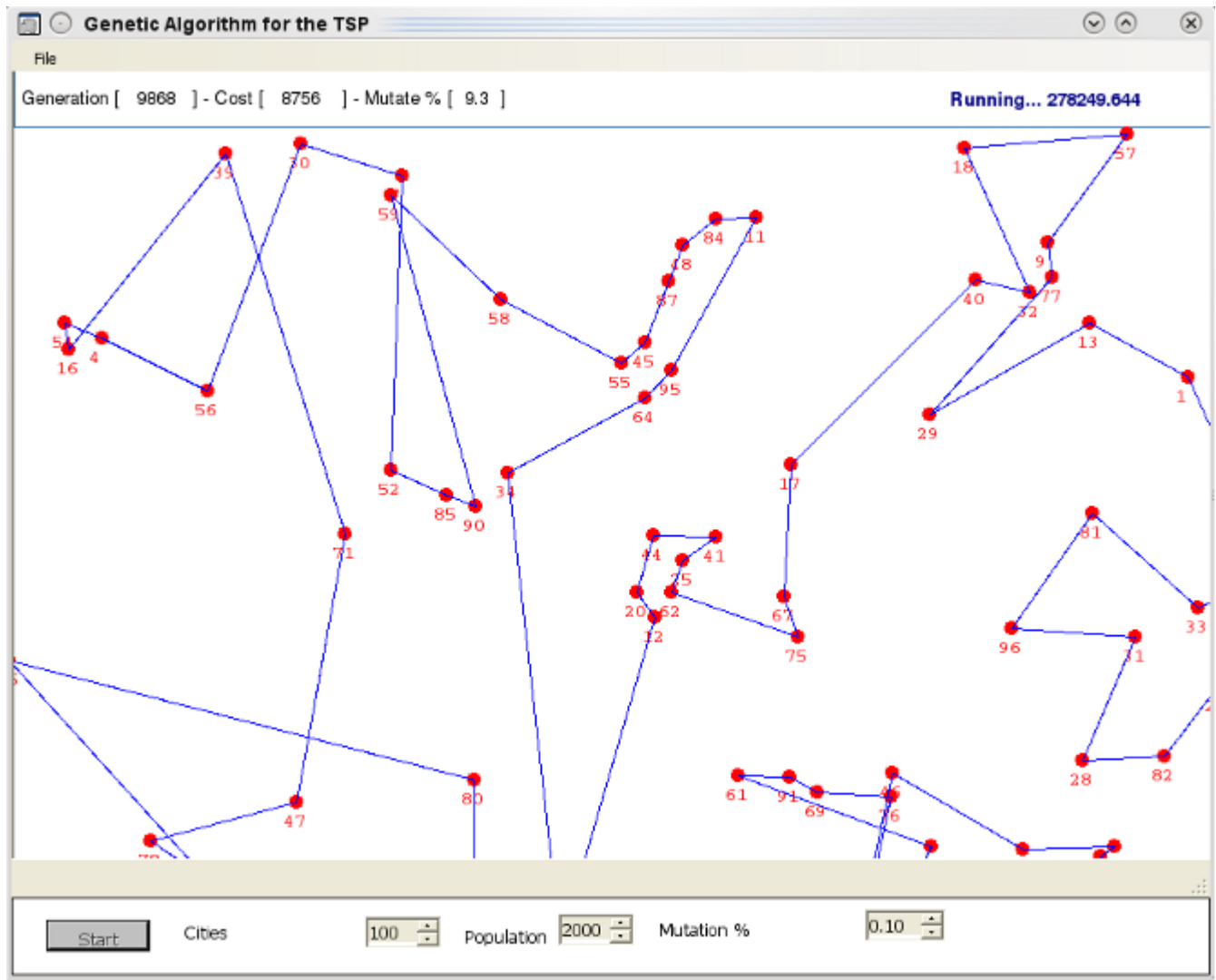


Fig 5 Resultado de la aplicación.



En su [web site](#) el autor *Jeff Heaton* además del código fuente del applet, ofrece libros acerca de inteligencia artificial y de programación con Java y C#.



[Descarga el código fuente en un proyecto para MonoDevelop o Visual Studio](#)