Universidad Nacional de La Plata
Facultad de Ciencias Exactas
Departamento de Matemática

Trabajo de Tesis Doctoral
**Un método de filtros sin derivadas
para programación no lineal**

Raúl P. Vignau
Directora: Nélide Echebest
Codirectora: María Laura Schuverdt

2014

Índice general

1. Introducción	3
2. Sistemas indeterminados de ecuaciones no lineales	7
2.1. Búsqueda lineal no monótona sin derivadas	9
2.2. Algoritmo espectral sin derivadas (DF-SAUNE)	13
2.3. Algoritmo cuasi-Newton sin derivadas (DF-QNB)	18
2.3.1. Análisis del caso $M = 1$	24
2.4. Experimentación numérica	26
2.4.1. Problemas test	26
2.4.2. Implementación	27
2.4.3. Resultados numéricos	28
3. Sistemas indeterminados de ecuaciones no lineales con restricciones de cotas en las variables	31
3.1. Algoritmo general y búsqueda lineal no monótona sin derivadas	32
3.2. Algoritmo cuasi-Newton sin derivadas con cotas en las variables (BCDF-QNB)	35
3.3. Experimentación numérica	41
3.3.1. Problemas test	41
3.3.2. Implementación	43
3.3.3. Resultados numéricos	44
4. Algoritmo de restauración inexacta sin derivadas utilizando filtros	45
4.1. Algoritmo de filtros sin derivadas (DFF)	48
4.2. Algoritmos internos	55
4.2.1. Fase de restauración	55
4.2.2. Fase de optimalidad	58
4.3. Resultados de convergencia	65
4.3.1. Convergencia a puntos Karush-Kuhn-Tucker	68
4.4. Experimentación numérica	71
4.4.1. Detalles sobre la implementación del Algoritmo DFF	71
4.4.2. Problemas test	72

4.4.3. Resultados numéricos	74
5. Conclusiones y trabajo futuro	79
A. Proyecciones ortogonales	81
Bibliografía	84

Capítulo 1

Introducción

En este trabajo de tesis estamos interesados en resolver el siguiente problema de programación no lineal:

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeto a } c(x) = 0 \end{aligned} \tag{1.1}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es llamada función objetivo y $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, es una función vectorial cuyas componentes son funciones $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ para $i = 1, \dots, m$. La función c define la región factible $\{x \in \mathbb{R}^n : c(x) = 0\}$ y todo punto perteneciente a este conjunto será llamado punto factible. Asumimos en este trabajo que las funciones f y c_i , para $i = 1, \dots, m$, son continuamente diferenciables pero sus derivadas no están disponibles.

Problemas en los cuales se desconocen las derivadas pueden provenir de diferentes campos de la vida real, como la física y la ingeniería, en los cuales es necesario minimizar funciones cuyas derivadas no están disponibles (ver por ejemplo [2, 10, 23]). Técnicas irrestrictas basadas en exploraciones locales, en búsquedas lineales o en modelos cuadráticos han sido adecuadamente adaptadas para resolver problemas de optimización sin derivadas con restricciones de cotas en las variables y restricciones lineales [3, 9, 11, 24, 27, 28, 42, 43]. Problemas de optimización con restricciones más generales son más difíciles de resolver ya que se necesita obtener optimalidad y factibilidad controlando el número de evaluaciones funcionales, tanto de la función objetivo f como de las restricciones no lineales $c_1, \dots, c_m$. La idea del método de Lagrangiano Aumentado ha sido considerada en [14, 30, 31] para resolver problemas de optimización sin derivadas con restricciones más generales.

Los métodos de restauración inexacta para problemas de optimización con restricciones constan de dos fases [19, 34, 35, 36]. Una primera fase llamada fase de restauración o factibilidad y una segunda fase llamada fase de optimalidad o minimización. En la fase de restauración, se mejora la infactibilidad sin realizar evaluaciones de la función objetivo. En la fase de optimalidad, la función objetivo o una función Lagrangiana es minimizada. Las fases de restauración y optimalidad pueden combinarse usando funciones de penalidad, Lagrangianos Aumentados o pueden tratarse independientemente. Los algoritmos de restauración

inexacta descriptos por Martínez [34] y por Martínez y Pilotta [35, 36], miden el progreso del método utilizando una función de mérito. Gonzaga, Karas y Vanti [19] propusieron un algoritmo de restauración inexacta que usa la estrategia de filtros para la elección de los próximos iterados. Esta idea fue propuesta originalmente por Fletcher y Leyffer [17] en otro contexto.

Un artículo reciente [8] usa un método de restauración inexacta para resolver un problema sin derivadas de optimización no lineal donde las derivadas de las restricciones están disponibles pero no lo están las derivadas de la función objetivo. En este caso, la fase de minimización debe ser resuelta usando un método sin derivadas. Para tal propósito fue empleado un algoritmo introducido por Kolda, Lewis y Torczon [24] para optimización sin derivadas con restricciones lineales.

En este trabajo de tesis, como mencionamos anteriormente, estamos interesados en resolver el problema (1.1) con una metodología sin derivadas. El método propuesto es un método de restauración inexacta que emplea el concepto de filtros para la aceptación de los siguientes iterados.

Para comenzar con el desarrollo de esta tesis, en el Capítulo 2, nos abocamos al estudio de resolver el problema de encontrar un punto x^* que satisfaga el sistema de ecuaciones no lineales $c(x) = 0$. Es decir, trabajamos sobre la fase de restauración, diseñando dos algoritmos sin derivadas que resuelvan el sistema $c(x) = 0$. En el Capítulo 3, extendimos uno de los métodos presentados en el Capítulo 2 para resolver el sistema $c(x) = 0$ incluyendo restricciones de cotas en las variables. Tal método es finalmente utilizado en la fase de restauración en el Capítulo 4 donde se desarrolla un algoritmo de restauración inexacta sin derivadas para resolver el problema (1.1). En cada uno de los capítulos mencionados se incluyen los respectivos teoremas de convergencia y resultados numéricos de los métodos presentados.

Finalmente, en el Capítulo 5, se presentan las conclusiones sobre los resultados de los métodos introducidos en esta tesis. Así como también las posibles líneas de trabajo a futuro.

Notaciones:

- $\|\cdot\|$ denota la norma euclídea.
- Dada una matriz $B \in \mathbb{R}^{m \times n}$, $\mathcal{N}(B)$ denota el espacio nulo o núcleo de la matriz B .
- Para $i = 1, \dots, n$; e^i es el vector canónico en $\mathbb{R}^n$.
- I_n denota la matriz identidad en $\mathbb{R}^{n \times n}$.
- Cuando $h(x) = \frac{1}{2}\|c(x)\|^2$ denotaremos por $g(x)$ al gradiente de $h(x)$ y en este caso $g(x) = \nabla h(x) = J_c(x)^T c(x)$.
- Dada una sucesión $\{x^k\}_{k \in \mathbb{N}}$, para $k = 1, 2, \dots$, una subsucesión de $\{x^k\}_{k \in \mathbb{N}}$ será indicada por $\{x^k\}_{k \in K}$ donde K es algún conjunto infinito de índices.
- Dadas dos funciones no negativas $g_1, g_2 : X \rightarrow \mathbb{R}$, $X \subset \mathbb{R}^n$, denotamos $g_1(x) = O(g_2(x))$ en $\Gamma \subset X$ (o equivalentemente $g_2(x) = \Omega(g_1(x))$) si existe una constante $M > 0$ tal que $g_1(x) \leq M g_2(x)$ para todo $x \in \Gamma$.

Capítulo 2

Sistemas indeterminados de ecuaciones no lineales

En este capítulo consideramos el problema de determinar una solución $x^* \in \mathbb{R}^n$ que verifique el sistema de ecuaciones no lineales

$$c(x) = 0 \tag{2.1}$$

donde $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $c = (c_1, \dots, c_m)$ y cada función c_i , $i = 1, \dots, m$, es continuamente diferenciable y el número de ecuaciones es menor al número de incógnitas, es decir, $m \leq n$. Estamos interesados en sistemas para los cuales la matriz Jacobiana de c , denotada por $J_c(x)$, no está disponible o requiere una cantidad prohibitiva de almacenamiento. Esta situación es común cuando los valores funcionales provienen de medidas experimentales de la física, química o economía por citar algunos ejemplos.

Tal tipo de problemas también aparece como el conjunto factible de problemas de programación no lineal. Nuestra principal motivación es la aplicación, en el capítulo 4, del método propuesto como subalgoritmo para encontrar puntos factibles en algoritmos sin derivadas para resolver problemas de programación no lineal, tales como algoritmos de dos fases [6, 20, 33, 48], métodos factibles [1, 26, 38, 39, 40, 44, 45, 46, 47] y métodos de restauración inexacta [34, 35].

La resolución de sistemas cuadrados de ecuaciones no lineales empleando métodos sin derivadas han sido tratados usando aproximaciones del residuo espectral en [25] y métodos cuasi-Newton en [29]. Algunas de las ideas que aparecen en estos trabajos son incorporadas en el presente capítulo. En [25] los autores definen un algoritmo sin derivadas para resolver sistemas de ecuaciones cuadrados no lineales llamado DF-SANE (derivative-free spectral algorithm for nonlinear equations). Además demuestran la convergencia global del método usando una búsqueda lineal no monótona sin derivadas que combina la estrategia de Grippo, Lampariello y Lucidi [21] con el esquema de Li y Fukushima [29]. En [29] un método cuasi-

Newton basado en la actualización de Broyden y una búsqueda lineal no monótona sin derivadas es definido para el caso de sistemas cuadrados. También en [7] fue presentado un método cuasi-Newton inexacto que utiliza una técnica de búsqueda lineal similar a la introducida en [29] y usando la matriz $B_k = J_c(x^k)$ cada cierto número fijo de iteraciones. Bajo hipótesis apropiadas los autores demostraron propiedades de convergencia global y convergencia superlineal en ambos trabajos.

En este capítulo vamos a presentar dos métodos para resolver sistemas no cuadrados de ecuaciones no lineales basados en las ideas que aparecen en [25] y [29]. Los algoritmos propuestos usan una generalización de la búsqueda lineal no monótona sin derivadas definida en [25]. La dirección de búsqueda en [25] es calculada usando el vector residual $c(x^k)$ y un coeficiente espectral [4]. En uno de los algoritmos propuestos para resolver sistemas indeterminados el iterado actual es calculado considerando un número fijo de L_m direcciones que son la solución de algún sistema cuadrado usando un coeficiente espectral siguiendo las ideas de [25]. La dirección en el segundo algoritmo es calculada como una solución aproximada de un sistema lineal usando la fórmula adaptada de Broyden no cuadrada. Está bien establecido en la literatura que para el caso cuadrado esta fórmula es la aproximación secante más usada y tiene muy buenos resultados de convergencia local [13]. Como mencionamos antes la fórmula de Broyden fue previamente usada por Li y Fukushima. En [29] la dirección actual es la solución del sistema lineal

$$B_k d + c(x^k) = 0 \quad (2.2)$$

y la actualización de la matriz B_{k+1} está definida por

$$B_{k+1} = B_k + \beta_k \frac{(y^k - B_k s^k)(s^k)^T}{\|s^k\|^2}, \quad (2.3)$$

si $s^k \neq 0$, donde $y^k = c(x^k) - c(x^{k-1})$, $s^k = x^k - x^{k-1}$ y el parámetro β_k es elegido de manera que B_{k+1} sea no singular siendo $|\beta_k - 1| \leq \bar{\beta} < 1$. En nuestro algoritmo, cuando existe una solución del sistema lineal (2.2), esta es usada como dirección de búsqueda. Cuando no existe solución resolvemos el sistema lineal aproximadamente como fue propuesto en [32]. Consecuentemente se evita así la necesidad de elegir apropiadamente el parámetro β_k .

Bajo hipótesis adecuadas demostramos propiedades de convergencia global para las sucesiones generadas por ambos métodos. Los resultados de convergencia global obtenidos para el algoritmo basado en las ideas espectrales extienden los resultados de [25]. Para el método cuasi-Newton que usa la fórmula de actualización de Broyden obtenemos resultados de convergencia global usando una condición de tipo Dennis-Moré [13]. Mostraremos también que esta condición puede ser omitida para un caso particular de la búsqueda lineal sin derivadas. Así, ambos algoritmos pueden ser vistos como extensiones de los métodos definidos en [25] y [29] para el caso de sistemas no lineales no cuadrados.

2.1. Búsqueda lineal no monótona sin derivadas

En esta sección presentamos la búsqueda lineal no monótona sin derivadas que usaremos en los métodos definidos en las siguientes secciones. Consideramos la función de mérito continuamente diferenciable $h : \mathbb{R}^n \rightarrow \mathbb{R}$, que consiste en una medida del residuo de c en x , $h(x) = \frac{1}{2}\|c(x)\|^2$. Como mencionamos antes la estrategia está basada en la búsqueda lineal propuesta en [25]. Dados el iterado actual x^k y una dirección de búsqueda d^k , el algoritmo busca una longitud de paso α_k tal que

$$h(x^k + \alpha_k d^k) \leq \max_{0 \leq j \leq M-1} h(x^{k-j}) + \eta_k - \gamma \alpha_k^2 h(x^k) \quad (2.4)$$

donde M es un entero no negativo, $0 < \gamma < 1$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$ siendo $\eta_k > 0$ para todo $k \geq 0$.

Este procedimiento combina la búsqueda lineal no monótona para optimización sin restricciones introducida por Grippo, Lampariello y Lucidi [21]:

$$h(x^k + \alpha_k d^k) \leq \max_{0 \leq j \leq M-1} h(x^{k-j}) + \gamma \alpha_k \nabla h(x^k)^T d^k, \quad (2.5)$$

con el esquema sin derivadas de Li y Fukushima [29]:

$$\|c(x^k + \alpha_k d^k)\| \leq (1 + \eta_k) \|c(x^k)\| - \gamma \alpha_k^2 \|d^k\|^2. \quad (2.6)$$

Esta estrategia combinada (2.4) produce una búsqueda lineal no monótona robusta sin derivadas que toma en cuenta las ventajas de ambos esquemas. La búsqueda lineal (2.4) está fuertemente basada en el hecho que la dirección de búsqueda proviene del vector residual. Para más detalles ver el trabajo [25]. En dicho trabajo implementaron y testearon esta estrategia usando un gran conjunto de experimentos numéricos que mostraron su competitividad para sistemas de ecuaciones cuadrados no lineales.

En este trabajo, dada una dirección de búsqueda arbitraria d^k , basados en (2.5) y (2.6), consideramos la siguiente condición de búsqueda:

$$h(x^k + \alpha_k d^k) \leq \max_{0 \leq j \leq M-1} h(x^{k-j}) + \eta_k - \gamma \alpha_k^2 \|d^k\|^2, \quad (2.7)$$

donde $\|d^k\|^2$ toma el lugar de $h(x^k)$ en (2.4), obteniendo así una estrategia más general. Por completitud establecemos aquí la implementación del proceso.

Algoritmo 2.1.1. *Búsqueda lineal no monótona*

Dados $x^k, d \in \mathbb{R}^n$, $0 < \tau_{min} < \tau_{max} < 1$, $0 < \gamma < 1$, $M \in \mathbb{N}$, $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\eta_k > 0$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$

Paso 1 Calcular $\bar{h}_k = \max\{h(x^k), \dots, h(x^{\max\{0, k-M+1\}})\}$

$$\alpha_+ = \alpha_- = 1$$

Paso 2 Si $h(x^k + \alpha_+ d) \leq \bar{h}_k + \eta_k - \gamma \alpha_+^2 \|d\|^2$,

$$\text{definir } d^k = d, \alpha_k = \alpha_+, x^{k+1} = x^k + \alpha_k d^k$$

si no

$$\text{si } h(x^k - \alpha_- d) \leq \bar{h}_k + \eta_k - \gamma \alpha_-^2 \|d\|^2,$$

$$\text{definir } d^k = -d, \alpha_k = \alpha_-, x^{k+1} = x^k + \alpha_k d^k$$

si no

$$\text{elegir } \alpha_{+new} \in [\tau_{min} \alpha_+, \tau_{max} \alpha_+], \alpha_{-new} \in [\tau_{min} \alpha_-, \tau_{max} \alpha_-]$$

$$\alpha_+ = \alpha_{+new}, \alpha_- = \alpha_{-new} \text{ e ir al paso 2.}$$

Proposición 2.1.2. *El Algoritmo 2.1.1 está bien definido.*

Demostración: Como cada η_k es positivo, la condición

$$h(x^k + \alpha_+ d) \leq \bar{h}_k + \eta_k - \gamma \alpha_+^2 \|d\|^2$$

necesariamente se cumple después de un número finito de reducciones de α_+ .

$$\text{Análogamente para la condición } h(x^k - \alpha_- d) \leq \bar{h}_k + \eta_k - \gamma \alpha_-^2 \|d\|^2. \quad \square$$

Los nuevos algoritmos para resolver el problema (2.1) siguen el siguiente procedimiento.

Algoritmo 2.1.3. *Algoritmo general*

Dados x^0 , $M \in \mathbb{N}$, $0 < \tau_{min} < \tau_{max} < 1$, $0 \leq \epsilon < 1$, $0 < \gamma < 1$, $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\eta_k > 0$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$.
Considerar $k \leftarrow 0$.

Paso 1 Si $\|c(x^k)\| \leq \epsilon \max\{1, \|c(x^0)\|\}$ detener el proceso.

Paso 2 Calcular una dirección de búsqueda d^k .

Paso 3 Encontrar α_k usando el Algoritmo 2.1.1 y definir $x^{k+1} = x^k + \alpha_k d^k$.

Actualizar $k \leftarrow k + 1$ e ir al paso 1.

Considerando el proceso anterior es posible obtener resultados que serán usados en las próximas secciones para demostrar resultados de convergencia global. Las demostraciones de las Proposiciones 2.1.4 y 2.1.5 que siguen a continuación están basadas en las Proposiciones 2 y 3 de [25] adaptadas para la condición de búsqueda lineal (2.7).

Proposición 2.1.4. Si se considera, para cada $k \in \mathbb{N}$, $U_k = \max\{h(x^{(k-1)M+1}), \dots, h(x^{kM})\}$ y se define $\nu(k) \in \{(k-1)M+1, \dots, kM\}$ el índice para el cual $h(x^{\nu(k)}) = U_k$, entonces para todo $k = 1, 2, \dots$ se verifica que

$$h(x^{\nu(k+1)}) \leq h(x^{\nu(k)}) + \eta$$

donde $\eta = \sum_{i=0}^{\infty} \eta_i$.

Demostración: Tenemos que

$$\begin{aligned} h(x^{kM+1}) &\leq \max\{h(x^{(k-1)M+1}), \dots, h(x^{kM})\} + \eta_{kM} - \gamma \alpha_{kM}^2 \|d^{kM}\|^2 \\ &= U_k + \eta_{kM} - \gamma \alpha_{kM}^2 \|d^{kM}\|^2 \leq U_k + \eta_{kM} \end{aligned}$$

entonces

$$\begin{aligned} h(x^{kM+2}) &\leq \max\{h(x^{(k-1)M+2}), \dots, h(x^{kM+1})\} + \eta_{kM+1} - \gamma \alpha_{kM+1}^2 \|d^{kM+1}\|^2 \\ &\leq \max\{U_k, h(x^{kM+1})\} + \eta_{kM+1} \leq U_k + \eta_{kM} + \eta_{kM+1}. \end{aligned}$$

Así por argumento inductivo obtenemos que:

$$h(x^{kM+l}) \leq U_k + \sum_{j=0}^{l-1} \eta_{kM+j} - \gamma \alpha_{kM+l-1}^2 \|d^{kM+l-1}\|^2,$$

para $l = 1, 2, \dots$

Como $\nu(k+1) \in \{kM+1, \dots, kM+M\}$ tenemos que

$$U_{k+1} = h(x^{\nu(k+1)}) \leq U_k + \sum_{j=0}^{M-1} \eta_{kM+j} - \gamma \alpha_{\nu(k+1)-1}^2 \|d^{\nu(k+1)-1}\|^2.$$

Entonces para todo $k = 1, 2, \dots$ obtenemos que

$$h(x^{\nu(k+1)}) \leq h(x^{\nu(k)}) + \sum_{j=0}^{M-1} \eta_{kM+j} - \gamma \alpha_{\nu(k+1)-1}^2 \|d^{\nu(k+1)-1}\|^2 \leq h(x^{\nu(k)}) + \eta$$

como queríamos demostrar. $\square$

Proposición 2.1.5.

$$\lim_{k \rightarrow \infty} \alpha_{\nu(k)-1}^2 \|d^{\nu(k)-1}\|^2 = 0.$$

Demostración: Para todo $k = 1, 2, \dots$ tenemos que

$$h(x^{\nu(k+1)}) \leq h(x^{\nu(k)}) + \sum_{j=0}^{M-1} \eta_{kM+j} - \gamma \alpha_{\nu(k+1)-1}^2 \|d^{\nu(k+1)-1}\|^2.$$

Escribiendo esta última desigualdad para $k = 1, 2, \dots, L$ y sumando estas L desigualdades obtenemos que

$$h(x^{\nu(L+1)}) \leq h(x^{\nu(1)}) + \sum_{j=M}^{(L+1)M-1} \eta_j - \gamma \sum_{j=1}^L \alpha_{\nu(j+1)-1}^2 \|d^{\nu(j+1)-1}\|^2.$$

Por lo tanto

$$\gamma \sum_{j=1}^L \alpha_{\nu(j+1)-1}^2 \|d^{\nu(j+1)-1}\|^2 \leq h(x^{\nu(1)}) + \sum_{j=M}^{(L+1)M-1} \eta_j - h(x^{\nu(L+1)}) \leq h(x^{\nu(1)}) + \eta.$$

Así la serie $\sum_{j=1}^{\infty} \alpha_{\nu(j+1)-1}^2 \|d^{\nu(j+1)-1}\|^2$ es convergente y $\lim_{k \rightarrow \infty} \alpha_{\nu(k)-1}^2 \|d^{\nu(k)-1}\|^2 = 0$ como queríamos demostrar. $\square$

Proposición 2.1.6. *La sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo general está contenida en*

$$\Gamma = \{x \in \mathbb{R}^n : h(x) \leq h(x^{\nu(1)}) + \eta\}.$$

Demostración: De la Proposición 2.1.4 tenemos que $h(x^{\nu(k+1)}) \leq h(x^{\nu(1)}) + \eta$ para todo $k \geq 1$.

Por lo tanto $h(x^{k+1}) \leq h(x^{\nu(k+1)}) \leq h(x^{\nu(1)}) + \eta$ como queríamos demostrar. $\square$

Observación 2.1.7. *Los resultados obtenidos hasta ahora dependen fuertemente de la técnica de búsqueda lineal sin tener en cuenta la manera en la cual la dirección d^k del paso 2 del Algoritmo 2.1.3 fue calculada.*

De ahora en adelante consideramos el conjunto

$$K = \{\nu(1) - 1, \nu(2) - 1, \nu(3) - 1, \dots\} \quad (2.8)$$

y de la Proposición 2.1.5 obtenemos que

$$\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0. \quad (2.9)$$

Observar que de las demostraciones de las Proposiciones 2.1.5 y 2.1.6 obtenemos los siguientes resultados:

Proposición 2.1.8. *Si se considera $M = 1$ en el Algoritmo 2.1.1 entonces*

- *la sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo general está contenida en $\Gamma_0 = \{x \in \mathbb{R}^n : h(x) \leq h(x^0) + \eta\}$,*
- *la serie $\sum_{k=1}^{\infty} \alpha_k^2 \|d^k\|^2$ es convergente.*

2.2. Algoritmo espectral sin derivadas (DF-SAUNE)

En esta sección desarrollamos un método sin derivadas basado en el Algoritmo DF-SANE [25] adaptado para el caso de sistemas indeterminados de ecuaciones no lineales. DF-SANE es un método sin derivadas para resolver sistemas de ecuaciones cuadrados no lineales que usa el vector residual n -dimensional como dirección de búsqueda junto con la longitud del paso espectral y una estrategia de globalización que produce un proceso no monótono. El coeficiente espectral es el inverso de una aproximación del cociente de Rayleigh con respecto a la aproximación secante de la Jacobiana:

$$\sigma_k = \frac{\langle s^k, s^k \rangle}{\langle y^k, s^k \rangle}, \quad (2.10)$$

donde $y^k = c(x^k) - c(x^{k-1})$ y $s^k = x^k - x^{k-1}$ (ver [4, 25]).

El proceso iterativo en DF-SANE puede ser visto como un método cuasi-Newton considerando la matriz $B_k = \frac{1}{\sigma_k} I_n$ junto con la iteración $x^{k+1} = x^k - \alpha_k B_k^{-1} c(x^k) = x^k - \alpha_k \sigma_k c(x^k)$ donde α_k es calculada usando la estrategia no monótona sin derivadas (2.4).

Para resolver el caso indeterminado proponemos combinar la idea del algoritmo de Jacobiano aumentado (ver por ejemplo [52]), con el vector residual espectral explicado anterior-

mente. Para esto consideramos L_m como el número parte entera de $\frac{n}{m}$, esto es,

$$L_m = \begin{cases} \frac{n}{m} & \text{si } \frac{n}{m} \in \mathbb{N} \\ \lfloor \frac{n}{m} \rfloor + 1 & \text{si no.} \end{cases}$$

Para cada $j = 0, \dots, L_m - 1$ definimos las matrices $E_j \in \mathbb{R}^{m \times n}$ de la siguiente manera. Si $L_m = \frac{n}{m}$ entonces E_j es la matriz cuyas filas son los m vectores canónicos $e^{jm+1}, e^{jm+2}, \dots, e^{j(m+m)}$ de $\mathbb{R}^n$. Si $L_m = \lfloor \frac{n}{m} \rfloor + 1$ entonces para $j = 0, \dots, L_m - 2$, E_j son las mismas matrices que se definieron para el caso anterior y E_{L_m-1} es la matriz cuyas filas son los m vectores canónicos $e^{(L_m-1)m+1}, \dots, e^n, e^1, \dots, e^{L_m m - n}$ de $\mathbb{R}^n$.

Por ejemplo, si el número de incógnitas de nuestro sistema fuera $n = 6$ y el número de ecuaciones $m = 2$, tendríamos que $L_m = 3$ y definiríamos entonces las matrices E_0, E_1 y E_2 de la siguiente manera:

$$E_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, E_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Consideremos por ejemplo que el número de incógnitas de nuestro sistema fuera $n = 6$ y el número de ecuaciones $m = 4$. En este caso $\frac{n}{m} = \frac{6}{4} = 1,5 \notin \mathbb{N}$, por lo tanto $L_m = 2$ y definiríamos entonces las siguientes matrices:

$$E_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, E_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Dado x^k , para $j = 0, \dots, L_m - 1$ consideramos la dirección $d^{k+j} \in \mathbb{R}^n$ como la solución del sistema cuadrado lineal

$$\frac{1}{\sigma_{k+j}} \begin{pmatrix} E_j \\ V_j \end{pmatrix} d^{k+j} = \begin{pmatrix} -c(x^{k+j}) \\ 0 \end{pmatrix} \quad (2.11)$$

donde E_j fue definida anteriormente, $V_j \in \mathbb{R}^{(n-m) \times n}$ es la matriz cuyas filas son los vectores canónicos de $\mathbb{R}^n$ que generan a $\mathcal{N}(E_j)$ y σ_{k+j} es un coeficiente espectral. Podemos observar que $d^{k+j} = -\sigma_{k+j}^{-1} E_j^T c(x^{k+j})$ y

$$\|d^{k+j}\|^2 = \sigma_{k+j}^2 \|c(x^{k+j})\|^2. \quad (2.12)$$

Una vez que la dirección d^{k+j} es obtenida, realizamos la búsqueda lineal usando el Algoritmo 2.1.1.

Observar que (2.11) es un sistema cuadrado que reemplaza al usado en DF-SANE para obtener la dirección de búsqueda.

Dado un punto inicial arbitrario $x^0 \in \mathbb{R}^n$, el Algoritmo DF-SAUNE (derivative-free spectral algorithm for underdetermined nonlinear equations) que nos permite obtener el próximo iterado es el siguiente:

Algoritmo 2.2.1. *Algoritmo DF-SAUNE*

Dados x^0 , $0 < \gamma < 1$, $0 < \tau_{min} < \tau_{max} < 1$, $\sigma_0 = 1$,

$0 < \sigma_{min} < \sigma_{max} < \infty$, $0 \leq \epsilon < 1$.

Considerar $k \leftarrow 0$.

Paso principal Dados x^k , σ_k .

1. Si $\|c(x^k)\| \leq \epsilon \max\{1, \|c(x^0)\|\}$, detener el proceso.

2. Para $j = 0 : L_m - 1$

- Calcular $d^{k+j} = -\sigma_{k+j} E_j^T c(x^{k+j})$
- Encontrar α_j usando el Algoritmo 2.1.1 y definir $x^{k+j+1} = x^{k+j} + \alpha_j d^{k+j}$.
- Si $\|c(x^{k+j+1})\| \leq \epsilon \max\{1, \|c(x^0)\|\}$, detener el proceso.
- Calcular $s = x^{k+j+1} - x^{k+j}$,
 $y = c(x^{k+j+1}) - c(x^{k+j})$ y $\sigma = \frac{\langle E_j s, E_j s \rangle}{\langle y, E_j s \rangle}$.
- Si $|\sigma| \in [\sigma_{min}, \sigma_{max}]$ definir $\sigma_{k+j+1} = \sigma$.
Si no, elegir σ_{k+j+1} tal que $|\sigma_{k+j+1}| \in [\sigma_{min}, \sigma_{max}]$

Fin.

3. Actualizar $k \leftarrow k + L_m$ y repetir el paso principal.

Fin.

Por (2.9) tenemos que

$$\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0.$$

Así, usando (2.12) y considerando que cada $|\sigma_k| \in [\sigma_{min}, \sigma_{max}]$ obtenemos que

$$\lim_{k \in K} \alpha_k^2 h(x^k) = 0. \quad (2.13)$$

En el siguiente teorema probamos el principal resultado de convergencia asociado al Algoritmo DF-SAUNE. La demostración sigue las ideas del Teorema 1 de [25] adaptada para nuestro algoritmo.

Teorema 2.2.2. Sea $\{x^k\}_{k \in \mathbb{N}}$ una sucesión generada por el Algoritmo 2.2.1. Entonces para cada punto de acumulación x^* de $\{x^k\}_{k \in K}$ existe un índice $j_0 \in \{0, \dots, L_m - 1\}$ tal que

$$\langle J_c(x^*)^T c(x^*), E_{j_0}^T c(x^*) \rangle = 0. \quad (2.14)$$

Demostración: Sea x^* un punto de acumulación de $\{x^k\}_{k \in K}$. Entonces existe un conjunto infinito de índices $K_1 \subset K$ tal que

$$\lim_{k \in K_1} x^k = x^*.$$

Por (2.13) tenemos que

$$\lim_{k \in K_1} \alpha_k^2 h(x^k) = 0. \quad (2.15)$$

Consideramos las siguientes dos posibilidades:

1. La sucesión $\{\alpha_k\}_{k \in K_1}$ no tiende a cero.
2. La sucesión $\{\alpha_k\}_{k \in K_1}$ tiende a cero.

En el primer caso existe una sucesión infinita de índices $K_2 \subset K_1$ tal que $\alpha_k \geq c > 0$ para todo $k \in K_2$. Entonces por (2.15) obtenemos que

$$\lim_{k \in K_2} h(x^k) = 0.$$

Como h es continua esto implica que $h(x^*) = 0$. Por lo tanto $c(x^*) = 0$ y (2.14) se verifica para todo $j \in \{0, \dots, L_m - 1\}$.

Supongamos ahora que ocurre el caso 2. Una vez que la dirección d^k es calculada en el paso 2 del Algoritmo 2.1.1, se testea la desigualdad

$$h(x^k + \alpha_+ d^k) \leq \bar{h}_k + \eta_k - 2\gamma \alpha_+^2 \sigma_k^2 h(x^k). \quad (2.16)$$

Si esta desigualdad no se verifica se testea la desigualdad

$$h(x^k - \alpha_- d^k) \leq \bar{h}_k + \eta_k - 2\gamma \alpha_-^2 \sigma_k^2 h(x^k). \quad (2.17)$$

Los primeros valores de las longitudes de paso en (2.16)-(2.17) son $\alpha_+ = \alpha_- = 1$. Como $\lim_{k \in K_1} \alpha_k = 0$, existe $k_0 \in K_1$ tal que $\alpha_k < 1$ para todo $k \in K_1, k \geq k_0$. Por lo tanto para aquellas iteraciones k existen longitudes de paso α_k^+ y α_k^- que no verifican (2.16) y (2.17) respectivamente y además se tiene que

$$\lim_{k \in K_1} \alpha_k^+ = \lim_{k \in K_1} \alpha_k^- = 0.$$

Por lo tanto para todo $k \in K_1, k \geq k_0$ tenemos que

$$h(x^k + \alpha_k^+ d^k) > \bar{h}_k + \eta_k - 2\gamma(\alpha_k^+)^2 \sigma_k^2 h(x^k), \quad (2.18)$$

$$h(x^k - \alpha_k^- d^k) > \bar{h}_k + \eta_k - 2\gamma(\alpha_k^-)^2 \sigma_k^2 h(x^k). \quad (2.19)$$

Como $|\sigma_k| \in [\sigma_{min}, \sigma_{max}]$ obtenemos que (2.18) implica que

$$h(x^k + \alpha_k^+ d^k) > \bar{h}_k + \eta_k - \bar{\gamma}(\alpha_k^+)^2 h(x^k) \quad (2.20)$$

y (2.19) implica que

$$h(x^k - \alpha_k^- d^k) > \bar{h}_k + \eta_k - \bar{\gamma}(\alpha_k^-)^2 h(x^k) \quad (2.21)$$

donde $\bar{\gamma} = 2\gamma\sigma_{max}^2$.

La desigualdad (2.20) implica que

$$h(x^k + \alpha_k^+ d^k) > h(x^k) - \bar{\gamma}(\alpha_k^+)^2 h(x^k).$$

Por lo tanto

$$h(x^k + \alpha_k^+ d^k) - h(x^k) > -\bar{\gamma}(\alpha_k^+)^2 h(x^k).$$

Por la Proposición 2.1.6, $\{h(x^k)\}_{k \in \mathbb{N}}$ es una sucesión acotada superiormente por una constante $C > 0$. Así

$$h(x^k + \alpha_k^+ d^k) - h(x^k) \geq -\bar{\gamma}C(\alpha_k^+)^2 \quad (2.22)$$

lo que implica que

$$\|c(x^k + \alpha_k^+ d^k)\|^2 - \|c(x^k)\|^2 \geq -2\bar{\gamma}C(\alpha_k^+)^2.$$

Entonces

$$\frac{\|c(x^k + \alpha_k^+ d^k)\|^2 - \|c(x^k)\|^2}{\alpha_k^+} \geq -2\bar{\gamma}C\alpha_k^+.$$

Por el Teorema del Valor Medio existe $\xi_k^+ \in [0, 1]$ tal que

$$\langle g(x^k + \xi_k^+ \alpha_k^+ d^k), d^k \rangle \geq -2\bar{\gamma}C\alpha_k^+.$$

Como L_m es finito existe $j_0 \in \{0, \dots, L_m - 1\}$, $K_2 \subset K_1$ tal que para todo $k \in K_2$

$$d^k = -\sigma_k \begin{pmatrix} E_{j_0}^T & V_{j_0}^T \end{pmatrix} \begin{pmatrix} c(x^k) \\ 0 \end{pmatrix}.$$

Así tenemos que

$$-\sigma_k \left\langle g(x^k + \xi_k^+ \alpha_k^+ d^k), \begin{pmatrix} E_{j_0}^T & V_{j_0}^T \end{pmatrix} \begin{pmatrix} c(x^k) \\ 0 \end{pmatrix} \right\rangle \geq -2\bar{\gamma}C\alpha_k^+.$$

Si $\sigma_k > 0$ para un número infinito de índices $k \in K_2$ la última desigualdad implica que para $k \in K_2, k \geq k_0$

$$\left\langle g(x^k + \xi_k^+ \alpha_k^+ d^k), \begin{pmatrix} E_{j_0}^T & V_{j_0}^T \end{pmatrix} \begin{pmatrix} c(x^k) \\ 0 \end{pmatrix} \right\rangle \leq \frac{2\bar{\gamma}C\alpha_k^+}{\sigma_k} \leq \frac{2\bar{\gamma}C\alpha_k^+}{\sigma_{min}}. \quad (2.23)$$

Usando (2.21) y procediendo de la misma manera obtenemos que para $k \in K_2, k \geq k_0$,

$$\left\langle g(x^k - \xi_k^- \alpha_k^- d^k), \begin{pmatrix} E_{j_0}^T & V_{j_0}^T \end{pmatrix} \begin{pmatrix} c(x^k) \\ 0 \end{pmatrix} \right\rangle \geq -\frac{2\bar{\gamma}C\alpha_k^-}{\sigma_k} \geq -\frac{2\bar{\gamma}C\alpha_k^-}{\sigma_{min}}. \quad (2.24)$$

para algún $\xi_k^- \in [0, 1]$.

Como $\{\sigma_k\}$ y $\{h(x^k)\}$ son acotadas tenemos que $\{\|d^k\|\}$ es acotada.

Entonces usando que $\alpha_k^+ \rightarrow 0, \alpha_k^- \rightarrow 0$ y tomando límite en (2.23) y (2.24) obtenemos que

$$\left\langle g(x^*), \begin{pmatrix} E_{j_0}^T & V_{j_0}^T \end{pmatrix} \begin{pmatrix} c(x^*) \\ 0 \end{pmatrix} \right\rangle = \langle J_c(x^*)^T c(x^*), E_{j_0}^T c(x^*) \rangle = 0.$$

Si $\sigma_k < 0$ para un número infinito de índices, procediendo de una manera análoga obtenemos la misma ecuación. Por lo tanto en ambos casos obtenemos el resultado que queríamos demostrar. $\square$

Observación 2.2.3. Hemos probado que existe un índice $j_0 \in \{0, \dots, L_m - 1\}$ tal que el gradiente de $\|c(x)\|^2$ en x^* es ortogonal al residuo $E_{j_0}^T c(x^*)$. Notar que cuando $m = n$, el resultado del Teorema 2.2.2 coincide con el resultado de convergencia obtenido en [25]. Así podemos decir que el Algoritmo DF-SAUNE es una extensión de DF-SANE.

Corolario 2.2.4. Si x^* es un punto de acumulación de una sucesión $\{x^k\}_{k \in K}$ generada por el Algoritmo 2.2.1 y para todo $j \in \{0, \dots, L_m - 1\}$ se tiene que $\langle J_c(x^*) E_j^T v, v \rangle \neq 0$ para todo $v \in \mathbb{R}^n, v \neq 0$, entonces $c(x^*) = 0$.

2.3. Algoritmo cuasi-Newton sin derivadas (DF-QNB)

En esta sección presentamos un método cuasi-Newton basado en la fórmula Broyden para actualizar las matrices y que utiliza la búsqueda lineal no monótona sin derivadas definida en la Sección 2.1. También establecemos los resultados de convergencia global obtenidos para este método.

Definimos la dirección de búsqueda como una solución aproximada del sistema lineal $B_k d = -c(x^k)$ y usaremos la fórmula de actualización de Broyden de rango uno:

$$B_{k+1} = B_k + \frac{(y^k - B_k s^k)(s^k)^T}{(s^k)^T s^k} \quad (2.25)$$

donde $s^k = x^{k+1} - x^k$ y $y^k = c(x^{k+1}) - c(x^k)$.

Como mencionamos previamente, el propósito principal es resolver el sistema lineal de manera precisa cuando sea posible y de forma aproximada en caso contrario como fue considerado en [32].

Algoritmo 2.3.1. *Algoritmo DF-QNB*

Dados x^0 , $B_0 \in \mathbb{R}^{m \times n}$, $0 < \gamma < 1$, $0 \leq \theta_0 < 1$, $\Delta > 0$,

$0 < \tau_{min} < \tau_{max} < 1$, $0 \leq \epsilon < 1$.

Considerar $k \leftarrow 0$.

Paso 1 Si $\|c(x^k)\| \leq \epsilon \max\{1, \|c(x^0)\|\}$ detener el proceso.

Paso 2 Cálculo de la dirección d^k

Paso 2.1 Encontrar d tal que

$$B_k d + c(x^k) = 0 \quad y \quad \|d\| \leq \Delta. \quad (2.26)$$

Si tal dirección d es encontrada definir $d^k = d$, $\theta_{k+1} = \theta_k$ e ir al paso 3.

Paso 2.2 Encontrar una solución d de $\min_{d \in \mathbb{R}^n} \|B_k d + c(x^k)\|$.

Si d satisface

$$\|B_k d + c(x^k)\| \leq \theta_k \|c(x^k)\| \quad y \quad \|d\| \leq \Delta \quad (2.27)$$

definir $d^k = d$, $\theta_{k+1} = \theta_k$ e ir al paso 3.

Si no considerar $d^k = 0$, $x^{k+1} = x^k$, $\theta_{k+1} = \frac{\theta_k + 1}{2}$ e ir al paso 5.

Paso 3 Encontrar α_k usando el Algoritmo 2.1.1 y definir $x^{k+1} = x^k + \alpha_k d^k$.

Paso 4 Actualizar B_{k+1} usando (2.25).

Paso 5 Actualizar $k \leftarrow k + 1$ e ir al paso 1.

En [32], el subproblema en el paso 2 fue definido usando dos matrices diferentes.

Observación 2.3.2. Cuando $m = n$ el Algoritmo 2.3.1 es similar al algoritmo descrito en [29], en el sentido que ambos usan la fórmula de actualización de Broyden. En [29] los autores resuelven el sistema lineal (2.2) usando la actualización (2.3). Sabemos que la resolución de sistemas lineales de gran escala puede ser muy costosa aún cuando estos tengan solución. Teniendo esto en cuenta es que tratamos de encontrar, en estos casos, una solución aproximada en el sentido de (2.27).

Los siguientes teoremas establecen las hipótesis necesarias para obtener los resultados de convergencia. En el primero de ellos el resultado de convergencia se obtiene usando que $\{\|c(x^k)\|\}$ es acotada (Proposición 2.1.6) y una condición más débil que la condición de Dennis-Moré:

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - J_c(x^k))d^k\|}{\|d^k\|} = 0. \quad (2.28)$$

Teorema 2.3.3. Suponiendo que el Algoritmo 2.3.1 genera una sucesión infinita $\{x^k\}_{k \in \mathbb{N}}$, si

$$\lim_{k \rightarrow \infty} \langle (B_k - J_c(x^k))d^k, c(x^k) \rangle = 0 \quad (2.29)$$

y existe $k_0 \in \mathbb{N}$ tal que para todo $k \geq k_0$, $\theta_k = \bar{\theta} < 1$, entonces cada punto de acumulación de $\{x^k\}_{k \in K}$ es una solución de (2.1), donde K es el conjunto dado en (2.8).

Demostración: Sea x^* un punto de acumulación de $\{x^k\}_{k \in K}$, entonces existe $K_1 \subset K$ tal que $\lim_{k \in K_1} x^k = x^*$.

Sabemos que $\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0$, entonces $\lim_{k \in K_1} \alpha_k^2 \|d^k\|^2 = 0$.

Consideramos dos casos: a) $\lim_{k \in K_1} \alpha_k \neq 0$ y b) $\lim_{k \in K_1} \alpha_k = 0$.

Supongamos que estamos en el caso a. Entonces tenemos que

$$\lim_{k \in K_1} \|d^k\| = 0. \quad (2.30)$$

- Primero asumamos que en el proceso del Algoritmo 2.3.1 en un número finito de iteraciones la dirección fue calculada resolviendo el sistema lineal $B_k d + c(x^k) = 0$.

Entonces existe $k_1 \in K_1$ tal que para todo $k \geq k_1$, $k \in K_1$, d^k verifica la fórmula $\|B_k d + c(x^k)\| \leq \theta_k \|c(x^k)\|$. Así para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ obtenemos que

$$\begin{aligned} \|B_k d^k + c(x^k)\|^2 &= \langle B_k d^k + c(x^k), B_k d^k + c(x^k) \rangle \\ &= \langle B_k d^k, B_k d^k \rangle + 2\langle B_k d^k, c(x^k) \rangle + \langle c(x^k), c(x^k) \rangle \\ &= \|B_k d^k\|^2 + 2\langle B_k d^k, c(x^k) \rangle + \|c(x^k)\|^2 \leq \bar{\theta}^2 \|c(x^k)\|^2. \end{aligned}$$

Por lo tanto, como $\bar{\theta} < 1$, para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ se tiene que

$$\|B_k d^k\|^2 + 2\langle B_k d^k, c(x^k) \rangle \leq (\bar{\theta}^2 - 1)\|c(x^k)\|^2 < 0.$$

Entonces, como $\|B_k d^k\|^2 \geq 0$, para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ obtenemos que

$$\langle B_k d^k, c(x^k) \rangle \leq \frac{\bar{\theta}^2 - 1}{2}\|c(x^k)\|^2.$$

Esto implica que

$$\langle (B_k - J_c(x^k))d^k, c(x^k) \rangle + \langle J_c(x^k)d^k, c(x^k) \rangle \leq \frac{\bar{\theta}^2 - 1}{2}\|c(x^k)\|^2.$$

Tomando límite para $k \in K_1$, $k \geq \max\{k_0, k_1\}$ en la última expresión, usando la continuidad de c y J_c y por (2.29)-(2.30) tenemos que

$$0 \leq \frac{\bar{\theta}^2 - 1}{2}\|c(x^*)\|^2.$$

Como $\bar{\theta} < 1$ obtenemos que $\|c(x^*)\| = 0$ como queríamos probar.

- Asumamos ahora que en infinitas iteraciones la dirección fue obtenida resolviendo el sistema lineal $B_k d + c(x^k) = 0$.

Entonces existe $K_2 \subset K_1$ tal que para todo $k \in K_2$ se verifica $B_k d^k + c(x^k) = 0$.

Así para todo $k \in K_2$,

$$\langle B_k d^k + c(x^k), c(x^k) \rangle = 0.$$

Por lo tanto, para todo $k \in K_2$ se tiene que

$$\langle B_k d^k + c(x^k) \rangle + \|c(x^k)\|^2 = 0,$$

lo que implica que

$$\langle (B_k - J_c(x^k))d^k, c(x^k) \rangle + \langle J_c(x^k)d^k, c(x^k) \rangle = -\|c(x^k)\|^2.$$

Tomando límite para $k \in K_2$, $k \geq k_0$ en la última expresión, usando la continuidad de c y J_c y por (2.29)-(2.30) obtenemos que $\|c(x^*)\| = 0$ como queríamos probar.

Supongamos ahora que estamos bajo la condición del caso b . Como la sucesión $\{d^k\}_{k \in \mathbb{N}}$ es acotada existen $K_2 \subset K_1$ y $\bar{d} \in \mathbb{R}^n$ tales que $\lim_{k \in K_2} d^k = \bar{d}$.

En la búsqueda lineal, para elegir la longitud del paso α_k el algoritmo testea las siguientes desigualdades

$$h(x^k + \alpha_+ d^k) \leq \bar{h}_k + \eta_k - \gamma \alpha_+^2 \|d^k\|^2 \quad (2.31)$$

$$h(x^k - \alpha_- d^k) \leq \bar{h}_k + \eta_k - \gamma \alpha_-^2 \|d^k\|^2. \quad (2.32)$$

Los valores iniciales de α_+ and α_- son 1. Como $\lim_{k \in K_2} \alpha_k = 0$ existe $\bar{k} \in K_2$ tal que $\alpha_k < 1$ para todo $k \in K_2, k \geq \bar{k}$. Así para aquellas iteraciones k existen longitudes de paso α_k^+ y α_k^- que no verifican (2.31) y (2.32) respectivamente y $\lim_{k \in K_2} \alpha_k^+ = \lim_{k \in K_2} \alpha_k^- = 0$. Por lo tanto tenemos que

$$h(x^k + \alpha_k^+ d^k) > \bar{h}_k + \eta_k - \gamma (\alpha_k^+)^2 \|d^k\|^2 \quad (2.33)$$

$$h(x^k - \alpha_k^- d^k) > \bar{h}_k + \eta_k - \gamma (\alpha_k^-)^2 \|d^k\|^2. \quad (2.34)$$

La desigualdad (2.33) implica que

$$h(x^k + \alpha_k^+ d^k) > h(x^k) - \gamma (\alpha_k^+)^2 \|d^k\|^2.$$

Como $\|d^k\| \leq \Delta$ obtenemos que

$$\frac{h(x^k + \alpha_k^+ d^k) - h(x^k)}{\alpha_k^+} > -\gamma \alpha_k^+ \Delta^2.$$

Por el Teorema del Valor Medio existe $\xi_k^+ \in [0, 1]$ tal que

$$\langle g(x^k + \xi_k^+ \alpha_k^+ d^k), d^k \rangle > -\gamma \alpha_k^+ \Delta^2. \quad (2.35)$$

La desigualdad (2.34) implica que

$$h(x^k - \alpha_k^- d^k) > h(x^k) - \gamma (\alpha_k^-)^2 \|d^k\|^2.$$

Como $\|d^k\| \leq \Delta$ tenemos que

$$\frac{h(x^k - \alpha_k^- d^k) - h(x^k)}{\alpha_k^-} > -\gamma \alpha_k^- \Delta^2.$$

Por el Teorema del Valor Medio existe $\xi_k^- \in [0, 1]$ tal que

$$\langle g(x^k - \xi_k^- \alpha_k^- d^k), d^k \rangle < \gamma \alpha_k^- \Delta^2. \quad (2.36)$$

Tomando límite en (2.35) y (2.36) cuando $k \rightarrow \infty, k \in K_2$ obtenemos que $\langle g(x^*), \bar{d} \rangle = \langle J_c(x^*)^T c(x^*), \bar{d} \rangle = 0$.

Así

$$\langle J_c(x^*) \bar{d}, c(x^*) \rangle = 0. \quad (2.37)$$

Ahora, como hicimos antes, consideramos dos casos:

- (i) en un número finito de iteraciones la dirección es obtenida como solución del sistema lineal $B_k d + c(x^k) = 0$; y
- (ii) en infinitas iteraciones la dirección es obtenida como solución del sistema lineal $B_k d + c(x^k) = 0$.

Procediendo de manera análoga a la realizada cuando $\lim_{k \in K_1} \alpha_k \neq 0$ y usando (2.29) y (2.37) obtenemos que

$$\|c(x^*)\| = 0$$

como queríamos probar. □

La condición de Dennis-Moré (2.28) es conocida como una condición necesaria y suficiente para obtener convergencia q -superlineal de los métodos cuasi-Newton clásicos [13].

Si no podemos garantizar que $\theta_k = \bar{\theta} < 1$ para $k \geq k_0$ entonces podemos demostrar el siguiente resultado, que es similar al obtenido en [32].

Teorema 2.3.4. *Suponiendo que en el Algoritmo 2.3.1 el parámetro θ_k es aumentado un número infinito de veces, si se define*

$$K_* = \{k \in \mathbb{N} \mid \theta_{k+1} > \theta_k\},$$

y se supone que

$$\lim_{k \in K_*} \|B_k - J_c(x^k)\| = 0, \tag{2.38}$$

entonces cada punto de acumulación x^ de la sucesión $\{x^k\}_{k \in K_*}$ es una solución del problema (2.1) o es un minimizador global de $\|c(x^*) + J_c(x^*)(x - x^*)\|$.*

Demostración: Si $c(x^*) = 0$ entonces x^* es solución del problema (2.1). Asumamos entonces que $\|c(x^*)\| > 0$. Supongamos que x^* no es un minimizador global de $\|c(x^*) + J_c(x^*)(x - x^*)\|$. Por lo tanto existe d tal que $\|d\| \leq \frac{\Delta}{2}$ y

$$\|c(x^*) + J_c(x^*)d\| < \|c(x^*)\|,$$

entonces tenemos que

$$\frac{\|c(x^*) + J_c(x^*)d\|}{\|c(x^*)\|} \leq r < 1.$$

Por (2.38) y la continuidad de c y J_c obtenemos que

$$\frac{\|c(x^k) + B_k(x^* - x^k + d)\|}{\|c(x^k)\|} \leq \frac{r+1}{2} \quad (2.39)$$

para todo k suficientemente grande, $k \in K_*$. Pero como $\|d\| \leq \frac{\Delta}{2}$ y $\lim_{k \in K_*} x^k = x^*$ tenemos que para $k \in K_*$ suficientemente grande, $\|x^* - x^k + d\| \leq \Delta$. Por lo tanto (2.39) contradice el hecho que $\theta_k \rightarrow 1$ y una dirección que verifica (2.27) puede ser obtenida. Por lo tanto x^* es un minimizador global de $\|c(x^*) + J_c(x^*)(x - x^*)\|$. $\square$

Un punto x^* que es minimizador global de la función $\|c(x^*) + J_c(x^*)(x - x^*)\|$ puede ser visto como una solución del problema de cuadrados mínimos

$$\min_{x \in \mathbb{R}^n} \|A(x - x^*) - b\| \quad (2.40)$$

donde $A = J_c(x^*)$ y $b = -c(x^*)$. La función lineal es el modelo afín de la función c alrededor de x^* . En general no podemos esperar encontrar x^* tal que $c(x^*) = 0$ ya que el problema podría no tener solución. Tampoco podemos esperar encontrar x^* tal que $c(x^*) + J_c(x^*)(x - x^*) = 0$ ya que estamos trabajando con un sistema indeterminado de ecuaciones lineales y $J_c(x^*)$ podría no tener rango completo. Por estas causas creemos que es razonable encontrar minimizadores globales de (2.40) cuando el problema no tiene solución.

2.3.1. Análisis del caso $M = 1$

En esta sección analizamos el caso en el cual tomamos $M = 1$ en la búsqueda lineal no monótona sin derivadas usada en el Algoritmo DF-QNB. Por la presencia de η_k la búsqueda sigue siendo no monótona pero impone un comportamiento casi monótono de la función de mérito cuando estamos cerca de una solución. Así es apropiado considerar $M = 1$ cuando estamos trabajando con un método cuasi-Newton. $M = 1$, como fue señalado en [25], puede resultar inconveniente para el método del residuo espectral ya que se comporta altamente no monótono aún cerca de una solución aislada.

Demostramos a continuación que en el caso en $M = 1$ nuestro algoritmo verifica la hipótesis (2.29).

Usando la Proposición 2.1.8 se puede observar que

$$\sum_{k=1}^{\infty} \|s^k\|^2 = \sum_{k=1}^{\infty} \alpha_k^2 \|d^k\|^2 < \infty. \quad (2.41)$$

Luego definimos la matriz

$$A_{k+1} = \int_0^1 J_c(x^k + ts^k) dt.$$

Por lo tanto $A_{k+1}s^k = y^k$ y

$$B_{k+1} = B_k + \frac{(A_{k+1} - B_k)s^k(s^k)^T}{\|s^k\|^2}.$$

Finalmente podemos usar el resultado del Lema 2.6 de [29], que enunciamos a continuación por completitud.

Lema 2.3.5. [Lema 2.6, [29]] Suponiendo que el conjunto $\Gamma_0 = \{x \in \mathbb{R}^n : h(x) \leq h(x^0) + \eta\}$ es acotado y que $J_c(x)$ es Lipschitz continua en Γ_0 , si (2.41) se verifica entonces

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=0}^{k-1} \rho_i^2 = 0$$

donde

$$\rho_k = \frac{\|(A_{k+1} - B_k)d^k\|}{\|d^k\|}.$$

En particular existe una subsucesión de $\{\rho_k\}$ que tiende a cero.

Así podemos demostrar el siguiente resultado de convergencia.

Teorema 2.3.6. Asumiendo que el Algoritmo 2.3.1 genera una sucesión $\{x^k\}_{k \in \mathbb{N}}$, considerando $M = 1$ en la búsqueda lineal, y asumiendo que se verifican las hipótesis del Lema 2.3.5, si existe k_0 tal que $\theta_k = \bar{\theta} < 1$ para todo $k \geq k_0$ entonces existe un punto de acumulación x^* de $\{x^k\}_{k \in \mathbb{N}}$ que es solución de (2.1).

Demostración: Por la Proposición 2.1.8 tenemos que (2.41) se verifica. Por lo tanto por el Lema 2.3.5 existe un subconjunto $\tilde{K} \subset \mathbb{N}$ tal que

$$\lim_{k \in \tilde{K}} \frac{\|(A_{k+1} - B_k)d^k\|}{\|d^k\|} = 0. \quad (2.42)$$

Sea x^* un punto de acumulación de la sucesión $\{x^k\}_{k \in \tilde{K}}$. Notar que

$$\begin{aligned} |\langle (B_k - J_c(x^k))d^k, c(x^k) \rangle| &\leq \|(B_k - J_c(x^k))d^k\| \|c(x^k)\| \\ &\leq (\|(A_{k+1} - B_k)d^k\| + \|(J_c(x^k) - A_{k+1})d^k\|) \|c(x^k)\|. \end{aligned}$$

Por la Proposición 2.1.8 tenemos que $\{\|c(x^k)\|\}$ es acotada. Así tomando límite cuando $k \in \tilde{K}, k \rightarrow \infty$ y usando que $\{d^k\}$ es acotada y (2.42), obtenemos que $\|(A_{k+1} - B_k)d^k\| \rightarrow 0$. Además por definición de A_{k+1} tenemos que $\|(J_c(x^k) - A_{k+1})d^k\| \rightarrow 0$. Por lo tanto hemos probado que (2.29) se verifica para $k \in \tilde{K}$ y el final de la demostración se sigue del Teorema 2.3.3. $\square$

Observar que esta búsqueda lineal particular mejora los resultados de convergencia obtenidos en el Teorema 2.3.3.

2.4. Experimentación numérica

En esta sección presentamos algunos resultados computacionales obtenidos con implementaciones en Fortran 77 de los métodos DF-SAUNE y DF-QNB. Todas las experimentaciones fueron corridas en una computadora personal con INTEL(R) Core (TM) 2 Duo CPU E8400 de 3.00 GHz y 3.23 GB de RAM.

Como es usual en artículos de optimización sin derivadas estamos interesados en el número de evaluaciones funcionales para ambos algoritmos. También reportamos para problemas de mediano tamaño el tiempo de CPU obtenido para ambos códigos e incluimos una comparación con el Algoritmo NEWUOA desarrollado por M.J.D. Powell para problemas de optimización sin restricciones [42]. En estos experimentos, NEWUOA resuelve el siguiente problema de cuadrados mínimos: $\min \|c(x)\|^2$.

2.4.1. Problemas test

Los problemas usados para estos experimentos numéricos, de la forma $c(x) = 0, c: \mathbb{R}^n \rightarrow \mathbb{R}^m, m \leq n$, son un conjunto de problemas definidos como conjuntos factibles de problemas de programación no lineal del libro de Hock y Schittkowski [22], donde el número de variables se encuentra entre 2 y 10, y el número de restricciones varía entre 1 y 4. También, en lo referente a problemas de dimensión grande, testamos algunos problemas descritos en [12]. Algunos de los problemas analizados aquí han sido previamente examinados en [18] para resolver sistemas de ecuaciones no lineales indeterminados con restricciones de cotas en las variables pero usando derivadas.

En la Tabla 2.1 mostramos los datos de los problemas. En la columna 1 mostramos el número del problema, en la columna 2 de donde fueron extraídos los problemas y en las últimas columnas mostramos el número de variables (n) y el número de ecuaciones (m) de cada problema.

Los puntos iniciales son los mismos que se usaron en las citas correspondientes.

Problema	Fuente	n	m
1	Problema 6 de [22]	2	1
2	Problema 7 de [22]	2	1
3	Problema 8 de [22]	2	2
4	Problema 26 de [22]	3	1
5	Problema 27 de [22]	3	1
6	Problema 39 de [22]	4	2
7	Problema 40 de [22]	4	3
8	Problema 42 de [22]	4	2
9	Problema 46 de [22]	5	2
10	Problema 47 de [22]	5	3
11	Problema 48 de [22]	5	2
12	Problema 53 de [22]	5	3
13	Problema 56 de [22]	7	4
14	Problema 61 de [22]	3	2
15	Problema 63 de [22]	3	2
16	Problema 77 de [22]	5	2
17	Problema 78 de [22]	5	3
18	Problema 79 de [22]	5	3
19	Problema 81 de [22]	5	3
20	Problema 111 de [22]	10	3
21	Problema 2(a) de [12]	300	150
22	Problema 2(b) de [12]	300	150
23	Problema 4(a) de [12]	300	150
24	Problema 4(b) de [12]	300	150
25	Problema ARWHEAD de [42]	150	149
26	Problema ARWHEAD de [42]	300	299

Tabla 2.1: Datos de los problemas

Observación 2.4.1. *El caso (a) en los problemas 2 y 4 de [12] corresponde a usar el punto inicial $x^0(1 : n) = 2$. El caso (b) en los problemas 2 y 4 de [12] corresponde a usar el punto inicial $x^0(1 : n) = 150$.*

2.4.2. Implementación

En la implementación de los Algoritmos DF-SAUNE y DF-QNB se usaron los siguientes parámetros:

1. Los parámetros del Algoritmo 2.1.1 fueron:

$$M = 2, \tau_{min} = 0.1, \tau_{max} = 0.5, \gamma = 10^{-4}, \eta_0 = 1,$$

$$\bullet \forall k \in \mathbb{N}, k \geq 1 \quad \eta_k = \frac{\|c(x^0)\|}{2^k} \quad \text{para problemas de pequeño tamaño.}$$

- $\forall k \in \mathbb{N}, k \geq 1, \quad \eta_k = \frac{\|c(x^0)\|}{(1+k)^2}$ para problemas de mediano tamaño.

2. Los parámetros utilizados en el Algoritmo DF-SAUNE fueron:

$$\epsilon = 10^{-6}, \sigma_0 = 1, \sigma_{min} = 10^{-10}, \sigma_{max} = 10^{10}.$$

3. Los parámetros utilizados en el Algoritmo DF-QNB fueron:

$$\epsilon = 10^{-6}, \theta_0 = 0.95, \Delta = 10^{12}.$$

4. En DF-QNB, la primera matriz B_0 fue calculada por diferencias finitas como una aproximación de la matriz Jacobiana en x^0 .

5. En DF-QNB, hemos usado el Algoritmo ACCIM, descrito en [50], para encontrar una solución del sistema lineal $B_k d = -c(x^k)$. Para resolver el problema de cuadrados mínimos $\min_{\|d\| \leq \Delta} \|B_k d + c(x^k)\|^2$ hemos usado el Algoritmo BVLS descrito en [51].

6. El criterio de finalización para ambos algoritmos fue:

$$\|c(x^k)\| \leq 10^{-6} \max\{1, \|c(x^0)\|\}.$$

7. El número máximo permitido de evaluaciones funcionales fue:

- $MAX_{FE} = 5000$, para problemas de pequeño tamaño,
- $MAX_{FE} = 15000$, cuando $n = 150$,
- $MAX_{FE} = 30000$, cuando $n = 300$.

En el método DF-QNB hemos incluido las evaluaciones requeridas para calcular la matriz inicial B_0 .

La implementación de NEWUOA es la versión original de M.J.D. Powell [42] con sus respectivos criterios de finalización, esto es, el algoritmo se detiene cuando el radio de la región de confianza es menor a la tolerancia $\rho_{end} = 10^{-6}$. Como mencionamos anteriormente, NEWUOA resuelve el problema de cuadrados mínimos $\min \|c(x)\|^2$.

2.4.3. Resultados numéricos

En la Tabla 2.2 mostramos los resultados obtenidos por DF-SAUNE (DF-S) y DF-QNB (DF-B) teniendo en cuenta el valor final $\|c(x^{end})\|$ y el número de evaluaciones funcionales. Los resultados corresponden al criterio de finalización correspondiente o a condiciones internas que no nos permiten mejorar los resultados.

La Tabla 2.2 muestra también el número de problema (columna 1), el número de iteraciones (Iter, columna 2), el número de evaluaciones funcionales (c -eval, columna 3), y el valor funcional final obtenido para ambos códigos ($\|c(x^{end})\|$, columna 4).

Estos resultados ilustran la eficiencia de DF-QNB, aunque DF-SAUNE también ha tenido un comportamiento satisfactorio. Podemos observar que en 15 de los 20 problemas test DF-QNB ha usado menos evaluaciones funcionales que DF-SAUNE. Es también importante mencionar que cuando DF-QNB requiere más evaluaciones funcionales que DF-SAUNE, tal diferencia es significativa.

Problema	Iter		c -eval		$\ c(x^{end})\ $	
	DF-S	DF-B	DF-S	DF-B	DF-S	DF-B
1	83	4	85	7	2.555333D-08	1.460382D-08
2	44	62	70	844	5.616128D-08	8.074138D-06
3	52	10	54	19	1.852329D-07	1.029118D-06
4	49	61	85	929	1.487057D-07	1.188074D-07
5	1	251	2	4815	0.0D+00	1.341756D-06
6	57	30	105	76	5.707520D-08	3.311382D-08
7	122	5	327	10	8.140108D-09	7.144642D-09
8	56	15	137	20	9.481413D-09	8.774760D-07
9	143	19	448	25	9.368079D-08	8.287063D-06
10	80	8	133	14	1.483448D-08	6.553991D-07
11	1	1	4	7	0.0D+00	5.264796D-06
12	1	1	3	7	0.0D+00	1.026234D-09
13	92	5	163	13	5.098483D-08	3.366883D-07
14	101	10	206	22	3.382254D-08	1.863196D-09
15	67	7	134	16	7.382254D-08	2.947364D-07
16	54	10	125	16	3.911728D-07	2.765711D-07
17	318	6	1176	12	2.243582D-06	1.839195D-07
18	68	7	169	13	6.362777D-08	1.753428D-06
19	696	8	5000	14	1.137512D-01	1.641720D-08
20	132	12	461	29	1.200486D-08	4.219128D-09

Tabla 2.2: Problemas de pequeño tamaño

En el problema 5 hemos observado que, en varias iteraciones, la norma de la dirección d^k , solución del sistema lineal $B_k d + c(x^k) = 0$ es mayor que Δ y, en esas iteraciones, DF-QNB debió resolver el subproblema (2.27). Por esta razón el decrecimiento de la función de mérito $\|c(x)\|$ es muy lento y el algoritmo requiere muchas evaluaciones funcionales.

En el problema 19, DF-SAUNE detiene el proceso sin obtener un decrecimiento satisfactorio en el residuo al alcanzar el número máximo permitido de evaluaciones funcionales. Creemos que este desempeño poco satisfactorio de DF-SAUNE está relacionado con la estrategia usada para definir las matrices E_j cuando $\frac{n}{m} \notin \mathbb{N}$. Pensamos también que es interesante como trabajo futuro estudiar mejores estrategias para completar la última matriz E_j en estos casos específicos.

En la Tabla 2.3 mostramos el número de iteraciones, el número de evaluaciones funcionales y el tiempo de CPU en segundos requerido por los Algoritmos DF-SAUNE, DF-QNB

y NEWUOA para los problemas de mediano tamaño 21, 22, 23, 24, 25 y 26.

En general los resultados numéricos muestran que los valores residuales finales son similares para todos los métodos.

Observamos que DF-SAUNE realiza una mayor cantidad de evaluaciones funcionales en los últimos dos problemas. Debe remarcarse que el tiempo de CPU fue siempre menor en DF-SAUNE que en DF-QNB. La razón de esto es que el Algoritmo DF-QNB tiene que resolver un sistema lineal de ecuaciones o un problema de cuadrados mínimos en cada iteración.

Finalmente corrimos el código NEWUOA para tener una medida comparativa del número de evaluaciones funcionales. Aunque NEWUOA fue diseñado para resolver problemas de optimización sin restricciones podemos concluir que nuestros métodos se comportan satisfactoriamente.

Problema	Método	Iter	c -eval	$\ c(x^{end})\ $	CPU
21	DF-SAUNE	105	123	1.096064D-06	0.01
	DF-QNB	2	303	4.544647D-11	0.05
	NEWUOA	3768	5765	4.681655D-10	245.12
22	DF-SAUNE	440	566	1.656464D-08	0.04
	DF-QNB	1	302	2.002649D-05	0.05
	NEWUOA	4010	6043	4.284213D-10	270.10
23	DF-SAUNE	147	185	5.481636D-06	0.01
	DF-QNB	128	649	4.283559D-06	2.43
	NEWUOA	15472	30000	8.203095D-07	1286.79
24	DF-SAUNE	94	122	9.933035D-07	0.01
	DF-QNB	394	731	7.489404D-05	9.67
	NEWUOA	18874	29205	1.334399D-07	1248.00
25	DF-SAUNE	3476	9848	2.696620D-07	0.34
	DF-QNB	13	164	3.628420D-05	4.23
	NEWUOA	7533	15000	1.092628D-09	124.15
26	DF-SAUNE	3001	8664	1.393098D-06	0.33
	DF-QNB	13	314	5.139963D-05	52.20
	NEWUOA	15072	30000	1.453638D-08	1237.54

Tabla 2.3: Problemas de mediano tamaño

Capítulo 3

Sistemas indeterminados de ecuaciones no lineales con restricciones de cotas en las variables

El conjunto factible de muchos problemas de programación no lineal es usualmente representado por un conjunto de ecuaciones no lineales, $c(x) = 0$ donde $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Eventualmente aparece también la necesidad de encontrar un punto factible sujeto a que se cumplan restricciones de cotas en las variables. Por este motivo en este capítulo consideraremos el problema de determinar una solución $x^* \in \mathbb{R}^n$ que verifica

$$\begin{aligned} c(x) &= 0 \\ x &\in \Omega \end{aligned} \tag{3.1}$$

donde $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$, $c = (c_1, \dots, c_m)$ y cada c_i , para $i = 1, \dots, m$, es una función continuamente diferenciable. Estamos interesados en sistemas para los cuales la matriz Jacobiana de c , denotada por $J_c(x)$, no está disponible o su cálculo requiere demasiado costo. Problemas con estas características provienen por ejemplo de la Ingeniería Química, Física o Economía.

El objetivo principal de este capítulo es extender el método de cuasi-Newton sin derivadas definido en el capítulo anterior para encontrar soluciones del sistema pero dentro de la caja Ω . En tal método, la dirección de búsqueda es calculada como una solución aproximada de un sistema lineal utilizando la fórmula adaptada de Broyden de rango 1 para las matrices. Está bien establecido en la literatura que, para el caso cuadrado, esta es la aproximación secante más usada del Jacobiano y funciona bien localmente [13]. La búsqueda lineal sin derivadas utilizada es una generalización de la definida en el Algoritmo DF-SANE para resolver sistemas no lineales cuadrados sin derivadas, ver [25].

La dirección de búsqueda utilizada en el método que presentamos en este capítulo será cal-

culada de manera semejante a la presentada en el capítulo anterior con la salvaguarda de considerar direcciones que no nos permitan salir de la caja Ω . Considerando la función de mérito usual $h : \mathbb{R}^n \rightarrow \mathbb{R}$, del residuo $c(x)$ en x , $h(x) = \frac{1}{2}\|c(x)\|^2$, la condición de suficiente decrecimiento será:

$$h(x^k + \alpha_k d^k) \leq \max_{0 \leq j \leq M-1} h(x^{k-j}) + \eta_k - \gamma \alpha_k^2 \|d^k\|^2 \quad (3.2)$$

donde M es un entero no negativo, $0 < \gamma < 1$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$. Como hemos mencionado anteriormente la condición anterior no utiliza derivadas y genera un proceso de descenso no monótono.

3.1. Algoritmo general y búsqueda lineal no monótona sin derivadas

En esta sección presentamos el Algoritmo general para resolver el problema (3.1) y la búsqueda lineal no monótona sin derivadas que utilizaremos en este esquema. Como mencionamos antes, la estrategia está basada en la búsqueda lineal sin derivadas propuesta en [25]. Dados el iterado actual x^k y una dirección de búsqueda d^k , el algoritmo busca una longitud de paso α_k tal que

$$h(x^k + \alpha_k d^k) \leq \max_{0 \leq j \leq M-1} h(x^{k-j}) + \eta_k - \gamma \alpha_k^2 \|d^k\|^2, \quad (3.3)$$

donde M es un entero no negativo, $0 < \gamma < 1$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$, $\eta_k > 0$.

Presentamos a continuación la implementación del proceso.

Algoritmo 3.1.1. Búsqueda lineal no monótona

Dados $d^k \in \mathbb{R}^n$, $0 < \tau_{min} < \tau_{max} < 1$, $0 < \gamma < 1$, $M \in \mathbb{N}$, $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\eta_k > 0$ y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$

Paso 1: Calcular $\bar{h}_k = \max\{h(x^k), \dots, h(x^{max\{0, k-M+1\}})\}$

$\alpha = 1$

Paso 2:

Si $h(x^k + \alpha d^k) \leq \bar{h}_k + \eta_k - \gamma \alpha^2 \|d^k\|^2$,

definir $\alpha_k = \alpha$, $x^{k+1} = x^k + \alpha_k d^k$

si no

elegir $\alpha_{new} \in [\tau_{min}\alpha, \tau_{max}\alpha]$,

$\alpha = \alpha_{new}$ e ir al paso 2.

La principal diferencia entre el algoritmo previo y el Algoritmo 2.1.1 del capítulo anterior es que en este nuevo procedimiento buscamos un punto en la dirección factible d^k evitando la búsqueda en la dirección opuesta $-d^k$.

Proposición 3.1.2. *El Algoritmo 3.1.1 está bien definido.*

Demostración: Ver demostración de la Proposición 2.1.2.

El nuevo Algoritmo general para resolver el problema (3.1) sigue el siguiente procedimiento.

Algoritmo 3.1.3. *Algoritmo general*

*Dados $x^0 \in \Omega$, $M \in \mathbb{N}$, $0 < \tau_{min} < \tau_{max} < 1$, $0 \leq \epsilon < 1$, $0 < \gamma < 1$, $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\eta_k > 0$
y $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$, $\Delta > 0$.*

Considerar $k \leftarrow 0$.

Paso 1: Si $\|c(x^k)\| \leq \epsilon$, detener el proceso.

Paso 2: Calcular una dirección de búsqueda d^k tal que $x^k + d^k \in \Omega$ y $\|d^k\| \leq \Delta$.

Paso 3: Encontrar α_k usando el Algoritmo 3.1.1 y definir $x^{k+1} = x^k + \alpha_k d^k$.

Actualizar $k \leftarrow k + 1$ e ir al paso 1.

Observar que como $x^k + d^k \in \Omega$ y la longitud de paso α_k obtenida por el Algoritmo 3.1.1 es menor o igual a 1, entonces $x^{k+1} = x^k + \alpha_k d^k \in \Omega$. Considerando el procedimiento anterior es posible obtener resultados que serán usados para demostrar los teoremas de convergencia. Las demostraciones de las proposiciones que enunciamos a continuación son análogas a las demostraciones de las Proposiciones 2.1.4, 2.1.5, 2.1.6 y 2.1.8 que desarrollamos en el capítulo anterior.

Proposición 3.1.4. *Se se considera $U_k = \max\{h(x^{(k-1)M+1}), \dots, h(x^{kM})\}$ para todo $k = 1, 2, \dots$ y se define $\nu(k) \in \{(k-1)M+1, \dots, kM\}$ el índice para el cual $h(x^{\nu(k)}) = U_k$, entonces para todo $k = 1, 2, \dots$*

$$h(x^{\nu(k+1)}) \leq h(x^{\nu(k)}) + \eta$$

donde $\eta = \sum_{i=0}^{\infty} \eta_i$.

Proposición 3.1.5.

$$\lim_{k \rightarrow \infty} \alpha_{\nu(k)-1}^2 \|d^{\nu(k)-1}\|^2 = 0.$$

Proposición 3.1.6. *La sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo general está contenida en*

$$\Gamma = \{x \in \Omega : h(x) \leq h(x^{\nu(1)}) + \eta\}.$$

Proposición 3.1.7. *Si se considera $M = 1$ en el Algoritmo 3.1.1 entonces*

- *la sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo general está contenida en $\Gamma_0 = \{x \in \Omega : h(x) \leq h(x^0) + \eta\}$,*
- *la serie $\sum_{k=1}^{\infty} \alpha_k^2 \|d^k\|^2$ es convergente.*

Los resultados obtenidos hasta aquí dependen fuertemente en la técnica de búsqueda lineal sin tener en cuenta la manera en la cual la dirección d^k en el paso 2 del Algoritmo 3.1.3 fue calculada.

De ahora en adelante consideramos el conjunto de índices

$$K = \{\nu(1) - 1, \nu(2) - 1, \nu(3) - 1, \dots\} \quad (3.4)$$

y de la Proposición 3.1.5 tenemos que

$$\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0. \quad (3.5)$$

3.2. Algoritmo cuasi-Newton sin derivadas con cotas en las variables (BCDF-QNB)

En esta sección presentamos un método cuasi-Newton para resolver el problema (3.1), basado en la fórmula de actualización de Broyden junto con la búsqueda lineal sin derivadas no monótona definida en el Algoritmo 3.1.1 y establecemos los resultados de convergencia. Como mencionamos anteriormente el algoritmo está fuertemente basado en el Algoritmo DF-QNB, desarrollado previamente.

El objetivo del método que proponemos es resolver en forma exacta cuando sea posible, y aproximadamente en caso contrario, el sistema lineal $B_k d = c(x^k)$ con la restricción $x^k + d \in \Omega$, usando la fórmula de actualización de Broyden de rango uno (2.25) como fue considerado en el capítulo anterior.

Algoritmo 3.2.1. BCDF-QNB

Dados $x^0 \in \Omega$, $0 < \gamma < 1$, $0 \leq \theta_0 < \bar{\theta} < 1$, $0 < \tau_{min} < \tau_{max} < 1$, $0 \leq \epsilon < 1$, $\Delta > 0$, $ind = 0$, $imax \in \mathbb{N}$, $imax > 0$.

Considerar $k \leftarrow 0$.

Paso 1: Si $\|c(x^k)\| \leq \epsilon$ detener el proceso.

Paso 2: Cálculo de la matriz B_k

Si $k = 0$ o $ind = imax$ calcular B_k por diferencias finitas como una aproximación de la matriz Jacobiana en x^k .

Si no, actualizar B_k usando (2.25).

Paso 3: Cálculo de la dirección d^k

Paso 3.1: Encontrar d tal que

$$B_k d + c(x^k) = 0, \quad x^k + d \in \Omega \quad y \quad \|d\| \leq \Delta. \quad (3.6)$$

Si tal dirección d es encontrada, definir $d^k = d$, $\theta_{k+1} = \theta_k$, $ind = 0$ e ir al paso 4.

Paso 3.2: Encontrar una solución aproximada de $\min_{x^k + d \in \Omega} \|B_k d + c(x^k)\|$.

Si d satisface

$$\|B_k d + c(x^k)\| \leq \theta_k \|c(x^k)\| \quad y \quad \|d\| \leq \Delta \quad (3.7)$$

definir $d^k = d$, $\theta_{k+1} = \theta_k$, $ind = 0$ e ir al paso 4.

Paso 3.3: Considerar $d^k = 0$, $x^{k+1} = x^k$, $\theta_{k+1} = \frac{\theta_k + \bar{\theta}}{2}$.

Si $\text{ind} = \text{imax}$, definir $\bar{\theta} = \frac{\bar{\theta} + 1}{2}$. Considerar $\text{ind} = 0$ e ir al paso 5.

Si $\text{ind} < \text{imax}$, considerar $\text{ind} = \text{ind} + 1$ e ir al paso 5.

Paso 4: Encontrar α_k usando el Algoritmo 3.1.1 y definir $x^{k+1} = x^k + \alpha_k d^k$.

Paso 5: Actualizar $k \leftarrow k + 1$ e ir al paso 1.

Observar que cuando una solución aproximada del problema con restricciones

$$\min_{x^k + d \in \Omega} \|B_k d + c(x^k)\|$$

verifica (3.7) aceptamos esa solución como dirección de búsqueda. Pero, cuando esto no ocurre, pensamos que es todavía posible encontrar una solución que verifique (3.7) incrementando el parámetro θ_k . Esta es la principal razón por la cual incluimos el paso 3.3. Cuando aumentamos el parámetro θ_k un número imax de veces sin poder encontrar una solución que satisfaga (3.7) el parámetro $\bar{\theta}$ es incrementado. En este caso consideramos que la matriz B_k no es una buena aproximación de la verdadera Jacobiana y entonces calculamos la nueva matriz usando diferencias finitas.

Los siguientes teoremas establecen las hipótesis necesarias para obtener los resultados de convergencia.

Teorema 3.2.2. *Asumiendo que el Algoritmo 3.2.1 genera una sucesión infinita $\{x^k\}_{k \in \mathbb{N}}$, si*

$$\lim_{k \rightarrow \infty} \langle (B_k - J_c(x^k))d^k, c(x^k) \rangle = 0 \quad (3.8)$$

y $\bar{\theta}$ es incrementado un número finito de veces, entonces cada punto de acumulación de la sucesión $\{x^k\}_{k \in K}$ es una solución de $c(x) = 0$ con $x \in \Omega$, donde K está definido en (3.4).

Demostración: Sea x^* un punto de acumulación de $\{x^k\}_{k \in K}$. Entonces existe $K_1 \subset K$ tal que $\lim_{k \in K_1} x^k = x^*$.

Como $\bar{\theta}$ es aumentado un número finito de veces existe $k_0 \in \mathbb{N}$ tal que para todo $k \geq k_0$, $\bar{\theta} = \tilde{\theta} < 1$.

Consideramos los siguientes casos:

a) Supongamos primero que en el proceso del Algoritmo 3.2.1 en un número finito de iteraciones la dirección fue obtenida resolviendo el sistema lineal $B_k d + c(x^k) = 0$ con $x^k + d \in \Omega$ y $\|d\| \leq \Delta$.

Entonces existe $k_1 \in K_1$ tal que para todo $k \geq k_1, k \in K_1$, d^k verifica la fórmula $\|B_k d + c(x^k)\| \leq \theta_k \|c(x^k)\|$, $x^k + d \in \Omega$ con $\|d\| \leq \Delta$. Así para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ tenemos que

$$\begin{aligned} \|B_k d^k + c(x^k)\|^2 &= \langle B_k d^k + c(x^k), B_k d^k + c(x^k) \rangle \\ &= \langle B_k d^k, B_k d^k \rangle + 2\langle B_k d^k, c(x^k) \rangle + \langle c(x^k), c(x^k) \rangle \\ &= \|B_k d^k\|^2 + 2\langle B_k d^k, c(x^k) \rangle + \|c(x^k)\|^2 \leq \tilde{\theta}^2 \|c(x^k)\|^2. \end{aligned}$$

Por lo tanto, como $\tilde{\theta} < 1$, para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ se tiene que

$$\|B_k d^k\|^2 + 2\langle B_k d^k, c(x^k) \rangle \leq (\tilde{\theta}^2 - 1)\|c(x^k)\|^2 < 0.$$

Entonces, como $\|B_k d^k\|^2 \geq 0$, para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$ obtenemos que

$$\langle B_k d^k, c(x^k) \rangle \leq \frac{\tilde{\theta}^2 - 1}{2} \|c(x^k)\|^2.$$

Esto implica que

$$\langle (B_k - J_c(x^k))d^k, c(x^k) \rangle + \langle J_c(x^k)d^k, c(x^k) \rangle \leq \frac{\tilde{\theta}^2 - 1}{2} \|c(x^k)\|^2 \quad (3.9)$$

para todo $k \in K_1$, $k \geq \max\{k_0, k_1\}$.

Sabemos, de la Proposición 3.1.5, que $\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0$ por lo tanto $\lim_{k \in K_1} \alpha_k^2 \|d^k\|^2 = 0$.

Si $\lim_{k \in K_1} \alpha_k \neq 0$ entonces

$$\lim_{k \in K_1} \|d^k\| = 0. \quad (3.10)$$

Tomando límite para $k \in K_1, k \geq \max\{k_0, k_1\}$ en (3.9), y usando (3.8), (3.10) y la continuidad de c y J_c , obtenemos que

$$0 \leq \frac{\tilde{\theta}^2 - 1}{2} \|c(x^*)\|^2.$$

Como $\tilde{\theta} < 1$, tenemos que $\|c(x^*)\| = 0$ como queríamos probar.

Si $\lim_{k \in K_1} \alpha_k = 0$, como la sucesión $\{d^k\}_{k \in K_1}$ es acotada existe $K_2 \subset K_1$ y $\bar{d} \in \mathbb{R}^n$ tal que $\lim_{k \in K_2} d^k = \bar{d}$.

En la búsqueda lineal, para calcular la longitud de paso α_k , el Algoritmo 3.2.1 testea la siguiente desigualdad

$$h(x^k + \alpha d^k) \leq \bar{h}_k + \eta_k - \gamma \alpha^2 \|d^k\|^2. \quad (3.11)$$

El valor inicial de α es 1. Como $\lim_{k \in K_2} \alpha_k = 0$ existe $\bar{k} \in K_2$ tal que $\alpha_k < 1$ para todo

$k \in K_2, k \geq \bar{k}$. Así, para esas iteraciones k existen longitudes de pasos $\bar{\alpha}_k$ que no satisfacen (3.11) y $\lim_{k \in K_2} \bar{\alpha}_k = 0$. Por lo tanto tenemos que

$$h(x^k + \bar{\alpha}_k d^k) > \bar{h}_k + \eta_k - \gamma(\bar{\alpha}_k)^2 \|d^k\|^2 > \bar{h}_k - \gamma(\bar{\alpha}_k)^2 \|d^k\|^2.$$

Como $\|d^k\| \leq \Delta$, $\bar{h}_k \geq h(x_k)$ y $\bar{\alpha}_k > 0$ obtenemos que

$$\frac{h(x^k + \bar{\alpha}_k d^k) - h(x^k)}{\bar{\alpha}_k} > -\gamma \bar{\alpha}_k \Delta^2.$$

Por el Teorema del Valor Medio existe $\xi_k \in [0, 1]$ tal que

$$\langle g(x^k + \xi_k \bar{\alpha}_k d^k), d^k \rangle > -\gamma \bar{\alpha}_k \Delta^2. \quad (3.12)$$

Tomando límite en (3.12) cuando $k \rightarrow \infty$, $k \in K_2$, tenemos que

$$\langle g(x^*), \bar{d} \rangle \geq 0.$$

Por lo tanto se tiene que $\langle J_c(x^*) \bar{d}, c(x^*) \rangle \geq 0$.

Luego tomando límite en (3.9) cuando $k \rightarrow \infty$, $k \in K_2$, y usando (3.8), obtenemos que

$$0 \leq \langle J_c(x^*) \bar{d}, c(x^*) \rangle \leq \frac{\tilde{\theta}^2 - 1}{2} \|c(x^*)\|^2 \leq 0. \quad (3.13)$$

Por lo tanto, por (3.13), concluimos que $\|c(x^*)\| = 0$ como queríamos demostrar.

b) Supongamos ahora que en el proceso del Algoritmo 3.2.1 en infinitas iteraciones la dirección fue obtenida resolviendo el sistema lineal $B_k d + c(x^k) = 0$, con $x^k + d \in \Omega$ y $\|d\| \leq \Delta$.

Entonces existe $K_2 \subset K_1$ tal que para todo $k \in K_2$ se tiene que $B_k d^k + c(x^k) = 0$. Así para todo $k \in K_2$,

$$\langle B_k d^k + c(x^k), c(x^k) \rangle = 0.$$

Por lo tanto, para todo $k \in K_2$ se tiene que

$$\langle B_k d^k + c(x^k) \rangle + \|c(x^k)\|^2 = 0,$$

lo que implica que

$$\langle (B_k - J_c(x^k)) d^k, c(x^k) \rangle + \langle J_c(x^k) d^k, c(x^k) \rangle = -\|c(x^k)\|^2. \quad (3.14)$$

Sabemos que $\lim_{k \in K} \alpha_k^2 \|d^k\|^2 = 0$, por lo tanto $\lim_{k \in K_2} \alpha_k^2 \|d^k\|^2 = 0$.

Si $\lim_{k \in K_2} \alpha_k \neq 0$ entonces

$$\lim_{k \in K_2} \|d^k\| = 0. \quad (3.15)$$

Tomando límite en (3.14) cuando $k \rightarrow \infty$, $k \in K_2$, y usando (3.8), (3.15) y la continuidad de c y J_c , obtenemos que $\|c(x^*)\| = 0$ como queríamos probar.

Si $\lim_{k \in K_2} \alpha_k = 0$, como la sucesión $\{d^k\}_{k \in K_2}$ es acotada existe $K_3 \subset K_2$ y $\bar{d} \in \mathbb{R}^n$ tal que $\lim_{k \in K_3} d^k = \bar{d}$.

Como hicimos en el caso $a)$ tenemos que $\langle J_c(x^*)\bar{d}, c(x^*) \rangle \geq 0$, y tomando límite en (3.14) y usando (3.8) obtenemos que $\langle J_c(x^*)\bar{d}, c(x^*) \rangle = -\|c(x^*)\|^2 \leq 0$. Por lo tanto

$$\|c(x^*)\| = 0$$

como queríamos demostrar. $\square$

Observar que, de acuerdo a la Proposición 3.1.6, tenemos que $\{\|c(x^k)\|\}$ es acotada. Por lo tanto si

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - J_c(x^k))d^k\|}{\|d^k\|} = 0$$

obtenemos la hipótesis (3.8). La última es una condición necesaria y suficiente para obtener convergencia q -superlineal de los métodos cuasi-Newton clásicos [13].

Si no podemos garantizar que $\bar{\theta}$ es aumentado un número finito de veces entonces podemos demostrar el siguiente resultado.

Teorema 3.2.3. *Suponiendo que en el Algoritmo 3.2.1 el parámetro $\bar{\theta}$ es aumentado infinitas veces y definiendo el conjunto K_* al conjunto de índices para los cuales esto ocurre, si*

$$\lim_{k \in K_*} \|B_k - J_c(x^k)\| = 0 \quad (3.16)$$

entonces cada punto de acumulación x^ de la sucesión $\{x^k\}_{k \in K_*}$ es una solución del problema (3.1) o es un minimizador global de $\|c(x^*) + J_c(x^*)(x - x^*)\|$ sujeto a $x \in \Omega$.*

Demostración: Ver demostración del Teorema 2.3.4 del capítulo anterior.

Observación 3.2.4. *Si la matriz B_k , $k \in K_*$, es calculada por diferencias finitas usando un paso h que satisface $h \leq \min\{\epsilon, 1 - \bar{\theta}\}$ entonces la hipótesis (3.16) puede ser omitida.*

Un punto $x^* \in \Omega$ que es un minimizador global de la función $\|c(x^*) + J_c(x^*)(x - x^*)\|$ sujeto a $x \in \Omega$ puede ser visto como una solución del problema de cuadrados mínimos

$$\min_{x \in \Omega} \|A(x - x^*) - b\| \quad (3.17)$$

donde $A = J_c(x^*)$ y $b = -c(x^*)$. La función lineal es el modelo afín de la función c alrededor de x^* . No podemos esperar, en general, encontrar $x^* \in \Omega$ tal que $c(x^*) = 0$ ya que el problema podría no tener solución. Tampoco podemos esperar encontrar $x^* \in \Omega$ tal que $c(x^*) + J_c(x^*)(x - x^*) = 0$ ya que este es un sistema de ecuaciones lineales indeterminado y $J_c(x^*)$ podría no tener rango completo. Por este motivo parece razonable encontrar minimizadores globales de (3.17) cuando el problema no tiene solución.

El caso $M = 1$ en la búsqueda lineal no monótona sin derivadas usado en el Algoritmo 3.2.1 merece comentarios adicionales. Como fue notado en el capítulo anterior para el Algoritmo DF-QNB, la presencia de η_k impone un comportamiento casi monótono de la función de mérito cuando x^k está cerca de una solución.

Como fue hecho en el capítulo anterior, bajo esta condición es posible demostrar que nuestro algoritmo verifica la hipótesis (3.8). Así esta búsqueda particular mejora los resultados del Teorema 3.2.2. Por completitud enunciamos esos resultados.

Lema 3.2.5. *[Lema 2.6, [29]] Suponiendo que el conjunto $\Gamma = \{x \in \mathbb{R}^n : h(x) \leq h(x^0) + \eta\}$ es acotado y que $J_c(x)$ es Lipschitz continua en Γ , si*

$$\sum_{k=1}^{\infty} \|s^k\|^2 = \sum_{k=1}^{\infty} \alpha_k^2 \|d^k\|^2 < \infty,$$

entonces

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=0}^{k-1} \rho_i^2 = 0$$

donde

$$\rho_k = \frac{\|(A_{k+1} - B_k)d^k\|}{\|d^k\|}$$

y

$$A_{k+1} = \int_0^1 J_c(x^k + ts^k) dt.$$

En particular existe una subsucesión de $\{\rho_k\}$ que tiende a cero.

Teorema 3.2.6. *Asumiendo que el Algoritmo 3.2.1 genera una sucesión infinita $\{x^k\}_{k \in \mathbb{N}}$, considerando $M = 1$ en la búsqueda lineal y asumiendo que las hipótesis del Lema 3.2.5 se verifican, si $\bar{\theta}$ es aumentado un número finito de veces entonces existe un punto de acumulación x^* de $\{x^k\}_{k \in \mathbb{N}}$ que es solución de (3.1).*

3.3. Experimentación numérica

En esta sección presentamos algunos resultados obtenidos con una implementación en Fortran 77 de Algoritmo BCDF-QNB. Todas las experimentaciones fueron pasadas en una computadora personal con INTEL(R) Core (TM) 2 Duo CPU E8400 de 3.00 GHz y 3.23 GB de RAM.

Comparamos el desempeño de nuestro algoritmo con los siguientes dos códigos: BOBYQA [43] y el Algoritmo 1 de [18], que llamaremos FKM de ahora en adelante. BOBYQA es un algoritmo iterativo sin derivadas para encontrar un mínimo de una función objetivo sujeto a restricciones de cotas en las variables. En estos experimentos BOBYQA resuelve el problema de cuadrados mínimos: $\min \|c(x)\|^2$, sujeto a $x \in \Omega$. FKM es un método de punto interior para resolver sistemas indeterminados con restricciones de caja que usa la función de mérito $h(x) = \frac{1}{2}\|c(x)\|^2$. La implementación en [18] permite el uso de diferencias finitas para aproximar la matriz Jacobiana en todas las iteraciones.

Somos conscientes que ambos códigos pertenecen a diferentes clases de algoritmos sin derivadas. BOBYQA usa una función general sin tener en cuenta las ventajas de la estructura de la función que aparece en el problema de cuadrados mínimos y usa interpolación polinomial para construir modelos aproximantes de la función objetivo. La comparación con BOBYQA no parece entonces ser justa ya que no hace uso de la estructura de la función de cuadrados mínimos. Por causa de esto consideramos al Algoritmo FKM, que toma en cuenta la función de cuadrados mínimos pero aproxima las derivadas por diferencias finitas en todas las iteraciones. Esta misma idea de comparación fue previamente usada en [53] con el objetivo de comparar el desempeño de un algoritmo sin derivadas para resolver un problema de cuadrados mínimos sin restricciones.

Como es usual en artículos de optimización sin derivadas estamos interesados en el número necesario de evaluaciones funcionales que se necesitan para satisfacer los criterios de parada.

3.3.1. Problemas test

Hemos usado una colección de problemas definidos como conjuntos factibles de problemas de programación no lineal del libro de Hock y Schittkowski [22] y otros problemas que fueron

presentados en un trabajo de Ryoo y Sahinidis [49]. También testeamos algunos problemas descriptos en [12].

Algunos de estos problemas fueron previamente usados en [18] para resolver el mismo problema pero con el uso de derivadas.

En la Tabla 3.1 mostramos los datos de los problemas. En la columna 1 mostramos el número del problema, en la columna 2 de donde fue extraído el problema y en la última columna el número de variables (n) y el número de ecuaciones (m).

Problema	Fuente	n	m
1	Problema 46 de [22]	5	2
2	Problema 53 de [22]	5	3
3	Problema 55 de [22]	6	6
4	Problema 56 de [22]	7	4
5	Problema 60 de [22]	3	1
6	Problema 63 de [22]	3	2
7	Problema 77 de [22]	5	2
8	Problema 79 de [22]	5	3
9	Problema 81 de [22]	5	3
10	Problema 111 de [22]	10	3
11	Problema 1 de [49]	10	7
12	Problema 3 de [49]	12	9
13	Problema 4 de [49]	4	2
14	Problema 5 de [49]	5	3
15	Problema 10 de [49]	2	1
16	Problema 11 de [49]	3	2
17	Problema 2 de [12]	300	150
18	Problema 4 de [12]	300	150

Tabla 3.1: Datos de los problemas

En los problemas 2, 5, 6, 9 y 10 usamos los puntos iniciales y cotas originales. Los problemas 1, 4, 7 y 8 originalmente no tenían restricciones de cota. Para estos problemas hemos usado las cotas artificiales que han sido utilizadas en [18] manteniendo los puntos iniciales originales. En el problema 3 hemos cambiado el punto inicial ya que FKM requiere puntos iniciales interiores. En este caso consideramos $x^0 = (0.5, 2, 0.5, 0.5, 0.5, 2)^T$. En los problemas 11, 12, 13, 14, 15, y 16 hemos usado las cotas originales y los siguientes puntos iniciales: $x^0 = (1.5, 1.5, 1, 1.5, 1, 90, 90, 3.5, 3, 150)^T$, $x^0 = \frac{1}{2}(u + l)$, $x^0 = (10, 40, -10, 200)^T$, $x^0 = (1000, 1000, 1000, 120, 120)^T$, $x^0 = \frac{1}{2}(u + l)$ y $x^0 = (5, 5, 200)^T$ respectivamente.

En los problemas 17 y 18 hemos considerado dos elecciones de puntos iniciales y cotas (caso (a) y caso (b)). El caso (a) corresponde al uso del punto inicial $x^0(1 : n) = 2$ con cotas $u_i = 100$, $l_i = -100$, $i = 1, \dots, n$. Y el caso (b) corresponde a usar $x^0(1 : n) = 150$ como punto inicial y cotas $u_i = 300$, $l_i = -300$, $i = 1, \dots, n$.

3.3.2. Implementación

Detalles de la implementación del Algoritmo BCDF-QNB:

1. Los parámetros usados en el Algoritmo 3.1.1 fueron:

$$M = 2, \tau_{min} = 0.1, \tau_{max} = 0.5, \gamma = 10^{-4}, \epsilon = 10^{-6}, \theta_0 = 0.5, \bar{\theta} = 0.999, \Delta = 10^{12}, \\ imax = 10, \eta_0 = 1,$$

- $\forall k \in \mathbb{N}, k \geq 1, \eta_k = \frac{\|c(x^0)\|}{(1+k)^2}$ cuando $n = 300$.
- $\forall k \in \mathbb{N}, k \geq 1, \eta_k = \frac{\|c(x^0)\|}{2^k}$ en otro caso.

2. La matriz inicial B_0 fue calculada por diferencias finitas como una aproximación de la matriz Jacobiana en x^0 .
3. Como hicimos en el Algoritmo DF-QNB del capítulo anterior, hemos usado el Algoritmo DACCIM, descrito en [50], para encontrar una solución del sistema lineal $B_k d = -c(x^k)$ con $x^k + d \in \Omega$. Para resolver el problema de cuadrados mínimos $\min_{x^k+d \in \Omega} \|B_k d + c(x^k)\|^2$ hemos usado el Algoritmo BVLS, descrito en [51].
4. La condición de finalización para nuestro algoritmo fue: $\|c(x^k)\| \leq \epsilon$.
5. El número máximo de evaluaciones funcionales permitadas fue $MAX_{FE} = 10000$ (considerando aquí las evaluaciones realizadas para calcular la matriz inicial B_0 y las matrices B_k que fueron calculadas por diferencias finitas cuando en el procedimiento de BCDF-QNB fuera requerido).

La implementación de BOBYQA, en Fortran 77, es la versión original de M.J.D. Powell [43] con sus respectivos criterios de finalización: el algoritmo finaliza cuando el radio de la región de confianza alcanza su cota inferior, $\rho_{end} = 10^{-6}$, o cuando se alcanza el número máximo permitido de evaluaciones funcionales. También adicionamos el siguiente criterio de finalización: el algoritmo se detiene si el valor $\|c(x^k)\|$ es menor que el obtenido por el Algoritmo BCDF-QNB.

La implementación de FKM en Matlab es la versión original de Francisco, Krejić y Martínez [18]. Como en BCDF-QNB el número máximo de evaluaciones funcionales fue 10000 y el algoritmo se detiene cuando $\|c(x^k)\| \leq 10^{-6}$. En esta implementación han sido consideradas las evaluaciones usadas para aproximar la matriz Jacobiana en todas las iteraciones.

3.3.3. Resultados numéricos

En la Tabla 3.2 mostramos los resultados obtenidos tomando en cuenta el número de evaluaciones funcionales y el valor final $\|c(x^{end})\|$ para los códigos: BCDF-QNB (BCDF), BOBYQA (BQA) y FKM. También mostramos el número del problema (columna 1), el número de evaluaciones funcionales (c -eval, columna 2) y el valor funcional final ($\|c(x^{end})\|$, columna 3).

Prob.	c -eval			$\ c(x^{end})\ $		
	BCDF	BQA	FKM	BCDF	BQA	FKM
1	28	111	49	1.841433D-07	3.418782D-08	1.297465D-07
2	7	59	13	1.026235D-09	1.250089D-10	8.441465D-09
3	8	94	15	2.101248D-09	1.399019D-09	1.989612D-08
4	13	76	25	3.366884D-07	5.284583D-09	5.070781D-07
5	12	66	61	3.043343D-10	2.722729D-15	9.073879D-07
6	14	55	21	6.312838D-09	2.607497D-10	1.140705D-10
7	16	119	37	2.765712D-07	1.147991D-08	1.254718D-11
8	14	89	25	6.017154D-08	4.784848D-08	4.388764D-09
9	13	63	25	9.827276D-07	2.075567D-07	5.815963D-13
10	26	250	57	4.219129D-09	3.109068D-09	1.724752D-10
11	20	57	282	2.468423D-07	1.515770D-07	3.638097D-12
12	30	115	274	4.372665D-08	1.434150D-09	2.239968D-08
13	11	141	21	5.922943D-07	2.128820D-07	1.931713D-07
14	30	360	55	2.363432D-08	1.157911D-03	6.177695D-09
15	9	30	13	1.610945D-08	2.088413D-09	7.235173D-07
16	7	48	13	6.664413D-08	1.918797D-08	5.982311D-07
17(a)	303	696	2409	4.544648D-11	6.324139D-11	3.140185D-16
18(a)	337	10000	2108	4.107578D-07	5.715843D-03	1.707833D-08
17(b)	303	794	2409	2.166159D-14	9.206523D-11	3.080427D-13
18(b)	853	10000	5118	9.871958D-07	★	5.625058D-12

Tabla 3.2: Con ★ indicamos que BOBYQA no alcanza una solución aproximada dentro del número máximo permitido de evaluaciones funcionales.

Estos resultados muestran la efectividad del método BCDF-QNB. Debemos mencionar que para esta colección de problemas, BCDF-QNB ha usado diferencias finitas sólo para realizar el cálculo de la matriz inicial B_0 . Esta es la razón principal por la cual BCDF-QNB ha realizado una cantidad menor de evaluaciones funcionales que FKM. También notamos que nuestro algoritmo realiza menos evaluaciones funcionales que BOBYQA en todos estos problemas mientras que FKM lo hace en 16 problemas. Creemos que este comportamiento se debe al hecho que BOBYQA no toma en cuenta las ventajas de la estructura del problema.

Por otro lado, cuando consideramos $\|c(x^{end})\|$ como una medida del desempeño de los algoritmos, pudimos observar que FKM da mejores resultados que BCDF-QNB en 13 de los 20 problemas. Creemos que estos resultados están asociados a que FKM usa una mejor aproximación de la matriz Jacobiana.

Capítulo 4

Algoritmo de restauración inexacta sin derivadas utilizando filtros

En este capítulo estamos interesados en resolver el siguiente problema de programación no lineal:

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeto a } c(x) = 0 \end{aligned} \tag{4.1}$$

siendo $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $c = (c_1, \dots, c_m)$ y las funciones f y c_i , para $i = 1, \dots, m$, son continuamente diferenciables pero sus derivadas no están disponibles. Denotamos por $J_c(\cdot)$ a la matriz Jacobiana de c y consideraremos la función $h(x) = \|c(x)\|$ que mide la infactibilidad de las restricciones en cada $x \in \mathbb{R}^n$, donde $\|\cdot\|$ denota la norma dos o norma euclídea.

En este capítulo proponemos un método sin derivadas basado en el algoritmo de restauración inexacta introducido en [19]. En ese trabajo los autores definen un método de filtros globalmente convergente para problemas de optimización no lineal considerando que las derivadas de la función objetivo y de las restricciones están disponibles. Ese método de filtros pertenece a la clase de metodologías que tratan a f y h como dos objetivos independientes. Cada iteración está compuesta de dos fases: la fase de restauración o factibilidad en la cual se intenta mejorar la factibilidad sin que intervenga la función objetivo y la fase de optimalidad o minimización en la cual se debe disminuir el valor de la función objetivo sobre una aproximación tangente a las restricciones. Como es mencionado en [19], los algoritmos de filtros definen una región prohibida memorizando los pares $(f(x^k), h(x^k))$ obtenidos por iteraciones previas, y luego evitando pares que esten dominados por los pares memorizados utilizando la regla Pareto de dominación:

$$(f(x), h(x)) \text{ domina a } (f(y), h(y)) \text{ si y sólo si } f(y) \geq f(x) \text{ y } h(y) \geq h(x).$$

Un filtro es una conjunto de pares $(f(x), h(x))$ tal que ningún par del conjunto es dominado

por otro. En una iteración k , el par $(f(x^k), h(x^k))$ es candidato a ingresar al filtro si éste no está dominado por ningún par del filtro.

A modo ilustrativo, supongamos que tenemos un filtro $F = \{(f_1, h_1), (f_2, h_2), (f_3, h_3)\}$. En la Figura 4.1 pueden observarse los pares del filtro y la región sombreada es la región prohibida que define el filtro.

Figura 4.1: Filtro F y región prohibida

Como puede observarse en la Figura 4.2, el par $(\tilde{f}, \tilde{h})$ está dominado por el par (f_2, h_2) con lo cual no será incluido en el filtro.

Figura 4.2: El par $(\tilde{f}, \tilde{h})$ no puede ser incluido en el filtro

En la Figura 4.3, el par $(\bar{f}, \bar{h})$ no está dominado por ningún par del filtro y en este caso podría ser incluido en el filtro.

Figura 4.3: El par $(\bar{f}, \bar{h})$ no puede ser incluido en el filtro

Cada vez que un nuevo par ingresa al filtro son eliminados de él todos los pares que son dominados por la nueva entrada generando una nueva región prohibida. Siguiendo con el ejemplo, los pares (f_2, h_2) y (f_3, h_3) deben ser removidos del filtro. Obtenemos de esta manera un nuevo filtro para la siguiente iteración, $F = \{(f_1, h_1), (\bar{f}, \bar{h})\}$, como puede observarse en la Figura 4.4.

Figura 4.4: Nuevo filtro y región prohibida

Más detalles sobre métodos de filtros pueden encontrarse en [16, 17, 19] y las referencias que allí se mencionan.

El algoritmo desarrollado en este capítulo está basado en la construcción de modelos de interpolación multivariados de la función objetivo y de las funciones que definen a las restricciones [11], lo cual es una de las principales diferencias con [19].

La fase de restauración debe resolver, con una metodología sin derivadas, un sistema indeterminado de ecuaciones no lineales con restricciones de cotas en las variables. En nuestra implementación realizamos esta fase utilizando el método sin derivadas que fue presentado en el capítulo anterior.

Por otro lado, la fase de optimización debe resolver un problema de optimización sin derivadas con restricciones lineales. Utilizaremos en esta fase un algoritmo de región de confianza con restricciones lineales que utiliza un modelo de interpolación lineal como aproximación de la función objetivo.

4.1. Algoritmo de filtros sin derivadas (DFF)

Desarrollamos un algoritmo que genera sucesiones $\{x^k\}_{k \in \mathbb{N}}$, $\{z^k\}_{k \in \mathbb{N}}$ y para probar los resultados de convergencia global asumimos las siguientes hipótesis:

Hipótesis generales:

- (H1) Los iterados x^k y z^k se mantienen dentro de un conjunto compacto convexo $X \subset \mathbb{R}^n$.
- (H2) Las funciones f , c_i , para $i = 1, \dots, m$, son continuamente diferenciables en un conjunto abierto que contiene a X .
- (H3) Las funciones ∇f , ∇c_i , para $i = 1, \dots, m$, son Lipschitz continuas en un conjunto abierto que contiene a X con constantes L_1 , $L_2 > 0$ respectivamente, es decir,

$$\|\nabla f(x) - \nabla f(y)\| \leq L_1 \|x - y\|$$

$$\|\nabla c_i(x) - \nabla c_i(y)\| \leq L_2 \|x - y\|, \quad \text{para } i = 1, \dots, m$$

para todo x, y en un conjunto abierto que contiene a X .

Antes de seguir adelante con los detalles del algoritmo introducimos algunos conceptos y resultados de modelos de interpolación polinomial multivariados de la función objetivo y de las restricciones. Estos conceptos y resultados serán usados a lo largo de este capítulo y pueden ser encontrados con más detalle en el libro “*Introduction to derivative-free optimization*” [10].

Cada conjunto de interpolación $Y = \{y^0, y^1, \dots, y^n\} \subset \mathbb{R}^n$, contenido en la bola $B(y^0, \Delta(Y))$ de radio $\Delta(Y) = \max_{1 \leq i \leq n} \|y^i - y^0\|$, se dice *equilibrado* para interpolación lineal si la matriz de direcciones $S = [y^1 - y^0 \quad y^2 - y^0 \quad \dots \quad y^n - y^0]^T$ es no singular. La definición de conjunto equilibrado es independiente de la base usada para el espacio lineal de polinomios de grado

menor o igual a 1. Por lo tanto si Y es un conjunto equilibrado para la base natural, entonces es equilibrado para cualquier otra base que se elija [10, Ch. 2].

El *gradiente simplex* de f en y^0 está definido por

$$\nabla_s f(y^0) = S^{-1} \delta f(Y)$$

donde $\delta f(Y) = (f(y^1) - f(y^0), f(y^2) - f(y^0), \dots, f(y^n) - f(y^0))^T$.

Si consideramos $m_f(x) = f(y^0) + g_f^T(x - y^0)$ el modelo lineal de interpolación de $f(x)$ sobre el conjunto Y entonces se tiene que $g_f = \nabla_s f(y^0)$. Por lo tanto el gradiente simplex de f está cercanamente relacionado con la interpolación polinomial lineal multivariada.

Las propiedades geométricas de Y determinan la calidad del correspondiente gradiente simplex g_f como una aproximación del gradiente exacto de la función objetivo f . Estamos interesados en la calidad de $m_f(x)$ y g_f en la bola $B(y^0, \Delta(Y))$ con centro en y^0 y radio $\Delta(Y)$.

Para todo $x \in B(y^0, \Delta(Y))$, considerando la matriz escalada $\bar{S} = \frac{S}{\Delta(Y)}$, se obtiene que

$$|f(x) - m_f(x)| \leq \kappa_{ef} \Delta^2(Y), \quad (4.2)$$

$$\|\nabla f(x) - \nabla m_f(x)\| \leq \kappa_{eg} \Delta(Y), \quad (4.3)$$

donde $\kappa_{eg} = L_1(1 + \frac{\sqrt{n}}{2} \|\bar{S}^{-1}\|)$ y $\kappa_{ef} = \kappa_{eg} + \frac{L_1}{2}$.

Analógamente, bajo las hipótesis anteriores, si consideramos para todo $j = 1, \dots, m$, $m_{c_j}(x) = c_j(y^0) + g_{c_j}^T(x - y^0)$ como el modelo lineal de interpolación de $c_j(x)$ sobre Y , se tiene que $g_{c_j} = \nabla_s c_j(y^0)$ así como también las siguientes cotas de error

$$|c_j(x) - m_{c_j}(x)| \leq \kappa_{ec} \Delta^2(Y), \quad (4.4)$$

$$\|\nabla c_j(x) - \nabla m_{c_j}(x)\| \leq \kappa_{eg_c} \Delta(Y), \quad (4.5)$$

donde $\kappa_{eg_c} = L_2(1 + \frac{\sqrt{n}}{2} \|\bar{S}^{-1}\|)$ y $\kappa_{ec} = \kappa_{eg_c} + \frac{L_2}{2}$.

Si ahora consideramos como una aproximación de $J_c(y)$ a la matriz $A(y)$, cuya j -ésima fila es la traspuesta de $\nabla m_{c_j}(y)$, obtenemos que

$$\|J_c(y) - A(y)\| \leq \kappa_{eJ_c} \Delta(Y), \quad (4.6)$$

donde $\kappa_{eJ_c} = \sqrt{m} \kappa_{eg_c}$.

Dado un iterado z^k consideraremos la siguiente hipótesis

(H4) El gradiente simplex usado para aproximar al gradiente de la función objetivo satisface la siguiente cota de error: $\|\nabla f(z^k) - \nabla_s f(z^k)\| \leq k_{eg} \Delta_f^k$ donde Δ_f^k es el radio de la bola que contiene los puntos de interpolación.

Las derivadas simplex usadas para aproximar a la verdadera matriz Jacobiana satisfacen la siguiente cota de error: $\|J_c(z^k) - A(z^k)\| \leq k_{eJ_c} \Delta_c^k$ donde Δ_c^k es el radio de la bola que contiene los puntos de interpolación.

Los resultados de convergencia global del método presentado en [19] son obtenidos sin discutir los detalles de los algoritmos usados en las fases internas. Los autores prueban que su algoritmo produce puntos factibles $\bar{x}$ que satisfacen

$$\liminf_{x \rightarrow \bar{x}} \|P_{T(x)}(x - \nabla f(x)) - x\| = 0, \quad (4.7)$$

donde $P_{T(z)}(w)$ es la proyección ortogonal de $w \in \mathbb{R}^n$ sobre el conjunto cerrado

$$T(z) = \{x \in \mathbb{R}^n : J_c(z)(x - z) = 0\}$$

que es una linealización del conjunto $\{x \in \mathbb{R}^n : c(x) = c(z)\}$ en el punto z .

La dirección $P_{T(z)}(z - \nabla f(z)) - z$ aparece como una condición de optimalidad secuencial en la condición AGP (Approximate Gradient Projected) definida por Martínez y Svaiter en [37].

Como estamos trabajando con problemas en los cuales las derivadas de las funciones no están disponibles, no podemos calcular de manera exacta el conjunto $T(z)$ y el gradiente de la función objetivo. Por lo tanto, en este contexto, probaremos que nuestro algoritmo de filtros sin derivadas genera una sucesión $\{x^k\}_{k \in \mathbb{N}}$ que posee un punto límite factible $\bar{x} \in \mathbb{R}^n$, $\bar{x} = \lim_{k \in \mathcal{K}} x^k$ para algún sunconjunto infinito $\mathcal{K}$ de $\mathbb{N}$, que satisface

$$\lim_{k \in \mathcal{K}} \|d_c(x^k)\| = 0, \quad (4.8)$$

donde $d_c(z) = P_{L(z)}(z - \nabla_s f(z)) - z$ y $L(z) = \{x \in \mathbb{R}^n : A(z)(x - z) = 0\}$.

Estos puntos factibles $\bar{x}$ serán llamados puntos *cuasi-estacionarios* a lo largo de este trabajo.

Siguiendo las ideas establecidas en [19], presentamos el algoritmo de filtros sin derivadas sin especificar cuales son los algoritmos internos utilizados.

Este algoritmo construye una sucesión $\{F_k\}$ de conjuntos de filtros compuestos por pares $(f_j, h_j) \in \mathbb{R}^2$. El algoritmo también tiene en cuenta una sucesión de conjuntos que definen una región prohibida $\mathcal{F}_k \subset \mathbb{R}^n$, $\mathcal{F}_k = \{x \in \mathbb{R}^n : f(x) \geq f_j, h(x) \geq h_j, \text{ para algún } (f_j, h_j) \in F_k\}$, los cuales son formalmente definidos en cada paso del algoritmo pero nunca son construidos. Cada iteración empieza con un filtro y su correspondiente región prohibida.

Generalmente en los algoritmos de filtros en vez de considerar los pares $(f(x^k), h(x^k))$ se tienen en cuenta los pares $(f(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k))$ para definir el filtro como puede observarse en la figura 4.5. De esta manera, gracias a este margen extra, se pueden probar los teoremas de convergencia global de los métodos de filtros.

Figura 4.5: Gráfico aproximado del filtro

Dado un iterado x^k , se define la holgura del filtro en x^k como

$$H_k = \min\{1, \min\{h_j : (f_j, h_j) \in F_k, f_j \leq f(x^k)\}\}. \quad (4.9)$$

Figura 4.6: Holgura del filtro

Algoritmo 4.1.1. *Algoritmo de Filtros sin derivadas (DFF).*

Dados $x^0 \in \mathbb{R}^n$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$, $\alpha \in (0, 1)$, $\beta > 0$, $\epsilon_f > 0$, $\epsilon_I > 0$, $\{\delta_k\}_{k \in \mathbb{N}}$, $\delta_k > 0$, $\delta_k \rightarrow 0$. Considerar $k \leftarrow 0$.

Paso 1 : Definir $(\tilde{f}, \tilde{h}) = (f(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k))$.

Construir el conjunto $\overline{F}_k = F_k \cup \{(\tilde{f}, \tilde{h})\}$.

Definir el conjunto $\overline{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n : f(x) \geq \tilde{f}, h(x) \geq \tilde{h}\}$.

Paso 2: Fase de restauración

Si $h(x^k) = 0$ definir $z^k = x^k$.

Si no calcular $z^k \notin \overline{\mathcal{F}}_k$ tal que $h(z^k) < (1 - \alpha)h(x^k)$ and $\|z^k - x^k\| \leq \beta h(x^k)$.

Si esto no es posible detener el proceso sin éxito. **FIN**.

Paso 3: Fase de optimalidad

3.1 Construir o actualizar $Y_c^k = \{z^k, y_c^1, \dots, y_c^n\}$, un conjunto de puntos de interpolación centrado en z^k tal que el radio $\Delta_c^k = \max_{i=1, \dots, n} \{\|y_c^i - z^k\|\}$ verifique $\Delta_c^k \leq \beta \min\{\max\{h(x^k), H_k\}, \delta_k\}$.

Calcular $A_k = A(z^k)$ usando derivadas simplex interpolando sobre Y_c^k .

Definir $L(z^k) = \{x \in \mathbb{R}^n : A_k(x - z^k) = 0\}$.

Construir o actualizar $Y_f^k = \{z^k, y_f^1, \dots, y_f^n\}$, un conjunto de puntos de interpolación centrado en z^k tal que el radio $\Delta_f^k = \max_{i=1, \dots, n} \{\|y_f^i - z^k\|\}$ verifique $\Delta_f^k \leq \delta_k$.

Calcular $\nabla_s f(z^k)$, interpolando sobre Y_f^k , y $d_c(z^k) = P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k$.

3.2 Si $h(x^k) = 0$, $\max\{\Delta_f^k, \Delta_c^k\} < \epsilon_I$ y $\|d_c(z^k)\| < \epsilon_f$ finalizar el proceso con convergencia finita. **FIN**.

3.3 Calcular, con un algoritmo sin derivadas, $x_T \notin \overline{\mathcal{F}}_k$ tal que $x_T \in L(z^k)$ y $f(x_T) \leq f(z^k)$.

Si $z^k = x^k$ y no existe x_T tal que $f(x_T) < f(z^k)$ considerar $\Delta_f^k = \alpha \Delta_f^k$, $\Delta_c^k = \alpha \Delta_c^k$ e ir al Paso 3.1.

Si no definir $x^{k+1} = x_T$.

Paso 4 : Actualización del filtro

Si $f(x^{k+1}) < f(x^k)$ considerar $F_{k+1} = F_k$, $\mathcal{F}_{k+1} = \mathcal{F}_k$ (*f-iteración*).

Si no considerar $F_{k+1} = \overline{F}_k - \{(f, h) \in F_k \text{ dominados por } (\tilde{f}, \tilde{h})\}$, $\mathcal{F}_{k+1} = \overline{\mathcal{F}}_k$ (*h-iteración*).

$k \leftarrow k + 1$, e ir al Paso 1.

Observar que al inicio de cada iteración el par $(f(x^k) - \alpha h(x^k), h(x^k) - \alpha h(x^k))$ es introducido temporariamente al filtro. Después de una iteración exitosa este par se convierte en una entrada permanente sólo si la iteración no produce un decrecimiento en la función objetivo.

Si existe z^k tal que $h(z^k) < (1 - \alpha)h(x^k)$, para aceptar este z^k , sólo es necesario verificar que $z^k \notin \mathcal{F}_k$. Como $x^k \notin \mathcal{F}_k$ y $\mathcal{F}_k$ es un conjunto cerrado, existe un entorno de x^k tal que $y \notin \mathcal{F}_k$ para todo y en ese entorno. Basados en estas consideraciones el algoritmo tiene posibilidades de encontrar un punto restaurado z^k .

Además, cuando $h(x^k) = 0$ es necesario encontrar x^{k+1} que satisfaga $f(x^{k+1}) < f(x^k)$, para cumplir la condición que $x^{k+1} \notin \overline{\mathcal{F}}_k$. Como no estamos trabajando con las derivadas verdaderas la dirección $d_c(z^k)$ puede no ser una dirección de descenso de f en z^k sobre $L(z^k)$, aunque esta sea no nula. Esto puede pasar porque los gradientes simplex pueden no ser aproximaciones suficientemente buenas de los gradientes verdaderos. Consecuentemente el proceso usado en la fase de optimalidad puede no ser capaz de encontrar un punto x_T tal que $f(x_T) < f(z^k)$. Si $z^k \neq x^k$, como $z^k \notin \overline{\mathcal{F}}_k$, es posible aceptar $x_T = z^k$ y $x^{k+1} = z^k$. Pero cuando $z^k = x^k$ y no existe x_T tal que $f(x_T) < f(z^k)$, proponemos recomenzar la fase de optimalidad recalculando los gradientes simplex de f y la matriz A_k reduciendo los radios Δ_f^k y Δ_c^k de los conjuntos de interpolación.

El siguiente lema garantiza bajo que condiciones $d_c(z^k)$ es una dirección de descenso de f en z^k sobre el conjunto $L(z^k)$.

Lema 4.1.2. *Dados $\epsilon > 0$ y $z^k \in \mathbb{R}^n$, si $\|d_c(z^k)\| = \|P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k\| > \epsilon$ y $\|\nabla f(z^k) - \nabla_s f(z^k)\| < \frac{\epsilon}{4}$ entonces*

$$\nabla^T f(z^k) d_c(z^k) < -\frac{1}{4} \|d_c(z^k)\|^2. \quad (4.10)$$

Demostración: Como la proyección $P_{L(z^k)}$ es no expansiva,

$$\|P_{L(z^k)}(z^k - \nabla f(z^k)) - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| \leq \|\nabla f(z^k) - \nabla_s f(z^k)\|,$$

entonces obtenemos que

$$\|d_c(z^k)\| \leq \|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| + \|\nabla f(z^k) - \nabla_s f(z^k)\|. \quad (4.11)$$

Por lo tanto, como $\|d_c(z^k)\| > \epsilon$ y $\|\nabla f(z^k) - \nabla_s f(z^k)\| < \frac{\epsilon}{4}$, se tiene que

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| \geq \|d_c(z^k)\| - \|\nabla f(z^k) - \nabla_s f(z^k)\| > \epsilon - \frac{1}{4} \epsilon = \frac{3}{4} \epsilon > 0.$$

Como $\nabla^T f(z^k) d_c(z^k) = (\nabla f(z^k) - \nabla_s f(z^k))^T d_c(z^k) + \nabla_s^T f(z^k) d_c(z^k)$, entonces

$$\begin{aligned} \nabla^T f(z^k) d_c(z^k) &\leq |(\nabla f(z^k) - \nabla_s f(z^k))^T d_c(z^k)| + \nabla_s^T f(z^k) d_c(z^k) \\ &\leq \|\nabla f(z^k) - \nabla_s f(z^k)\| \|d_c(z^k)\| + \nabla_s^T f(z^k) d_c(z^k). \end{aligned}$$

Por lo tanto considerando la desigualdad

$$\nabla_s^T f(z^k) d_c(z^k) \leq -\frac{\|d_c(z^k)\|^2}{2}, \quad (4.12)$$

la cual es obtenida de una forma análoga a la obtenida en [35, Sec. 2.6, pag. 140] reemplazando $\nabla f(z^k)$ por $\nabla_s f(z^k)$ (ver en el Apéndice la demostración de la desigualdad), obtenemos que

$$\begin{aligned} \nabla^T f(z^k) d_c(z^k) &\leq \frac{\epsilon}{4} \|d_c(z^k)\| - \frac{\|d_c(z^k)\|^2}{2} \\ &= \left(\frac{\epsilon}{4 \|d_c(z^k)\|} - \frac{1}{2} \right) \|d_c(z^k)\|^2 < \left(\frac{1}{4} - \frac{1}{2} \right) \|d_c(z^k)\|^2 = -\frac{1}{4} \|d_c(z^k)\|^2. \end{aligned}$$

Por lo tanto $d_c(z^k)$ es una dirección de descenso de f en z^k . □

Observación 4.1.3. *Bajo las hipótesis del lema anterior si z^k no pertenece a $\overline{\mathcal{F}}_k$, el cual es un conjunto cerrado, entonces existen $\Delta > 0$ y $t > 0$ tal que si $t \|d_c(z^k)\| < \Delta$ entonces $z^k + t d_c(z^k)$ no pertenece a la región prohibida $\overline{\mathcal{F}}_k$ y $f(z^k + t d_c(z^k)) < f(z^k)$. Cuando $h(x^k) = 0$, por construcción tenemos que $z^k = x^k$ y $z^k \in \overline{\mathcal{F}}_k$. En este caso, como $z^k \notin \mathcal{F}_k$, el cual es un conjunto cerrado, existe $\Delta > 0$ y $t > 0$ tal que si $t \|d_c(z^k)\| < \Delta$ entonces $z^k + t d_c(z^k)$ no pertenece a la región $\mathcal{F}_k$. Además, como $f(z^k + t d_c(z^k)) < f(z^k)$ se obtiene que $z^k + t d_c(z^k) \notin \overline{\mathcal{F}}_k$.*

Lema 4.1.4. *El Algoritmo 4.1.1 está bien definido.*

Demostración: Si el método usado en la fase de restauración no es capaz de encontrar un punto z^k que satisfaga las condiciones requeridas entonces el Algoritmo 4.1.1 se detiene.

En la fase de optimalidad, cuando $z^k \neq x^k$ siempre existe $x_T \notin \overline{\mathcal{F}}_k$ tal que $f(x_T) \leq f(z^k)$ ya que $z^k \notin \overline{\mathcal{F}}_k$ y entonces es posible aceptar $x_T = z^k$.

Cuando x^k es factible, $z^k = x^k$, si es posible encontrar x_T con $f(x_T) < f(z^k)$ entonces x^{k+1} puede definirse. Si esto no es posible, el algoritmo recommienza la fase de optimalidad con radios menores Δ_f^k y Δ_c^k , con el fin de mejorar las aproximaciones de los gradientes verdaderos de f y c_i , para $i = 1, \dots, m$. En este caso, en un número finito de iteraciones los radios Δ_f^k y Δ_c^k serán lo suficientemente pequeños y si $\|d_c(z^k)\|$ es suficientemente grande, por el Lema 4.1.2 y la Observación 4.1.3, será posible obtener $x_T \notin \overline{\mathcal{F}}_k$ tal que $f(x_T) < f(x^k)$

y entonces x^{k+1} podrá ser definido. Por otro lado, si $\|d_c(z^k)\| < \epsilon_f$ y $\max\{\Delta_f^k, \Delta_c^k\} < \epsilon_I$ el Algoritmo 4.1.1 finaliza porque se satisface el criterio de parada. $\square$

Como fue mencionado en [19] hay algunos resultados que se desprenden directamente de la construcción del algoritmo:

(R1) Dado $k \in \mathbb{N}$, $x^{k+p} \notin \mathcal{F}_{k+1}$ para todo $p \geq 1$.

(R2) Dado $k \in \mathbb{N}$, al menos una de las siguientes situaciones debe ocurrir:

1. $f(x^{k+1}) < f(x^k) - \alpha h(x^k)$.
2. $h(x^{k+1}) < (1 - \alpha)h(x^k)$.

(R3) Dado $k \in \mathbb{N}$, $h_j > 0$ para todo $j \in \mathbb{N}$ tal que $(f_j, h_j) \in F_k$. Consecuentemente $H_k > 0$ para todo $k \in \mathbb{N}$.

Observación 4.1.5. Cuando x^k está en un entorno de un punto factible, se verifica que $h(x^k) < 1$ y entonces $h(x^k) \leq H_k$. Por definición de H_k , $H_k \leq 1$. Así si $H_k = 1$ entonces $h(x^k) \leq H_k$. Cuando $H_k < 1$, existe $h_j < 1$ tal que $f_j \leq f(x^k)$, $(f_j, h_j) \in F_k$ y $H_k = h_j$. En este caso, como $x^k \notin \mathcal{F}_k$, debe ocurrir que $h(x^k) < h_j$. Por lo tanto $h(x^k) \leq H_k$.

4.2. Algoritmos internos

La metodología de restauración inexacta nos brinda la posibilidad de utilizar diferentes métodos para resolver cada fase. En esta sección describiremos los algoritmos que usaremos en cada fase. Mostraremos también que los mismos verificarán las condiciones necesarias para obtener la global convergencia del Algoritmo DFF.

4.2.1. Fase de restauración

Para resolver esta fase utilizamos el Algoritmo BCDF-QNB, descrito en el capítulo anterior, como subalgoritmo del Algoritmo DFF. BCDF-QNB es un método sin derivadas para resolver sistemas de ecuaciones indeterminados con cotas en las variables.

Dado un iterado x^k aplicamos BCDF-QNB, comenzando del punto inicial $y^0 = x^k$, hasta que encontremos un nuevo punto $z^k \notin \overline{\mathcal{F}}_k$ que satisfaga la condición de descenso $h(z^k) < (1 - \alpha)h(x^k)$ y la condición $\|z^k - x^k\|_\infty \leq \beta h(x^k)$ donde $0 < \alpha < 1$, $\beta > 0$ son parámetros predeterminados del algoritmo principal.

Este algoritmo iterativo genera una sucesión $\{y^j\}_{j \in \mathbb{N}}$ tal que para $j = 0, 1, 2, \dots$, $y^j \in \Omega_k$, donde $\Omega_k = \{y \in \mathbb{R}^n : \|y - x^k\|_\infty \leq \beta h(x^k)\}$. Dado un iterado y^j , BCDF-QNB calcula una dirección considerando dos posibilidades. En un primer intento, como solución del siguiente sistema lineal con restricciones de cotas

$$B_j d + c(y^j) = 0 \quad \text{y} \quad y^j + d \in \Omega_k, \quad (4.13)$$

si esto es posible. Caso contrario la dirección es calculada como una solución aproximada del problema

$$\min_{y^j + d \in \Omega_k} \|B_j d + c(y^j)\| \quad (4.14)$$

donde la actualización de Broyden de la matriz B_j es la siguiente:

$$B_j = B_{j-1} + \frac{(w_j - B_{j-1}s^j)(s^j)^T}{\|s^j\|^2} \quad (4.15)$$

siendo $w_j = c(y^j) - c(y^{j-1})$, $s^j = y^j - y^{j-1}$.

Una vez que la dirección d_j es calculada, se realiza una búsqueda lineal no monótona sin derivadas para encontrar una longitud de paso $\lambda_j \leq 1$ tal que

$$h(y^j + \lambda_j d_j)^2 \leq \max_{0 \leq i \leq M-1} h(y^{j-i})^2 + \eta_j - \gamma \lambda_j^2 \|d_j\|^2 \quad (4.16)$$

donde M es un entero no negativo, $0 < \gamma < 1$ y $\sum_{j=0}^{\infty} \eta_j = \eta < \infty$, $\eta_j > 0$. Este procedimiento es una combinación de las estrategias para optimización sin restricciones introducidas por Grippo, Lampariello y Lucidi [21] con el esquema sin derivadas presentado por Li y Fukushima en [29].

Bajo hipótesis adecuadas hemos establecido en el capítulo anterior los resultados de convergencia global para el método BCDF-QNB.

Presentamos a continuación el Algoritmo BCDF-QNB modificado que fue adaptado para poder ser usado en la fase de restauración del algoritmo principal.

Algoritmo 4.2.1. *BCDF-QNB*

Dados $x^k \in \Omega_k$, $0 < \alpha < 1$, $\beta > 0$, W_k una aproximación de $J_c(x^k)$, $0 < \gamma < 1$, $0 \leq \theta_0 < \bar{\theta} < 1$, $0 < \tau_{min} < \tau_{max} < 1$, $ind = 0$, $imax > 0$, $imax \in \mathbb{N}$.

$j \leftarrow 0$, $y^0 = x^k$, $B_0 = W_k$.

Paso 1: Si $h(y^j) < (1 - \alpha)h(x^k)$ e $y^j \notin \bar{\mathcal{F}}_k$, definimos $z^k = y^j$. Regresamos con éxito.

Paso 2: Cálculo de la matriz B_j :

Si $j > 0$ e $ind < imax$ calcular B_j usando la actualización de Broyden (4.15). Si $ind = imax$ calcular B_j por diferencias finitas como una aproximación de la matriz Jacobiana de c en y^j .

Paso : Cálculo de la dirección d_j

Paso 3.1: Encontrar d tal que

$$B_j d + c(y^j) = 0, \quad y^j + d \in \Omega_k \quad (4.17)$$

Si tal dirección d es encontrada, definir $d_j = d$, $\theta_{j+1} = \theta_j$, $ind = 0$ e ir al paso 4.

Paso 3.2: Encontrar una solución aproximada de $\min_{y^j + d \in \Omega_k} \|B_j d + c(y^j)\|$.

Si d satisface

$$\|B_j d + c(y^j)\| \leq \theta_j \|c(y^j)\| \quad (4.18)$$

definir $d_j = d$, $\theta_{j+1} = \theta_j$, $ind = 0$ e ir al paso 4.

Paso 3.3: Considerar $d_j = 0$, $y^{j+1} = y^j$, $\theta_{j+1} = \frac{\theta_j + \bar{\theta}}{2}$.

Si $ind < imax$, considerar $ind \leftarrow ind + 1$ e ir al paso 5.

Si $ind = imax$, definir $\bar{\theta} = \frac{\bar{\theta} + 1}{2}$. Considerar $ind \leftarrow 0$ e ir al paso 5.

Paso 4: Encontrar λ_j usando el Algoritmo 3.1.1 y definir $y^{j+1} = y^j + \lambda_j d_j$.

Paso 5: $j \leftarrow j + 1$ e ir al paso 1.

La matriz W_0 es una aproximación de $J_c(x^0)$, la cual es obtenida por diferencias finitas. La matriz inicial W_k , $k > 0$, es la actualización de Broyden de la matriz A_{k-1} , donde A_{k-1} es la matriz definida en z^{k-1} en la fase de optimalidad.

Observación 4.2.2. Como $\{y^j\} \subset \Omega_k$, el iterado obtenido z^k satisface la condición $\|z^k - x^k\| \leq \beta h(x^k)$, $\beta > 0$.

Más formalmente, el procedimiento genera iterados que verifican la siguiente condición:

(C1) Condición del paso de restauración: En todas las iteraciones $k \in \mathbb{N}$, el paso de restauración satisface

$$\|z^k - x^k\| = O(h(x^k)). \quad (4.19)$$

Usando la condición (C1) y que ∇f es acotado en X , obtenemos que

$$|f(z^k) - f(x^k)| = O(\|z^k - x^k\|) = O(h(x^k)). \quad (4.20)$$

4.2.2. Fase de optimalidad

Dado el iterado $z^k \in X$, generado en la fase de restauración, la fase de optimalidad debe encontrar $x^{k+1} \notin \overline{\mathcal{F}}_k$ tal que $x^{k+1} \in L(z^k)$ y $f(x^{k+1}) \leq f(z^k)$ utilizando algún método sin derivadas.

Describimos un método de región de confianza que utiliza un modelo lineal y mostramos que el nuevo iterado satisface las hipótesis necesarias para obtener convergencia.

En cada iterado z^k , el algoritmo de región de confianza asociado al punto restaurado z^k utiliza el siguiente modelo lineal

$$m_k(x) = f(z^k) + \nabla_s^T f(z^k)(x - z^k)$$

donde consideramos el gradiente simplex de la función objetivo.

En cada iteración del método se usa un radio $\Delta > 0$ y se resuelve el problema

$$\begin{aligned} &\text{Minimizar} && m_k(x) \\ &\text{sujeto a} && x \in L(z^k) \\ &&& \|x - z^k\| \leq \Delta. \end{aligned}$$

Como el modelo es lineal sabemos que la solución de este problema es $z^k + d(z^k, \Delta)$ tal que

$$d(z^k, \Delta) = \Delta \frac{d_c(z^k)}{\|d_c(z^k)\|} \quad (4.21)$$

cuando $d_c(z^k) \neq 0$, donde la dirección del gradiente proyectado está definida por

$$d_c(z^k) = P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k.$$

Definimos la *reducción estimada* producida por el paso $d(z^k, \Delta)$ como

$$pred(z^k, \Delta) = m_k(z^k) - m_k(z^k + d(z^k, \Delta))$$

y la *reducción real* como

$$ared(z^k, \Delta) = f(z^k) - f(z^k + d(z^k, \Delta)).$$

El paso $d(z^k, \Delta)$ es aceptado sólo si se satisface la siguiente condición de suficiente decrecimiento:

$$ared(z^k, \Delta) > \eta \, pred(z^k, \Delta)$$

para algún $\eta \in (0, 1)$.

Entonces tenemos que

$$\begin{aligned} \text{pred}(z^k, \Delta) &= m_k(z^k) - m_k(z^k + d(z^k, \Delta)) = -\nabla_s^T f(z^k) d(z^k, \Delta) \\ &= -\Delta \nabla_s^T f(z^k) \frac{d_c(z^k)}{\|d_c(z^k)\|} = -\Delta \nabla_s^T f(z^k) \frac{P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k}{\|P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k\|}. \end{aligned}$$

Considerando (4.12), tenemos que

$$\nabla_s^T f(z)(P_{L(z)}(z - \nabla_s f(z)) - z) \leq -\frac{1}{2} \|P_{L(z)}(z - \nabla_s f(z)) - z\|^2,$$

y por lo tanto se verifica que

$$\text{pred}(z^k, \Delta) \geq \frac{\Delta}{2} \|P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k\| = \frac{\Delta}{2} \|d_c(z^k)\|. \quad (4.22)$$

Describimos brevemente el método de región de confianza que utilizamos en la fase de optimalidad.

Algoritmo 4.2.3. *Minimización sobre $L(z^k)$*

Dados $\eta \in (0, 1)$, $\Delta_{\min} > 0$, $z^k \notin \mathcal{F}_k$, $d_c(z^k)$, $\Delta \geq \Delta_{\min}$, $0 < \text{tol} < 1$

Considerar $x^+ = z^k$.

Mientras $(\|d_c(z^k)\| \Delta > \text{tol} \text{ y } f(x^+) \geq f(z^k))$ hacer:

Calcular $d = d(z^k, \Delta)$ como en (4.21).

Calcular $\text{pred}(z^k, \Delta)$ y $\text{ared}(z^k, \Delta) = f(z^k) - f(z^k + d)$.

Si $\text{ared}(z^k, \Delta) > \eta \text{ pred}(z^k, \Delta)$ y $z^k + d \notin \overline{\mathcal{F}}_k$, definir $x^+ = z^k + d$. **FIN**.

Si no considerar $\Delta = \frac{\Delta}{2}$.

FIN (mientras).

Si $f(x^+) < f(z^k)$ o $z^k \neq x^k$, definir $x_T = x^+$, $\Delta_k = \Delta$. **FIN**.

Caso contrario retornar sin éxito.

Este procedimiento termina en un número finito de pasos con $f(x^+) < f(z^k)$ o $x^+ = z^k$. En particular, finaliza en la primera iteración cuando $\|d_c(z^k)\| = 0$. Si termina el proceso con $x^+ = z^k$ y $z^k = x^k$, cuando x^k es factible, no es posible definir $x_T \notin \overline{\mathcal{F}}_k$ y por lo tanto

regresa al Algoritmo DFF sin éxito. En los otros casos retorna con éxito habiendo encontrado $x_T = x^+$.

En los lemas siguientes estudiamos el paso de optimalidad cerca de un punto no cuasi-estacionario $\bar{x} \in X$ (ver (4.8)).

Lema 4.2.4. *Sea $\bar{x} \in X$ un punto de acumulación factible no cuasi-estacionario de una sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF. Entonces existen un entorno $\tilde{V}$ de $\bar{x} \in X$, $\tilde{\Delta} > 0$ y una constante $\tilde{c} > 0$ tales que para cualquier $z^k \in \tilde{V}$ y cualquier $\Delta \in (0, \tilde{\Delta})$*

$$ared(z^k, \Delta) > \eta \text{ pred}(z^k, \Delta) \geq \eta \tilde{c} \Delta.$$

Demostración: Como $\bar{x}$ es un punto de acumulación no cuasi-estacionario, existe un entorno $\tilde{V}$ tal que para $z^k \in \tilde{V}$ se tiene que $\|d_c(z^k)\| \geq \tilde{\varepsilon} > 0$ para todo $k \geq k_0$.

Dado que f es diferenciable y ∇f es Lipschitz continuo, sabemos que

$$\begin{aligned} ared(z^k, \Delta) &= f(z^k) - f(z^k + d(z^k, \Delta)) \geq (-\nabla f(z^k))^T d(z^k, \Delta) - L_1 \Delta^2 \\ &= (-\nabla f(z^k) + \nabla_s f(z^k))^T d(z^k, \Delta) - (\nabla_s f(z^k))^T d(z^k, \Delta) - L_1 \Delta^2. \end{aligned}$$

En particular, si $\|d_c(z^k)\| \geq \tilde{\varepsilon}$, usando (4.22), obtenemos que

$$-\nabla_s^T f(z^k) d(z^k, \Delta) = \text{pred}(z^k, \Delta) \geq \frac{\Delta}{2} \|d_c(z^k)\| \geq \frac{\Delta}{2} \tilde{\varepsilon}.$$

Entonces considerando que

$$-\nabla_s^T f(z^k) d(z^k, \Delta) = \eta(-\nabla_s^T f(z^k) d(z^k, \Delta)) + (1 - \eta)(-\nabla_s^T f(z^k) d(z^k, \Delta)),$$

obtenemos que

$$-\nabla_s^T f(z^k) d(z^k, \Delta) \geq \eta(-\nabla_s^T f(z^k) d(z^k, \Delta)) + (1 - \eta) \frac{\Delta}{2} \tilde{\varepsilon}.$$

Por lo tanto

$$\begin{aligned} ared(z^k, \Delta) &\geq \eta(-\nabla_s^T f(z^k) d(z^k, \Delta)) + \\ &+ (1 - \eta) \frac{\Delta}{2} \tilde{\varepsilon} + (-\nabla^T f(z^k) + \nabla_s^T f(z^k)) d(z^k, \Delta) - L_1 \Delta^2. \end{aligned}$$

Por (H4) sabemos que $\|-\nabla f(z^k) + \nabla_s f(z^k)\| \leq k_{eg} \Delta_f^k$.

Consideremos $k_1 \geq k_0$ suficientemente grande tal que para $k \geq k_1$, $k_{eg} \Delta_f^k < \frac{(1-\eta)}{4} \tilde{\varepsilon}$.

Entonces

$$\begin{aligned} \text{ared}(z^k, \Delta) &> \eta(-\nabla_s^T f(z^k) d(z^k, \Delta)) - \frac{(1-\eta)}{4} \tilde{\varepsilon} \|d(z^k, \Delta)\| + (1-\eta) \frac{\Delta}{2} \tilde{\varepsilon} - L_1 \Delta^2 \\ &\geq \eta(-\nabla_s^T f(x^k) d(z^k, \Delta)) - \frac{(1-\eta)}{4} \tilde{\varepsilon} \Delta + (1-\eta) \frac{\Delta}{2} \tilde{\varepsilon} - L_1 \Delta^2. \end{aligned}$$

Así $\text{ared}(z^k, \Delta) > \eta \text{pred}(z^k, \Delta) + (1-\eta) \frac{\Delta}{4} \tilde{\varepsilon} - L_1 \Delta^2$.

Por lo tanto si $\Delta < \tilde{\Delta} = \frac{(1-\eta)}{4L_1} \tilde{\varepsilon}$, obtenemos que

$$\text{ared}(z^k, \Delta) > \eta \text{pred}(z^k, \Delta)$$

y

$$\text{pred}(z^k, \Delta) \geq \frac{\Delta}{2} \|d_c(z^k)\| \geq \tilde{c} \Delta,$$

donde $\tilde{c} = \frac{\tilde{\varepsilon}}{2}$, como queríamos demostrar. $\square$

Observación 4.2.5. *En el lema anterior hemos visto que si z^k , el iterado hallado en la fase de restauración, está en un entorno de un punto factible no cuasi-estacionario, entonces es posible encontrar un paso $d(z^k, \Delta)$ sobre $L(z^k)$ tal que $f(z^k + d(z^k, \Delta)) < f(z^k)$. Además, si z^k no pertenece a $\overline{\mathcal{F}}_k$, el cual es un conjunto cerrado, entonces debe existir $\Delta \leq \tilde{\Delta}$ para el cual $z^k + d(z^k, \Delta)$ no cae en la región prohibida $\overline{\mathcal{F}}_k$. Cuando $h(x^k) = 0$ por construcción sabemos que $z^k = x^k$ y $z^k \in \overline{\mathcal{F}}_k$. Por el Lema 4.2.4, como $f(z^k + d(z^k, \Delta)) < f(z^k)$ para todo $\Delta \in (0, \tilde{\Delta})$, tenemos que $z^k + d(z^k, \Delta) \notin \{x \in \mathbb{R}^n : f(x) \geq f(z^k), h(x) > 0\}$. Entonces considerando que $z^k \in \mathcal{F}_k$, el cual es un conjunto cerrado, obtenemos un resultado similar al caso cuando z^k no pertenece a $\overline{\mathcal{F}}_k$. Por lo tanto, bajo las hipótesis del Lema 4.2.4, el Algoritmo 4.2.3 encuentra un punto $x^+ \notin \overline{\mathcal{F}}_k$ y luego define $x_T = x^+$.*

Lema 4.2.6. *Suponiendo que la matriz A_k , aproximación de $J_c(z^k)$, es calculada por derivadas simplex utilizando un radio Δ_c^k , entonces si $z^k + d \in L(z^k)$ se tiene que*

$$|h(z^k + d) - h(z^k)| \leq \sqrt{m} L_2 \|d\|^2 + \kappa_{eJ_c} \Delta_c^k \|d\|. \quad (4.23)$$

Demostración: Por un lado, por el Teorema del Valor Medio, sabemos que

$$c(z^k + d) - c(z^k) = J_c(\xi)d,$$

donde ξ es un punto que se encuentra en el segmento de recta que une z^k con $z^k + d$.

Por lo tanto

$$c(z^k + d) - c(z^k) - J_c(z^k)d = J_c(\xi)d - J_c(z^k)d = (J_c(\xi) - J_c(z^k))d.$$

Considerando (H3), obtenemos que

$$\begin{aligned} \|c(z^k + d) - c(z^k) - J_c(z^k)d\| &= \|(J_c(\xi) - J_c(z^k))d\| \\ &\leq \|J_c(\xi) - J_c(z^k)\| \|d\| \leq \sqrt{m} L_2 \|\xi - z^k\| \|d\|, \end{aligned}$$

donde la raíz cuadrada de m proviene de la equivalencia entre las normas 2 e infinito.

Como ξ está entre z^k y $z^k + d$ finalmente obtenemos que

$$\|c(z^k + d) - c(z^k) - J_c(z^k)d\| \leq \sqrt{m} L_2 \|d\|^2. \quad (4.24)$$

Por otro lado

$$\begin{aligned} |h(z^k + d) - h(z^k)| &= \left| \|c(z^k + d)\| - \|c(z^k)\| \right| \leq \|c(z^k + d) - c(z^k)\| \\ &\leq \|c(z^k + d) - c(z^k) - J_c(z^k)d\| + \|J_c(z^k)d\|. \end{aligned}$$

Luego utilizando (4.24) y que $A_k d = 0$, ya que $z^k + d \in L(z^k)$, obtenemos que

$$|h(z^k + d) - h(z^k)| \leq \sqrt{m} L_2 \|d\|^2 + \|J_c(z^k)d - A_k d\|.$$

Finalmente utilizando (H4), se tiene que

$$|h(z^k + d) - h(z^k)| \leq \sqrt{m} L_2 \|d\|^2 + \kappa_{eJ_c} \Delta_c^k \|d\|$$

como queríamos demostrar. □

Puede observarse que la cota en (4.23) es $O(\|d\|)$. Esto se debe a que no estamos trabajando con las derivadas verdaderas. Una cota similar aparece en [19], sección 4.3, cuando los autores proponen un paso tangencial simplificado.

Lema 4.2.7. *Sea $\bar{x} \in X$ un punto de acumulación factible no cuasi-estacionario de una sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF. Asumiendo que (C1) y la hipótesis del Lemas 4.2.6 se verifican, entonces existe un entorno V de $\bar{x}$ tal que si $x^k \in V$ entonces se verifica*

$$f(z^k) - f(x^{k+1}) = \Omega(\sqrt{H_k}), \quad (4.25)$$

donde $x^{k+1} = x_T$ fue calculado por el Algoritmo 4.2.3.

Demostración: Sea $\{x^k\}_{k \in \mathcal{K}}$ una subsucesión tal que $\lim_{k \in \mathcal{K}} x^k = \bar{x}$.

Por (C1), $\|x^k - z^k\| = O(h(x^k))$. Y como $h(x^k)$ tiende a cero, ya que $\bar{x}$ es factible, se sigue que $\lim_{k \in \mathcal{K}} z^k = \bar{x}$.

Sean $\tilde{V} \subset X$ y $\tilde{\Delta} > 0$ el entorno de $\bar{x}$ y el radio dados en el Lema 4.2.4. Entonces para $k \in \mathcal{K}$ suficientemente grande, $z^k \in \tilde{V}$ y para cualquier $\Delta \in (0, \tilde{\Delta})$, $ared(z^k, \Delta) > \eta \text{pred}(z^k, \Delta) \geq \eta \tilde{c} \Delta$.

El Algoritmo 4.2.3 comienza con un radio $\Delta \geq \Delta_{min}$ y calcula $d(z^k, \Delta_j)$, $\Delta_j = 2^{-j} \Delta$ para $j = 0, 1, \dots$, hasta que $z^k + d(z^k, \Delta_j) \notin \bar{\mathcal{F}}_k$ y $ared(z^k, \Delta_j) > \eta \text{pred}(z^k, \Delta_j)$. Entonces define $\Delta_k = \Delta_j$.

Definamos $\hat{\Delta}$ como el primer Δ_j tal que

$$ared(z^k, \Delta_j) > \eta \text{pred}(z^k, \Delta_j), \text{ y} \quad (4.26)$$

$$z^k + d(z^k, \Delta_j) \notin \bar{\mathcal{F}}_k \quad \text{o} \quad f(z^k + d(z^k, \Delta_j)) \geq \tilde{f}, \quad (4.27)$$

donde $(\tilde{f}, \tilde{h}) = (f(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k))$ es la entrada temporaria del filtro.

Denotamos $\hat{d} = d(z^k, \hat{\Delta})$ y $\hat{x} = z^k + \hat{d}$. Notemos que $\hat{\Delta} \geq \Delta_k$, y $\hat{\Delta} > \Delta_k$ ocurre sólo cuando $f(\hat{x}) \geq \tilde{f}$.

Observar que por el Lema 4.2.6, para un Δ fijo, se tiene la siguiente desigualdad

$$|h(z^k + d(z^k, \Delta)) - h(z^k)| \leq \kappa_{eJ_c} \Delta_c^k \|d(z^k, \Delta)\| + \sqrt{m} L_2 \|d(z^k, \Delta)\|^2.$$

Por la Observación 4.1.5 sabemos que si x^k está en un entorno de un punto factible se verifica que $h(x^k) \leq H_k$. Por lo tanto, considerando que $\|d(z^k, \Delta)\| \leq \Delta$ y que $\Delta_c^k \leq \beta \min\{\max\{h(x^k), H_k\}, \delta_k\}$ tenemos que

$$|h(z^k + d(z^k, \Delta)) - h(z^k)| \leq \kappa_{eJ_c} \beta H_k \Delta + \sqrt{m} L_2 \Delta^2. \quad (4.28)$$

Consideremos $\bar{\Delta}$ tal que $\bar{\Delta} \leq \frac{\alpha}{4\beta\kappa_{eJ_c}}$ y $\bar{\Delta} < \frac{\tilde{\Delta}}{2}$.

(i) Supongamos que $\hat{\Delta} \geq \bar{\Delta}$. Entonces, por (4.22), tenemos que

$$\text{pred}(z^k, \hat{\Delta}) \geq \frac{\hat{\Delta}}{2} \|d_c(z^k)\| \geq \frac{\tilde{\varepsilon}}{2} \hat{\Delta}.$$

Considerando $\tilde{c} = \frac{\tilde{\varepsilon}}{2}$ como en la demostración del Lema 4.2.4 tenemos que

$$\text{pred}(z^k, \hat{\Delta}) \geq \tilde{c} \hat{\Delta} \geq \tilde{c} \bar{\Delta}.$$

Por definición de $\hat{\Delta}$, se verifica (4.26) y entonces

$$f(z^k) - f(\hat{x}) > \eta \text{pred}(z^k, \hat{\Delta}) \geq \eta \tilde{c} \bar{\Delta} = \Omega(1).$$

Por lo tanto, como $H_k \leq 1$, se obtiene que

$$f(z^k) - f(\hat{x}) = \Omega(\sqrt{H_k}).$$

(ii) Supongamos que $\hat{\Delta} < \bar{\Delta}$. Entonces $2\hat{\Delta} < 2\bar{\Delta} < \tilde{\Delta}$ y $2\hat{\Delta}$ no verifica (4.27). Por el Lema 4.2.4,

$$\text{ared}(z^k, d(z^k, 2\hat{\Delta})) > \eta \text{pred}(z^k, d(z^k, 2\hat{\Delta}))$$

y por (4.27) obtenemos que $z^k + d(z^k, 2\hat{\Delta}) \in \overline{\mathcal{F}}_k$ y $f(z^k + d(z^k, 2\hat{\Delta})) < \tilde{f}$. Consecuentemente por la definición de H_k se tiene que $h(z^k + d(z^k, 2\hat{\Delta})) \geq H_k$.

Por construcción $h(z^k) < (1 - \alpha)h(x^k) \leq (1 - \alpha)H_k$. Así

$$h(z^k + d(z^k, 2\hat{\Delta})) - h(z^k) \geq \alpha H_k.$$

Entonces, usando (4.28),

$$\alpha H_k \leq h(z^k + d(z^k, 2\hat{\Delta})) - h(z^k) \leq \kappa_{eJ_c} \beta H_k 2\hat{\Delta} + 4\sqrt{m} L_2 \hat{\Delta}^2,$$

obtenemos que

$$H_k \leq \frac{2\beta}{\alpha} \kappa_{eJ_c} H_k \hat{\Delta} + O(\hat{\Delta}^2) \leq \frac{1}{2} H_k + O(\hat{\Delta}^2).$$

Por lo tanto

$$\frac{1}{2} H_k = O(\hat{\Delta}^2) \quad \text{o} \quad \hat{\Delta} = \Omega(\sqrt{H_k}).$$

Usando el Lema 4.2.4 con $\hat{\Delta} < \bar{\Delta} < \tilde{\Delta}$, obtenemos que

$$f(z^k) - f(\hat{x}) = \text{ared}(z^k, \hat{\Delta}) \geq \eta \tilde{c} \hat{\Delta} = \eta \tilde{c} \Omega(\sqrt{H_k}). \quad (4.29)$$

Así, para ambos casos, tenemos que $f(z^k) - f(\hat{x}) = \Omega(\sqrt{H_k})$. Entonces el paso $\hat{d}$ satisface la condición del lema.

Para finalizar la demostración, debemos probar que para $k \in \mathcal{K}$ suficientemente grande, $f(\hat{x}) < \tilde{f}$ lo que implica que $\hat{x} \notin \overline{\mathcal{F}}_k$ y así $\hat{x} = x^{k+1}$. Por (4.29) existe una constante positiva M tal que

$$f(z^k) - f(\hat{x}) \geq M \sqrt{H_k}$$

y

$$f(\hat{x}) \leq f(z^k) - M \sqrt{H_k}.$$

De (4.20) existe una constante positiva N tal que

$$f(z^k) \leq f(x^k) + N h(x^k).$$

Combinando las dos últimas desigualdades obtenemos que

$$\begin{aligned} f(\widehat{x}) &\leq f(x^k) + Nh(x^k) - M\sqrt{H_k} \leq f(x^k) + Nh(x^k) - M\sqrt{h(x^k)} \\ &= f(x^k) - \sqrt{h(x^k)}(M - N\sqrt{h(x^k)}) \end{aligned}$$

y para $k \in \mathcal{K}$ suficientemente grande tal que $M - N\sqrt{h(x^k)} > \alpha\sqrt{h(x^k)}$, que equivale a tener $\sqrt{h(x^k)} < \frac{M}{N+\alpha}$, tenemos que $f(\widehat{x}) < f(x^k) - \alpha h(x^k) = \tilde{f}$, completando la demostración. $\square$

El resultado del lema anterior establece que el procedimiento propuesto genera iterados que verifican la siguiente condición:

(C2) Condición del paso de optimalidad: Dado un punto factible no cuasi-estacionario $\bar{x} \in X$, existe un entorno V de $\bar{x}$ tal que para cualquier iterado $x^k \in V$ se verifica que

$$f(z^k) - f(x^{k+1}) = \Omega(\sqrt{H_k}). \quad (4.30)$$

4.3. Resultados de convergencia

En esta sección, basados en las condiciones (C1), (C2) y considerando las hipótesis generales, mostramos los resultados de convergencia del Algoritmo DFF a puntos cuasi-estacionarios.

Como fue hecho en [19], puede probarse que (C1) y (C2) implican la siguiente condición:

(C3) Dado un punto factible no cuasi-estacionario $\bar{x} \in X$, existe un entorno V de $\bar{x}$ tal que para cualquier iterado $x^k \in V$, se verifica que

$$f(x^k) - f(x^{k+1}) = \Omega(\sqrt{H_k}) \quad (4.31)$$

donde H_k es la holgura del filtro en x^k definida en (4.9).

La diferencia entre las condiciones (C2) y (C3) y sus análogas en [19] es que aquí están definidas en un entorno de un punto no cuasi-estacionario mientras que en [19] se definen en entornos de puntos no estacionarios.

Lema 4.3.1. *(C1) y (C2) implican (C3).*

Demostración: Sea $\bar{x}$ un punto factible no cuasi-estacionario y sea V_1 el entorno definido en (C2). Como $\|z^k - x^k\| = O(h(x^k))$ y $\bar{x}$ es un punto factible existe un entorno $V_2 \subset V_1$ of $\bar{x}$ tal que para $x^k \in V_2$, $z^k \in V_1$. Consideremos un iterado $x^k \in V_2$. Por (4.20) existe una

constante positiva N tal que $|f(z^k) - f(x^k)| \leq Nh(x^k)$ y $f(x^k) - f(z^k) \geq -Nh(x^k)$. Por (C2) existe una constante positiva M tal que $f(z^k) - f(x^{k+1}) \geq M\sqrt{H_k}$. Entonces considerando que $h(x^k) \leq H_k$ obtenemos que

$$\begin{aligned} f(x^k) - f(x^{k+1}) &= f(x^k) - f(z^k) + f(z^k) - f(x^{k+1}) \geq M\sqrt{H_k} - Nh(x^k) \\ &= M\sqrt{H_k} - N\sqrt{h(x^k)}\sqrt{h(x^k)} \geq M\sqrt{H_k} - N\sqrt{H_k}\sqrt{h(x^k)}. \end{aligned}$$

Así

$$f(x^k) - f(x^{k+1}) \geq (M - N\sqrt{h(x^k)})\sqrt{H_k}.$$

Por continuidad de h en el punto factible $\bar{x}$, existe un entorno $V \subset V_2$ tal que para cualquier $x \in V$, $\sqrt{h(x)} \leq 0.5\frac{M}{N}$. Por lo tanto para cualquier iterado $x^k \in V$, $f(x^k) - f(x^{k+1}) \geq 0.5M\sqrt{H_k}$, lo que completa la demostración. $\square$

Los siguientes lemas son adaptaciones de los Lemas 2.5 y 2.6 de [19] para el caso de puntos cuasi-estacionarios y métodos sin derivadas. Estos resultados son obtenidos considerando la validez de la condición (C3).

Lema 4.3.2. *Sea $\bar{x} \in X$ un punto límite no cuasi-estacionario de una sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF. Entonces existen $\bar{k} \in \mathbb{N}$ y un entorno V de $\bar{x}$ tales que siempre que $k > \bar{k}$ y $x^k \in V$, la iteración k es una f -iteración, es decir, $f(x^{k+1}) < f(x^k)$.*

Demostración: Si $\bar{x} \in X$ es un punto factible, por (C2) y (R3) existe un entorno V de $\bar{x}$ tal que para todo $x^k \in V$,

$$f(x^k) - f(x^{k+1}) = \Omega(\sqrt{H_k}) > 0$$

y entonces k es una f -iteración.

Supongamos ahora que $\bar{x} \in X$ es un punto infactible, entonces $h(\bar{x}) > 0$. Asumamos, por contradicción, que existe un conjunto infinito $\mathcal{K} \subset \mathbb{N}$ tal que $\lim_{k \in \mathcal{K}} x^k = \bar{x} \in X$ y todas las iteraciones en $\mathcal{K}$ son h -iteraciones. Por continuidad, tenemos que

$$\lim_{k \in \mathcal{K}} h(x^k) = h(\bar{x}) \quad \text{y} \quad \lim_{k \in \mathcal{K}} f(x^k) = f(\bar{x}).$$

Entonces existe $k_1 \in \mathcal{K}$ tal que para todo $k \in \mathcal{K}, k \geq k_1$

$$|h(x^k) - h(\bar{x})| < \frac{\alpha}{2}h(x^{k_1}) \quad \text{y} \quad |f(x^k) - f(\bar{x})| < \frac{\alpha}{2}h(x^{k_1}). \quad (4.32)$$

Para cualquier $k_2 \in \mathcal{K}$ tal que $k_2 > k_1$,

$$|h(x^{k_2}) - h(\bar{x})| < \frac{\alpha}{2}h(x^{k_1}) \quad \text{y} \quad |f(x^{k_2}) - f(\bar{x})| < \frac{\alpha}{2}h(x^{k_1}). \quad (4.33)$$

Entonces, usando (4.32)-(4.33), tenemos que

$$|h(x^{k_2}) - h(x^{k_1})| < \alpha h(x^{k_1}) \quad \text{y} \quad |f(x^{k_2}) - f(x^{k_1})| < \alpha h(x^{k_1}).$$

Por lo tanto $x^{k_2} \in \mathcal{F}_{k_1+1}$ contradiciendo (R1), lo que completa la demostración. $\square$

Lema 4.3.3. *Si la sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF no tiene puntos de acumulación cuasi-estacionarios, entonces para k suficientemente grande todas las iteraciones son f -iteraciones.*

Demostración: Asumimos por contradicción que existe un número infinito de h -iteraciones. Entonces existe un conjunto infinito $\mathcal{K}_1 \subset \mathbb{N}$ tal que para todo $k \in \mathcal{K}_1$, la iteración k es una h -iteración. Por (H1), $\{x^k\}_{k \in \mathcal{K}_1}$ es acotada y existen $\mathcal{K}_2 \subset \mathcal{K}_1$ y $\bar{x}$ tales que $\lim_{k \in \mathcal{K}_2} x^k = \bar{x}$. Por hipótesis $\bar{x}$ es un punto no cuasi-estacionario y esto contradice el resultado del lema previo. $\square$

Finalmente presentamos el resultado principal de convergencia global del Algoritmo DFF.

Teorema 4.3.4. *La sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF tiene al menos un punto de acumulación cuasi-estacionario.*

Demostración: Asumamos por contradicción que $\{x^k\}_{k \in \mathbb{N}}$ no tiene puntos de acumulación cuasi-estacionarios. Entonces por el Lema 4.3.3 para k suficientemente grande, $k \geq k_1$, todas las iteraciones son f -iteraciones, $f(x^k)$ decrece y

$$f(x^k) - f(x^{k+1}) \rightarrow 0. \quad (4.34)$$

Para $k \geq k_1$, $F_k = F_{k_1}$ y usando (R3) tenemos que $H_k \geq H_{k_1} > 0$.

Veamos que la sucesión $\{x^k\}$ no puede tener puntos de acumulación factibles. Por (C2), si existe $\mathcal{K}_1$ y un punto factible $\bar{x} \in X$ tal que $\lim_{k \in \mathcal{K}_1} x^k = \bar{x}$, entonces para $k > k_1$ suficientemente grande tenemos que

$$f(x^k) - f(x^{k+1}) = \Omega(\sqrt{H_{k_1}}) > 0,$$

contradiendo (4.34). Por lo tanto $\{x^k\}$ no tiene puntos de acumulación factibles.

Ahora, supongamos que en algún conjunto infinito $\mathcal{K}_2$,

$$h(x^{k+1}) \geq (1 - \alpha)h(x^k).$$

Usando (R2), para $k \in \mathcal{K}_2$,

$$f(x^{k+1}) < f(x^k) - \alpha h(x^k).$$

Por (4.34) tenemos que $\lim_{k \in \mathcal{K}_2} h(x^k) = 0$ lo que contradice que $\{x^k\}$ no tiene puntos de acumulación factibles. Por lo tanto se verifica que

$$h(x^{k+1}) < (1 - \alpha)h(x^k)$$

y $\lim_{k \in \mathcal{K}_2} h(x^k) = 0$ y esto contradice que $\{x^k\}$ no tiene puntos de acumulación factibles, completando la demostración. $\square$

4.3.1. Convergencia a puntos Karush-Kuhn-Tucker

Sabemos, de una sección anterior, que la sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el Algoritmo DFF tiene un punto de acumulación cuasi-estacionario $\bar{x}$. Entonces existe $\mathcal{K} \subset \mathbb{N}$ tal que $\lim_{k \in \mathcal{K}} x^k = \bar{x}$. Además, por (C1), tenemos que $\lim_{k \in \mathcal{K}} z^k = \bar{x}$ y por lo tanto

$$\lim_{k \in \mathcal{K}} \|P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k\| = 0. \quad (4.35)$$

En esta sección probamos que usando la condición de calidad de independencia lineal (LICQ) $\bar{x}$ es un punto KKT de (4.1).

El siguiente lema muestra que (4.35) se sigue verificando cuando reemplazamos $\nabla_s f(z^k)$ por $\nabla f(z^k)$ pero manteniendo la proyección sobre $L(z^k)$.

Lema 4.3.5. *Sea $\{x^k\}_{k \in \mathbb{N}}$ una sucesión generada por el Algoritmo DFF. Entonces existe $\mathcal{K} \subset \mathbb{N}$ tal que*

$$\lim_{k \in \mathcal{K}} \|P_{L(z^k)}(z^k - \nabla f(z^k)) - z^k\| = 0. \quad (4.36)$$

Demostración: De la condición (H4),

$$\|\nabla f(z^k) - \nabla_s f(z^k)\| \leq k_{eg} \Delta_f^k \leq k_{eg} \delta_k, \quad (4.37)$$

donde la sucesión $\{\delta_k\}$ tiende a cero. Entonces considerando

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| = \|z^k - P_{L(z^k)}(z^k - \nabla f(z^k) - \nabla_s f(z^k) + \nabla_s f(z^k))\| \quad (4.38)$$

y usando (4.11) tenemos que

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| \leq \|z^k - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| + \|\nabla f(z^k) - \nabla_s f(z^k)\|.$$

Por lo tanto, usando (4.35) y (4.37) y tomando límite cuando k tiene a infinito, $k \in \mathcal{K}$, obtenemos (4.36) como queríamos probar. $\square$

La principal diferencia entre la condición (4.7) y la condición (4.35) es que en la última sólo se están tomando aproximaciones de las derivadas verdaderas.

En [19, Lemma 1.1] los autores prueban que la condición (4.7), junto con la condición de calidad de Mangasarian-Fromovitz [5], es equivalente a la condición KKT.

En este trabajo, dado que el problema original (4.1) tiene sólo restricciones de igualdad, pudimos demostrar que si un punto de acumulación cuasi-estacionario $\bar{x}$ de la sucesión generada por el Algoritmo DFF verifica la condición de calidad de independencia lineal entonces este punto es un punto KKT del problema (4.1). Es decir, $\bar{x}$ es factible y existen multiplicadores $\mu \in \mathbb{R}^m$ tales que

$$\nabla f(\bar{x}) + \sum_{i=1}^m \mu_i \nabla c_i(\bar{x}) = 0.$$

Teorema 4.3.6. *Sean $\{x^k\}_{k \in \mathbb{N}}$ una sucesión generada por el Algoritmo DFF y $\bar{x}$ un punto de acumulación cuasi-estacionario de $\{x^k\}_{k \in \mathbb{N}}$ que satisface la condición de calidad de independencia lineal. Entonces $\bar{x}$ es un punto KKT de (4.1).*

Demostración: Como $\bar{x}$ es un punto de acumulación cuasi-estacionario de $\{x^k\}_{k \in \mathbb{N}}$, entonces existe $\mathcal{K} \subset \mathbb{N}$ tal que $\lim_{k \in \mathcal{K}} x^k = \bar{x}$.

Sea $\tilde{z}^k = P_{L(z^k)}(z^k - \nabla f(z^k))$. Entonces, por definición, $\tilde{z}^k$ es solución del problema

$$\begin{aligned} \min \quad & \|z - (z^k - \nabla f(z^k))\|^2 \\ \text{sujeto a} \quad & A_k(z - z^k) = 0. \end{aligned} \tag{4.39}$$

Como $\bar{x}$ es un punto de acumulación cuasi-estacionario, usando el lema previo tenemos que

$$\lim_{k \in \mathcal{K}} (\tilde{z}^k - z^k) = 0.$$

Como el conjunto factible de (4.39) está definido por restricciones lineales sabemos que existe $\bar{\mu}^k \in \mathbb{R}^m$ tal que

$$\begin{aligned} -(\tilde{z}^k - (z^k - \nabla f(z^k))) &= A_k^T \bar{\mu}^k \\ A_k(\tilde{z}^k - z^k) &= 0. \end{aligned}$$

Entonces

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i=1}^m \bar{\mu}_i^k a_i^k$$

donde a_i^k denota la i -ésima columna de A_k^T .

Por un teorema de Carathéodory (por ejemplo ver [5], página 689), para cada $k \in \mathcal{K}$ existen $I_k \subset \{1, \dots, m\}$ y $\{\mu^k\}_{k \in \mathcal{K}} \subset \mathbb{R}^m$ tales que

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i \in I_k} \mu_i^k a_i^k$$

donde el conjunto $\{a_i^k\}_{i \in I_k}$ es linealmente independiente.

Como el número de posibles conjuntos I_k es finito, existe $\mathcal{K}_1 \subset \mathcal{K}$ tal que para todo $k \in \mathcal{K}_1$,

$$I_k = I \subset \{1, \dots, m\}$$

y

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i \in I} \mu_i^k a_i^k, \quad (4.40)$$

donde el conjunto $\{a_i^k\}_{i \in I}$ es linealmente independiente.

Si la sucesión $\{\mu^k\}_{k \in \mathcal{K}}$ no es acotada, consideremos $M_k = \|\mu^k\|_\infty$. Entonces $\lim_{k \in \mathcal{K}_1} M_k = \infty$ y podemos tomar una subsucesión apropiada tal que $\lim_{k \in \mathcal{K}_2} \frac{\mu^k}{M_k} = \mu \neq 0$, donde $\mathcal{K}_2 \subset \mathcal{K}_1$. Entonces

$$\frac{z^k - \tilde{z}^k}{M_k} = \frac{\nabla f(z^k)}{M_k} + \sum_{i \in I} \frac{\mu_i^k}{M_k} a_i^k. \quad (4.41)$$

Así usando (H4) y tomando límite en (4.41) cuando k tiende a infinito, $k \in \mathcal{K}_2$, obtenemos que

$$\sum_{i \in I} \mu_i \nabla c_i(\bar{x}) = 0$$

lo cual contradice la condición de calidad de independencia lineal.

Por lo tanto la sucesión $\{\mu^k\}_{k \in \mathcal{K}}$ debe ser acotada. Entonces existen $\mathcal{K}_3 \subset \mathcal{K}_1$ y $\mu \in \mathbb{R}^m$ tales que $\lim_{k \in \mathcal{K}_3} \mu^k = \mu$. Entonces usando (H4) y tomando límite en (4.40) cuando k tiende a infinito, $k \in \mathcal{K}_3$, tenemos que

$$\nabla f(\bar{x}) + \sum_{i \in I} \mu_i \nabla c_i(\bar{x}) = 0.$$

Así $\bar{x}$ es un punto KKT de (4.1) como queríamos demostrar. $\square$

4.4. Experimentación numérica

En esta sección presentamos algunos resultados computacionales obtenidos con una implementación en Fortran 77 del Algoritmo DFF. Estos experimentos fueron corridos en una computadora personal con INTEL(R) Core (TM) 2 Duo CPU E8400 de 3.00 GHz y 3.23 GB de RAM.

Como es usual en problemas de optimización sin derivadas estamos interesados en el número de evaluaciones funcionales necesarios para satisfacer los criterios de finalización de los algoritmos.

4.4.1. Detalles sobre la implementación del Algoritmo DFF

Hemos considerado dos versiones del Algoritmo DFF: DFF1 y DFF2. La única diferencia entre ellos es la forma en la cual calculamos la matriz A_k . En la primer versión, DFF1, A_k fue calculada por derivadas simplex como fue descrito en una sección previa. En la segunda versión, DFF2, una vez que z^k fue hallado en la fase de restauración, A_k fue calculada como una nueva matriz de Broyden actualizando la última matriz obtenida en el proceso.

Los parámetros utilizados en los experimentos numéricos para evaluar el desempeño de DFF1 y DFF2 fueron $\alpha = 0.1$, $\beta = 100$, $\epsilon_f = 10^{-6}$ y $\epsilon_I = 10^{-6}$. Y hemos considerado que x^k es un punto cuasi-estacionario cuando $h(x^k) \leq \epsilon_f$, $\max\{\Delta_f^k, \Delta_c^k\} \leq \epsilon_I$ y $\|d_c(z^k)\| \leq \epsilon_f$.

Los parámetros utilizados en el Algoritmo BCDF-QNB fueron los mismos que se consideraron en el capítulo anterior.

Los parámetros utilizados en el Algoritmo 4.2.3 fueron los siguientes: $\eta = 0.1$, $tol = 10^{-16}$ y $\Delta_{min} = 0.5$.

En la implementación de la fase de optimalidad se usaron las subrutinas DLSVRR, de la Librería IMSL Fortran, la cual calcula la descomposición en valores singulares (USV) de la matriz A_k para obtener la proyección de $-\nabla_s f(z^k)$ sobre $\mathcal{N}(A_k)$.

En el Paso 3 del Algoritmo DFF se requiere el cálculo de los gradientes simplex de c_j , para $j = 1, \dots, m$, para lo cual se necesita seleccionar un conjunto de puntos de interpolación. En la primera iteración construimos el conjunto $Y_c^0 = \{z^0, y_c^1, \dots, y_c^n\}$ para obtener los modelos lineales $m_{c_j}(x) = c_j(z^0) + \nabla_s c_j(z^0)^T(x - z^0)$, $j = 1, \dots, m$, generando de este modo la matriz A_0 , como una aproximación de $J_c(z^0)$. Consideramos $y_c^i - z^0 = \rho_0 e^i$ y los correspondientes valores $c_j(y_c^i)$, para $i = 1, \dots, n$ y $j = 1, \dots, m$, $\rho_0 < \beta \max\{\delta_0, h(x^0)\}$. También necesitamos calcular el modelo $m_f(x) = f(z^k) + \nabla_s f(z^k)^T(x - z^k)$. En la primera iteración utilizamos

los vectores de la matriz V de la descomposición en valores singulares de la matriz A_0 para obtener el modelo $m_f(x) = f(z^0) + \nabla_s f(z^0)^T(x - z^0)$, considerando el conjunto de interpolación $Y_f^0 = \{z^0, y_f^1, \dots, y_f^n\}$, donde $y_f^i = z^0 + \rho_0 v_i$ y $f(y_f^i)$, para $i = 1, \dots, n$.

En las siguientes iteraciones los conjuntos Y_c^k e Y_f^k son actualizados incorporando el punto z^k como el centro de los conjuntos de interpolación y eliminando el punto y_t que está mas alejado del centro manteniendo la independencia de las direcciones, es decir, de modo que los conjuntos sigan siendo equilibrados. En esta primera implementación, en algunas iteraciones, los conjuntos de interpolación son contruídos nuevamente mientras que en otras iteraciones son actualizados a partir de los conjuntos de la iteración anterior. La construcción es llevada a cabo en la primera iteración y cuando no es posible preservar la independencia lineal de las direcciones.

4.4.2. Problemas test

Hemos usado un conjunto de problemas de programación no lineal que se encuentran en el libro de Hock y Schittkowski [22]. También hemos considerado el problema utilizado en [19]. El conjunto de problemas seleccionados de [22] está formado por aquellos que tienen restricciones de igualdad. También consideramos algunos problemas que tienen restricciones de desigualdad. Como en estos problemas las restricciones de desigualdad son activas en el punto solución, estas han sido consideradas como restricciones de igualdad. Recordemos que una restricción de desigualdad, $c_i(x) \leq 0$, se dice activa en un punto y si la restricción se cumple por igualdad cuando es evaluada en y , es decir, si se verifica que $c_i(y) = 0$.

En la Tabla 4.1 mostramos los datos de los problemas. El número de variables se encuentra entre 2 y 10 y el número de restricciones de igualdad entre 1 y 4. Los puntos iniciales son los mismos que se dan en las citadas referencias.

A modo ilustrativo mostramos el comportamiento del Algoritmo DFF cuando consideramos el problema en [19] que enunciamos a continuación:

$$\begin{aligned} &\text{Minimizar } x_2 \\ &\text{s. a } x_2 + (2 + x_1) \cos(x_1) = 0 \end{aligned}$$

En la Tabla 4.2 se muestran los valores $(f(x^k), h(x^k))$ obtenidos después de realizar 6 iteraciones, comenzando desde el iterado $(0.5, 0)$. En la Figura 4.7 se muestran los iterados x^k obtenidos.

Aún cuando el problema tiene dimensiones pequeñas podemos observar un desempeño satisfactorio ya que la sucesión tiende a la solución óptima $(0.39, -2.2)$.

Problema	n	m	Problema	n	m
HS 6 of [22]	2	1	HS 47 of [22]	5	3
HS 7 of [22]	2	1	HS 48 of [22]	5	2
HS 8 of [22]	2	2	HS 52 of [22]	5	3
HS 9 of [22]	2	1	HS 53 of [22]	5	3
HS 14 of [22]	2	2	HS 56 of [22]	7	4
HS 22 of [22]	2	2	HS 60 of [22]	3	1
HS 26 of [22]	3	1	HS 61 of [22]	3	2
HS 27 of [22]	3	1	HS 63 of [22]	3	2
HS 29 of [22]	3	1	HS 77 of [22]	5	2
HS 35 of [22]	3	1	HS 78 of [22]	5	3
HS 39 of [22]	4	2	HS 79 of [22]	5	3
HS 40 of [22]	4	3	HS 80 of [22]	5	3
HS 42 of [22]	4	2	HS 81 of [22]	5	3
HS 43 of [22]	4	3	HS 111 of [22]	10	3
HS 46 of [22]	5	2	Ejemplo de [19]	2	1

Tabla 4.1: Datos de los problemas

iteración	$f(x^k)$	$h(x^k)$
0	0	2.1900
1	-2.0350	0.4100
2	-1.8000	0.2000
3	-2.0400	0.1000
4	-2.1700	0.0300
5	-2.1900	0.0200
6	-2.2000	0.0003

Tabla 4.2: Pares $(f(x^k), h(x^k))$

Figura 4.7: Iterados x^k

4.4.3. Resultados numéricos

En la Tabla 4.3 mostramos los resultados obtenidos tomando en cuenta el número de iteraciones (Iter), el número de evaluaciones funcionales de la función objetivo (ObjEval), el número de evaluaciones funcionales de las restricciones (ConstEval), los valores funcionales finales obtenidos $f(x^{end})$ y los valores finales de la medida la infactibilidad en el punto final $h(x^{end})$.

Prob	Iter		ObjEval		ConstEval		$f(x^{end})$		$h(x^{end})$	
	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2
HS 6	24	49	76	151	103	103	3.050E-05	3.023E-05	8.644E-10	2.383E-11
HS 7	10	10	33	33	46	24	-1.732E00	-1.732E00	2.620E-13	9.645E-11
HS 8	3	7	5	9	18	12	-1.000E00	-1.000E00	2.764E-12	3.157E-07
HS 9	27	49	58	101	36	53	-5.000E-01	-5.000E-01	5.329E-15	3.695E-09
HS 14	3	5	5	7	16	9	1.393E00	1.393E00	2.428E-10	6.547E-09
HS 22	3	4	5	6	15	8	1.000E00	1.000E00	2.085E-09	5.116E-07
HS 26	15	18	74	107	93	72	8.787E-07	5.367E-08	1.858E-12	4.981E-11
HS 27	6	45	81	355	270	267	4.001E-02	4.005E-02	2.254E-11	1.113E-11
HS 29	19	24	80	148	106	104	-2.263E01	-2.262E01	1.040E-11	9.216E-08
HS 35	41	45	210	186	255	97	1.111E-01	1.111E-01	9.159E-09	5.271E-10
HS 39	31	36	127	156	193	202	-1.000E00	-9.999E-01	1.036E-07	1.644E-08
HS 40	21	14	68	87	138	76	-2.500E-01	-2.500E-01	9.352E-11	1.522E-08
HS 42	18	45	123	269	165	181	1.386E01	1.386E01	9.108E-13	3.686E-14
HS 43	24	32	97	156	181	187	-4.400E01	-4.400E01	1.882E-08	1.740E-07
HS 46	31	37	182	214	249	105	5.774E-05	5.265E-05	1.323E-08	4.828E-09
HS 47	30	40	129	182	222	105	1.461E-05	2.582E-05	1.852E-09	1.824E-07
HS 48	57	62	249	317	165	133	7.521E-09	1.150E-09	9.108E-09	7.326E-09
HS 52	41	41	289	286	230	207	5.327E00	5.327E00	1.959E-08	9.141E-09
HS 53	19	19	87	81	87	46	4.093E00	4.093E00	8.408E-09	8.067E-09
HS 56	58	79	364	437	685	207	-3.456E00	-3.346E00	8.545E-07	1.217E-05(*)
HS 60	11	18	67	105	85	70	3.257E-02	3.257E-02	4.679E-11	2.839E-08
HS 61	16	18	67	94	115	82	-1.436E02	-1.436E02	4.715E-09	2.985E-10
HS 63	12	30	43	93	78	67	9.617E02	9.617E02	3.141E-10	1.806E-10
HS 77	25	26	133	270	190	198	2.415E-01	2.415E-01	1.021E-11	4.608E-07
HS 78	5	30	27	167	50	110	-2.919E00	-2.919E00	5.694E-09	1.824E-08
HS 79	8	10	41	51	73	34	7.878E-02	7.878E-02	6.064E-12	1.570E-07
HS 80	11	10	48	44	89	27	5.395E-02	5.396E-02	7.441E-09	1.282E-07
HS 81	11	11	48	48	89	29	5.395E-02	5.395E-02	1.215E-08	1.835E-07
HS 111	66	101	595	1084	805	397	-4.776E01	-4.764E01	3.849E-07	8.760E-07
Ex. of [19]	11	20	46	64	70	45	-2.210E00	-2.211E00	1.278E-09	1.783E-09

(*): la solución final no alcanza un suficiente decrecimiento de la medida de la infactibilidad.

Tabla 4.3: Resultados de los problemas test

Se puede observar que la versión DFF1 ha realizado menos iteraciones que DFF2 en un 70 % de los problemas. Creemos que este comportamiento se debe al hecho que DFF1 usa una mejor aproximación de $J_c(z^k)$ en varias iteraciones, y como consecuencia la matriz inicial actualizada en la fase de restauración es mejor. Cuando consideramos $h(x^{end})$ como una medida del desempeño de los algoritmos podemos observar que con DFF1 se obtienen mejores resultados que con DFF2 en un 70 % de los problemas.

Para la comparación algorítmica usamos *perfiles de desempeño*, descritos en [15], y *perfiles de datos* para optimización sin derivadas, presentados en [41].

El perfil de desempeño de un resolutor s está definido como la fracción de problemas

donde el radio de desempeño es al menos α , es decir, $\rho_s(\alpha) = \frac{1}{|\mathcal{P}|} \text{size}\{p \in \mathcal{P} : r_{p,s} \leq \alpha\}$, donde $r_{p,s} = \frac{t_{p,s}}{\{\min t_{p,s} : s \in \mathcal{S}\}}$, $t_{p,s}$ es el número de evaluaciones requeridas para satisfacer el criterio de finalización, $\mathcal{P}$ es el conjunto de problemas y $|\mathcal{P}|$ denota la cardinalidad de $\mathcal{P}$.

También estamos interesados en el porcentaje de problemas que puede ser resuelto por un resolutor s con un número particular de evaluaciones funcionales. El porcentaje de problemas que puede ser resuelto con α evaluaciones funcionales es calculado por $d_s(\alpha) = \frac{1}{|\mathcal{P}|} \text{size}\{p \in \mathcal{P} : t_{p,s} \leq \alpha\}$.

Como es mencionado en [41], la definición de d_s es independiente del número de variables del problema $p \in \mathcal{P}$. Sin embargo sabemos que el número de evaluaciones funcionales crece cuando el número de variables aumenta. Consideramos entonces el perfil de datos de un resolutor s por $d_s(\alpha) = \frac{1}{|\mathcal{P}|} \text{size}\{p \in \mathcal{P} : \frac{t_{p,s}}{n+1} \leq \alpha\}$, donde n es el número de variables del problema $p \in \mathcal{P}$. El valor de $d_s(\alpha)$ puede ser interpretado como el porcentaje de problemas que puede ser resuelto con el equivalente de α estimaciones de gradientes simplex, considerando que $n+1$ es el número de evaluaciones necesarias para calcular un estimado del gradiente por diferencias finitas [41].

Analizamos separadamente el número de las evaluaciones de la función objetivo (ObjEval) y el número de las evaluaciones de las restricciones (ConstEval).

En las siguientes figuras comparamos DFF1 y DFF2 usando el número de evaluaciones de la función objetivo como medida del desempeño.

En el perfil de desempeño de la Figura 4.8, podemos observar que DFF1 fue el mejor resolutor en más del 80 % de los problemas, mientras que DFF2 lo fue aproximadamente en el 20 % de los problemas. La diferencia del desempeño entre DFF1 y DFF2 es aproximadamente 20 % cuando el radio del desempeño es 2.

Figura 4.8: Perfil de desempeño: evaluaciones de la función objetivo

El perfil de datos de la Figura 4.9 (a) muestra que DFF1 resuelve el mayor porcentaje de problemas de todos los tamaños del número evaluaciones de la función objetivo. Podemos

(a)

(b)

Figura 4.9: Perfiles de datos: evaluaciones de la función objetivo

observar que DFF1 resuelve el 80 % de los problemas con 200 evaluaciones, mientras que DFF2 resuelve aproximadamente el 70 % de los problemas. La mayor diferencia es 30 % y ocurre cuando el número de evaluaciones es aproximadamente 180. Creemos que este comportamiento se debe al hecho que DFF1 utiliza mejores aproximaciones de $J_c(z^k)$ en muchas iteraciones, además de realizar menos iteraciones que DFF2.

La Figura 4.9 (b) muestra que DFF1 resuelve el mayor porcentaje de problemas de todos los tamaños del número de estimaciones de gradientes simplex ($\text{ObjEval}/(n+1)$). Con 60 evaluaciones, DFF1 resuelve el 100 % de los problemas, mientras que DFF2 requiere 100 evaluaciones para resolver la totalidad de los problemas. La mayor diferencia entre DFF1 y DFF2 ocurre cuando el número de evaluaciones funcionales es aproximadamente 30 y en este caso DFF1 resuelve el 80 % de los problemas, mientras que DFF2 resuelve aproximadamente el 50 % de los problemas.

En las siguientes figuras comparamos DFF1 y DFF2 teniendo en cuenta el número de evaluaciones de las restricciones como medida del desempeño.

En el perfil de desempeño de la Figura 4.10 podemos observar que DFF2 fue el mejor método en aproximadamente 80 % de los problemas mientras que DFF1 lo fue en más de 20 % de los problemas.

En la Figura 4.11 (a) el perfil de datos muestra que DFF2 resuelve el mayor porcentaje de los problemas de todos los tamaños del número de evaluaciones de las restricciones. Creemos que este resultado está asociado al hecho que DFF2 no requiere nuevas evaluaciones de las restricciones para calcular la matriz A_k , ya que esta es actualizada a partir de la última matriz usada en la fase de restauración. Con 400 evaluaciones DFF2 resuelve todos los problemas que hemos considerado mientras que DFF1 necesita 800 evaluaciones para resolver el mismo

Figura 4.10: Perfil de desempeño: evaluaciones de las restricciones

conjunto de problemas.

La Figura 4.11 (b) muestra que DFF2 resuelve el mayor porcentaje de problemas de todos los tamaños de gradientes simplex ($\text{ConstEval}/(n+1)$). Con 70 evaluaciones DFF2 resuelve casi el 100 % de los problemas, mientras que DFF1 resuelve aproximadamente el 90 % de los problemas. La mayor diferencia entre DFF1 y DFF2 ocurre cuando el número de evaluaciones de las restricciones es 20 y en este caso DFF2 resuelve el 60 % de los problemas, mientras que DFF1 resuelve aproximadamente el 40 % de los problemas.

(a)

(b)

Figura 4.11: Perfiles de datos: evaluaciones de las restricciones

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo hemos considerado el problema de programación no lineal de minimizar una función objetivo sujeto a restricciones no lineales de igualdad donde las derivadas de las funciones que definen el problema no están disponibles. Hemos desarrollado un algoritmo sin derivadas de restauración inexacta que utiliza la estrategia de filtros para la selección de los siguientes iterados. Esta metodología es llevada a cabo realizando dos fases. En la primer fase se busca mejorar la factibilidad y en la segunda fase se pretende disminuir el valor funcional de la función objetivo.

Así, para realizar la primer fase, se estudió la manera de resolver sistemas indeterminados de ecuaciones no lineales. En el capítulo 2 presentamos dos algoritmos sin derivadas. Uno de ellos está basado en la aproximación del residuo espectral (DF-SAUNE) y el otro es un método cuasi-Newton que utiliza la actualización de Broyden (DF-QNB).

Desde el punto de vista teórico fuimos capaces de obtener resultados de convergencia global para ambos métodos considerando hipótesis usuales.

El método DF-SAUNE puede ser visto como una extensión del método DF-SANE definido en [25] para resolver sistemas indeterminados de ecuaciones no lineales.

Para el método DF-QNB se obtuvieron resultados de convergencia global bajo condiciones del tipo Dennis-Moré. Y hemos demostrado que esta hipótesis puede ser omitida para una búsqueda lineal particular.

Los experimentos numéricos sugieren que los algoritmos se comportan promisoriamente.

En el capítulo 3 desarrollamos el Algoritmo BCDF-QNB, que resuelve sistemas de ecuaciones no lineales con restricciones de cotas en las variables.

Así como lo hicimos en el capítulo dos, pudimos demostrar resultados de convergencia para el Algoritmo BCDF-QNB. Bajo hipótesis usuales sobre la matriz Jacobiana del sistema y una condición del tipo Dennis-Moré establecimos la convergencia global del método y mostramos que, para una búsqueda lineal particular, la misma puede ser omitida.

Los experimentos y comparaciones numéricas sugieren que el algoritmo se comporta satisfactoriamente. Analizando el número de evaluaciones funcionales se obtuvieron mejores

resultados con el Algoritmo BCDF-QNB. Estos resultados son razonables y muestran la importancia de desarrollar algoritmos que tomen en cuenta la estructura del problema. Sin embargo, debemos testear el desempeño del método propuesto en el capítulo 3 con una colección más amplia de problemas.

Finalmente en el capítulo 4 presentamos el algoritmo sin derivadas de filtros de restauración inexacta (DFF) para resolver el problema de programación no lineal con restricciones de igualdad (4.1). Uno de los hechos principales de esta tesis es la introducción de filtros en el contexto de optimización sin derivadas para métodos de restauración inexacta, en lugar del uso de una función de mérito.

Desde el punto de vista teórico, bajo hipótesis adecuadas, fuimos capaces de demostrar convergencia global a puntos cuasi-estacionarios. Además pudimos probar que si un punto de acumulación satisface la condición de calidad de independencia lineal, este resulta ser un punto Karush-Kuhn-Tucker de (4.1).

Desde el punto de vista práctico, implementamos dos versiones del Algoritmo DFF que fueron testeados considerando una colección de problemas de pequeño tamaño. La principal diferencia entre estas dos versiones es la manera en la cual es calculada la aproximación de la verdadera matriz Jacobiana $J_c(z^k)$. Desde el punto de vista experimental debemos tener en cuenta dos aspectos principales:

1. Los experimentos sugieren que es admisible el uso de la actualización de Broyden para calcular las aproximaciones de la matriz Jacobiana y este hecho debe ser un tópico de estudio en el futuro.
2. Los algoritmos implementados se comportan según lo esperado. Sin embargo, como trabajo futuro debemos realizar la experimentación correspondiente con una colección de problemas de mayor tamaño. También debemos comparar el desempeño de los algoritmos propuestos con otros métodos sin derivadas que resuelvan el problema (4.1).

Finalmente, como el algoritmo propuesto pertenece a la clase de métodos de restauración inexacta, diferentes alternativas pueden ser estudiadas con el fin de resolver cada una de las fases. En particular, nos gustaría definir un algoritmo sin derivadas basado en un modelo cuadrático para resolver la fase de optimalidad. En este caso el uso de modelos cuadráticos debe ser consistente con la teoría, especialmente con la condición (C2), para así preservar la convergencia del método.

Apéndice A

Proyecciones ortogonales

En esta sección enunciaremos dos teoremas acerca de proyecciones ortogonales sobre conjuntos cerrados y convexos. En el primero además se define que es una proyección ortogonal y el segundo teorema nos permite demostrar la desigualdad

$$\nabla_s^T f(z^k) d_c(z^k) \leq -\frac{\|d_c(z^k)\|^2}{2},$$

la cual ha sido utilizada en varias oportunidades en el capítulo 4.

Teorema A.0.1. *Sea $Y \subset \mathbb{R}^n$ un conjunto cerrado y convexo y sea $\|\cdot\|$ la norma euclídea.*

- (i) *Para todo $x \in \mathbb{R}^n$ existe un único vector en Y que minimiza $\|z - x\|$ sobre todo $z \in Y$. Este vector será llamado proyección de x sobre Y y será denotado por $P(x)$, es decir,*

$$P(x) = \operatorname{argmin}_{z \in Y} \|z - x\|.$$

- (ii) *Dado $x \in \mathbb{R}^n$, un vector $z \in Y$ es igual a $P(x)$ si y sólo si*

$$(y - z)^T (x - z) \leq 0.$$

- (iii) *La función P es continua y no expansiva, es decir,*

$$\|P(x) - P(y)\| \leq \|x - y\|$$

para todo $x, y \in \mathbb{R}^n$.

Demostración: Ver demostración en el libro de Bertsekas [5].

Teorema A.0.2. Sea P una proyección ortogonal sobre un conjunto convexo y cerrado $Y \subset \mathbb{R}^n$. Si $y \in Y$ entonces se verifica la siguiente desigualdad

$$(P(y - x) - y)^T x \leq -\frac{\|P(y - x) - y\|^2}{2} \quad (\text{A.1})$$

para todo $x \in \mathbb{R}^n$.

Demostración: Por ser P lineal y como $y \in Y$ tenemos que

$$\|(y - x) - P(y - x)\|^2 = \|y - x - y + P(x)\|^2 = \|P(x) - x\|^2.$$

Considerando $P^\perp$ la proyección sobre $Y^\perp$ y usando que $x = P(x) + P^\perp(x)$ tenemos que

$$\|(y - x) - P(y - x)\|^2 = \|P(x) - x\|^2 = \|P^\perp(x)\|^2.$$

Por lo tanto, como $\|P\| = \|P^\perp\| = 1$, obtenemos que

$$\|(y - x) - P(y - x)\|^2 \leq \|x\|^2. \quad (\text{A.2})$$

Por otro lado tenemos que

$$\begin{aligned} \|(y - x) - P(y - x)\|^2 &= \|(y - P(y - x)) - x\|^2 = \|(P(y - x) - y) + x\|^2 \\ &= (P(y - x) - y)^T (P(y - x) - y) + P(y - x)^T P(y - x) + x^T x + 2(P(y - x) - y)^T x \\ &= \|P(y - x) - y\|^2 + \|x\|^2 + 2(P(y - x) - y)^T x. \end{aligned}$$

Luego, utilizando (A.2) tenemos que

$$\|(y - x) - P(y - x)\|^2 = \|P(y - x) - y\|^2 + \|x\|^2 + 2(P(y - x) - y)^T x \leq \|x\|^2.$$

Por lo tanto

$$\|P(y - x) - y\|^2 + 2(P(y - x) - y)^T x \leq 0,$$

de lo que se deduce que

$$(P(y - x) - y)^T x \leq -\frac{\|P(y - x) - y\|^2}{2}$$

como queríamos demostrar. □

Observación A.0.3. Como $z^k \in L(z^k)$ y $d_c(z^k) = P_{L(z^k)}(z^k - \nabla_s^T f(z^k)) - z^k$ si consideramos en (A.1): $y = z^k$, $x = \nabla_s^T f(z^k)$, $Y = L(z^k)$ y $P = P_{L(z^k)}$, obtenemos la desigualdad (4.12),

es decir,

$$\nabla_s^T f(z^k) d_c(z^k) \leq -\frac{\|d_c(z^k)\|^2}{2}.$$

Bibliografía

- [1] J. ABADIE, J. CARPENTIER, *Generalization of the Wolfe Reduced-Gradient Method to the Case of Nonlinear Constraints*, Optimization, Edited by R. Fletcher, Academic Press, New York , pp. 37-47, (1968).
- [2] N. ALEXANDROV, M.Y. HUSSAINI, *Multidisciplinary Design Optimization: state of the art*, SIAM Proceedings Series, SIAM, Philadelphia, (1997).
- [3] M.B. AROUXÉT, N. ECHEBEST, E.A. PILOTTA, *Active-set strategy in Powell's method for optimization without derivatives*, Computational and Applied Mathematics, 30 (1), pp. 171-196, (2011).
- [4] J. BARZILAI, J.M. BORWEIN, *Two-point stepsize gradient methods*, IMA Journal of Numerical Analysis 8, pp. 141-148, (1988).
- [5] D.P. BERTSEKAS, *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.
- [6] R.H. BIELSCHOWSKY, F.A.M. GOMES, *Dynamic control of infeasibility in equality constrained optimization*, SIAM Journal on Optimization 19 (3), pp. 1299-1325, (2008).
- [7] E.G. BIRGIN, N. KREJIĆ, J.M. MARTÍNEZ, *Globally convergent inexact quasi-Newton methods for solving nonlinear systems*, Numerical Algorithms 32, pp. 249-260, (2003).
- [8] L.F. BUENO, A. FRIEDLANDER, J.M. MARTÍNEZ, F.N.C. SOBRAL, *Inexact restoration method for derivative-free optimization with smooth constraints*, SIAM Journal on Optimization, 23 (2), pp. 1189-1213, (2013).
- [9] A.R. CONN, K. SCHEINBERG, PH.L. TOINT, *Recent progress in unconstrained nonlinear optimization without derivatives*, Mathematical Programming, 79 (3), pp. 397-414, (1997).
- [10] A. CONN, K. SCHEINBERG, L.N. VICENTE, *Introduction to derivative-free optimization*. SIAM Book Series on Optimization, Philadelphia, 2009.
- [11] A.L. CUSTODIO, L.N. VICENTE, *Using sampling and simplex derivatives in pattern search methods*, SIAM Journal on Optimization, 18 (2), pp. 537-555, (2007).

- [12] N. DAN, N. YAMASHITA, M. FUKUSHIMA, *Convergence properties of inexact Levenberg-Marquardt method under local error bound conditions*, Optimization Methods and Software 17, pp. 605-626, (2002).
- [13] J.E. DENNIS, R.B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, 1983.
- [14] M.A. DINIZ-EHRHARDT, J.M. MARTÍNEZ, L.G. PEDROSO, *Derivative-free methods for nonlinear programming with general lower-level constraints*, Computational and Applied Mathematics, 30 (1), pp. 19-52, (2011).
- [15] E. DOLAN, J. MORÉ, *Benchmarking optimization software with performance profiles*, Mathematical Programming 91, pp. 201-213, (2002).
- [16] R. FLETCHER, N.I.M. GOULD, S. LEYFFER, PH.L. TOINT, A. WÄCHTER, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM Journal on Optimization, 13, pp. 635-659, (2002).
- [17] R. FLETCHER, S. LEYFFER, *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2), pp. 239-269, (2002).
- [18] J.B. FRANCISCO, N. KREJIĆ, J.M. MARTÍNEZ, *An interior point method for solving box-constrained underdetermined nonlinear systems*, Journal of Computational and Applied Mathematics 177, pp. 67-88, (2005).
- [19] C.C. GONZAGA, E.W. KARAS, M. VANTI, *A globally convergent filter method for nonlinear programming*, SIAM Journal on Optimization, 14 (3), pp. 646-669, (2004).
- [20] N.I.M. GOULD, PH. L. TOINT, *Nonlinear programming without a penalty function or a filter*, Mathematical Programming 122(1), pp. 155-196, (2010).
- [21] L. GRIPPO, F. LAMPARIELLO, S. LUCIDI, *A nonmonotone line search technique for Newton's method*, SIAM Journal on Numerical Analysis 23, pp.707-716, (1986).
- [22] W. HOCK, K. SCHITTKOWSKI, *Test Examples for Nonlinear Programming Codes*, Springer Series Lectures Notes in Economics Mathematical Systems, 1981.
- [23] T.G. KOLDA, R.M. LEWIS, V. TORCZON, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Review, 45, pp. 85-482 (2003).
- [24] T.G. KOLDA, R.M. LEWIS, V. TORCZON, *Stationarity results for generating set search for linearly constrained optimization*, SIAM Journal on Optimization, 17 (4), pp. 943- 968, (2006).
- [25] W. LA CRUZ, J.M. MARTÍNEZ, M. RAYDAN, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Mathematics of Computation 75, pp. 1429-1448, (2006).

- [26] L.S. LASDON, *Reduced Gradient Methods*, Nonlinear Optimization 1981, edited by M. J. D. Powell, Academic Press, New York, pp. 235-242, (1982).
- [27] R.M. LEWIS, V. TORCZON, *Pattern search algorithms for bound constrained minimization*, SIAM Journal on Optimization, 9 (4), pp. 1082-1099, (1999).
- [28] R.M. LEWIS, V. TORCZON, *Pattern search algorithms for linearly constrained minimization*, SIAM Journal on Optimization, 10 (3), pp. 917-941, (2000).
- [29] D.H. LI, M. FUKUSHIMA, *A derivative-free line search and global convergence of Broyden-like method for nonlinear equations*, Optimization Methods and Software 13, pp. 181-201, (2000).
- [30] R.M. LEWIS, V. TORCZON, *A globally convergent Augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM Journal on Optimization, 12 (4), pp. 1075-1089, (2002).
- [31] R.M. LEWIS, V. TORCZON, *A direct search approach to nonlinear programming problems using an Augmented Lagrangian method with explicit treatment of linear constraints*, Technical Report WMCS- 2010-01, College of William & Mary, Department of Computer Sciences, (2010).
- [32] J.M. MARTÍNEZ, *Quasi-Inexact Newton methods with global convergence for solving constrained nonlinear systems*, Nonlinear Analysis 30, pp. 1-7, (1997).
- [33] J.M. MARTÍNEZ, *Two-Phase Model Algorithm with Global Convergence for Nonlinear Programming*, Journal of Optimization Theory and Applications 96, pp. 397-436, (1998).
- [34] J.M. MARTÍNEZ, *Inexact Restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming*, Journal of Optimization Theory and Applications 111 (1), pp. 39-58, (2001).
- [35] J.M. MARTÍNEZ, E.A. PILOTTA, *Inexact Restoration algorithms for constrained optimization*, Journal of Optimization Theory and Applications 104 (1), pp. 135-163, (2000).
- [36] J.M. MARTÍNEZ, E.A. PILOTTA, *Inexact Restoration methods for nonlinear programming: Advances and perspectives*. In: Qi, Teo, Yang (eds.) Optimization and Control with Applications. Springer, Berlin, pp. 271-292, (2005).
- [37] J.M. MARTÍNEZ, B.F. SVAITER, *A practical optimality condition without constraint qualifications for nonlinear programming*, Journal of Optimization Theory and Applications, 118 (1), pp. 117-133, (2003).
- [38] A. MIELE, H.Y. HUANG, J.C. HEIDEMAN, *Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions*, Ordinary and Conjugate Gradient Version, Journal of Optimization Theory and Applications 4, pp. 213-246, (1969).

- [39] A. MIELE, A.V. LEVY, E.E. CRAGG, *Modifications and Extensions of the Conjugate-Gradient Restoration Algorithm for Mathematical Programming Problems*. Journal of Optimization Theory and Applications 7 , pp. 450-472, (1971).
- [40] A. MIELE, E.M. SIMS, V.K. BASAPUR, *Sequential Gradient-Restoration Algorithm for Mathematical Programming Problems with Inequality Constraints, Part 1, Theory*. Rice University, Aero-Astronautics Report No. 168, (1983).
- [41] J.J. MORÉ, S.M. WILD, *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (1), pp. 172-191 (2009).
- [42] M.J.D. POWELL, *The NEWUOA software for unconstrained optimization without derivatives*, Nonconvex Optimization and its Applications 83, pp. 255-297, (2006).
- [43] M.J.D. POWELL, *The BOBYQA algorithm for bound constrained optimization without derivatives*. Technical Report, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England, (2009).
- [44] M. ROM, M. AVRIEL, *Properties of the Sequential Gradient-Restoration Algorithm (SGRA), Part 1: Introduction and Comparison with Related Methods*, Journal of Optimization Theory and Applications 62, pp. 77-98, (1989).
- [45] M. ROM, M. AVRIEL, *Properties of the Sequential Gradient-Restoration Algorithm (SGRA), Part 2: Convergence Analysis*, Journal of Optimization Theory and Applications 62, pp. 99-126, (1989).
- [46] J.B. ROSEN, *The Gradient Projection Method for Nonlinear Programming, Part 1, Linear Constraints*, SIAM Journal on Applied Mathematics 8, pp. 181-217, (1960).
- [47] J.B. ROSEN, *The Gradient Projection Method for Nonlinear Programming, Part 2, Nonlinear Constraints*, SIAM Journal on Applied Mathematics 9, pp. 514-532, (1961).
- [48] J.B. ROSEN, *Two-Phase Algorithm for Nonlinear Constraint Problems*, Nonlinear Programming 3, Edited by O. L. Mangasarian, R. R. Meyer and S. M. Robinson, Academic Press, London and New York, pp. 97-124, (1978).
- [49] N.V. RYOO, N.D. SAHINIDIS, *Global Optimization of Nonconvex NLPs and MINLPs with applications in process design*, Computers & Chemical Engineering 19, pp. 551-566, (1995).
- [50] H.D. SCOLNIK, N. ECHEBEST, M.T. GUARDARUCCI, M.C. VACCHINO, *Incomplete Oblique Projections for Solving Large Inconsistent Linear Systems*, Mathematical Programming B 111, pp. 273-300, (2008).
- [51] P.B. STARK, R.L. PARKER, *Bounded Variable Least Squares: An Algorithm and Applications*, Journal Computational Statistics 10, pp. 129-141, (1995).

- [52] H.F. WALKER, L.T. WATSON, *Least-Change secant update methods for underdetermined systems*, SIAM Journal on Numerical Analysis 27 (5), pp. 1227-1262, (1990).
- [53] H. ZHANG, A.R. CONN, K. SCHEINBERG, *A derivative-free algorithm for least-squares minimization*, SIAM Journal on Optimization 20 (6), pp. 3555-3576, (2010).