

Algoritmo genético para la optimización del picking en almacenes.

Cecilia Fili, Mía Torres López, Gonzalo Romero Román
Inteligencia Artificial II, Facultad de Ingeniería, Universidad Nacional de Cuyo

Resumen — En este resumen se busca describir la implementación de algoritmos genéticos para el ordenamiento de productos en un almacén. Los resultados obtenidos exhiben el funcionamiento del algoritmo y su efectividad para evitar máximos locales. Se consiguen así resultados suficientes para resolver el problema en cuestión.

Palabras claves — *Algoritmo Genético, Almacén, Optimización.*

I. INTRODUCCION

Los algoritmos genéticos son métodos adaptativos que pueden utilizarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de evolución de los organismos vivos. 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 relacionado con la idoneidad de dicha solución. Cuanto mayor sea la adaptación de un individuo al problema mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, produciendo nuevos individuos para actualizar la nueva población hasta conseguir una solución al problema.

En esta monografía aplicamos algoritmos genéticos al problema de optimizar el picking en un almacén.

II. PLANTEAMIENTO DEL PROBLEMA

El objetivo del problema es saber cuál es el mejor emplazamiento de productos en un almacén, teniendo en cuenta que el layout del almacén es fijo y sólo hay que decidir dónde colocar los productos, definiendo sus coordenadas. El almacén se ordenará de acuerdo a las órdenes recibidas, que deberán ser despachadas en su totalidad. Además, el picking debe comenzar y terminar en una misma bahía de carga y el costo de esta acción es proporcional a la distancia recorrida.

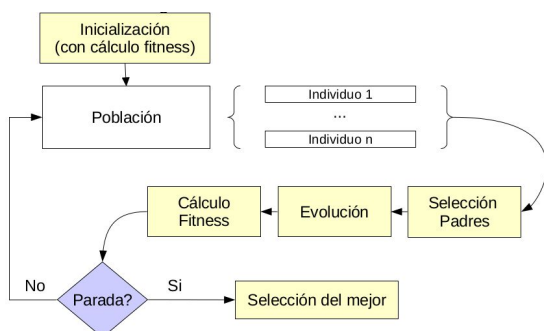
El algoritmo genera un conjunto de órdenes ficticias, simulando órdenes reales que el almacén debe satisfacer. Dichas órdenes tendrán los distintos artículos que se deben buscar en la acción de picking.

En la siguiente imagen se observa el layout del almacén, cuyo tamaño es modificable, pudiendo achicarse o ampliarse al quitar o agregar estanterías, cada una de las cuales almacena 8 productos distintos.

1	2	25	26
3	4	27	28
5	6	29	30
7	8	31	32
9	10	33	34
11	12	35	36
13	14	37	38
15	16	39	40
17	18	41	42
19	20	43	44
21	22	45	46
23	24	47	48
⋮			

III. DISEÑO Y METODOLOGÍA

Para resolver el problema se realizaron los siguientes pasos:



Se inicializa con una población aleatoria de almacenes. A través de una función de fitness (idoneidad) se calculan los mejores padres y se realiza la evolución por medio de acciones de cruce y mutación, obteniendo una nueva población con la misma cantidad de individuos. Ese ciclo se repite hasta cumplir con el criterio de parada. Una vez alcanzado dicho criterio finaliza la ejecución del programa obteniendo los resultados deseados.

Para poder resolver el problema de interés de esta monografía, se implementaron dos algoritmos genéticos.

El primero para poder determinar la mejor distribución de productos en el layout del almacén (cada posible distribución es un individuo de este algoritmo) y otro AG que se se utiliza como fitness del primero.

Este segundo algoritmo genético recibe una posible distribución de productos en el layout del almacén y tiene una población donde cada individuo es uno de los pedidos que recibe el almacén y cada gen es un producto a buscar. Entonces ordena los productos de cada pedido de modo tal que se busquen recorriendo la distancia más corta. La función de idoneidad que utiliza el segundo algoritmo es la distancia Manhattan entre los productos a buscar y hasta la bahía de carga. De este modo, devuelve al primer algoritmo el valor del camino más corto.

Utilizando el valor que recibe del segundo AG, el primer algoritmo puede hacer evolucionar su población al usar ese valor como fitness de cada individuo.

Para la selección de padres se utilizaron dos algoritmos diferentes y se comparó el desempeño de ambos. En una primera instancia se seleccionaron N padres (donde N es la población) a partir del método de la ruleta. Una vez seleccionados se crearon $2N$ hijos de los cuales se seleccionaron los mejores N para ser la próxima población. Debido al gran tiempo de ejecución, se implementó un algoritmo diferente con menor costo computacional. Se seleccionaron $N/2$ padres que a la vez forman parte de la nueva población, el $N/2$ restante se calcula a partir de dichos padres usando permutación de cruce de orden.

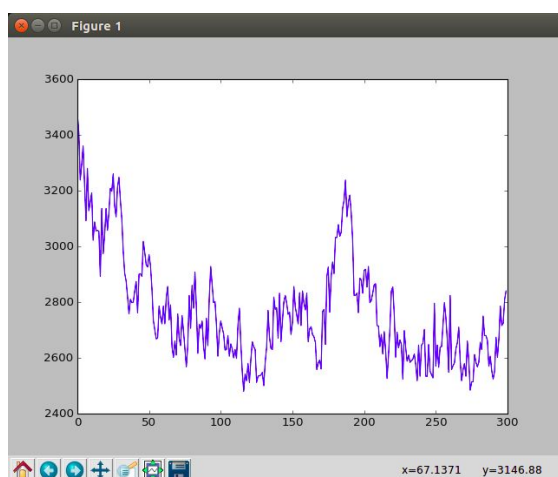
En cuanto a la mutación, se realiza una conmutación por intercambio de un solo

gen a partir de una probabilidad definida por usuario.

Como criterio de parada se utilizó la cantidad de iteraciones. Se almacenó en cada iteración el menor valor obtenido. De ésta forma aunque no se haya alcanzado un óptimo global se alcanza al menos el menor óptimo local de todas las combinaciones obtenidas.

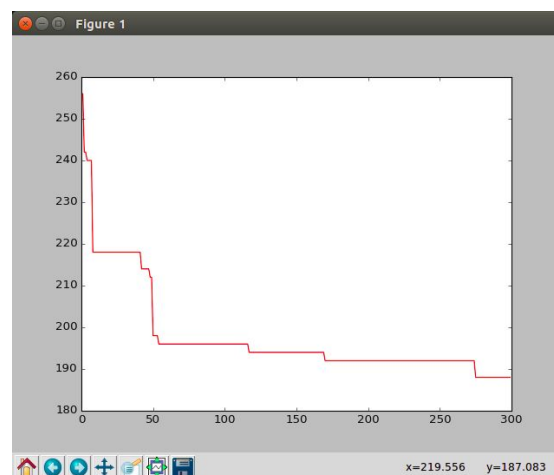
IV. RESULTADOS Y EVALUACIÓN

Se ejecutó el programa realizando trescientas iteraciones. Los resultados obtenidos se muestran a continuación en los siguientes gráficos:



En el esquema anterior, se grafica la suma de los valores de fitness de todos los almacenes que componen la población en función de las iteraciones realizadas. Se nota una tendencia a la disminución de dichos valores, lo que representa una menor distancia recorrida. Debido a la función de fitness utilizada para el segundo algoritmo genético es una simplificación muy relajada del problema (impuesta para disminuir el tiempo de cálculo), el algoritmo genético no converge a una solución óptima.

Para obtener mejores resultados, se busca en cada iteración comprobar si hay un valor de fitness menor al menor alcanzado hasta el momento y en caso de existir se actualizar dicho valor. A continuación se muestra los valores mínimos obtenidos en cada iteración.



Se puede apreciar una mejora apreciable de los resultados a medida que se ejecuta el algoritmo. Para un almacén de 16 productos y a partir de órdenes aleatorias calculadas al inicio del programa se obtiene el mejor almacén para dichas órdenes, este almacén se muestra a continuación.

```
[ '-', '-', '-', '-', '-', '-', '-']  
[ '-', 1, 2, '-', 3, 4, '-']  
[ '-', 16, 7, '-', 6, 5, '-']  
[ '-', 15, 14, '-', 13, 11, '-']  
[ '-', 8, 9, '-', 12, 10, '-']  
[ '-', '-', '-', '-', '-', '-', '-']
```

V. CONCLUSIÓN

En esta monografía se ha detallado la implementación de dos algoritmos genéticos que trabajan en conjunto para solucionar el problema de optimización del picking en un almacén. A partir de las simulaciones realizadas se observó que, si bien el algoritmo da resultados

aceptables, los resultados dependen del nivel de simplificación del problema. Con las condiciones impuestas para este trabajo, utilizando la distancia de Manhattan entre los productos del almacén como función de idoneidad del segundo algoritmo, se observó una mejora en los resultados pero sólo hasta cierto punto y luego comienza a oscilar y no converge, por lo que se debe utilizar como criterio de parada la cantidad de iteraciones que se realizan.

Como desarrollo futuro, se considera que agregando restricciones al problema se puede mejorar en gran medida la convergencia de las soluciones.

REFERENCIAS

[1]

<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>

[2]

B. Martín del Brío y A. Sanz Molina "Redes neuronales y sistemas borrosos"

[3]

M. Marchetta, "Sistemas evolutivos"
Apuntes de Cátedra, Inteligencia Artificial II