Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Técnicas de inteligencia artificial

Reporte: Primeros ejemplos con AG

BUAP

Docente: Abraham Sánchez López

Alumno	Matrícula
Taisen Romero Bañuelos	202055209

Algoritmos genéticos en R

En el documento se aborda una introducción breve a los algoritmos genéticos. Lo que más me gustó fueron los gráficos, me parecieron muy bonitos y el gráfico que se actualizaba con cada iteración me gustó mucho, no me esperaba ver eso. Me decepcionó un poco que el gráfico 3D no se pudiera mover (en realidad era 2D pero con un dibujo 3D).

Al inicio no terminé de entender bien qué representaban algunas cosas de los comandos nuevos, pero lo volví a leer y tampoco entendí pero me di cuenta de que lo que no entendía era la relación que tenían ciertos comandos desconocidos con los fundamentos teóricos de los AG, así que sólo fue cosa de leer el manual para darme una mejor idea de qué era lo que hacía cada cosa. Hecho esto procedí a modificar algunos parámetros para observar qué sucedía. Estos fueron los cambios realizados:

- Aumento de la población, de 100 → 200.
- Mayor número de iteraciones, de 100 → 300 generaciones.
- Cruce: de 0.8 → 0.9 para tener más combinaciones.
- Mutación: de 0.1 → 0.2 para evitar la convergencia prematura.
- Si la solución excede 10,000 kg, se reduce su aptitud a -500 (en lugar de 0) para forzar la exploración de soluciones viables (algo así como la penalización de errores más significativos).

Para empezar, el tiempo de ejecución fue significativamente mayor. En mi PC pasó de 1.01 segundos a 22007.52 segundos (equivalente a casi 6 horas. Realmente no tardó eso, pero fue lo que marcó la función toc()). Al parecer, el aumento de la población y las generaciones es computacionalmente costosísimo. El gráfico pese a verse más aburrido que el anterior, muestra que la mejor solución se estabiliza rápidamente alrededor de 10,000 desde las primeras generaciones. También, al principio hay más variabilidad, pero se reduce con el tiempo hasta que tiene un repunte al final (véase la sombra verde). Y finalmente, se muestra una tabla con los ítems seleccionados por el algoritmo. Como se ve, el motor ("Engine") representa la mayor parte del peso total.

```
> toc()
22137.09 sec elapsed
```

```
> df_sol
# A tibble: 5 × 4
# Groups:   item [5]
  item    weight  freq total_weight
  <chr>    <dbl> <int>        <dbl>
1 Bummper     17    24          408
2 Chasis     100    27         2700
3 Engine     158    35         5530
4 Seat        30    37         1110
5 Tires        7    36          252
> sum(df_sol$total_weight)
[1] 10000
```