Práctica de laboratorio

Optimización multiobjetivo con restricciones

Objetivos

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

Descripción

Consideremos el problema CF6 definido por la minimización de las funciones:

$$f_1(\boldsymbol{x}) = x_1 + \sum_{j \in J_1} y_j^2$$

$$f_2(\boldsymbol{x}) = (1 - x_1)^2 + \sum_{j \in J_2} y_j^2$$

donde:

$$J_1 = \{ j \mid j \text{ es impar y } 2 \le j \le n \}$$

$$J_2 = \{ j \mid j \text{ es par y } 2 \le j \le n \}$$

y

$$y_j = \begin{cases} x_j - 0.8 x_1 \cos(6\pi x_1 + \frac{j\pi}{n}) & \text{si } j \in J_1 \\ x_j - 0.8 x_1 \sin(6\pi x_1 + \frac{j\pi}{n}) & \text{si } j \in J_2 \end{cases}$$

Las restricciones son:

$$x_2 - 0.8 x_1 \sin(6\pi x_1 + \frac{2\pi}{n}) - \operatorname{sgn}(0.5(1 - x_1) - (1 - x_1)^2) \sqrt{\left| 0.5(1 - x_1) - (1 - x_1)^2 \right|} \ge 0$$

$$x_4 - 0.8 x_1 \sin(6\pi x_1 + \frac{4\pi}{n}) - \operatorname{sgn}(0.25\sqrt{(1 - x_1)} - 0.5(1 - x_1)) \sqrt{\left| 0.25\sqrt{1 - x_1} - 0.5(1 - x_1) \right|} \ge 0$$

El espacio de búsqueda es

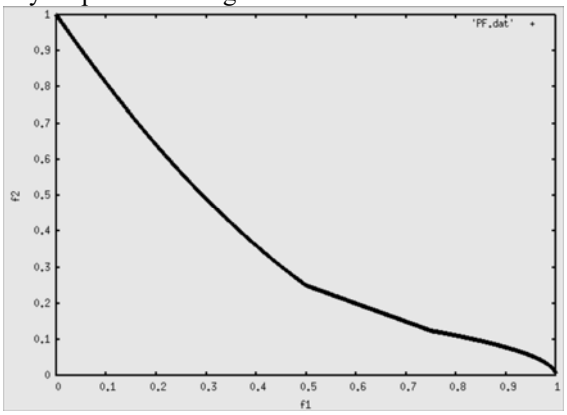
$$0 \le x_1 \le 1$$

$$-2 \le x_i \le 2 \quad i > 1$$

El frente Pareto óptimo es conocido y viene dado por la siguiente ecuación analítica:

$$f_2 = \begin{cases} (1-f_1)^2 & \text{si } 0 \le f_1 \le 0.5 \\ 0.5(1-f_1) & \text{si } 0.5 < f_1 \le 0.75 \\ 0.25\sqrt{(1-f_1)} & \text{si } 0.75 < f_1 \le 1 \end{cases}$$

cuya representación gráfica es:



Un muestreo de puntos del frente Pareto óptimo se encuentra en el fichero "PF.dat".

1. En primer lugar, consideraremos la función con un espacio de búsqueda de 4 dimensiones (n=4). La función a optimizar se encuentra almacenada en el fichero "cf6_4d.c". Algoritmo y función se encuentran ya compilados en el fichero "nsga2_cf6_4d". 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. Con un presupuesto de evaluaciones de las funciones del problema cifrado en 4000, pretendemos utilizar el número de generaciones y tamaño de la población que deseemos (limitado al presupuesto impuesto) y obtener un buen resultado de la optimización.
    a) Comience considerando una población de 100 individuos y 40 generaciones. Ejecute el algoritmo NSGAII con una semilla 0.5 mediante la instrucción

       ./nsga2_cf6_4d 0.5

       Considere los parámetros de configuración por defecto del algoritmo NSGAII y visualice la evolución de las soluciones a lo largo del número de generaciones. La herramienta de visualización está preparada para visualizar simultáneamente las soluciones junto con el frente ideal. Las soluciones se dibujan en color rojo cuando no se cumplen las restricciones y en negro cuando se cumplen. En apenas unas pocas generaciones las soluciones que no cumplen restricciones desaparecen porque NSGAII utiliza un mecanismo de manejo de restricciones basado en selección, en el cual se le da prioridad al cumplimiento de las restricciones. Puede corroborar esto editando el fichero *all_pop.out*, en el cual se muestran las soluciones de todas las generaciones. La columna *constr_violation* contiene la violación de restricciones y tiene el valor 0.0 cuando se cumplen.
       Compruebe que obtiene el mismo resultado cuando ejecuta:

       ./nsga2_cf6_4d 0.5 <cf6p100g40.in

       por lo que puede ejecutar variaciones más rápidamente simplemente realizando cambios sobre este fichero.

A continuación, pretendemos realizar la misma operación con otras combinaciones de tamaño de la población y número de generaciones. Se puede ejecutar para una población de 200 individuos y 20 generaciones mediante:

./nsga2_cf6_4d 0.5 <cf6p200g20.in

y para 40 individuos y 100 generaciones mediante:

./nsga2_cf6_4d 0.5 <cf6p40g100.in

Observe y discuta los resultados.

b) Vamos a aplicar métricas para comparar entre sí algunos resultados del apartado anterior. Copie los ficheros de resultados para métricas *all_popm.out* del apartado anterior al directorio de métricas con nombre *cf6_4d_all_popmp100g40.out*, *cf6_4d_all_popmp200g20.out* y *cf6_4d_all_popmp40g100.out*, respectivamente.
Ejecute la comparación de métricas entre parejas de las ejecuciones anteriores en función del número de generaciones. Puede ejecutar, por ejemplo, la comparación entre la ejecución con 100 individuos y 40 generaciones con 200 individuos y 20 generaciones con el fichero de configuración *cf6_4dp200g20vsp100g40stallCS.in* mediante:

./metrics < cf6_4dp200g20vsp100g40stallCS.in

Discuta los resultados.
Ejecute ahora la comparación entre métricas únicamente para la última generación de cada ejecución. Puede ejecutar, por ejemplo, la comparación entre la ejecución con 100 individuos y 40 generaciones con 200 individuos y 20 generaciones con el fichero de configuración *cf6_4dp200g20vsp100g40stsingleCS.in* mediante:

./metrics < cf6_4dp200g20vsp100g40stsingleCS.in

Discuta los resultados.

c) Puesto que no se pueden extraer conclusiones de una sola ejecución, generaremos varias ejecuciones para cada combinación de tamaño de la población y número de generaciones. El proceso se ha automatizado en los scripts *cf6_4dp100g40script*, *cf6_4dp200g20script* y *cf6_4dp40g100script*. Compruebe en dichos scripts que simplemente se ejecuta NSGAII con los mismos ficheros de entrada con distintas semillas y los resultados se almacenan con un nombre distinto. Para ejecutar, por ejemplo, el primero de ellos solo tiene que lanzar el primer script:

./cf6_4dp100g40script

Compruebe la evolución y resultados de las distintas ejecuciones. Discuta los resultados.

d) Ejecute el software de métricas para estudiar estadísticamente las ejecuciones con distintas combinaciones de tamaño de la población y número de generaciones. Considere en primer lugar la evolución con el número de generaciones. Para ayudarle en dicho estudio dispone de los siguientes scripts[1] de ejemplo:
cf6_4dp100g40vsp200g20vsp40g100allscript[2]: calcula spacing e hipervolumen con un punto de referencia común para todas las ejecuciones y todas las generaciones de los ficheros de datos considerados en ese script. Las gráficas se representan separadamente para cada ejecución. Puede representarlas conjuntamente ejecutando gnuplot y en el prompt del mismo ejecutar el script:

gnuplot> load 'plotmethv_cf6_4d'

para el hipervolumen y

gnuplot> load 'plotmetsp_cf6_4d'

para el spacing. En esos scripts se representan las 10 ejecuciones de las tres combinaciones de tamaño de la población y número de generaciones. Modifique adecuadamente dichos ficheros para representar un subconjunto de ellas.

cf6_4dp200g20vsp100g40allCSscript: calcula spacing, coverage set e hipervolumen con un punto de referencia común para todas las generaciones de las parejas de ficheros de datos considerados en ese script.
cf6_4dp200g20vsp40g100allCSscript: calcula spacing, coverage set e hipervolumen con un punto de referencia común para todas las generaciones de las parejas de ficheros de datos considerados en ese script.
cf6_4dp100g40vsp40g100allCSscript: calcula spacing, coverage set e hipervolumen con un punto de referencia común para todas las generaciones de las parejas de ficheros de datos considerados en ese script.
La descripción de dichos scripts se encuentra en el documento "Descripción de implementación scripts masivo métricas en problema de optimización con restricciones" en Enseñanza Virtual. No olvide que en los scripts anteriores se asume que los ficheros de datos de las múltiples ejecuciones de NSGAII se han almacenado en una localización determinada.
Discuta los resultados obtenidos.

[1] 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 lo scripts.

[2] 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:
LC_NUMERIC="en_US" bash

A continuación, considere la comparación de métricas considerando únicamente la última generación de cada ejecución.

Para ayudarle en dicho estudio dispone de los siguientes scripts[3] de ejemplo:

cf6_4dp100g40vsp200g20vsp40g100singlescript: calcula spacing e hipervolumen con un punto de referencia común para la última generación de todas las ejecuciones de los ficheros de datos considerados en ese script. También se obtiene el valor medio y desviación estándar de dichas métricas para cada combinación de tamaño de la población y número de generaciones

cf6_4dp200g20vsp100g40singleCSscript: calcula spacing, coverage set hipervolumen con un punto de referencia común para la última generación de las parejas de ficheros de datos considerados en ese script. El coverage set se almacena en los ficheros tmp_cf6_4d/cs_total.out y tmp_cf6_4d/cs2_total.out por si se quieren procesar.

cf6_4dp200g20vsp40g100singleCSscript: calcula spacing, coverage set hipervolumen con un punto de referencia común para la última generación de las parejas de ficheros de datos considerados en ese script. El coverage set se almacena en los ficheros tmp_cf6_4d/cs_total.out y tmp_cf6_4d/cs2_total.out por si se quieren procesar.

cf6_4dp100g40vsp40g100singleCSscript: calcula spacing, coverage set hipervolumen con un punto de referencia común para la última generación de las parejas de ficheros de datos considerados en ese script. El coverage set se almacena en los ficheros tmp_cf6_4d/cs_total.out y tmp_cf6_4d/cs2_total.out por si se quieren procesar.

La descripción de dichos scripts se encuentra en el documento "Descripción de implementación scripts masivo métricas en problema de optimización con restricciones" en Enseñanza Virtual.

Discuta los resultados obtenidos.

2. Repita y compare los resultados cuando el espacio de búsqueda tiene 16 dimensiones (n=16). El presupuesto de evaluaciones de las funciones del problema es 10000. La función a optimizar se encuentra almacenada en el fichero "cf6_16d.c". Algoritmo y función se encuentran ya compilados en el fichero "nsga2_cf6_16d". 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. Para ayudarle en dicho estudio dispone de ficheros de entrada y scripts de ejecución masiva de NSGAII y cálculo de métricas con una nomenclatura similar al apartado anterior, con las lógicas adaptaciones.

[3] 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 lo scripts.