

Práctica de laboratorio

Algoritmos de optimización multiobjetivo

Objetivos

- Adquirir experiencia práctica en el uso de algoritmos de optimización multiobjetivo para problemas sin restricciones.
- Adquirir experiencia sobre la utilización de métricas para la comparación de resultados.

Descripción

Consideremos la función ZDT3. Es un problema con un espacio de búsqueda de 30 dimensiones ($n=30$).

Las funciones a minimizar son:

$$f_1(\mathbf{x}) = x_1$$

$$f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x}))$$

donde:

$$h(f_1(\mathbf{x}), g(\mathbf{x})) = 1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})} - (f_1(\mathbf{x})/g(\mathbf{x})) \sin(10\pi f_1(\mathbf{x}))$$

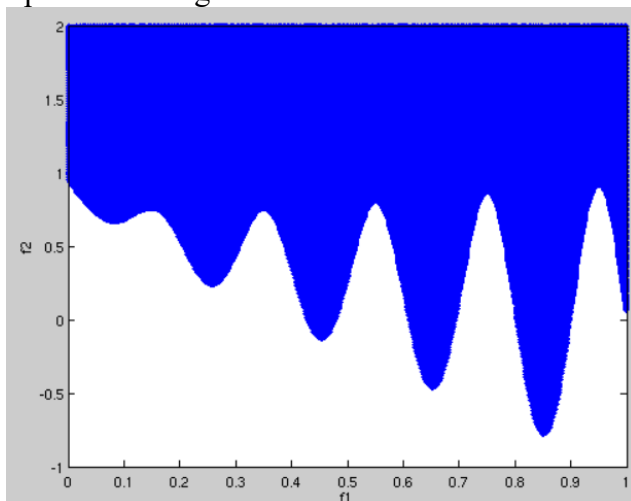
y

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

y el espacio de búsqueda es

$$0 \leq x_i \leq 1$$

La representación gráfica de dichas funciones es:



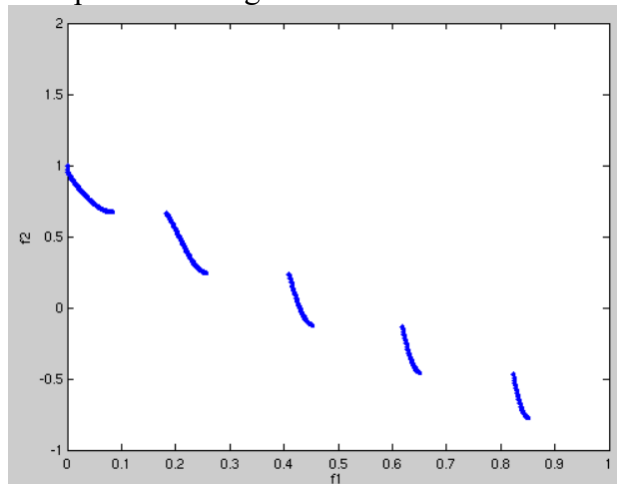
El frente Pareto óptimo es conocido y se localiza en

$$(f_1, 1 - \sqrt{f_1} - f_1 \sin(10\pi f_1))$$

con f_1 definido en:

$$f_1 \in [0.0, 0.0830015349] \cup (0.1822287280, 0.2577623634] \cup (0.4093136748, 0.4538821041] \cup (0.6183967944, 0.6525117038] \cup (0.8233317983, 0.8518328654]$$

La representación gráfica de dicho frente Pareto es:



Un muestreo de puntos del frente Pareto se encuentra en el fichero “PF.dat”.

- (a) En primer lugar vamos a tratar de obtener puntos de una aproximación a dicho frente Pareto mediante la utilización de un algoritmo conocido de un único objetivo, por ejemplo el algoritmo genético utilizado en prácticas anteriores. La función a optimizar se encuentra almacenada en el fichero “zdt3_ga.c”. Como se trata de un algoritmo de un único objetivo, transforme el problema de dos objetivos en un único objetivo mediante la suma con peso de los dos objetivos (edite el fichero anterior para ver un ejemplo en que los pesos de ambos objetivos es 0.5). Ensaye con distintos pesos de los objetivos, tratando de obtener distintos puntos de dicho frente. Utilice una población de 100 individuos y 100 generaciones. Realice 30 ejecuciones de cada prueba y examine los valores obtenidos de los objetivos f_1 y f_2 (línea “Subobjectives” en el fichero *ga_result.out* que se genera en la ejecución). ¿Qué conclusiones obtiene de los resultados?

Ahora representaremos en la misma gráfica las soluciones que hemos obtenido junto con el muestreo de puntos del frente Pareto. El enlace “Muestreo frente Pareto ideal” en el directorio GA en EV contiene un número elevado de puntos del frente idea. Por su parte, el fichero en el enlace “Puntos de varias ejecuciones de GA” contiene las soluciones obtenidas para varias ejecuciones con distintas combinaciones de pesos. Para representarlo se puede utilizar el siguiente comando de gnuplot:

plot 'PF.dat' using 1:2 with points pt 7 lt 1, 'ga.out' using 1:2 lt 3 pt 5

- (b) Utilice ahora el algoritmo NSGAI para obtener una aproximación al frente Pareto utilizando el paquete de ficheros disponible en EV. La función a optimizar se encuentra almacenada en el fichero “zdt3_nsga.c”. Algoritmo y función se encuentran ya compilados en el fichero “nsga2_zdt3”. Si por alguna razón este ejecutable no funcionara en su entorno local, compile de nuevo siguiendo las

instrucciones del documento adjunto de descripción de la implementación. Ejecute el algoritmo con una semilla 0.5 y una población de 100 individuos durante 100 generaciones, y utilice los parámetros de configuración por defecto de cada algoritmo descritos en el correspondiente documento de “Descripción implementación algoritmo NSGAI” disponibles en Enseñanza Virtual. Para ello, ejecute

```
.nsga2_zdt3 seed
```

donde *seed* es un número real entre 0.0 y 1.0. Utilice por ejemplo 0.5.

Haga que el software le muestre la evolución del conjunto de soluciones a lo largo de todas las generaciones (tenga en cuenta que el programa está intencionadamente ralentizado para que la visualización sea adecuada).

Por otra parte, visualice los ficheros de resultados de la ejecución. El fichero *all_pop.out* contiene la información de todas las soluciones de todas las generaciones, incluyendo objetivos, restricciones, variables del espacio de búsqueda, violación de restricciones (definida como: $CV = \sum_i |g_i(\mathbf{x})| \quad \forall i | g_i(\mathbf{x}) < 0$), orden del frente y distancia de crowding. El orden de dichas variables se encuentra en las primeras líneas del fichero. El fichero *final_pop.out* contiene la misma información, pero restringida a la última generación. Finalmente, el fichero *all_popm.out* contiene la misma información del fichero *all_pop.out* pero incluyendo únicamente las columnas relativas a objetivos y violación de restricciones, y eliminando todas las líneas de texto explicativas o que separan secciones. Es un fichero especialmente concebido para su uso como datos de entrada del software de métricas.

Compruebe que obtiene el mismo resultado utilizando:

```
.nsga2_zdt3 0.5 <zdt3p100g100.in
```

donde el fichero *zdt3p100g100.in* contiene las mismas respuestas que ha dado a las preguntas en la ejecución anterior.

- (c) A continuación utilizaremos el software de métricas (disponible en EV) para evaluar la calidad de los resultados. Tenga en cuenta que solo tiene sentido utilizar el software de métricas sobre resultados buenos. Primero tenemos que proporcionar la información necesaria para el software de métricas: cogeremos el fichero de resultados *all_popm.out* y lo copiaremos al directorio del software de métricas y le cambiaremos el nombre a *zdt3_all_popmp100g100.out*. En primer lugar, obtenga la métrica hypervolume y spacing de la última generación. Para ello, ejecute el software:

```
.metrics
```

y conteste a las preguntas guiándose por el contenido del fichero *zdt3p100g100one.in*.

Como podrá imaginar obtendrá los mismos resultados si ejecuta:

```
.metrics < zdt3p100g100one.in
```

Observe que por pantalla aparece el punto de referencia obtenido automáticamente para el cálculo del hipervolumen. Dicho punto se ha almacenado también en el fichero *hvrefer.out*. Observe que el fichero *zdt3p100g100singlerefer.in* contiene la misma información pero que el punto de referencia se introduce de forma manual. Compruebe que se obtiene el mismo resultado si ejecuta:

```
.lmetrics < zdt3p100g100singlerefer.in
```

Más adelante veremos ocasiones en que es necesario introducir manualmente el punto de referencia.

A continuación, obtenga hipervolumen y spacing en función del número de generaciones utilizando como datos de entrada del software metrics el fichero *zdt3p100g100all.in*. Compare con los resultados obtenidos para la última generación y discuta.

- (d) En el paquete de ficheros de trabajo en EV dispone de los resultados de 10 ejecuciones de otro algoritmo desconocido (UNKNOWN) para varios conjuntos de tamaño de población de soluciones y número de iteraciones. Copie los resultados de una de las ejecuciones para 100 soluciones y 100 iteraciones (directorio UNKNOWN) y asígnele el nombre *zdt3_all_popmp100g100_unknown.out*.

En primer lugar visualizaremos los frentes. Del fichero anterior, extraiga las últimas 100 líneas y almacénelas en un fichero de nombre *zdt3_last_unknown.out*. Para representar este frente junto con el frente ideal almacenado en el fichero *PF.dat*, arranque gnuplot y teclee:

```
gnuplot> plot 'PF.dat' using 1:2 with points pt 7 lt 2, 'zdt3_last_unknown.out' w points pt 6 ps 1 lt -1
```

Si desea comparar con el resultado de NSGAII extraiga las últimas 100 líneas del fichero *zdt3_all_popmp100g100.out* y almacénelas en el fichero *zdt3_last_nsgaii.out*. Para representarlos todos juntos teclee:

```
gnuplot> plot 'PF.dat' using 1:2 with points pt 7 lt 2, 'zdt3_last_unknown.out' w points pt 6 ps 1 lt -1, 'zdt3_last_nsgaii.out' w points pt 6 ps 1 lt 1
```

Una vez comprobado que los frentes son aceptables pasamos a utilizar métricas para la comparación de dos frentes entre sí. Obtenga el spacing, hipervolumen con un punto de referencia común y coverage set entre los dos frentes de la última generación utilizando el software de métricas y los datos de entrada almacenados en el fichero *zdt3p100g100stsingleCS.in*. Para ejecutar el software de métricas en este caso renombre el fichero de resultados *zdt3_all_popmp100g100.out* obtenido con NSGAII como *zdt3_all_popmp100g100_nsgaii.out*

Discuta los resultados.

Realice ahora la misma operación, pero calculando la evolución de las distintas métricas a lo largo de las distintas generaciones. Para ello ejecute el software de métricas y los datos de entrada almacenados en el fichero *zdt3p100g100stallCS.in*. Discuta los resultados.

- (e) Nunca se pueden extraer conclusiones con una única ejecución por lo que es necesario hacer un estudio estadístico. Ejecute el algoritmo con una población de 100 individuos durante 100 generaciones con distintas semillas. Para ello puede utilizar el script *zdt3p100g100script*. Si edita ese fichero podrá ver que ejecuta NSGAI1 10 veces con distintas semillas y va copiando los ficheros de resultados a un directorio (EVAL10000/P100G100) con distinto nombre para no sobrescribir los resultados de ejecuciones anteriores. Para ejecutarlo solo tiene que teclear:

```
./zdt3p100g100script
```

Observe los resultados de las ejecuciones.

Compare los resultados de NSGAI1 y los proporcionados para el algoritmo desconocido usando las métricas *hypervolume*, *coverage set* y *spacing*. Para ello primero tendrá que mover/copiar el directorio EVAL10000/P100G100 al directorio del software de métricas.

Para comparar el *spacing* y el *hypervolume* de todas las iteraciones de los dos algoritmos con un punto de referencia común dispone del script *zdt3p100g100allscript* en el directorio del software de métricas. Edite ese fichero para entender cómo se está haciendo el proceso. Para visualizar simultáneamente todas las gráficas de *hypervolume* y *spacing*, para ambos algoritmos a la vez o separadamente dispone de los scripts de gnuplot: *plotmethv*, *plotmethv_nsgai1*, *plotmethv_unknown*, *plotmetasp*, *plotmetasp_nsgai1* y *plotmetasp_unknown*. Para ejecutar el primero de ellos por ejemplo teclee dentro de gnuplot:

```
gnuplot> load 'plotmethv'
```

Para comparar el *spacing* y el *hypervolume* de la última iteración de los dos algoritmos con un punto de referencia común dispone del script *zdt3p100g100singlescript*¹ en el directorio del software de métricas². Edite ese fichero para entender cómo se está haciendo el proceso. Verá que el script calcula también el valor medio y la desviación estándar de *hypervolume* y *spacing*.

Finalmente, compare el *coverage set* de ambos algoritmos. El *coverage set* implica dos frentes por lo que es imprescindible seleccionar parejas de frentes, uno del algoritmo desconocido y uno de NSGAI1. Con el script *zdt3p100g100allCSscript* se obtienen las métricas de 10 parejas obtenidas con una selección una a una de cada algoritmo en función del número de iteraciones. Al igual que en el apartado anterior para cada pareja se obtiene no solamente el *coverage set* sino también el *spacing* de cada frente y el *hypervolume* de cada frente con un punto de referencia

¹ Tenga en cuenta que el script utiliza procesamiento con notación científica. Para garantizar que funciona correctamente Linux debe estar configurado adecuadamente. El procesamiento correcto se obtiene cuando LC_NUMERIC está establecido en configuración americana. Para ello, si está trabajando en bash simplemente tiene que poner en su ventana de comandos antes de ejecutar el script:
export LC_NUMERIC="en_US.UTF-8"

² Tenga en cuenta que los ficheros se han de adaptar a cada circunstancia particular. Además, es muy importante tener en cuenta que se está haciendo uso de scripts estándar de Linux para procesar los datos y ejecutar el software de métricas de forma repetida automáticamente, pero esta no es la única manera de hacerlo. Es lícito utilizar cualquier mecanismo que consiga los mismos resultados de ejecución repetida del software de métricas proporcionado. En documento adjunto dispone de una descripción detallada de los scripts.

común para esa pareja. Obviamente este hypervolume, limitado a una pareja, no tiene sentido en un contexto de comparación estadística.

Con el script *zdt3p100g100singleCSScript* puede realizar la misma comparación pero limitado a la última iteración.

- (f) Ejecute el algoritmo NSGAIi con distintas semillas y una población de 200 individuos durante 50 generaciones (mismo número de evaluaciones totales que en el caso anterior). Realice las mismas comparaciones con el algoritmo desconocido. Para ayudarse dispone de scripts similares a los apartados anteriores. Repita con una población de 40 individuos y 250 generaciones. Compare también la evolución temporal de la métrica coverage set, hypervolume y spacing con ejecuciones individuales de NSGAIi con el mismo número de evaluaciones totales. Discuta los resultados.