Un esquema general de búsqueda local en programación entera. Evaluación computacional

Mayor Gallego, José A.
Ruiz Canales, Pascual
Dpto. Estadística e I.O. Universidad de Sevilla. Spain. **E-mail:mayor@cica.es**

En este trabajo, estudiamos un procedimiento general de búsqueda local basad0 en un sistema probabilístico de entornos, que puede ser combinada con técnicas de "annealing" simulado y de banda inferior.

Para ello, se estudia un sistema de entornos definidos sobre conjuntos discretos y que incorporan información proporcionada por el gradiente de la función objetivo.

Adicionalmente, se estudia el comportamiento de este tipo de entornos en combinación con técnicas de "simulated annealing" en lo que respecta a la convergencia asintótica a óptimos globales.

Finalmente se exponen una serie de resultados computacionales obtenidos al aplicar estas técnicas a varios problemas de programación entera, tanto lineal como no lineal.

Introdución

Las técnicas de búsqueda local permiten obtener soluciones aproximadas a problemas de optimización en condiciones muy generales, siendo de especial interés el caso en el cual el conjunto factible es discreto, lo que incluye la optimización combinatoria y la programación entera en general.

Consideraremos el problema siguiente,

$$\min_{x \in S \cap Z^n} f(x) \qquad \text{con} \qquad S \subseteq R^n$$

para el cual suponemos,

$$\exists\ \nabla f(x)\ \forall x \in S$$

Denotando el conjunto factible, $S \cap Z^n$ por F, podemos dar las siguientes definiciones,

DEFINICION 1 *Un sistema probabilístico de entornos sobre F es una función,*

$$\theta : F \times F \longrightarrow [0,1]$$

verificando,

$$\sum_{t \in F} \theta(x,t) = 1 \qquad \forall x \in F$$

DEFINICION 2 *Dado $x \in F$, al conjunto,*

$$N(\theta, x) = \{t \in F | \theta(x,t) > 0\}$$

le denominaremos θ-entorno de x.

El θ-entorno formaliza la idea de accesibilidad desde un punto del conjunto factible a otro, cuantificando el grado de accesibilidad mediante una probabilidad. En la práctica, la distribución de probabilidad $\theta(x,t)$ es utilizada para obtener un elemento $t \in F$ usando algún procedimiento de selección con probabilidades variables como el método acumulativo o el de Lahiri. En general, supondremos que $x \notin N(\theta, x)$.

El siguiente teorema indica como es posible obtener un sistema probabilístico de entornos sobre el conjunto F a partir de un sistema probabilístico de entornos sobre un conjunto que contiene a F.

TEOREMA 1 *Sea $U \subset Z^n$ tal que $F \subset U$, y sea θ_U un sistema probabilístico de entornos sobre U. Dado $x \in F$, supongamos que $F \cap N(\theta_U, x) \neq \emptyset$, y consideremos el siguiente procedimiento para obtener un elemento $t \in F$. Generamos $t \in U$ usando $\theta_U(x,t)$. Si $t \in F$ lo aceptamos, en caso contrario repetimos la generación hasta obtener un elemento $t \in F$. Entonces, este procedimiento induce un sistema probabilístico de entornos sobre F dado por,*

$$\theta(x,t) = \frac{\theta_U(x,t)}{P[F \cap N(\theta_U, x)]}$$

DEFINICION 3 *Al sistema probabilístico de entornos θ, obtenido en el teorema anterior se le denomina* **reducción** *a F del sistema probabilístico θ_U.*

Dado un sistema probabilístico de entornos, θ, vamos a considerar el siguiente proceso de búsqueda local **generalizada**, donde la notación $u \sim U[0,1)$ indica que u es una realización de una variable aleatoria uniforme en $[0,1)$,

ALGORITMO 1 (BUSQUEDA LOCAL GENERALIZADA)

1. *Obtener $x \in F$.*
2. *Si se verifica el CRITERIO DE DETENCIÓN, parar el proceso, siendo x la solución. En caso contrario, ir al paso 3.*
3. *Usando la distribución θ generar $t \in N(\theta, x)$.*
4. *Generar $u \sim U[0,1)$. Si $u \leq G(f(t), f(x))$ hacer $x := t$.*
5. *Ir al paso 2.*

Observemos que el criterio introducido en el paso 4 generaliza al criterio usual de simple mejora, para el cual se tiene,

$$G(a,b) = \begin{cases} 1 & a \leq b \\ 0 & a > b \end{cases}$$

y para el método de "annealing" simulado,

$$G(a,b) = \begin{cases} 1 & a \leq b \\ e^{-(a-b)/T} & a > b \end{cases}$$

Es evidente que la elección adecuada de θ puede influir sobre la eficiencia del procedimiento. Una solución simple consiste en tomar,

$$\theta(x,t) = \frac{1}{|N(\theta, x)|}$$

es decir, la distribución de probabilidad uniforme sobre $N(\theta, x)$. Sin embargo, podemos pensar que la incorporación de información sobre f al sistema de entorno puede originar mejores resultados. A continuación estudiaremos una clase de sistema probabilístico de entornos que incorpora información de la función objetivo basada en el gradiente de esta.

Entornos de gradiente

Volviendo al problema de minimización planteado, vamos a definir el θ-entorno de un punto $x \in F$ de la siguiente forma.

0.1. Soporte

Denotando,

$$\mathcal{T}^l = \{T \in \{0,1,-1\}^n | \sum_{i=1}^{n} |T_i| = l\} \qquad l \leq n$$

definimos,

$$C^l(x) = \{x + T | T \in \mathcal{T}^l\} \cap F \qquad l \leq n$$

que nos sirven para construir el "soporte" del entorno,

$$B^k(x) = \cup_{l=1}^{k} C^l(x) \qquad 1 \leq k \leq n$$

La construcción del sistema de entornos se realiza de una forma inversa pues primero hemos definido su soporte, es decir, el conjunto de puntos "accesibles" desde x.

En los siguientes diagramas se indican los conjuntos anteriores para $n = 2$.

$B^1(x) = C^1(x)$

$$\begin{array}{ccc} & \bullet & \\ \bullet & x & \bullet \\ & \bullet & \end{array}$$

$C^{(}x)2$

$$\begin{array}{ccc} \bullet & & \bullet \\ & x & \\ \bullet & & \bullet \end{array}$$

$B^2(x) = C^1(x) \cup C^2(x)$

$$\begin{array}{ccc} \bullet & \bullet & \bullet \\ \bullet & x & \bullet \\ \bullet & \bullet & \bullet \end{array}$$

Este tipo de soporte ya ha sido definido anteriormente en Beato et al. (1992), con distribución de probabilidad uniforme sobre el mismo, y ha sido aplicado a problemas de programación lineal entera.

Distribución de probabilidad

El siguiente paso consiste en asignar las probabilidades. Para ello, vamos a considerar que, en base a minimizar la función objetivo, las mejores "direcciones" de movimiento son aquellas para las cuales el gradiente es más negativo. Ello nos induce a construir el vector T de la siguiente forma, para el soporte $B^k(x)$,

1. Hacer $T := (0, 0, \ldots, 0)$.

2. Generar aleatoriamente un número entero, l, entre 1 y k, que será el número de coordenadas que van a ser modificadas, en +1 ó -1.

3. Dado l, seleccionar, sin reemplazamiento, y probabilidades p_i, $1 \leq i \leq n$, l componentes del vector T, siendo $p_i > 0$, $\forall i$. Para el cálculo de p_i proponemos usar

una función que sea tanto mayor cuanto más negativo sea la componente i-ésima del gradiente. Una función de este tipo es, por ejemplo,

$$f(x) \propto e^{-\lambda x}$$

siendo λ una constante.

4. Una vez seleccionadas las componentes, hay que asignar el "sentido" de movimiento, pues resulta deseable que incluso movientos desfavorables puedan ser seleccionados, aunque con menor probabilidad.

   Para ello, consideramos una variable aleatoria de Bernoulli, $Z \sim Be(p)$, y para cada componente de T seleccionada en el paso anterior fue i, hacemos,

$$T_i = 2 \times Z - 1$$

   siendo el parámetro p de la forma,

$$p = \begin{cases} 1-\alpha & \nabla_i f(x) \leq 0 \\ \alpha & \nabla_i f(x) > 0 \end{cases}$$

   siendo α un parámetro próximo a cero. En el caso extremo $\alpha = 0$, T_i será +1 si el gradiente es mayor que cero, y -1 en caso contrario.

Una vez construido T, haremos $t := x + T$. En el caso de que $N(\theta, x)$ tenga elementos fuera del conjunto factible, se aplicará la reducción.

Al entorno definido del la forma mencionada le denotaremos θ_k^α.

Propiedades del entorno

1. Si F es tal que $\forall x \in F$, $\exists T \in \mathcal{T}^1$ tal que $x + T \in F$ y $\alpha > \beta$, siendo β una constante estrictamente positiva, entonces el entorno θ_k^α es **fuertemente conexo**, es decir,

$$\forall x, y \in F \quad \exists x_0, x_1, \ldots, x_n \in F \quad \text{siendo} \quad x_0 = x \quad x_n = y \quad \text{verificando}$$

$$x_l \in N(\theta_k^\alpha, x_{l-1}) \qquad l = 1, 2, \ldots, n$$

   así, todo elemento de F es accesible desde cualquier otro elemento mediante movimientos locales.

2. Si $\alpha > \beta$, siendo β una constante estrictamente mayor que cero, entonces el entorno es **débilmente simétrico**, es decir,

$$\theta_k^\alpha(x,t) > 0 \Longleftrightarrow \theta_k^\alpha(t,x) > 0$$

Planteamos ahora el estudio del comportamiento asintótico del entorno definido, en combinación con la técnica de "simulated annealing". Como es sabido, si el entorno es simétrico y fuertemente conexo, dicho método converge a un mínimo global.

En nuestro caso, tenemos un sistema de entornos débilmente simétrico (relajación de la simetría), no obstante, se sigue verificando la convergencia asintótica.

TEOREMA 2 *Dado α fijo, si θ_k^α es fuertemente conexo y débilmente simétrico, la cadena de Markov definida por la matriz $\{P_k^\alpha(x,t;\lambda)\}_{x,t \in F}$ dada por,*

$$P_k^\alpha(x,t;\lambda) = \begin{cases} \theta_k^\alpha(x,t) & f(t) \leq f(x) \quad x \neq t \\ \theta_k^\alpha(x,t) e^{z(\lambda)(f(t)-f(s))} & f(t) > f(x) \quad s \neq t \\ 1 - \sum_{t \neq x} P_k^\alpha(x,t;\lambda) & \qquad\qquad\quad x = t \end{cases}$$

tiene una distribución estacionaria $\pi_k^\alpha(\cdot,\lambda)$ que verifica,

$$\lim_{\lambda \longmapsto 0} \pi_k^\alpha(x;\lambda) > 0 \qquad x \in \mathcal{M}$$

$$\lim_{\lambda \longmapsto 0} \pi_k^\alpha(x;\lambda) = 0 \qquad x \notin \mathcal{M}$$

siendo $\mathcal{M}$ el conjunto de mínimos globales de la función f en F y $z(\lambda)$ una función real de variable real verificando,

$$\lim_{\lambda \longmapsto 0} z(\lambda) = -\infty$$

Resultados computacionales

Hemos aplicado la búsqueda local con el sistema de entornos indicado a varios problemas test. En todos ellos, empleamos θ_1^α y $\alpha = 0{,}25$.

Para evitar la convergencia a óptimos locales, usamos una técnica de banda inferior, consistente en tomar la función,

$$G(a,b) = G(a,b;\varepsilon) = \begin{cases} 1 & a \leq b+\varepsilon \\ \\ 0 & a > b+\varepsilon \end{cases}$$

con ε tendiendo a 0.

Para modificar ε, en la k-ésima iteración calculamos $\varepsilon_k = \varepsilon_{k-1} \times r$, donde r es práximo a 1. Por ejemplo, $r \in [0{,}99, 0{,}9999]$.

En los resultados, P indica el tanto por ciento de ejecuciones en las que se ha obtenido el óptimo, μ la media de iteraciones necesarias y ε_0 el valor inicial de este parámetro.

Fuentes : Beato et al. (1992), Hock and Schittkowski (1981) y Schittkowski (1987).

Adicionalmente, consideramos el problema de afijación óptima en muestreo estratificado.

TIPO: Programación lineal entera.

FUENTE: Beato et al. (1992)

PROBLEMA:

```
MAX          14 X1 + 7 X2 - 20 X3 - 11 X4 - 6 X5

SUJETO A

             16 X1 +  9 X2 +  6 X3 +  7 X4 +  4 X5  <= 85

             -6 X1 + 11 X2 -  8 X3 - 15 X4 + 14 X5  <= 39

            -20 X1 -  3 X2 + 10 X3 + 19 X4 + 16 X5  <= 93

                    Xi >= 0 i=1,2,3,4,5
```

UNA SOLUCIÓN ÓPTIMA ENTERA: (3,4,0,0,0)

VALOR ÓPTIMO: 70

RESULTADOS:

$$P = 100\,\% \qquad \mu = 6 \qquad \varepsilon_0 = 0 \qquad \lambda = 5$$

TIPO: Programación lineal entera.

FUENTE: Beato et al. (1992)

PROBLEMA:

```
MAX

-5 X1 -16 X2 -15 X3 -2 X4 -17 X5 +4 X6 -3 X7 -14 X8 +19 X9 +8 X10


SUJETO A
          9 X2 - 18 X3 - 17 X4 +  4 X5 - 19 X6 - 14 X7
                              - 13 X8         - 15 X10   <= 154

 1 X1 + 12 X2 - 11 X3 + 18 X4 + 11 X5         -  7 X7
                              - 10 X8 -  1 X9 +  4 X10   <= 113

 4 X1 +  1 X2 + 14 X3 + 23 X4 + 12 X5 + 13 X6 + 10 X7
                              + 19 X8 + 24 X9 +  1 X10   <= 30

10 X1 - 13 X2 + 16 X3 -  7 X4 + 14 X5 +  7 X6 - 20 X7
                              - 19 X8 +  2 X9 -  5 X10   <= 100

 9 X1 + 14 X2 -  1 X3 - 20 X4 - 11 X5 +  2 X6 +  3 X7
                                      +  1 X9 - 10 X10   <= 131


                       Xi >= 0 i=1,..10
```

UNA SOLUCIÓN ÓPTIMA ENTERA: (0,0,0,0,0,0,0,0,0,28)

VALOR ÓPTIMO: 224

RESULTS:

$$P = 100\,\% \qquad \mu = 260 \qquad \varepsilon_0 = 20 \qquad r = 0{,}995 \qquad \lambda = 15$$

TIPO: Programación entera no lineal.

FUENTE: Hock and Schittkowski (1981)

PROBLEMA:

```
MIN

                 2 - (X1 X2 X3 X4 X5) / 120


SUJETO A

                 0 <= Xi <= i   i=1,2,3,4,5
```

UNA SOLUCIÓN ÓPTIMA ENTERA: (1,2,3,4,5)

VALOR ÓPTIMO: 1

RESULTADOS:

$$P = 100\,\% \qquad \mu = 10 \qquad \varepsilon_0 = 0 \qquad \lambda = 1$$

TIPO: Programación no lineal entera.

FUENTE: Hock and Schittkowski (1981)

PROBLEMA:

```
MIN

                 2 - (X1 X2 X3 X4 X5) / 120


SUJETO A
                            2
                 0 <= Xi <= i      i=1,2,3,4,5
```

UNA SOLUCIÓN ENTERA: (1,4,9,16,25)

VALOR ÓPTIMO: -118

RESULTADOS:

$$P = 100\,\% \qquad \mu = 50 \qquad \varepsilon_0 = 0 \qquad \lambda = 10$$

TIPO: Programación no lineal entera.

FUENTE: AFIJACIÓN ÓPTIMA EN MUESTREO ALEATORIO ESTRATIFICADO

PROBLEMA:

$$\min \sum_{i=1}^{10} \frac{A_i}{X_i} \qquad A_i = 10,\ i = 1, \ldots, 10$$

SUJETO A

$$\sum_{i=1}^{10} X_i \leq 500 \qquad X_i \geq 2$$

SOLUCIÓN ÓPTIMA:

$$X_i = 50 \quad i = 1, \ldots, 10$$

VALOR ÓPTIMO: 2

RESULTADOS:

$$\varepsilon_0 = 5 \qquad \lambda = 1000$$

En el 100 % de las ejecuciones hemos obtenidos cuasi-soluciones, x, verificando,

$$\left|\frac{f(x) - f_{\text{opt}}}{f_{\text{opt}}}\right| < 0{,}001$$

con $\mu = 1{,}3 \times 10^4$

TIPO: Programación no lineal entera.

FUENTE: AFIJACION OPTIMA EN MUESTREO ALEATORIO ESTRATIFICADO

PROBLEMA:

$$\min \sum_{i=1}^{50} \frac{A_i}{X_i} \qquad A_i = 10,\ i = 1, \ldots, 10$$

SUJETO A

$$\sum_{i=1}^{50} X_i \leq 500 \qquad X_i \geq 2$$

SOLUCIÓN ÓPTIMA:

$$X_i = 10 \quad i = 1, \ldots, 50$$

VALOR ÓPTIMO: 50

RESULTADOS:

$$\varepsilon_0 = 5 \qquad \lambda = 1000$$

En el 100 % de las ejecuciones hemos obtenidos cuasi-soluciones, x, verificando,

$$\left|\frac{f(x) - f_{\text{opt}}}{f_{\text{opt}}}\right| < 0{,}001$$

con $\mu = 1{,}1 \times 10^4$

Referencias

[1] Aart, E., Korst, J. (1989). “Simulated Annealing and Boltzmann Machines”. John Wiley & Sons.

[2] Beato Moreno, A., Mayor Gallego, J.A., Rufián Lizana, A., Ruiz Canales, P. (1992). “On Integer Multicriteria Programming Problems”. EURO XII/TIMS XXXI, Helsinky.

[3] Hock, W., Schittkowski, K. (1981). “Test Examples for Nonlinear Programming Codes”. Springer-Verlag.

[4] Mayor Gallego, J.A. (1992). “Optimización con estructuras aleatorias”. Tésis Doctoral (no publicada). Universidad de Sevilla.

[5] Schittkowski, K. (1987). “More Test Examples for Nonlinear Programming Codes”. Springer-Verlag.