



Universidade de Vigo

Traballo Fin de Mestrado

Optimización baixo incerteza en redes de gas.

Ana Belén Buide Carballosa

Mestrado en Técnicas Estatísticas

Curso 2016-2017

Proposta de Trabajo Fin de Mestrado

<p>Título en galego: Optimización baixo incerteza en redes de gas.</p>
<p>Título en castelán: Optimización bajo incertidumbre en redes de gas.</p>
<p>English title: Optimization gas networks under uncertainty.</p>
<p>Modalidade: B</p>
<p>Autora: Ana Belén Buide Carballosa, Universidad de Santiago de Compostela</p>
<p>Director: Julio González Díaz, Universidad de Santiago de Compostela</p>
<p>Titor: Diego Rodríguez Martínez, Instituto Tecnológico de Matemática Industrial</p>
<p>Breve resumo do traballo:</p> <p>O traballo a realizar polo alumno pode dividirse en varias fases:</p> <p>I) Problema sen incerteza (determinista):</p> <ul style="list-style-type: none"> - Familiarización cun problema de optimización non linear no contexto das redes de transporte de gas. - Familiarización co algoritmo implementado dende ITMATI. <p>II) Problema con incerteza (estocástico):</p> <ul style="list-style-type: none"> - Familiarización coa extensión do problema anterior a situacións multietapa baixo incerteza. - Familiarización cos conceptos fundamentais de programación estocástica. <p>III) Resolución do problema con incerteza:</p> <ul style="list-style-type: none"> - Familiarización co algoritmo de programación estocástica implementado dende ITMATI. - Validación do algoritmo implementado dende ITMATI sobre unha batería de problemas asociados á Rede Nacional de Gas Natural.

Agradecementos

Ao director do TFM, Julio González Díaz pola confianza depositada para realizar este proxecto, pola súa axuda e interese. A Ángel González por estar sempre a disposición para calquera dúbida sobre GANESO. Tamén a todos os compañeiros do ITMATI pola boa acollida durante a estancia das prácticas.

Como non a meus pais, o mellor exemplo de superación, a meus irmáns e ao anano da familia, Antón.

Para finalizar, a Carlos polo apoio e ánimo que me deu e pola inmensa paciencia que tivo durante estes últimos anos.

Índice xeral

Resumo	XI
Introdución	XIII
1. Nocións teóricas de programación matemática	1
1.1. Introducción á programación matemática	1
1.2. Programación linear. Condicións de optimalidade	3
1.3. Programación non linear. Condicións de optimalidade	4
1.4. Dualidade	10
1.5. Optimización con restricións.	13
1.5.1. Método de penalización exterior	14
1.5.2. Método lagrangiano aumentado	16
1.5.3. Programación linear sucesiva	20
2. Problema de optimización en redes de gas (determinista)	25
2.1. Formulación do problema	25
2.2. Algoritmo implementado dende o ITMATI (SLP-NTR)	28
3. Introdución á programación estocástica	31
3.1. Exemplo: o problema do granxeiro	31
3.2. Formulación dun problema de programación estocástica	33
3.2.1. Métodos baseados na programación estocástica lagrangiana	41
4. Problema con incerteza no contexto das redes de gas	47
4.1. Formulación do problema	47
4.2. Resolución do problema estocástico	49
5. Validación	53
5.1. GANESO	53
5.2. Labor desenvolvido nas prácticas	55

5.3. Rede de gas galega	58
5.3.1. Resultados con escenario de partida galicia1	59
5.3.2. Resultados con escenario de partida galicia2	63
5.3.3. Resultados con escenario de partida galicia3	68
5.3.4. Comparación	72
5.4. Rede de gas española	73
5.5. Conclusións e observacións	80
A. Código AMPL	83
B. Código R	89
Bibliografía	103

Índice de táboas

3.1. Información sobre os cultivos.	32
3.2. Solución óptima baseada no rendemento esperado, por encima e por debaixo.	33
3.3. Solución óptima baseada no modelo estocástico 3.8.	36
5.1. Cabeceira das táboas recollendo os resultados.	57
5.2. Resultados DEM asociados á batería de casos obtida do escenario galicia1.	60
5.3. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia1 (DEM).	60
5.4. Etapa de construción da infraestrutura para a batería de casos do escenario galicia1 (DEM).	60
5.5. Resultados PHA asociados á batería de casos obtida do escenario galicia1.	61
5.6. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia1 (PHA).	62
5.7. Etapa de construción da infraestrutura para a batería de casos do escenario galicia1 (PHA).	62
5.8. Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia1 e número de iteracións do PHA.	63
5.9. Resultados DEM asociados á batería de casos obtida do escenario galicia2.	65
5.10. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia2 (DEM).	65
5.11. Etapa de construción da infraestrutura para a batería de casos do escenario galicia2 (DEM).	65
5.12. Resultados PHA asociados á batería de casos obtida do escenario galicia2.	66
5.13. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia2 (PHA).	67
5.14. Etapa de construción da infraestrutura para a batería de casos do escenario galicia2 (PHA).	67
5.15. Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia2 e número de iteracións do PHA.	68
5.16. Resultados DEM asociados a batería de casos obtida do escenario galicia3.	69

5.17. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida a partir do escenario galicia3 (DEM).	70
5.18. Etapa de construción da infraestrutura para a batería de casos obtida a partir do escenario galicia3 (DEM).	70
5.19. Resultados PHA asociados á batería de casos obtidos do escenario galicia3.	71
5.20. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida do escenario galicia3 (PHA).	71
5.21. Etapa de construción da infraestrutura para a batería de casos obtida do escenario galicia3 (PHA).	71
5.22. Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia3 e número de iteracións do PHA.	72
5.23. Resultados DEM asociados á batería de casos obtida do escenario da rede de gas española.	75
5.24. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida da rede de gas española (DEM).	76
5.25. Etapa de construción da infraestrutura para a batería de casos obtida da rede de gas española (DEM).	76
5.26. Resultados PHA asociados á batería de casos obtida do escenario da rede de gas española.	77
5.27. Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida da rede de gas española (PHA).	78
5.28. Etapa de construción da infraestrutura para a batería de casos obtida da rede de gas española (PHA).	79
5.29. Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario da rede de gas española e número de iteracións do PHA.	80

Resumo

Resumo en galego

Este traballo xorde pola necesidade de validar un algoritmo de programación estocástica empregado pola empresa Reganosa para a resolución de problemas de optimización no contexto das redes de gas. A continuación enumeramos os principais contidos deste traballo. Lévese a cabo unha revisión das nocións teóricas da programación matemática. Introdúcese o problema de optimización determinista en redes de gas e o algoritmo de programación linear sucesiva de dúas etapas implementado dende o ITMATI. Introdúcese a programación estocástica e o problema de programación estocástica no contexto das redes de transporte de gas. A formulación deste problema é o punto de partida para a validación do algoritmo implementado sobre unha batería de problemas asociados ás redes de gas galega e nacional. Finalmente, amósase o traballo e resultados obtidos durante o transcurso das prácticas desenvolvidas nas instalacións do ITMATI. Estes resultados amosan a utilidade do algoritmo implementado na práctica.

English abstract

This work arises from the need to validate a stochastic programming algorithm that Reganosa company uses to solve optimization problems under uncertainty in the context of gas networks. Next, we list the main contents. First, we carry out a revision of theoretical notions of mathematical programming. We introduce a deterministic optimization problem in gas networks and the two-step sequential linear programming algorithm (2SLP) implemented in ITMATI. We introduce stochastic programming and the stochastic problem in the context of gas transport networks. The formulation of this problem is the starting point to validate the algorithm implemented in a set of problems associated with the Galician and the Spanish gas networks. Last, we show the work and results that we get during the internship at ITMATI. These results show the usefulness in practice of the algorithm.

Introdución

A optimización está presente nunha gran variedade de campos da sociedade. Así, é común no noso día a día a busca da mellor das alternativas entre todas as posibles, por exemplo na planificación dunha viaxe, e en máis dunha ocasión buscamos minimizar custos ou maximizar beneficios atendendo a diversas restricións. En xeral, nun problema de optimización atoparémonos cunha función obxectivo, que se desexa optimizar, coas variables do problema, que representan as posibles decisións e con restricións, que representan o conxunto de relacións que terán que satisfacer algunha das variables. Cando algún parámetro ou restrición do problema non se coñece con certeza dise que o problema de optimización é estocástico.

A programación estocástica xurdiu en 1955 cos traballos de Dantzig [7] e Beale [2]. En 1981 púxose en marcha o Committee on Stochastic Programming (COSP), como unha rama da Mathematical Programming Society. Este comité puxo en funcionamento a páxina web <http://stoprog.org> no que se pode encontrar información e documentación sobre a programación estocástica.

A programación estocástica serve para modelar problemas de optimización que involucran incerteza, mentres que os problemas de optimización deterministas fórmulanse con parámetros coñecidos. Na programación estocástica non se coñecen os valores dos parámetros, pero si as súas distribucións de probabilidade ou supóñense coñecidas aproximacións discretas das mesmas.

Na práctica son comúns os problemas que presentan incerteza, xa sexa pola falta de datos ou porque os parámetros se refiren a información sobre o futuro. Por exemplo, no contexto das redes de gas xorden problemas estocásticos na toma de decisións relativas á planificación da construción de novas infraestruturas en redes de gas. A incerteza afecta á demanda de gas na rede, ao prezo por unidade de gas e a custo por infraestrutura, por exemplo, non é posible determinar con certeza cal vai ser a demanda do gas no vindeiro mes.

Existen unha ampla variedade de áreas nas que a incerteza está presente como por exemplo na planificación de rutas, de produción, de inversións, etc. En [4] preséntanse distintos exemplos de problemas estocásticos en diversas áreas.

Debido á dificultade que presentan os problemas con incerteza, na práctica adoitase resolver o problema determinista resultante de substituír os parámetros estocásticos polos seus valores medios. Isto pode levar a unha solución infactible nos diversos escenarios existentes, xa que a solución do problema anterior só é a solución óptima do escenario promedio que non ten nin porque existir. O emprego da programación estocástica na resolución garante que se satisfaga a factibilidade en todos os escenarios, xa que se teñen en conta todos na resolución.

Este traballo está vinculado a unhas prácticas desenvolvidas nas instalacións do ITMATI dentro do proxecto GANESO. O principal obxectivo destas é a validación do algoritmo implementado desde o ITMATI sobre unha batería de problemas con incerteza asociados a unha rede de gas natural. Debido a isto introdúcese o problema de optimización non linear determinista no contexto das redes de transporte de gas e a extensión deste a situacións multietapa baixo incerteza. Ademais de familiarizarnos cos algoritmos implementados dende o ITMATI para a resolución destes.

Primeiro, no Capítulo 1, abórdase a revisión de nocións teóricas de programación matemática. Faise unha breve introdución á programación matemática. Introdúcese o problema de programación linear e non linear, e os principais resultados de optimalidade que garanten a existencia de óptimos. Abórdase a revisión dos principais resultados de dualidade en programación non linear. Finalmente, introdúcense algoritmos para a resolución de problemas de optimización con restricións: o método de penalización exterior, o método Lagrangiano aumentado e a programación linear sucesiva. Esta última técnica ten especial interese, dado que o algoritmo de resolución, que empregan dende ITMATI para a resolución do problema determinista no contexto das redes de gas, fai uso dunha variante desta.

No Capítulo 2 abórdase a descrición do problema de optimización determinista no contexto das redes de gas e o algoritmo implementado dende o ITMATI para a súa resolución, a programación linear sucesiva de dúas etapas (2SLP). Esta técnica de resolución consiste en dúas etapas, na primeira dela aplícase unha variante da programación linear sucesiva, que principalmente consiste en eliminar a rexión de confianza para fixar as variables binarias e logo aplicar a programación linear sucesiva penalizada partindo da solución obtida na etapa anterior.

O Capítulo 3 céntrase na programación estocástica. Introdúcese o problema de programación estocástico de dúas etapas, o problema multietapa, o principio de non anticipación e métodos de resolución do problema estocástico baseados na programación estocástica lagrangiana, en concreto o algoritmo *progressive hedging* que se emprega na resolución do problema estocástico non linear no contexto das redes de gas. Todos estes conceptos ilústranse cun exemplo coñecido na literatura como problema do granxeiro. No Apéndice A recóllese a implementación en AMPL do problema determinista e estocástico asociado a este exemplo.

No Capítulo 4 fórmulase o problema con incerteza no contexto das redes de gas e como sería a resolución deste empregando o método determinista e o método *progressive hedging*. O uso de ambos métodos na resolución é validado no último capítulo deste traballo.

Finalmente, no Capítulo 5, abórdase a validación dos métodos de resolución do problema estocástico, introducidos no capítulo anterior. Introdúcese brevemente o proxecto no que se engloban as prácticas, a ferramenta GANESO e amósase parte do traballo desenvolvido. O obxectivo das prácticas é validar o algoritmo implementado no proxecto para a resolución do problema con incerteza. A validación consiste en resolver unha batería de problemas, resultantes da modificación dos parámetros de planificación na formulación do problema estocástico asociado a rede de gas galega e nacional, e analizar os resultados obtidos. Para isto elaboráronse varios scripts na linguaxe de programación R que se recollen no Apéndice B.

Capítulo 1

Nocións teóricas de programación matemática

Neste primeiro capítulo imos introducir conceptos básicos de programación matemática. Falaremos de programación linear e das principais condicións de optimalidade nestes problemas. Consideramos logo problemas de programación non linear e introduciremos condicións de optimalidade para problemas de programación convexa e as condicións de Fritz John e as condicións de Karush-Kuhn-Tucker para problemas con función obxectivo e restricións diferenciables. Logo introducimos a dualidade que veremos que está relacionada con algún dos algoritmos que trataremos. Na última sección presentamos algúns dos algoritmos que se empregan para resolver problemas de optimización con restricións. O obxectivo dos algoritmos que presentamos é identificar solucións que cumpran algunha condición de optimalidade. En concreto falaremos dos métodos de penalización, do método Lagrangiano aumentado e da programación linear sucesiva. Este último algoritmo ten especial importancia neste traballo, xa que, o algoritmo que se emprega para resolver o problema de optimización no contexto das redes de transporte de gas, do que falaremos no seguinte capítulo, resulta ser unha modificación deste.

Como se verá, centrámonos principalmente en problemas de optimización non lineais e con restricións de igualdade e desigualdade, dada a natureza dos problemas que tratamos durante as prácticas. Enunciamos os principais resultados pero sen entrar nas demostracións que se poden ver en [1]. A maioría delas tamén son tratadas nos apuntes da materia de Programación Matemática [10].

Como consideración relativa á notación, mencionar que ao longo deste capítulo, os vectores se denotarán con subíndices, e se usarán os superíndices para denotar os elementos dunha sucesión. Empregarase a letra k como índice dos elementos dunha sucesión ou das iteracións dun determinado algoritmo. Mediante T designase a trasposta dunha matriz ou vector.

1.1. Introducción á programación matemática

Os problemas de optimización aparecen en multitude de campos da vida real, en especial, no ámbito da enxeñería, proxectos empresariais, planificación de transportes, na industria petroleira, etc. O propósito destes é buscar a mellor decisión entre todas as posibles, tendo en conta os recursos dispoñibles, as restricións presentes, etc.

Definición 1.1. Un problema de optimización ven dado por un par (X, f) onde X é o conxunto de puntos factibles e f a función obxectivo ou función de custo, definida nos puntos de X , que queremos

minimizar. O problema consiste en atopar un *punto factible* $\bar{\mathbf{x}} \in X$ tal que

$$f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$$

para todo punto $\mathbf{x} \in X$. Dicimos que $\bar{\mathbf{x}}$ é un óptimo global do problema.

Un problema de programación matemática é un problema de optimización ao que se lle engade certa estrutura matemática. Consiste en atopar unha solución, entre todas as posibles, que satisfaga as restricións que definen o conxunto factible e minimize a función obxectivo f .

Definición 1.2. Un *problema de programación matemática* consiste en atopar unha solución ao problema:

$$\begin{aligned} &\text{minimizar } f(\mathbf{x}) \\ &\text{suxeito a } g_i(\mathbf{x}) \leq 0, \quad \forall i \in \{1, \dots, m\} = I^m \\ &\quad h_j(\mathbf{x}) = 0, \quad \forall j \in \{1, \dots, l\} = I^l, \end{aligned} \tag{1.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ para $i \in I^m$ e $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ para $j \in I^l$.

Como xa mencionamos, a función f recibe o nome de *función obxectivo* ou *función de custo*. Cada restrición $g_i(\mathbf{x}) \leq 0 \forall i \in I^m$ será unha *restrición de desigualdade* e cada restrición $h_j(\mathbf{x}) = 0 \forall j \in I^l$ será unha *restrición de igualdade*. No conxunto X normalmente inclúense os límites superior e inferior das variables e restricións “fáciles” que non teñen interese á hora de empregar algoritmos na resolución.

Un vector $\mathbf{x} \in X$ satisfacendo todas as restricións é unha *solución factible* do Problema 1.1. Polo que, o *conxunto factible* defínese como $X = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, h_j(\mathbf{x}) = 0, \text{ para } i \in I^m \text{ e } j \in I^l\}$. Un problema de programación matemática é *non acotado* cando X non está acotado, e *acotado* en caso contrario. Dise que é *non factible* cando $X = \emptyset$.

Dado o Problema 1.1, dicimos que o punto $\bar{\mathbf{x}} \in X$ é un *óptimo global* se, para todo $\mathbf{x} \in X$, $f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$. Dicimos que un punto $\bar{\mathbf{x}} \in X$ é un *óptimo local* se existe $\epsilon > 0$ tal que, para todo $\mathbf{x} \in B(\bar{\mathbf{x}}, \epsilon) \cap X$, $f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$; se esta última desigualdade é estrita, dicimos que $\bar{\mathbf{x}}$ é un *óptimo local estrito*.

Un problema de programación matemática dise que é un *problema determinista* se todos os seus elementos se poden coñecer con exactitude. Por outra banda, dirase que é un *problema estocástico* se algún dos elementos está suxeito a incerteza.

Está claro que un problema de programación será *linear* se as funcións que o definen son lineais e en caso contrario dirase que é *non linear*.

Dirase que un problema de programación matemática é *convexo* se as restricións de desigualdade son convexas, as restricións de igualdade son lineais e a función obxectivo é convexa.

Para finalizar esta sección definimos os problema de programación enteira ou problemas discretos, xa que nalgunha ocasión aparecen no contexto das redes de transporte de gas, como veremos no seguinte capítulo.

Definición 1.3. Un problema de programación enteira consiste en atopar unha solución ao problema:

$$\begin{aligned} &\text{minimizar } f(\mathbf{x}) \\ &\text{suxeito a } g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m \\ &\quad h_j(\mathbf{x}) = 0, \quad \forall j \in I^l \\ &\quad \mathbf{x}_t \in \mathbb{Z}, \quad t \in I \subseteq \{1, \dots, n\}, \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ para $i \in I^m$ e $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ para $j \in I^l$.

Cando $I = \{1, \dots, n\}$ e $x_t \in \{0, 1\}$ para todo $t \in I$, estas variables chámanse *variables binarias*.

1.2. Programación linear. Condicións de optimalidade

Un problema de programación linear consiste nun problema de programación matemática cuxo obxectivo é optimizar unha función linear no conxunto de puntos que verifican certas restricións lineais de igualdade e de desigualdade.

Definición 1.4. Un problema de programación linear LP é da forma:

$$\begin{aligned} &\text{minimizar } \mathbf{c}^\top \cdot \mathbf{x} \\ &\text{suxeito a } \mathbf{a}_i^\top \cdot \mathbf{x} \leq b_i, \quad i \in I^m \\ &\quad \mathbf{q}_j^\top \cdot \mathbf{x} = d_j, \quad \forall j \in I^l, \end{aligned}$$

onde $\mathbf{c}, \mathbf{a}_i, \mathbf{q}_j \in \mathbb{R}^n$ e $b_i, d_j \in \mathbb{R}$ para $i \in I^m$ e para $j \in I^l$. Denotamos por $\mathbf{x} \cdot \mathbf{y}$ o produto escalar de dous vectores. O conxunto factible ven dado por:

$$X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_i^\top \cdot \mathbf{x} \leq b_i, \mathbf{q}_j^\top \cdot \mathbf{x} = d_j, \text{ para } i \in I^m, j \in I^l\}.$$

Proposición 1.5. O conxunto factible X de LP é un conxunto convexo.

A convexidade do conxunto factible é unha propiedade moi importante, pois se a función obxectivo é convexa, entón todo óptimo local é un óptimo global.

Proposición 1.6. Dado un problema de programación linear LP, todo óptimo local é un óptimo global.

Outra propiedade moi interesante da programación linear é que dados dous óptimos, calquera combinación convexa de eles tamén é un óptimo.

Proposición 1.7. Dado un problema de programación linear LP e dous óptimos \mathbf{x} e \mathbf{y} , o punto $\mathbf{z} = \lambda \mathbf{x} + (1 - \lambda) \mathbf{y}$ para $\lambda \in (0, 1)$ tamén é un óptimo.

As dúas últimas proposicións tamén son certas para problemas de programación matemática convexa, é dicir, problemas de programación matemática non linear con restricións de desigualdade convexas, restricións de igualdade lineais e función obxectivo convexa.

A programación linear presenta dúas importantes vantaxes á hora de resolver o problema que non se dan en programación convexa non linear. A rexión factible X é un politopo, é dicir, a envoltura convexa dunha cantidade finita de puntos. Por outro lado, as curvas de nivel da función obxectivo son hiperplanos. Destas propiedades resulta o seguinte resultado:

Proposición 1.8. *Dado un problema de programación lineal LP, unha das seguintes afirmacións é certa:*

1. *O conxunto X é vacío e non hai óptimos.*
2. *O conxunto X é acotado e polo menos un dos seus vértices é un óptimo.*
3. *O conxunto X é non acotado, e ou ben non ten óptimos, ou un dos seus vértices é un óptimo.*

A busca dun óptimo local pode reducirse a un número finito de puntos, ao número de vértices de X . Isto converte ao problema nun problema de combinatoria. Pero cando o número de vértices é grande son necesarios algoritmos que minimicen a busca. Por exemplo, o *método do simplex*. Este algoritmo deseñado para resolver problemas de programación lineal baséase na idea de que se o valor da función obxectivo nun vértice é menor ou igual que o valor da función nos vértices veciños, entón estamos ante un óptimo (global). Consiste en que partindo dun vértice escolle entre os vértices veciños algún que toma un valor menor da función obxectivo, ata obter o óptimo.

1.3. Programación non linear. Condicións de optimalidade

Un problema de programación non linear, como xa mencionamos, é un problema de programación matemática no que algunha das funcións que definen o problema son non lineais. A diferenza da programación lineal, en problemas de programación non linear soe ser complicado encontrar óptimos globais, e habitualmente hai que conformarse con algunha condición de optimalidade local. Así, resulta útil caracterizar os óptimos locais mediante condicións necesarias e suficientes, tanto para facilitar a súa busca, como para contrastar a optimalidade local. Nesta sección imos ver as condicións de Fritz John e as condicións de Karush-Kuhn-Tucker para problemas de programación non linear nos que as funcións que definen o problema son diferenciables. Antes disto introduciremos as condicións de optimalidade para problemas de programación convexa dende o enfoque de direccións de descenso e direccións factibles.

Consideremos a formulación xeral dun problema de optimización $\min_{\mathbf{x} \in X} f(\mathbf{x})$, con $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Recordamos que os problemas nos que tanto a función obxectivo f como o conxunto factible X son convexos se denominan *problemas de programación convexa*. No caso de que a función obxectivo sexa cóncava, os resultados seguirán sendo certos pero para problemas de maximizar en vez de minimizar. Pois minimizar $f(\mathbf{x})$ é o mesmo que maximizar $-f(\mathbf{x})$ e $f(\mathbf{x})$ é convexa se e só se $-f(\mathbf{x})$ é cóncava.

O principal resultado de optimalidade en problemas de programación convexa é que un mínimo local tamén é global.

Teorema 1.9. *Sexa X un conxunto non vacío e convexo en \mathbb{R}^n , e $f : X \rightarrow \mathbb{R}$ convexa en X . Se $\bar{\mathbf{x}} \in X$ é un óptimo local do problema. Entón:*

1. *$\bar{\mathbf{x}}$ é un óptimo global.*
2. *Se $\bar{\mathbf{x}}$ é un óptimo local estrito ou f é estritamente convexa, $\bar{\mathbf{x}}$ é o único óptimo global.*

Definimos o conxunto de direccións de descenso e direccións factibles aos que nos referiremos nas condicións de optimalidade que trataremos posteriormente.

Definición 1.10. O conxunto de *direccións de descenso* da función f no punto \mathbf{x} , $D^D(f, \mathbf{x})$, ven dado por

$$D^D(f, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : f(\mathbf{x} + \lambda \mathbf{d}) < f(\mathbf{x}) \text{ para todo } \lambda \in (0, \delta), \text{ para un certo } \delta > 0\}.$$

Definición 1.11. O conxunto de *direccións factibles* do conxunto S no punto \mathbf{x} , $D^F(X, \mathbf{x})$, ven dado por

$$D^F(X, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d} \neq \mathbf{0} \text{ e } \mathbf{x} + \lambda \mathbf{d} \in X \text{ para todo } \lambda \in (0, \delta), \text{ para un certo } \delta > 0\}.$$

Denotamos os dous conxuntos simplemente como D^D e D^F e referímonos aos seus elementos como direccións de descenso e direccións factibles, respectivamente. As definicións anteriores dinnos que un vector $\mathbf{d} \in \mathbb{R}^n$ é unha dirección de descenso se ao realizar desprazamentos suficientemente pequenos partindo dun punto \mathbf{x} da rexión factible na dirección \mathbf{d} , conseguimos que o valor da función obxectivo sexa menor que $f(\mathbf{x})$, e é unha dirección factible se ao realizar desprazamentos suficientemente pequenos partindo de \mathbf{x} na dirección de \mathbf{d} nos movemos no interior do conxunto factible X .

A continuación presentamos algunha das condicións de optimalidade para o caso de problemas de programación convexa con función obxectivo diferenciable. Recordamos, a condición clásica de optimalidade que nos di que o gradiente se ten que anular nos puntos óptimos (do interior do dominio).

Proposición 1.12. $\bar{\mathbf{x}} \in X$ é un óptimo global do problema se e só se, para todo $\mathbf{x} \in X$, $\nabla f(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) \geq 0$. Ademais, se X é aberto, $\bar{\mathbf{x}}$ é un óptimo global do problema se e só se $f(\bar{\mathbf{x}}) = 0$.

O seguinte resultado permítenos relacionar o concepto de dirección de descenso co de derivada direccional. Ademais danos unha condición suficiente para que unha dirección sexa de descenso.

Proposición 1.13. Sexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$ unha función diferenciable. Dados un punto $\mathbf{x} \in \mathbb{R}^n$ e unha dirección de descenso $\mathbf{d} \in \mathbb{R}^n$. Se $\nabla f(\mathbf{x})^\top \mathbf{d} < 0$, entón \mathbf{d} é unha dirección de descenso en \mathbf{x} . Se ademais é convexa o recíproco é certo.

Nótese que o recíproco do resultado anterior non é certo se a función non é convexa, pois, para a función $f(x) = x^3$, a dirección $d = -1$ é unha dirección de descenso en $x = 0$ e, pola contra, $f'(0, -1) = \nabla f(0)(-1) = 0$.

A caracterización de óptimos globais dun problema de programación convexa pode verse como aqueles puntos nos que non hai ningunha dirección que sexa simultaneamente factible e descenso.

Teorema 1.14. Se $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é unha función convexa e $X \subseteq \mathbb{R}^n$ un conxunto non vacío e convexo. O punto $\bar{\mathbf{x}} \in X$ é un óptimo local (e global) do problema de optimización se e só se $D^D \cap D^F = \emptyset$.

Na práctica os conxuntos D^D e D^F son difíciles de manipular. Por isto, búscanse condicións necesarias sobre conxuntos máis fáciles de manipular analiticamente. Así, defínense os seguintes conxuntos asociados a $D^D(f, \mathbf{x})$:

$$D^D_-(f, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x})^\top \mathbf{d} < 0\} \quad \text{e} \quad D^D_+(f, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \nabla f(\mathbf{x})^\top \mathbf{d} \leq 0\}.$$

A continuación presentamos unha primeira condición necesaria de optimalidade local. Este resultado dinos que se nun punto dado temos $D^D_- \cap D^F \neq \emptyset$, entón hai algunha dirección que é simultaneamente de descenso e factible, polo que ese punto non pode ser mínimo local.

Teorema 1.15. Sexa a función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e o conxunto $X \subseteq \mathbb{R}^n$ un conxunto non vacío. Consideramos o seguinte problema de optimización:

$$\begin{aligned} & \text{minimizar} && f(\mathbf{x}) \\ & \text{suxeito a} && \mathbf{x} \in X. \end{aligned}$$

Se f é diferenciable no punto $\bar{\mathbf{x}} \in S$ e $\bar{\mathbf{x}}$ é un mínimo local, entón $D^D_- \cap D^F = \emptyset$.

O conxunto D_-^D calcúlase usando o gradiente de f , a pesar de que o conxunto de direccións factibles D^F non está definido en termos de gradientes. A condición necesaria de optimalidade local $D_-^D \cap D^F = \emptyset$ non resulta demasiado útil. Necesitamos dotar ao problema de estrutura matemática para poder obter condicións necesarias de optimalidade local nunha linguaxe alxebrica. Estas condicións coñécense co nome de condicións de Karush-Kuhn-Tucker.

Antes de presentar as condicións de Karush-Kuhn-Tucker (KKT) imos ver as condicións necesarias de Fritz John (FJ). Centrémonos nun problema de programación non linear tanto con restricións de igualdade como de desigualdade. Discusións previas destas condicións con problemas máis sinxelos e as probas dos resultados que introducimos poden verse en [1].

Para $\mathbf{x} \in X$, denotamos por $I(\mathbf{x}) = \{i \in I^m : g_i(\bar{\mathbf{x}}) = 0\}$ o conxunto de restricións de desigualdades activas ou saturadas en \mathbf{x} . As restricións asociadas a este conxunto son as que nos limitan para movernos dende o punto \mathbf{x} , pois estamos no interior do conxunto definido polas outras restricións. Definido este conxunto estamos en condicións de introducir os seguintes conxuntos ligados ao cono de direccións factibles, D^F :

$$D_-^F(X, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \nabla g_i(\mathbf{x})^\top \mathbf{d} < 0, \forall i \in I(\mathbf{x})\} \quad \text{e}$$

$$D_+^F(X, \mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \nabla g_i(\mathbf{x})^\top \mathbf{d} \leq 0, \forall i \in I(\mathbf{x})\}.$$

Teorema 1.16. *Consideremos o seguinte problema de optimización:*

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{suxeito a } g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m \\ & \quad h_j(\mathbf{x}) = 0, \quad \forall j \in I^l, \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ para $i \in I^m$ e $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ para $j \in I^l$. Supoñamos que $\bar{\mathbf{x}}$ é un punto factible tal que

- as funcións f e g_i con $i \in I$ son diferenciables en $\bar{\mathbf{x}}$ e
- as funcións g_i para cada $i \notin I$ son continuas en $\bar{\mathbf{x}}$ e cada h_j para $j \in I^l$ é continuamente diferenciable en $\bar{\mathbf{x}}$,

sendo $I(\bar{\mathbf{x}}) = \{i \in I^m : g_i(\bar{\mathbf{x}}) = 0\}$ o conxunto de restricións de desigualdades activas en $\bar{\mathbf{x}}$. Se $\nabla h_j(\bar{\mathbf{x}})$ para $j \in I^l$ son linearmente independentes, entón

$$D_-^D \cap D_-^F \cap H_0 = \emptyset,$$

onde $H_0 = \{\mathbf{d} : \nabla h_j(\bar{\mathbf{x}})^\top \mathbf{d} = 0 \text{ para } j \in I^l\}$.

Para problemas con unicamente restricións de igualdade a condición de optimalidade sería simplemente $D_-^D \cap D_-^F = \emptyset$. Se un punto $\bar{\mathbf{x}}$ verifica esta condición quere dicir que non existe ningún vector \mathbf{d} que forme un ángulo menor de 90° co gradiente en $\bar{\mathbf{x}}$ e, simultaneamente menor de 90° con todos os gradientes das restricións activas en $\bar{\mathbf{x}}$. Xeometricamente, isto está relacionado co feito de que $-\nabla f(\bar{\mathbf{x}})$ este no cono xerado polos gradientes de ditas restricións.

Presentamos agora as condicións necesarias de Fritz John que nos din que a condición xeométrica de optimalidade $D_-^D \cap D_-^F = \emptyset$ pode escribirse en termos de gradientes da función obxectivo, das restricións de desigualdade activas e das restricións de igualdade.

Teorema 1.17 (Condicións necesarias de Fritz-John). *Consideremos o seguinte problema de optimización:*

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{suxeito a } g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m \\ & \quad h_j(\mathbf{x}) = 0, \quad \forall j \in I^l \\ & \quad \mathbf{x} \in X. \end{aligned}$$

Supoñamos que $\bar{\mathbf{x}}$ é un punto factible tal que

- as funcións f e g_i con $i \in I(\bar{\mathbf{x}})$ son diferenciables en $\bar{\mathbf{x}}$ e
- as funcións g_i para cada $i \in I(\bar{\mathbf{x}})$ son continuas en $\bar{\mathbf{x}}$ e cada h_j para $j \in I^l$ é continuamente diferenciable en $\bar{\mathbf{x}}$,

sendo $I(\bar{\mathbf{x}}) = \{i \in I^m : g_i(\bar{\mathbf{x}}) = 0\}$ o conxunto de restricións activas ou saturadas en $\bar{\mathbf{x}}$. Se $\bar{\mathbf{x}}$ é un mínimo local, entón existen u_0 e u_i para todo $i \in I$ e v_j para $j \in I^l$ tales que

$$\begin{aligned} u_0 \nabla f(\bar{\mathbf{x}}) + \sum_{i \in I} u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla h_j(\bar{\mathbf{x}}) &= \mathbf{0} \\ u_0, u_i &\geq 0 && \forall i \in I \\ (u_0, \mathbf{u}_I, \mathbf{v}) &\neq (0, \mathbf{0}, \mathbf{0}), \end{aligned}$$

onde \mathbf{u}_I é o vector cuxas compoñentes son u_i para $i \in I$ e $\mathbf{v} = (v_1, \dots, v_l)^\top$. Se, ademais, as funcións g_i con $i \notin I$ son diferenciables en $\bar{\mathbf{x}}$, entón estas condicións poden ser escritas, de modo equivalente, como:

$$\begin{aligned} u_0 \nabla f(\bar{\mathbf{x}}) + \sum_{i=1}^m u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla h_j(\bar{\mathbf{x}}) &= \mathbf{0} \\ u_i g_i(\bar{\mathbf{x}}) &= 0 && \text{para } i \in I^m \\ u_0, u_i &\geq 0 && \text{para } i \in I^m \\ (u_0, \mathbf{u}, \mathbf{v}) &\neq (0, \mathbf{0}, \mathbf{0}). \end{aligned}$$

onde $\mathbf{u} = (u_1, \dots, u_m)^\top$ e $\mathbf{v} = (v_1, \dots, v_l)^\top$.

Os escalares u_0, u_i para $i \in I^m$ e v_j para $j \in I^l$ reciben o nome de *multiplicadores de Lagrange*. A condición de que $\bar{\mathbf{x}}$ sexa factible coñécese como *condición de factibilidade do primal* (FP) e o resto de condición como *condicións de factibilidade do dual* (FD). As condicións $u_i g_i(\bar{\mathbf{x}}) = 0$ coñécese como *condicións de folgura complementaria* (FC). (As condicións de folgura complementaria relacionan, as variables de folgura de cada restrición do problema primal, coas variables duais asociadas a ditas restricións e as folguras do dual, coas variables primais asociadas. Pódense interpretar como que dúas solucións posibles do primal e do dual son, respectivamente óptimas, se e só se, toda variable asociada a unha restrición con variable de folgura distinta de cero, é nula.) En particular, o multiplicador de Lagrange asociado as restricións que non se saturan en $\bar{\mathbf{x}}$ serán cero.

Calquera punto $\bar{\mathbf{x}}$ para o que existen multiplicadores de Lagrange $(\bar{u}_0, \bar{\mathbf{u}})$ cumprindo as condicións FP, FD e FC chamarase punto Fritz-John (FJ).

Na formulación do teorema anterior, observamos que calquera punto onde se anule ∇f ou algún gradiente dunha restrición activa nese punto ou das restricións h_i será automaticamente un punto FJ. En particular, sempre que $D^F = \emptyset$ teremos un punto FJ, independentemente do que diga a función obxectivo. Isto é un problema, pois pódese ter $D^F = \emptyset$ pero $D^F \neq \emptyset$, co que se podería ter direccións factibles. Se ademais estas direccións son de descenso, non temos optimalidade. Para resolver este problema xorden as condicións de KKT.

Teorema 1.18 (Condicións necesarias de Karush-Kuhn-Tucker). *Consideremos o seguinte problema de optimización:*

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{suxeito a } g_i(\mathbf{x}) \leq 0, \quad i \in I^m. \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j \in I^l. \end{aligned}$$

Supoñamos que $\bar{\mathbf{x}}$ é un punto factible tal que

- as funcións f e g_i con $i \in I(\bar{\mathbf{x}})$ son diferenciables en $\bar{\mathbf{x}}$,
- as funcións g_i con $i \notin I(\bar{\mathbf{x}})$ son continuas en $\bar{\mathbf{x}}$ e cada h_j para $j \in I^l$ é continuamente diferenciable en $\bar{\mathbf{x}}$,
- os vectores $\nabla g_i(\bar{\mathbf{x}})$, con $i \in I(\bar{\mathbf{x}})$, e $\nabla h_j(\bar{\mathbf{x}})$ con $j \in I^l$ son linearmente independentes,

sendo $I(\bar{\mathbf{x}}) = \{i \in I^m : g_i(\bar{\mathbf{x}}) = 0\}$ o conxunto de restricións activas ou saturadas en $\bar{\mathbf{x}}$.

Se $\bar{\mathbf{x}}$ é un mínimo local, entón $\bar{\mathbf{x}}$ é un punto KKT. Isto é, existen escalares únicos u_i para todo $i \in I(\bar{\mathbf{x}}) = I$ e v_j para todo $j \in I^l$ tales que

$$\begin{aligned} \nabla f(\bar{\mathbf{x}}) + \sum_{i \in I} u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla h_j(\bar{\mathbf{x}}) &= \mathbf{0} \\ u_i &\geq 0 \quad \forall i \in I. \end{aligned}$$

Se, ademais, as funcións g_i con $i \notin I(\bar{\mathbf{x}})$ son diferenciables en $\bar{\mathbf{x}}$, entón estas condicións poden ser escritas, de modo equivalente, como:

$$\begin{aligned} \nabla f(\bar{\mathbf{x}}) + \sum_{i=1}^m u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla h_j(\bar{\mathbf{x}}) &= \mathbf{0} \\ u_i g_i(\bar{\mathbf{x}}) &= 0 \quad \forall i \in I^m. \\ u_i &\geq 0 \quad \forall i \in I^m. \end{aligned}$$

A demostración do teorema anterior, como xa se mencionou, pode verse en [1].

Condicións KKT de segunda orde

Antes de presentar as condicións de segunda orde é interesante introducir o concepto de *función lagrangiana*, que é importante no desenvolvemento da teoría de dualidade que trataremos na seguinte sección e no algoritmo para a resolución de problemas con restricións, método lagrangiano aumentado.

Definición 1.19. Dado un problema de optimización da forma

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) \\ \text{suxeito a} \quad & g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m, \\ & h_j(\mathbf{x}) = 0, \quad \forall j \in I^l, \end{aligned} \tag{1.2}$$

onde todas as funcións son dúas veces diferenciáveis, a *función lagrangiana* defínese como

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}). \tag{1.3}$$

Esta función está moi relacionada coas condicións KKT pois, dado $\bar{\mathbf{x}}$ un punto KKT con multiplicadores $\bar{\mathbf{u}}$ e $\bar{\mathbf{v}}$ podemos falar da función lagrangiana restrinxida ao problema primal:

$$L^P(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^m \bar{u}_i g_i(\mathbf{x}) + \sum_{j=1}^l \bar{v}_j h_j(\mathbf{x}).$$

Previamente a presentar a condición KKT de segunda orde precisamos dividir o conxunto de restricións activas nun punto KKT $\bar{\mathbf{x}}$, $I(\bar{\mathbf{x}}) = \{i : g_i(\bar{\mathbf{x}}) = 0\}$. Se \mathbf{u} e \mathbf{v} son os multiplicadores asociados a $\bar{\mathbf{x}}$, consideramos:

Restricións fortemente activas: $I^+(\bar{\mathbf{x}}) = \{i : \bar{u}_i > 0\}$.

Restricións debilmente activas: $I^0(\bar{\mathbf{x}}) = \{i : \bar{u}_i = 0\}$.

Nas condicións KKT de segunda orde traballaremos coa hessiana da función lagrangiana reducida para un punto $\bar{\mathbf{x}}$ con multiplicadores dados por $\bar{\mathbf{u}}$ e $\bar{\mathbf{v}}$:

$$\nabla^2 L^P(\bar{\mathbf{x}}) = \nabla^2 f(\bar{\mathbf{x}}) + \sum_{i \in I(\bar{\mathbf{x}})} \bar{u}_i \nabla^2 g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l \bar{v}_j \nabla^2 h_j(\bar{\mathbf{x}}),$$

onde $\nabla^2 f(\bar{\mathbf{x}})$, $\nabla^2 g_i(\bar{\mathbf{x}})$ e $\nabla^2 h_j(\bar{\mathbf{x}})$ son as hessianas das funcións que definen o problema avaliadas no punto $\bar{\mathbf{x}}$.

Defínese o cono de direccións como

$$\begin{aligned} C(\bar{\mathbf{x}}) = \{ \mathbf{0} \neq \mathbf{d} \in \mathbb{R}^n : & \nabla^2 g_i(\bar{\mathbf{x}})^\top \mathbf{d} = 0 \quad \forall i \in I^+(\bar{\mathbf{x}}), \\ & \nabla^2 g_i(\bar{\mathbf{x}})^\top \mathbf{d} \leq 0 \quad \forall i \in I^0(\bar{\mathbf{x}}), \text{ e} \\ & \nabla^2 h_j(\bar{\mathbf{x}})^\top \mathbf{d} = 0 \quad \forall j \in I^l \} \end{aligned}$$

Teorema 1.20 (Condicións suficientes de Karush-Kunh-Tucker de segunda orde). *Consideremos o seguinte problema de optimización:*

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) \\ \text{suxeito a} \quad & g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m, \\ & h_j(\mathbf{x}) = 0, \quad \forall j \in I^l. \end{aligned}$$

Supoñamos que as funcións f, g_i con $i \in I^m$ e h_j con $j \in I^l$ son dúas veces diferenciables. Supoñamos que $\bar{\mathbf{x}}$ é un punto KKT con multiplicadores de Lagrange dados por $\bar{\mathbf{u}}$ e $\bar{\mathbf{v}}$. Dada a hessiana da función lagrangiana reducida, $\nabla^2 L^P(\bar{\mathbf{x}})$. Entón, se $\mathbf{d}^\top \nabla^2 L^P(\bar{\mathbf{x}}) \mathbf{d} > 0$ para todo $\mathbf{d} \in C(\bar{\mathbf{x}})$, o punto $\bar{\mathbf{x}}$ é un mínimo local estrito.

O anterior teorema é unha xeneralización da condición suficiente de optimalidade local para problemas de optimización sen restricións:

Proposición 1.21. *Sexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$ unha función dúas veces diferenciable en $\bar{\mathbf{x}}$. Se $\nabla f(\bar{\mathbf{x}}) = 0$ e a hessiana da función f , $H(\bar{\mathbf{x}})$, é definida positiva, entón $\bar{\mathbf{x}}$ é un mínimo local estrito.*

1.4. Dualidade

Nesta sección imos familiarizarnos cos resultados principais de dualidade en programación non linear. Introdúcese a dualidade lagrangiana da que fai uso o método lagrangiano aumentado, algoritmo que se emprega na resolución de problemas de optimización con restricións e que tratamos na Sección 1.9.

Recordamos que un problema de programación non linear consiste en encontrar unha solución ao problema:

Problema Primal P

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) \\ \text{suxeito a} \quad & g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m, \\ & h_j(\mathbf{x}) = 0, \quad \forall j \in I^l. \end{aligned} \tag{1.4}$$

Existe outro problema de programación non linear asociado ao anterior que chamamos *dual lagrangiano* que se formula como segue:

Problema Dual D

$$\begin{aligned} \text{maximizar} \quad & \theta(\mathbf{u}, \mathbf{v}) = \inf\{f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}) : \mathbf{x} \in X\} \\ \text{suxeito a} \quad & \mathbf{u} \geq 0. \end{aligned} \tag{1.5}$$

As restricións do problema primal pasan á función obxectivo do dual penalizadas cos u_i e v_j que se chaman *multiplicadores de Lagrange* ou *variables duais*. A un mesmo problema pódese asociar distintos duais lagrangianos, segundo o conxunto X que consideremos.

O dual lagrangiano busca as penalizacións que equilibran a tensión entre optimalidade e factibilidade, é dicir, os multiplicadores. Estes multiplicadores ou penalizacións son locais e varían con cada óptimo local do primal. Ademais, un multiplicador asociado a unha restrición que non está activa no óptimo é cero.

Por outro lado, a dualidade lagrangiana xeneraliza a dualidade en programación linear. Recordamos

a formulación do problema de programación linear

$$\begin{aligned} & \text{minimizar} && \mathbf{c}^T \mathbf{x} \\ & \text{suxeito a} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq 0. \end{aligned}$$

Tomando $X = \{\mathbf{x} : \mathbf{x} \geq 0\}$ temos o dual lagrangiano:

$$\text{maximizar} \quad \theta(\mathbf{v}) = \inf\{\mathbf{c}^T \mathbf{x} + \mathbf{v}^T(\mathbf{b} - \mathbf{Ax}) : \mathbf{x} \geq 0\} = \mathbf{v}^T \mathbf{b} + \inf\{(\mathbf{c}^T - \mathbf{v}^T \mathbf{A})\mathbf{x} : \mathbf{x} \geq 0\}.$$

Pola segunda igualdade, temos que

$$\theta(\mathbf{v}) = \begin{cases} \mathbf{v}^T \mathbf{b} & \text{si } (\mathbf{c}^T - \mathbf{v}^T \mathbf{A}) \geq 0 \\ -\infty & \text{en outro caso} \end{cases}$$

Entón, o dual lagrangiano pode ser reescrito como

$$\begin{aligned} & \text{maximizar} && \mathbf{v}^T \mathbf{b} \\ & \text{suxeito a} && \mathbf{A}^T \mathbf{v} \leq \mathbf{c}. \end{aligned}$$

Teoremas de dualidade

Imos agora enunciar os principais resultados sobre as relacións que existen entre o problema primal P e o dual D.

Teorema 1.22 (Teorema de dualidade débil). *Dado un par de solucións factibles \mathbf{x} e (\mathbf{u}, \mathbf{v}) dos problemas P e D, respectivamente, entón*

$$f(\mathbf{x}) \geq \theta(\mathbf{u}, \mathbf{v}).$$

O teorema de dualidade débil dinos que o valor da función obxectivo dunha solución factible para o problema dual é unha cota inferior para o valor da función obxectivo de calquera solución factible do problema primal.

Do teorema anterior séguense os seguintes corolarios:

Corolario 1.23. $\inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0\} \geq \sup\{\theta(\mathbf{u}, \mathbf{v}), \mathbf{u} \geq 0\}.$

Corolario 1.24. *Se $f(\bar{\mathbf{x}}) = \theta(\mathbf{u}, \mathbf{v})$, con $\bar{\mathbf{x}}$ factible en P e (\mathbf{u}, \mathbf{v}) factible en D, entón $\bar{\mathbf{x}}$ é óptimo en P e (\mathbf{u}, \mathbf{v}) óptimo en D.*

Noutras palabras, o anterior corolario dinos que se encontramos un par de solucións factibles do problema primal e dual co mesmo valor obxectivo, temos óptimos do primal e dual.

Corolario 1.25. *Se $\inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0\} = -\infty$, entón $\theta(\mathbf{u}, \mathbf{v}) = -\infty, \forall \mathbf{u} \geq 0$.*

Corolario 1.26. *Se $\sup\{\theta(\mathbf{u}, \mathbf{v}), \mathbf{u} \geq 0\} = \infty$, entón o problema P non ten solucións factibles.*

Se a desigualdade do Corolario 1.23 é estricta, dise que existe *dualidade gap*. É dicir, non existe un par primal-dual que garanta que se verifica a optimalidade global.

Moitos algoritmos baséanse en ir reducindo iterativamente a distancia entre as dúas cotas ($f(\mathbf{x})$ cota superior e $\theta(\mathbf{u}, \mathbf{v})$ cota inferior), dado que o gap entre as cotas pode permitir valorar a calidade da solución dada.

Imos ver agora o teorema de dualidade forte. O cal, nos dá as condicións baixo as cales non hai dualidade gap. Baixo hipóteses de convexidade e restricións axeitadas, o teorema asegura que o valores da función obxectivo no óptimo do problema primal e dual coinciden.

Teorema 1.27 (Teorema de dualidade forte). *Sexa $\emptyset \neq X \subset \mathbb{R}^n$ un conxunto convexo, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ convexas e $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^l$ afín, é dicir, $\mathbf{h}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. Se $\mathbf{0} \in \text{int}\mathbf{h}(X)$, onde $\mathbf{h}(X) = \{\mathbf{h}(\mathbf{x}) : \mathbf{x} \in X\}$ e $\hat{\mathbf{x}} \in X$ tal que $\mathbf{g}(\hat{\mathbf{x}}) < \mathbf{0}$ e $\mathbf{h}(\hat{\mathbf{x}}) = \mathbf{0}$ (condición de regularidade). Entón, non hai dualidade gap, é dicir,*

$$\inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\} = \sup\{\theta(\mathbf{u}, \mathbf{v}) : \mathbf{u} \geq \mathbf{0}\}.$$

Ademais, se o ínfimo é finito, logo $\sup\{\theta(\mathbf{u}, \mathbf{v}) : \mathbf{u} \geq \mathbf{0}\}$ alcánzase en $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ con $\bar{\mathbf{u}} \geq \mathbf{0}$. Se o ínfimo se alcanza en $\bar{\mathbf{x}}$, logo $\bar{\mathbf{u}}^\top \mathbf{g}(\bar{\mathbf{x}}) = 0$.

A condición $\bar{\mathbf{u}}^\top \mathbf{g}(\bar{\mathbf{x}}) = 0$ e a condición de regularidade pódense ver como as condicións FC e FD, respectivamente, do Teorema 1.18.

Unha condición necesaria e suficiente para ausencia de dualidade gap é a existencia dun punto de sela. Dado o problema primal P, recordamos que a función lagrangiana se define como

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \mathbf{u}^\top \mathbf{g}(\mathbf{x}) + \mathbf{v}^\top \mathbf{h}(\mathbf{x}).$$

Unha solución $(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{v}})$ dise que é un punto de sela da función lagrangiana anterior se $\bar{\mathbf{x}} \in X$, $\bar{\mathbf{u}} > \mathbf{0}$ e

$$L(\bar{\mathbf{x}}, \mathbf{u}, \bar{\mathbf{v}}) \leq L(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}) \leq L(\mathbf{x}, \bar{\mathbf{u}}, \bar{\mathbf{v}}),$$

para todo $\mathbf{x} \in X$ e todo $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ con $\bar{\mathbf{u}} > \mathbf{0}$.

A existencia dun punto de sela, $(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{v}})$, para a función lagrangiana está relacionada coa solución do problema primal $\bar{\mathbf{x}}$ e do dual $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ en ausencia de dualidade gap. Ademais, baixo convexidade resolver o dual lagrangiano, encontrar un punto de sela e encontrar un punto KKT do problema primal son cousas equivalentes. A formulación e probas deste resultados poden verse no Capítulo 6 de [1].

Propiedades da función dual

Os teoremas enunciados anteriormente dannos condicións baixo as que podemos resolver o problema primal resolvendo o dual. Se hai dualidade gap non se pode facer. Polo que é interesante coñecer as propiedades da función dual para obter a solución do problema dual.

Aquí imos asumir que o conxunto factible X é compacto. E combinamos os vectores \mathbf{u} e \mathbf{v} en \mathbf{w} e as funcións \mathbf{g} e \mathbf{h} en β .

Teorema 1.28. *Sexa $\emptyset \neq X \subset \mathbb{R}^n$ un conxunto compacto, e as función \mathbf{u} e \mathbf{u} continuas. Logo θ definida por*

$$\theta(\mathbf{w}) = \inf\{f(\mathbf{u}) + \mathbf{w}^\top \beta(\mathbf{x}) : \mathbf{x} \in X\}, \quad (1.6)$$

é cóncava sobre \mathbb{R}^{m+l} .

Recordamos que maximizar unha función cóncava é como minimizar unha función convexa, así, o teorema anterior garante que un óptimo local de θ sexa un óptimo global, polo Teorema 1.9. Pero

a función dual non está explicitamente dispoñible, xa que θ só pode ser avaliada nun punto despois de resolver un subproblema de minimización. Así, para maximizar a función dual resulta de axuda apoiarse na diferenciabilidade, nos subgradientes e nas direccións de ascenso da función dual θ .

Consideramos a función dual definida en 1.6. Dado $\mathbf{w} \in \mathbb{R}^{m+l}$, definimos o conxunto de solución óptimas como:

$$X(\mathbf{w}) = \{\mathbf{y} : \mathbf{y} \text{ minimiza } f(\mathbf{x}) + \mathbf{w}^\top \beta(\mathbf{x}) \text{ sobre } \mathbf{x} \in X\}.$$

Teorema 1.29 (Diferenciabilidade da función dual). *Sexa $\emptyset \neq X \subset \mathbb{R}^n$ compacto, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\beta : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ continuas. Sexa $\mathbf{w} \in \mathbb{R}^{m+l}$ tal que $X(\bar{\mathbf{w}}) = \{\bar{\mathbf{x}}\}$, entón θ é diferenciable en $\bar{\mathbf{w}}$ e o seu gradiente é $\nabla\theta(\bar{\mathbf{w}}) = \beta(\bar{\mathbf{x}})$.*

O seguinte teorema dinos que cada $\bar{\mathbf{x}} \in X(\bar{\mathbf{w}})$ proporciona un subgradiente de θ en $\bar{\mathbf{w}}$.

Teorema 1.30 (Subdiferenciabilidade e subgradientes). *Sexa $\emptyset \neq X \subset \mathbb{R}^n$ compacto, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\beta : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ continuas tal que para todo $\mathbf{w} \in \mathbb{R}^{m+l}$, $X(\bar{\mathbf{w}}) \neq \emptyset$. Se $\bar{\mathbf{x}} \in X(\bar{\mathbf{w}})$ entón $\beta(\bar{\mathbf{x}})$ é un subgradiente de θ en $\bar{\mathbf{w}}$.*

Teorema 1.31 (Caracterización de subgradientes). *Sexa $\emptyset \neq X \subset \mathbb{R}^n$ compacto, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\beta : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ continuas. Entón,*

$$\xi \text{ é un subgradiente de } \theta \text{ en } \bar{\mathbf{w}} \in \mathbb{R}^{m+l} \iff \xi \in \text{conv}\{\beta(y) : y \in X(\bar{\mathbf{w}})\}.$$

O resultado anterior serve para caracterizar as direccións de ascenso de θ , é dicir, aquelas direccións nas que θ aumenta o seu valor. Unha dirección de ascenso, \mathbf{d} , forma un ángulo menor de 90° con cada subgradiente. Ademais do anterior, permite caracterizar a dirección de máximo ascenso, que se corresponde co subgradiente de menor norma. É útil na definición de algoritmos de gradiente que se empregan para resolver o dual. A idea principal destes algoritmos é partindo dun punto inicial, determinar unha dirección de ascenso axeitada para θ e facer unha “busca en liña” (*line search*) nesa dirección para determinar a nova solución.

1.5. Algoritmos de resolución para problemas de optimización con restricións

O procedemento de resolución dos problemas de optimización consiste no emprego de algoritmos. Un *algoritmo* é un proceso iterativo, que seguindo unhas instrucións prefixadas, xera unha sucesión de puntos coa intención de que converxan a un punto coas propiedades desexadas. Un algoritmo, pode representarse mediante a chamada *correspondencia algorítmica*, \mathbf{A} , que a partir dun punto inicial \mathbf{x}^1 , xera unha sucesión $\mathbf{x}^1, \mathbf{x}^2, \dots$ onde $\mathbf{x}^{k+1} \in \mathbf{A}(\mathbf{x}^k)$. Cada unha destas aplicacións da correspondencia \mathbf{A} , o paso de \mathbf{x}^k a \mathbf{x}^{k+1} , constitúe unha iteración do algoritmo.

O desexable é que o algoritmo constrúa sucesións de puntos converxentes a un óptimo global do problema. Pero, como xa mencionamos, en problemas de programación non convexa, habitualmente só se aspira a óptimos locais, de maneira que os algoritmos deséñanse de tal maneira que terminen cando se alcancen puntos dun certo conxunto desexable, Ω (óptimos locais, puntos FJ, puntos KKT, puntos onde se anule o gradiente,...). De aí, que unha parte importante no desenvolvemento de algoritmos é asegurar, mediante demostracións matemáticas, que a sucesión xerada polo algoritmo sempre converxe a un punto do conxunto Ω . Aquí, imos omitir os resultados de converxencia que se poden ver en [1]. Unha vez que a converxencia é demostrada teoricamente, á hora de implementar o problema é importante definir un *criterio de parada* que especifique cando termina o algoritmo. A maioría de criterios de parada, apóianse en que toda sucesión converxente é de Cauchy, e por tanto, para todo $\epsilon > 0$ hai un término da sucesión tal que a distancia entre dous términos calquera contiguos é menor que $\epsilon > 0$.

A maioría dos algoritmos de optimización con restricións resolven o problema de partida transformándoo nunha sucesión de problemas sen restricións, xa sexa mediante o uso de multiplicadores de Lagrange e conceptos de dualidade ou mediante métodos de penalización. Unha introdución aos algoritmos de optimización sen restricións pode verse en [6].

Existe un gran abanico de algoritmos para problemas con restricións. Aquí centrámonos nos métodos de penalización, no método lagrangiano aumentado e na programación linear sucesiva. Este último algoritmo é o de principal interese neste traballo, xa que o algoritmo creado dende o ITMATI para resolver o problema de optimización no contexto das redes de transporte de gas, como veremos no seguinte capítulo, consiste nunha variante deste.

1.5.1. Método de penalización exterior

O *método de penalización exterior* consiste en engadir un término de penalización para cada violación das restricións de igualdade ou desigualdade que definen o problema. Así, o problema transfórmase nun problema sen restricións ou nunha sucesión de problemas sen restricións aos que se lles pode dar solución empregando os algoritmos existentes para resolver problemas sen restricións.

Formalmente, dise que un método de penalización exterior consiste en escoller unha sucesión de parámetros $\{\mu^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^+$ tal que $\lim_{k \rightarrow \infty} \mu^k = \infty$ e resolver problemas da forma:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \mu^k \alpha(\mathbf{x})$$

onde $\alpha(\mathbf{x})$ se coñece como *función de penalización* e debe cumprir:

- a) Para todo $\mathbf{x} \in \mathbb{R}^n$, $\alpha(\mathbf{x}) \geq 0$.
- b) $\alpha(\mathbf{x}) = 0$ se e só se $\{h(\mathbf{x}) = 0 \text{ e } g(\mathbf{x}) \leq 0\}$.
- c) $\alpha(\mathbf{x})$ é continua.

En xeral, unha función de penalización axeitada debe incorrer nunha penalización positiva para puntos infactibles e nunha non penalización para puntos factibles. Se as restricións son da forma $g_i(\mathbf{x}) \leq 0$ para $i = 1, \dots, m$ e $h_j(\mathbf{x}) = 0$ para $j = 1, \dots, l$, unha función de penalización α defínese como

$$\alpha(\mathbf{x}) = \sum_{i=1}^m \phi[g_i(\mathbf{x})] + \sum_{j=1}^l \psi[h_j(\mathbf{x})],$$

onde ϕ e ψ son funcións continuas satisfacendo o seguinte

$$\begin{aligned} \phi(y) &= 0 \quad \text{se } y \leq 0 \quad \text{e} \quad \phi(y) > 0 \quad \text{se } y > 0 \\ \psi(y) &= 0 \quad \text{se } y = 0 \quad \text{e} \quad \psi(y) > 0 \quad \text{se } y \neq 0 \end{aligned}$$

Normalmente, ϕ e ψ son da forma

$$\phi(y) = [\text{máx}\{0, y\}]^p \quad \psi(y) = |y|^p$$

onde p é un enteiro positivo. Así, a función de penalización α sería da forma

$$\alpha(\mathbf{x}) = \sum_{i=1}^m [\text{máx}\{0, g_i(\mathbf{x})\}]^p + \sum_{j=1}^l |h_j(\mathbf{x})|^p. \quad (1.7)$$

A idea do método de penalización exterior é sinxela, canto máis grande sexa μ^k máis peso terá a función de penalización $\alpha(\mathbf{x})$ no subproblema de minimización e máis difícil será que un punto non factible no problema orixinal poida ser solución deste subproblema.

Na práctica é habitual subir á función obxectivo unicamente as restricións máis difíciles, chegando a subproblemas penalizados que tamén son problemas de optimización con restricións, pero que son máis sinxelos de resolver dada a natureza das restricións. Neste caso, podemos pensar en que estamos resolvendo problemas da forma:

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{suxeito a: } g_i(\mathbf{x}) \leq 0, \quad \forall i \in I^m, \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad \forall j \in I^l, \\ & \quad \quad \quad \mathbf{x} \in X, \end{aligned} \tag{1.8}$$

onde X é un conxunto definido a partir de restricións “fáceis”. Entón, teríamos subproblemas

$$\min_{\mathbf{x} \in X} f(\mathbf{x}) + \mu^k \alpha(\mathbf{x})$$

onde a función de penalización α unicamente penaliza as infactibilidades orixinadas nas función \mathbf{g} e \mathbf{h} .

Este método podemos relacionalo co problema dual. En certo modo, a resolución do problema dual consiste en encontrar as penalizacións para as cales o problema de penalización toma un valor maior.

Na Figura 1.1 presentamos o esquema xeral dun algoritmo de penalización exterior. Desde o punto de vista teórico, os iterantes da sucesión $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ son independentes entre sí, dado que o problema a resolver só depende do parámetro de penalización.

O algoritmo que resolve os subproblemas define os seus puntos iniciais a partir de solucións obtidas nos pasos anteriores. Ademais, a principal limitación deste algoritmo está na dificultade de resolución dos subproblemas, especialmente a medida que μ^k se vai facendo máis grande.

MÉTODO DE PENALIZACIÓN EXTERIOR

Inicialización:

Escoller a función de penalización $\alpha(\mathbf{x})$ e o parámetro de terminación $\epsilon > 0$. Elixir un punto inicial \mathbf{x}^1 , un parámetro de penalización $\mu^1 > 0$ e un escalar $\beta > 1$. Definir $k = 1$.

Paso principal:

1. Sexa \mathbf{x}^{k+1} unha solución óptima do problema $\min_{\mathbf{x} \in X} f(\mathbf{x}) + \mu^k \alpha(\mathbf{x})$.
2. Se $\mu^k \alpha(\mathbf{x}^{k+1}) < \epsilon$. Fin. Devolvemos \mathbf{x}^{k+1} .
Noutro caso, definimos $\mu^{k+1} = \beta \mu^k$. Substituír k por $k + 1$. E volver ao paso 1.

Figura 1.1: Esquema do método de penalización exterior.

A continuación presentamos o principal resultado que serve de xustificación teórica dos métodos de penalización.

Teorema 1.32. *Supoñamos que as función que definen o problema de optimización dado pola Ecuación (1.8) son continuas e que a sucesión $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$, construída ao aplicarlle o método de penalización exterior, está contida nun subconxunto compacto de X . En particular, cada \mathbf{x}^k , $k > 1$, é un óptimo global do subproblema penalizado na iteración anterior, $k - 1$. Entón:*

- *Todo punto límite da sucesión $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ é un óptimo global do problema $\min_{\mathbf{x} \in X} \alpha(\mathbf{x})$.*
- *Ademais, se o problema orixinal ten algún punto factible, entón todo punto límite de $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ é un óptimo global de dito problema.*

O resultado anterior dinos que se o noso problema é infactible os puntos límite serán aqueles que minimicen a infactibilidade, medida mediante a función $\alpha(\mathbf{x})$. Por outro lado, se o problema ten unha rexión factible non baleira, os puntos límites serán óptimos globais. A demostración deste Teorema pode verse en [1].

Na práctica a implementación deste método ten varias dificultades asociadas:

- A solución do problema de penalización pode estar cerca da solución do problema orixinal escollendo un parámetro de penalización suficientemente grande. Pero valores altos deste parámetro da lugar a problemas mal condicionados (moita diferenza entre os autovalores da matriz hessiana) o que ocasiona varias dificultades prácticas: i) é fácil atoparse con problemas de precisión numérica, o que pode desestabilizar aos algoritmos, ii) a converxencia dos algoritmos pode ser lenta, e iii) a converxencia soe estar garantida para puntos onde se anula o gradiente, que poden non ser bos puntos se temos subproblemas con moitos óptimos locais.
- Non é fácil saber con antelación cal é a sucesión de parámetros de penalización máis axeitada. No caso do algoritmo presentado na Figura 1.1, isto tradúcese a que non é fácil saber cal é o valor axeitado para β .

1.5.2. Método lagrangiano aumentado

Para evitar o mal condicionamento derivado de ter que facer o parámetro de penalización infinitamente grande a medida que nos acercamos ao óptimo, xurdiron funcións de penalización exactas. As cales, tratan de conseguir o óptimo utilizando valores finitos do parámetro de penalización.

A función lagrangiana aumentada é unha función de penalización exacta, que consiste en engadir á lagrangiana orixinal, tratada na Sección 1.4, termos cadráticos para penalizar as restricións de igualdade e desigualdade.

Empezamos traballando con problemas que teñen unicamente restricións de igualdade:

$$\text{Problema (P1):} \quad \text{Minimizar} \{f(\mathbf{x}) : h_j(\mathbf{x}) = 0 \text{ para } j \in I^l\}.$$

Se empregamos unha función de penalización cadrática para obter o óptimo do problema, $f(\mathbf{x}) + \mu \sum_{j=1}^l h_j^2(\mathbf{x})$, necesitaríamos que o termo de penalización tenda a infinito. Se desprazamos a restrición orixinal a $\theta = (\theta_j, j \in I^l)$ e consideramos a función obxectivo penalizada $f(\mathbf{x}) + \mu \sum_{j=1}^l [h_j^2(\mathbf{x}) - \theta_j]^2$ con respecto ao problema no cal os lados dereitos das restricións son perturbados a θ_i dende 0, isto permitiría obter un mínimo do problema orixinal sen necesidade de que o parámetro de penalización tenda a infinito. A forma expandida, da última función obxectivo é $f(\mathbf{x}) + \sum_{j=1}^l 2\mu\theta_j h_j(\mathbf{x}) + \mu \sum_{j=1}^l h_j^2(\mathbf{x}) + \mu \sum_{j=1}^l \theta_j^2$. Denotando $v_j = -2\mu\theta_j$ para $j \in I^l$ e omitindo o termo constante, esta pode reescribirse como:

$$L_\mu^A(\mathbf{x}, \mathbf{v}) = f(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}) + \mu \sum_{j=1}^l h_j^2(\mathbf{x}). \quad (1.9)$$

Pode verse que a medida que o parámetro μ tende a infinito, o desprazamento converge a cero, polo que no límite estaríamos resolvendo o Problema (P1).

A función da Ecuación 1.9 é a función lagrangiana máis un termo de penalización cadrático, de aí o nome de lagrangiana aumentada. Esta pode verse como a función obxectivo que obteríamos ao

aplicar unha penalización cadrática usual con respecto ao problema de minimizar a función lagrangiana do Problema (P1):

$$\text{Problema (P2):} \quad \text{Minimizar} \left\{ f(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}) : h_j(\mathbf{x}) = 0 \text{ para } j \in I^l \right\}.$$

Alternativamente, tamén se pode ver a Ecuación 1.9 como a función lagrangiana da versión penalizada do problema (P1):

$$\text{Problema (P3):} \quad \text{Minimizar} \left\{ f(\mathbf{x}) + \mu \sum_{j=1}^l h_j^2(\mathbf{x}) : h_j(\mathbf{x}) = 0 \text{ para } j \in I^l \right\}.$$

Supoñamos agora que temos un punto que é un par primal-dual óptimo $(\bar{\mathbf{x}}, \bar{\mathbf{v}})$ ou, equivalentemente $\bar{\mathbf{x}}$ é un punto KKT do problema (P1) con multiplicadores de Lagrange asociados ás restricións de igualdade dados por $\bar{\mathbf{v}}$. Entón cúmprese que:

$$\nabla L_{\mu}^A(\bar{\mathbf{x}}, \bar{\mathbf{v}}) = \nabla f(\bar{\mathbf{x}}) + \sum_{j=1}^l \bar{v}_j \nabla h_j(\bar{\mathbf{x}}) + 2\mu \sum_{j=1}^l h_j(\bar{\mathbf{x}}) \nabla h_j(\bar{\mathbf{x}}) = \mathbf{0}.$$

A última igualdade é certa independentemente do valor da penalización μ . Por outro lado, traballando co método de penalización exterior teríamos que o gradiente da función penalizada en $\bar{\mathbf{x}}$ viría dado por $\nabla f(\bar{\mathbf{x}}) + \mu \sum_{j=1}^l h_j(\bar{\mathbf{x}}) \nabla h_j(\bar{\mathbf{x}}) = \nabla f(\bar{\mathbf{x}})$, que só se anula se $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$, o que fai en que xeral $\bar{\mathbf{x}}$ non sexa un óptimo do problema penalizado para ningún valor finito de μ .

Consideracións teóricas

A continuación amosamos o resultado no que se sustenta o método lagrangiano aumentado, que fai uso da condición suficiente de KKT de segunda orde (ver Teorema 1.20).

Teorema 1.33. *Supoñamos que $\bar{\mathbf{x}}$ é un punto KKT con multiplicadores $\bar{\mathbf{v}}$ que cumpre a condición suficiente de KKT de segunda orde. Entón existe $\bar{\mu} > 0$ tal que, para todo $\mu > \bar{\mu}$, a función $L_{\mu}^A(\cdot, \bar{\mathbf{v}})$ ten un mínimo local estrito en $\bar{\mathbf{x}}$.*

Se non se verifica a condición suficiente de segunda orde, non existe ningún valor finito de μ que permita obter un valor óptimo $\bar{\mathbf{x}}$ para o Problema (P1). Ademais, de cara ao deseño dun algoritmo para atopar un óptimo de (P1), o resultado anterior garante que non é necesario facer tender μ a infinito e que manteremos as condicións de diferenciabilidade do problema (P1) nos subproblemas. Pero existe un problema que debe resolver o algoritmo, e é que non coñecemos o valor de $\bar{\mathbf{v}}$. Entón, o algoritmo ten que ir actualizando iteración a iteración non só o valor de $\bar{\mathbf{x}}$, senón tamén o valor de $\bar{\mathbf{v}}$, o multiplicador de Lagrange asociado as restricións de igualdade.

Incorporando restricións de desigualdade

Supoñamos que temos agora un problema con restricións de desigualdade da forma

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) \\ &\text{suxeito a} && g_i(\mathbf{x}) \leq 0 \quad i \in I^m. \end{aligned}$$

As restricións de desigualdades pódense transformar en igualdades sen máis que engadir variables auxiliares z_i e poñer as restricións como $h_i(\mathbf{x}) = g_i(\mathbf{x}) + z_i^2 = 0$ para $i \in I^m$. Estas restricións teñen o inconveniente de que a derivada con respecto a z_i se anula cando $z_i = 0$ o que dificultará que o problema resultante teña boas propiedades de regularidade. Na práctica, esta transformación adóitase facer engadindo directamente as variables z_i ás restricións e esixindo que $z_i > 0$. Considerando o problema de minimizar $f(\mathbf{x})$ con estas restricións desprazadas a $-u_i/2\mu$, o lagrangiano aumentado queda da forma

$$L_\mu^A(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mu \sum_{i=1}^m (h_i(\mathbf{x}) + \frac{u_i}{2\mu})^2 + c = f(\mathbf{x}) + \mu \sum_{i=1}^m (g_i(\mathbf{x}) + z_i^2 + \frac{u_i}{2\mu})^2 + c, \quad (1.10)$$

onde c engloba os termos constantes.

O método lagrangiano aumentado ten como paso principal a minimización en \mathbf{x} de $L_\mu^A(\mathbf{x}, \mathbf{u})$ para distintos valores de \mathbf{u} . Neste caso, dado un vector \mathbf{x} teremos que

$$z_i^2 = \begin{cases} 0 & g_i(\mathbf{x}) + \frac{u_i}{2\mu} \geq 0 \\ -g_i(\mathbf{x}) - \frac{u_i}{2\mu} & \text{noutro caso} \end{cases}$$

Por tanto, cada término $(g_i(\mathbf{x}) + z_i^2 + \frac{u_i}{2\mu})^2$ tomará o valor $\max\{0, g_i(\mathbf{x}) + \frac{u_i}{2\mu}\}^2$. Entón, para traballar con restricións de desigualdade basta tomar

$$L_\mu^A(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mu \sum_{i=1}^m \max\{0, g_i(\mathbf{x}) + \frac{u_i}{2\mu}\}^2 + c.$$

Entón, para o problema de minimizar $f(\mathbf{x})$ suxeito a restricións de igualdade e desigualdade, temos que a función lagrangiana aumentada é da seguinte forma:

$$L_\mu^A(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \mu \left(\sum_{i=1}^m \max\{0, g_i(\mathbf{x}) + \frac{u_i}{2\mu}\}^2 + \sum_{j=1}^l (h_j(\mathbf{x}) + \frac{v_j}{2\mu})^2 \right) + c.$$

Descrición do método lagrangiano aumentado

Consideramos que queremos resolver un problema de programación no linear

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) \\ &\text{suxeito a} && g_i(\mathbf{x}) \leq 0 \quad i \in I^m. \\ &&& h_j(\mathbf{x}) = 0 \quad j \in I^l. \end{aligned}$$

A idea do método é a seguinte. En cada iteración k teremos un parámetro de penalización μ^k e aproximacións \mathbf{u}^k e \mathbf{v}^k dos multiplicadores de Lagrange asociados as restricións de desigualdade e igualdade, respectivamente, que estamos buscando. Logo, buscamos o mínimo en \mathbf{x} do lagrangiano aumentado $L_{\mu^k}^A(\mathbf{x}, \mathbf{u}^k, \mathbf{v}^k)$, que esquecéndonos do termo constante c é equivalente a minimizar

$$f(\mathbf{x}) + \mu^k \left(\sum_{i=1}^m \max\{0, g_i(\mathbf{x}) + \frac{u_i^k}{2\mu^k}\}^2 + \sum_{j=1}^l (h_j(\mathbf{x}) + \frac{v_j^k}{2\mu^k})^2 \right).$$

Supoñamos que \mathbf{x}^{k+1} é un mínimo deste problema de minimización. Na práctica, o problema de minimización da iteración k resolverase empregando, por exemplo, un método de Newton, cuasi-Newton ou gradiente conxugado (dependendo das características do problema). Ademais, este método de optimización sen restricións sóese resolver tomando como solución inicial \mathbf{x}^k , a solución da iteración anterior, o que pode mellorar a velocidade do algoritmo. Para o mínimo \mathbf{x}^{k+1} tense que $\nabla_x L_{\mu^k}^A(\mathbf{x}^{k+1}, \mathbf{u}, \mathbf{v}) = 0$ co que

$$f(\mathbf{x}^{k+1}) + \sum_{i=1}^m \max\{0, u_i^k + \mu^k g_i(\mathbf{x}^{k+1})\} \nabla g_i(\mathbf{x}^{k+1}) + \sum_{j=1}^l (v_j^k + h_j(\mathbf{x}^{k+1})) \nabla h_j(\mathbf{x}^{k+1}) = 0.$$

A condición anterior é equivalente a condición KKT (ver Teorema 1.18) do problema (P1) de partida. Non obstante, verifícase respecto a uns “multiplicadores” distintos de \mathbf{u}^k e \mathbf{v}^k que non se coñecían de antemán, pois dependen de \mathbf{x}^{k+1} .

O método lagrangiano aumentado actualizará os valores de cada u_i^k a $u_i^{k+1} = \max\{0, u_i^k + 2\mu^k g_i(\mathbf{x}^{k+1})\} \geq 0$ e os de cada v_j^k a $v_j^{k+1} = v_j^k + 2\mu^k h_j(\mathbf{x}^{k+1})$. Esta actualización, pode ser xustificada a partir das propiedades da función dual. Recordamos que a función dual vén dada por:

$$\theta(\mathbf{u}, \mathbf{v}) = \inf\{f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}) : \mathbf{x} \in X\},$$

e o Teorema 1.29 dinos que dado un óptimo $\bar{\mathbf{x}}$ de $\theta(\mathbf{u}, \mathbf{v})$, o vector $s = (\mathbf{g}(\bar{\mathbf{x}}), \mathbf{h}(\bar{\mathbf{x}}))$ é un subgradiente de $\theta(\mathbf{u}, \mathbf{v})$. Por tanto, a actualización dos multiplicadores pode verse como un paso na dirección do subgradiente da función dual.

Ademais de actualizar os multiplicadores para intentar que se acerquen ao seu valor óptimo buscado, temos que actualizar o parámetro μ^k para que os iterantes \mathbf{x}^k se vaian acercando á rexión factible.

Unha das claves do método lagrangiano aumentado é que non é necesario actualizar o parámetro de penalización en todas as iteracións. O que se fai é mirar se na última iteración se viola o cumprimento das condicións de factibilidade e complementariedade, e sendo así actualízase o parámetro μ . Entón, necesítase unha función que mida dalgunha maneira o grado de violación destas condicións. Para as restricións de igualdade sóese considerar $\|\mathbf{h}(\mathbf{x})\|$ e para as restricións de desigualdade tómasse $\|V^{k+1}(\mathbf{x}^{k+1})\|$, onde para cada restrición i , a medida de violación $V_i^{k+1}(\mathbf{x}^{k+1})$ ven dada por

$$V_i^{k+1}(\mathbf{x}^{k+1}) = \max\{g_i(\mathbf{x}^{k+1}), \frac{-u_i^k}{2\mu^k}\}.$$

Na Figura 1.2 presentamos o método lagrangiano aumentado que recolle todas as consideracións mencionadas. Comentar que se introducen os parámetros \mathbf{u}^{\max} , \mathbf{v}^{\min} e \mathbf{v}^{\max} para asegurar que os multiplicadores permanecen acoutados e que a penalización μ pode acabar dominándoos no caso de ser necesario. Por outra banda, os parámetros ϵ_1, ϵ_2 e ϵ_3 empregados no criterio de parada poden ser distintos.

Teorema 1.34. *Se aplicamos o método lagrangiano aumentado ao problema P1, entón as sucesións xeradas teñen as seguintes propiedades:*

- *Os puntos de acumulación da sucesión $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ son puntos KKT do seguinte problema:
minimizar $\sum_{i=1}^m \max\{0, g_i(\mathbf{x})\}^2 + \sum_{j=1}^l h_j^2(\mathbf{x})$.*
- *Se un punto de acumulación $\bar{\mathbf{x}}$ é factible e os vectores $\nabla g_i(\bar{\mathbf{x}})$, con $i \in I(\bar{\mathbf{x}})$, e $\nabla h_j(\bar{\mathbf{x}})$ con $j \in I^l$ son linearmente independentes, entón $\bar{\mathbf{x}}$ é un punto KKT e as sucesións $\{\mathbf{u}^k\}_{k \in \mathbb{N}}$ e $\{\mathbf{v}^k\}_{k \in \mathbb{N}}$ converxen aos multiplicadores de Lagrange asociados con $\bar{\mathbf{x}}$.*

- Se ademais en $\bar{\mathbf{x}}$ se verifica a condición suficiente de KKT de segunda orde, entón a sucesión $\{\mu^k\}_{k \in \mathbb{N}}$ está acoutada.

MÉTODO LAGRANGIANO AUMENTADO

Inicialización:

Elixir un parámetro de terminación $\epsilon > 0$. Elixir un parámetro inicial $\mu^1 > 0$ e escalares $\delta \in (0, 1)$ e $\gamma > 1$. Elixir un punto inicial \mathbf{x}^1 e valores $\mathbf{u}^1 \geq 0$ e \mathbf{v}^1 , $\mathbf{u}^{\text{máx}}$, $\mathbf{v}^{\text{mín}}$ e $\mathbf{v}^{\text{máx}}$ para os multiplicadores. Definir $k = 1$.

PASO 1. RESOLUCIÓN SUBPROBLEMA

Definir \mathbf{x}^{k+1} como unha solución óptima do problema $\min_{\mu^k} L_{\mu^k}^A(\mathbf{x}, \mathbf{u}^k, \mathbf{v}^k)$. Ir ao PASO 2.

PASO 2. ACTUALIZACIÓN DE PARÁMETROS

Calcular, para cada $i \in I^m$, $V_i^{k+1}(\mathbf{x}^{k+1}) = \max\{g_i(\mathbf{x}^{k+1}), -\frac{u_i^k}{\mu^k}\}$.

- Se $\max\{\|\mathbf{V}(\mathbf{x}^{k+1})\|, \|\mathbf{h}(\mathbf{x}^{k+1})\|\} \leq \delta \max\{\|\mathbf{V}(\mathbf{x}^k)\|, \|\mathbf{h}(\mathbf{x}^k)\|\}$, entón $\mu^{k+1} = \mu^k$.
- Noutro caso, $\mu^{k+1} = \gamma \mu^k$.

Para cada $i \in I^m$, sexa $u_i^{k+1} = \min\{u_i^{\text{máx}}, u_i^k + \mu^k g_i(\mathbf{x}^{k+1})\}$. Para cada $j \in I^l$:

- Se $v_j^k + \mu^k h_j(\mathbf{x}^{k+1}) \geq 0$, $v_j^{k+1} = \min\{v_j^{\text{máx}}, v_j^k + \mu^k h_j(\mathbf{x}^{k+1})\}$.
- Noutro caso, $v_j^{k+1} = \max\{v_j^{\text{mín}}, v_j^k + \mu^k h_j(\mathbf{x}^{k+1})\}$.

Ir ao PASO 3.

PASO 3. CRITERIO DE PARADA

Calculamos os seguintes parámetros:

$$\text{Factibilidade dual KKT: } C_1 = \nabla f(\mathbf{x}^{k+1}) + \sum_{i=1}^m u_i^{k+1} \nabla g_i(\mathbf{x}^{k+1}) + \sum_{j=1}^l v_j^{k+1} \nabla h_j(\mathbf{x}^{k+1}).$$

$$\text{Folguas complementarias: } C_2 = \max_{i \in I^m} \{|\min\{-g_i(\mathbf{x}^{k+1}), u_i^{k+1}\}|\}.$$

$$\text{Factibilidade primal: } C_3 = \max\{\max_{i \in I^m} g_i(\mathbf{x}^{k+1}), \max_{j \in I^l} |h_j(\mathbf{x}^{k+1})|\}.$$

- Se $C_1 < \epsilon$, $C_2 < \epsilon$ e $C_3 < \epsilon$, FIN. Devolvemos \mathbf{x}^{k+1} e os multiplicadores \mathbf{u}^{k+1} e \mathbf{v}^{k+1} .
- Noutro caso, substituír k por $k + 1$. Volver ao PASO 1.

Figura 1.2: Esquema do método lagrangiano aumentado.

1.5.3. Programación linear sucesiva

Outra técnica empregada para a resolución de problemas non lineais é a programación linear sucesiva, a cal designaremos como SLP (successive linear programming). A grandes trazos consiste en resolver, en cada iteración k , unha aproximación linear do problema orixinal entorno a un punto. Esta aproximación perde precisión a medida que nos afastamos do punto, polo que, os algoritmos baseados na SLP restrinxen a procura da solución a unha rexión de confianza entorno ao punto onde se lineariza o problema.

Unha das primeiras referencias publicadas referentes ao SLP foi o traballo de Griffith e Stewart desenvolvido en Shell Development Company en 1961, para dar solución a problemas relacionados coa industria petroleira, nel descríbese un método que denominaron Mathematical Approximation

Programming (MAP). Este traballo, non contén detalles do algoritmo, pero dálle importancia a imposición dunha rexión de confianza para evitar ciclos e asegurar a converxencia. Desde entón, os métodos baseados no SLP son aplicados a problemas relacionados coa industria química e petroleira.

En [14] podemos encontrar a primeira formulación matemática do SLP, que foi chamado RSLP (reject successive linear programming), a descrición detallada do algoritmo e propiedades teóricas sobre a converxencia. O problema non linear que empregan só está definido por restricións de igualdade e restricións de cota e ademais para obter a linearización do problema non linear orixinal empregan aproximacións de Taylor, polo que as funcións que definen o problema deben ser diferenciables.

Aquí consideraremos o problema de programación non linear NLP, tanto con restricións de igualdade como de desigualdade, definido como:

$$\begin{aligned}
 & \text{minimizar } f(\mathbf{x}) \\
 & \text{suxeito a: } g_i(\mathbf{x}) \leq 0, & i \in I^m \\
 & h_j(\mathbf{x}) = 0, & j \in I^l \\
 & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}
 \end{aligned} \tag{1.11}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ e $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ se asumen continuamente diferenciables e todas as restricións lineais se recollen no conxunto X .

Definición 1.35. Consideremos o problema de programación non linear NLP definido como 1.11. A aproximación de Taylor de orde un de NLP ao redor do punto $\bar{\mathbf{x}}$ con paso $\mathbf{s} \in \mathbb{R}^n$, TNLP($\bar{\mathbf{x}}, \mathbf{s}$), é o problema de programación linear seguinte:

$$\begin{aligned}
 & \text{minimizar } f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) \\
 & \text{suxeito a:} \\
 & g_i(\bar{\mathbf{x}}) + \nabla g_i(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) \leq 0, \quad i \in I^m \\
 & h_j(\bar{\mathbf{x}}) + \nabla h_j(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) = 0, \quad j \in I^l \\
 & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\} \\
 & -\mathbf{s} \leq \mathbf{x} - \bar{\mathbf{x}} \leq \mathbf{s} \qquad \text{Rexión de confianza}
 \end{aligned}$$

A función obxectivo pode substituírse por $f(\bar{\mathbf{x}})^\top \mathbf{x}$ pois $f(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}})^\top \bar{\mathbf{x}}$ é constante e ao resolver o problema de minimización o conxunto de óptimos é o mesmo.

No problema anterior, $\mathbf{d} = \mathbf{x} - \bar{\mathbf{x}}$ é a dirección de descenso na que nos movemos partindo do punto $\bar{\mathbf{x}}$ e impoñemos que o paso \mathbf{d} está limitado por \mathbf{s} , é dicir, movémonos dentro dunha rexión de confianza. Denotaremos por TNLP($\bar{\mathbf{x}}$) ao problema linear resultante da aproximación linear de primeira orde do problema NLP ao redor do punto $\bar{\mathbf{x}}$ sen incluír a restrición da rexión de confianza.

O algoritmo SLP consiste en: partindo dun punto dado $\mathbf{x}^0 \in \mathbb{R}^n$ e unha cota de paso $\mathbf{s} \in \mathbb{R}^n$ resolver o problema TNLP(\mathbf{x}^0, \mathbf{s}) obtendo unha solución \mathbf{x}^1 . Esta solución acéptase se verifica certo criterio e, nese caso, tómase $\mathbf{x}^0 = \mathbf{x}^1$, increméntase a cota de paso \mathbf{s} e repítese o proceso. Se o punto \mathbf{x}^1 non verifica o criterio considerado, rexéitase e vólvese a resolver o problema TNLP(\mathbf{x}^0, \mathbf{s}) pero cunha cota de paso \mathbf{s} menor. Normalmente, considérase como criterio de parada a condición de que a distancia entre dous iterantes sucesivos, $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|$, sexa suficientemente pequena.

Imos ver agora, unha versión do SLP que consiste en modificar o problema non linear, penalizando as restricións non lineais na función obxectivo, como se facía nos métodos de penalización que describimos anteriormente. Esta nova versión introdúcese en [1] co nome de *programación linear sucesiva penalizada* (PSLP).

Antes de detallar o pseudocódigo do algoritmo PSLP é necesario introducir algúns conceptos previos. Por simplicidade asúmese o mesmo parámetro de penalización para as restricións de igualdade e desigualdade.

Definición 1.36. Dado o problema non linear NLP, o problema penalizado PNP con parámetro de penalización $\mu > 0$, defínese como

$$\begin{aligned} \text{minimizar} \quad & p(\mathbf{x}) = f(\mathbf{x}) + \mu \left[\sum_{i=1}^m \max\{g_i(\mathbf{x}), 0\} + \sum_{j=1}^l |h_j(\mathbf{x})| \right] \\ \text{suxeito a} \quad & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\} \end{aligned} \quad (1.12)$$

Observamos que o problema penalizado PNP consiste nunha modificación do problema non linear NLP. Elimínanse as restricións non lineais e aplícase unha penalización na función obxectivo aos puntos que as violan. A función de penalización é da forma 1.7 con $p = 1$ e recibe o nome de función de penalización absoluta.

Teorema 1.37. *Sexa NLP un problema non linear e PNP o problema penalizado asociado con parámetro de penalización μ . As seguintes afirmacións son certas:*

1. *Se $\bar{\mathbf{x}}$ é un punto KKT de NLP con multiplicadores de Lagrange (u_i, v_j) para $i \in I^m$, $j \in I^l$ tales que $\mu_i > |u_i|$ para $i \in I^m$ e $\bar{\mu}_j > |v_j|$ para $j \in I^l$, entón $\bar{\mathbf{x}}$ é un punto KKT para o problema PNP.*
2. *Se $\bar{\mathbf{x}}$ é un punto KKT para PNP e $\bar{\mathbf{x}}$ é factible para NLP, entón $\bar{\mathbf{x}}$ é un punto KKT de NLP.*

No anterior teorema estamos considerando que o parámetro de penalización μ pode verse como un vector $(\mu_i, \bar{\mu}_j)$ de $m + l$ variables de maneira que podemos asociar a cada restrición unha penalización distinta.

O obxectivo agora é resolver o problema PNP en lugar da linearización do problema orixinal NLP. O algoritmo que se emprega para isto é a programación linear sucesiva penalizada. Para a súa aplicación, precisamos definir a aproximación linear de primeira orde da función obxectivo do problema PNP ao redor do punto $\bar{\mathbf{x}}$:

$$p_L(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) + \mu \left[\sum_{i=1}^m \max\{g_i(\bar{\mathbf{x}}) + \nabla g_i(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}), 0\} + \sum_{j=1}^l |h_j(\bar{\mathbf{x}}) + \nabla h_j(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}})| \right]$$

Definición 1.38. Dado o problema penalizado PNP con parámetro de penalización μ , a aproximación linear de primeira orde ao redor do punto $\bar{\mathbf{x}}$ con paso \mathbf{s} defínese como o problema TPNP($\bar{\mathbf{x}}, \mathbf{s}$):

$$\begin{aligned} \text{minimizar} \quad & p_L(\mathbf{x}) \\ \text{suxeito a} \quad & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\} \\ & -\mathbf{s} \leq \mathbf{x} - \bar{\mathbf{x}} \leq \mathbf{s}. \end{aligned}$$

O problema TPNP($\bar{\mathbf{x}}, \mathbf{s}$) sen termos non diferenciables (valor absoluto e función máximo) trátase dun problema de programación linear que se formula como segue:

$$\begin{aligned}
 \text{minimizar} \quad & p_L(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) + \mu \left[\sum_{i=1}^m \max\{\mathbf{y}_i\} + \sum_{j=1}^l (\mathbf{z}_j^+ + \mathbf{z}_j^-) \right] \\
 \text{suxeito a} \quad & g_i(\bar{\mathbf{x}}) + \nabla g_i(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) \leq \mathbf{y}_i, \quad i \in I^m \\
 & h_j(\bar{\mathbf{x}}) + \nabla h_j(\bar{\mathbf{x}})^\top (\mathbf{x} - \bar{\mathbf{x}}) = \mathbf{z}_j^+ - \mathbf{z}_j^-, \quad j \in I^l \\
 & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}, \\
 & -\mathbf{s} \leq \mathbf{x} - \bar{\mathbf{x}} \leq \mathbf{s}, \\
 & \mathbf{y}_i, \mathbf{z}_j^+, \mathbf{z}_j^- \geq 0 \quad i \in I^m, j \in I^l
 \end{aligned}$$

A continuación, na Figura 1.3 describimos o pseudocódigo da PSLP. Na práctica, dado un parámetro pequeno ϵ , o criterio de parada substitúese por $\Delta p_L < \epsilon$.

Algoritmo PSLP

- 1: Inicialización: seleccionar $\mathbf{x}^0 \in X$ factible coas restricións que definen a rexión de confianza, e o vector de paso $\mathbf{s}^0 \in \mathbb{R}^n$. Fixar o parámetro de penalización μ , os escalares $0 < \rho_0 < \rho_1 < \rho_2 < 1$, $0 < \beta < 1$ para o test da rexión de confianza e $\alpha > 0$ o menor límite de tolerancia para \mathbf{s}^0 . Sexa $k = 0$.
- 2: Resolver TPNLP($\mathbf{x}^k, \mathbf{s}^k$). Sexa $\bar{\mathbf{x}}^k$ a solución.
- 3: Calcular $\Delta p(\mathbf{x}) = p(\mathbf{x}^k) - p(\bar{\mathbf{x}}^k)$ e $\Delta p_L(\mathbf{x}) = p_L(\mathbf{x}^k) - p_L(\bar{\mathbf{x}}^k)$.
- 4: **if** $\Delta p_L(\mathbf{x}) = 0$ **then** STOP: Devolver $\bar{\mathbf{x}}^k$
- 5: **else** Calcular $r_k = \frac{\Delta p}{\Delta p_L}$
- 6: **end if**
- 7: **if** $r_k < \rho_0$ **then** $\mathbf{x}^{k+1} = \mathbf{x}^k$ e $\mathbf{s}^{k+1} = \beta \mathbf{s}^k \rightarrow$ Ir ao paso 2.
- 8: **else** $\mathbf{x}^{k+1} = \bar{\mathbf{x}}^k$ e actualizamos $\mathbf{s}^{k+1} = \begin{cases} \beta \mathbf{s}^k & \text{se } r_k < \rho_1 \\ \frac{\mathbf{s}^k}{\beta} & \text{se } r_k > \rho_2 \\ \mathbf{s}^k & \text{resto} \end{cases}$
- 9: **end if**
- 10: Tomar $\mathbf{s}^{k+1} = \max\{\mathbf{s}^{k+1}, \alpha\}$, $k = k + 1 \rightarrow$ Ir ao paso 2.

Figura 1.3: Esquema do algoritmo PSLP.

A sucesión $\{\mathbf{x}^k\}$ obtida co algoritmo PSLP non é sempre converxente, pero tense que calquera subsucesión converxente o fai a un punto KKT do problema penalizado PNP. A proba deste resultado pode verse en [1].

Teorema 1.39. *Sexa $\{\mathbf{x}^k\}$ a sucesión obtida ao aplicar o algoritmo PSLP ao problema PNP. Se o conxunto $C = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \in X, p(\mathbf{x}) \leq p(\mathbf{x}^0)\}$ é acoutado, entón calquera punto de acumulación de $\{\mathbf{x}^k\}$ é un punto KKT de PNP. Se ademais o punto límite é factible para NLP, entón é un punto KKT de NLP.*

Capítulo 2

Problema de optimización en redes de gas (determinista)

Neste capítulo imos falar do problema determinista no contexto das redes de transporte de gas e do algoritmo implementado dende o ITMATI para a súa resolución. Na primeira sección preséntase a formulación do problema de optimización determinista no contexto das redes de transporte de gas. Para isto empregouse [11] e [12]. Centrarémonos principalmente no problema simplificado, é dicir, sen introducir ningún elemento que requira o uso de variables binarias que converten ao problema de programación non linear nun problema de programación non linear mixto-enteiro. Na segunda sección introdúcese o algoritmo implementado dende o ITMATI para a resolución do problema anterior, a programación linear sucesiva sen rexión de confianza. Os resultados teóricos referentes a este algoritmo preséntanse en [8].

2.1. Problema de optimización non linear no contexto das redes de transporte de gas

A rede de transporte de gas pódese modelar mediante un grafo dirixido $G = (N, A)$, onde N representa o conxunto de nodos e A o conxunto de arestas. Cada elemento de A é un par ordenado de elementos de N . Os nodos poden ser puntos de subministración, N^s , puntos de consumo, N^c , almacéns subterráneos, puntos de interconexión entre as tubaxes de gas, etc. En canto ás arestas, que representan os gasodutos da rede, téñense os seguintes tipos: A^n (tubaxe da rede de gas), A^c (estacións de compresión), A^{pcv} (arestas con válvulas de control da presión), A^{vc} (arestas con válvulas de peche) e A^{vr} (arestas con válvulas de regulación).

A continuación descríbense os parámetros principais e as variables que interveñen no modelado do problema.

Os parámetros asociados a cada nodo $i \in N$ son:

- Presión mínima e máxima: \underline{p}_i e \bar{p}_i .
- Consumo ou capacidade fixo: c_i . Se é positivo denota consumo senón emisión.
- Altura á que está situado: h_i .
- Temperatura: T_i (Suponse constante igual a 15°).

Por outro lado, os parámetros asociados a cada aresta $a \in A$ son os seguintes:

- Caudal mínimo e máximo: q_a, \bar{q}_a .
- Lonxitude: L_a .
- Diámetro: D_a .

As variables que se consideran no problema serán:

- A presión [Pa] nos nodos: $p_i, i \in N$. De feito, considerarase a presión ao cadrado [Pa²], p_i^{sq} , xa que permite eliminar unha parte da non linearidade das restricións de perda de gas, que veremos máis adiante.
- O caudal máscico [kg/s]: $q_a, a \in A$. Denotando por $a = (s, t) \in A$ a aresta que vai do nodo s ao nodo t , q_a será positivo se o fluxo é na dirección marcada pola aresta e negativo se vai en dirección oposta (de t a s).

O problema de optimización determinista co que traballan consiste en minimizar o gas consumido polas estacións de compresión g_a . A función dunha estación de compresión é elevar a presión do gas, que se perde ao recorrer grandes distancias, co fin de subministrar a enerxía necesaria para o seu transporte. Para levar a cabo a súa función, a estación de compresión fai uso do propio gas que circula pola rede, polo que interesa reducir o seu consumo para así poder satisfacer a demanda.

Por outra banda, as restricións que interveñen no problema son:

- Restricións de cota: tanto a presión nos nodos como o caudal nas tubaxes, teñen unha cota mínima e máxima que non poden exceder.

$$\begin{aligned} p_i^{sq} \leq p_i^{sq} \leq \bar{p}_i^{sq} \quad \forall i \in N \\ q_a \leq q_a \leq \bar{q}_a \quad \forall a \in A \end{aligned}$$

- Restricións de conservación de fluxo: establecen que, a suma dos caudais máscicos entrantes debe ser igual á suma dos caudais máscicos saíntes.

$$\underline{c}_i \leq \sum_{a \in A_i^{ini}} q_a - \sum_{a \in A_i^{fin}} q_a \leq \bar{c}_i \quad \forall i \in N^S$$

Se o nodo i representa unha planta de regasificación terase que $\underline{c}_i = 0$ e \bar{c}_i será igual a capacidade da planta (c_i). Se se trata dun nodo i de consumo, $\underline{c}_i = \bar{c}_i < 0$.

- Restricións de perda de presión: A perda de presión débese ao rozamento do gas coas tubaxes polas que circula. Está relacionada tanto coa lonxitude L_a como co diámetro D_a da tubaxe:

$$p_i^{sq} - p_j^{sq} = K_1 \frac{L_a}{D_a} |q_a| q_a + K_2 \frac{p_i^{sq} + p_j^{sq}}{2} (h_j - h_i) \quad \forall a = (i, j) \in A^n$$

onde K_1 e K_2 son constantes que engloban as propiedades do gas que circula pola aresta como a temperatura, a constante universal do gas, a constante de gravitación universal, o factor de fricción e o factor de compresión.

- Restricións de compresión: as estacións de compresión necesitan unha fonte de enerxía e xeralmente abastécense do propio gas que circula pola rede, como xa se mencionou. Este gas consumido pode calcularse como segue:

$$g_a = K_3 \frac{\gamma}{\gamma - 1} \left(\left(\frac{p_j}{p_i} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_a \quad \forall a = (i, j) \in A^c$$

onde K_3 é unha constante calculada a partir do factor de eficiencia, do poder calorífico do gas, do factor de compresión do gas, da constante universal do gas e da temperatura, e γ o coeficiente adiabático do gas.

Ademais, as estacións de compresión permiten un incremento da presión na dirección do fluxo do gas, isto é:

$$p_j^{sq} \geq p_i^{sq} \quad \forall a = (i, j) \in A^c$$

Nas expresións das restricións de perda de presión e nas restricións de compresión obsérvase a non linearidade que converten ao problema de optimización resultante nun problema non linear.

Resumindo, o problema de optimización de redes de transporte de gas, será o problema de programación non linear non convexo seguinte:

Función obxectivo:

$$\text{mín} \sum_{a \in A^c} g_a$$

Suxeito a:

$$\begin{aligned} p_i^{sq} &\leq p_i^{sq} \leq \bar{p}_i^{sq} && \forall i \in N \\ q_a &\leq q_a \leq \bar{q}_a && \forall a \in A \\ c_i &\leq \sum_{a \in A_i^{ini}} q_a - \sum_{a \in A_i^{fin}} q_a \leq \bar{c}_i && \forall i \in N \\ p_i^{sq} - p_j^{sq} &= K_1 \frac{L_a}{D_a} |q_a| q_a + K_2 \frac{p_i^{sq} + p_j^{sq}}{2} (h_j - h_i) && \forall a = (i, j) \in A^n \\ g_a &= K_3 \frac{\gamma}{\gamma - 1} \left(\left(\frac{p_j}{p_i} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_a && \forall a = (i, j) \in A^c \\ p_j^{sq} - p_i^{sq} &\geq 0 && \forall a = (i, j) \in A^c \end{aligned}$$

Para resolver este problema poderíase empregar o algoritmo de programación linear sucesiva tratado na Sección 1.5.3 do Capítulo 1. Pero, a formulación anterior é unha simplificación do problema xeral, xa que, ademais das restricións detalladas anteriormente, hai elementos na rede que para o seu modelado se precisa facer uso de variables binarias, como por exemplo as válvulas de control do fluxo do gas, que engaden máis restricións ao problema. Así, pasaríase a ter un problema de programación non linear mixto-enteiro (MINLP). Existen diferentes tipos de válvulas de control como as válvulas de control da presión ou as válvulas de peche. Aquí, amósanse como serían as restricións para as válvulas de control da presión. Estas, permiten reducir a presión do gas na dirección de fluxo polas tubaxes. Para unha válvula $v = (i, j) \in A^{pcv}$, introdúcese unha variable binaria y_v^{pcv} para determinar a dirección de fluxo:

$$\begin{aligned} q_v &\leq M y_v^{pcv} \\ q_v &\geq M (y_v^{pcv} - 1), \end{aligned}$$

onde M é unha constante suficientemente grande. E, para cada $v = (i, j) \in A^{pcv}$, as restricións son

$$\begin{aligned} p_i^{sq} - p_j^{sq} &\leq M y_v^{pcv} \\ p_i^{sq} - p_j^{sq} &\geq M (y_v^{pcv} - 1), \end{aligned}$$

Omítese aquí o detalle da descrición das restricións que engadiría o modelado do autoconsumo das plantas de regasificación (*boil-off*), a inclusión de outro tipo de válvulas ou os diagramas de operación asociados aos turbocompresores activos nas estacións de compresión presentes na rede. Unha descrición máis detallada do problema pode verse en [3] e [11].

Como xa se mencionou, temos que ademais da non linearidade, un problema real de optimización da rede de gas presenta elementos que só se poden modelar facendo uso de variables binarias. Isto dálle un carácter discreto a rexión factible. A idea orixinal para resolver o problema non linear era empregar un algoritmo do tipo SLP. Pero, a presenza de variables discretas non resulta compatible coa busca local dun óptimo nunha rexión de confianza, dado o carácter discreto do conxunto factible. Así xurdiu o SLP-NTR, a programación linear sucesiva sen rexión de confianza, da que falamos na seguinte sección.

2.2. Algoritmo implementado dende o ITMATI (SLP-NTR)

Nesta sección preséntase o algoritmo de programación linear sucesiva de dúas etapas, ao que nos referimos como 2SLP, para resolver o problema MINLP de transporte de gas pola rede.

O algoritmo 2SLP consta de dúas etapas. A primeira etapa enfócase en obter unha configuración óptima das variables binarias facendo uso dunha variante da programación linear sucesiva, que recibe o nome de programación linear sucesiva sen rexión de confianza (SLP-NTR) e se describe a continuación. Entón, a primeira etapa do algoritmo incorpora todos os elementos enteiros do modelo e consiste en decidir a configuración óptima destes elementos. Fixadas as variables binarias, a segunda etapa consiste en resolver un problema de programación non linear facendo uso do algoritmo PSLP que introducimos na Sección 1.5.3.

Recordamos, que a PLS clásica consiste nunha aproximación linear do problema non linear, linealizando as restricións e a función obxectivo entorno a un punto inicial, \mathbf{x}^0 . Logo, o problema linear é resolto baixo a condición adicional de que a solución do problema anterior, \mathbf{x}^1 , estea dentro dunha rexión de confianza, o que garante que a linearización do problema en \mathbf{x}^0 non é unha mala aproximación do problema en \mathbf{x}^1 . Dependendo de se é unha boa aproximación ou no, emprégase \mathbf{x}^1 como novo punto entorno ao que linearizar o problema e actualízase a rexión de confianza, para repetir o procedemento anterior co óptimo obtido de resolver o problema non linear entorno ao novo punto. Existen distintos enfoques para actualizar a rexión de confianza en cada iteración, por exemplo, o PSLP, onde se compara o valor da función obxectivo penalizada linearizada e a orixinal penalizada entre dous iterantes consecutivos do algoritmo. O inconveniente deste algoritmo para resolver o problema de optimización no contexto das redes de gas é acomodar as variables binarias, polo que se emprega unha alternativa deste algoritmo omitindo a restrición que ten que ver coa rexión de confianza e os parámetros que teñen que ver coa súa actualización no algoritmo PSLP, que se coñece, como mencionabamos antes, como algoritmo SLP-NTR. Así, a principal diferenza do algoritmo SLP-NTR respecto ao anterior é que non introduce restricións de paso e permite mover a nova solución en cada iteración.

Considerando o problema de programación non linear (sen variables binarias) definido como:

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{suxeito a: } g_i(\mathbf{x}) \leq 0, \quad i \in I^m \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j \in I^l \end{aligned} \tag{2.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ e $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ se asumen continuamente diferenciables.

Sexa $\bar{\mathbf{x}} \in \mathbb{R}^n$, considérase unha aproximación linear do problema non linear definido por 2.1 entorno

a $\bar{\mathbf{x}}$, por exemplo a aproximación linear de Taylor de orde 1 (ver Definición 1.35). Denotamos por $TNLP(\bar{\mathbf{x}})$ esta aproximación linear entornao ao punto $\bar{\mathbf{x}}$.

O algoritmo SLP-NTR parte dun punto $\mathbf{x}^0 \in \mathbb{R}^n$ (non necesariamente factible para o problema non linear) e resólvese o problema $TNLP(\mathbf{x}^0)$ obtendo a solución \mathbf{x}^1 . Se a distancia entre \mathbf{x}^1 e \mathbf{x}^0 é suficientemente pequena, o algoritmo para. Noutro caso, volve a resolverse a linearización do problema ao redor de \mathbf{x}^1 . Na Figura 2.1 descríbese o pseudocódigo deste algoritmo.

Algoritmo SLP-NTR

```

1: Inicializamos  $\mathbf{x}^0 \in \mathbb{R}^n$ ,  $k = 0$ .
2: Resolvemos o  $TNLP(\mathbf{x}^k, \infty)$ . Sexa  $\bar{\mathbf{x}}^k$  a solución.
3: Actualizamos  $\mathbf{x}^{k+1} = \bar{\mathbf{x}}^k$ .
4: if  $\|\mathbf{x}^k - \mathbf{x}^{k+1}\| < \epsilon$  :  $\mathbf{x}^{k+1}$  é próximo a  $\mathbf{x}^k$  then
5:   STOP: Devolver  $\mathbf{x}^{k+1}$ 
6: else
7:    $k = k + 1 \rightarrow$  Ir ao paso 2.
8: end if

```

Figura 2.1: Esquema do algoritmo SLP-NTR.

Se o algoritmo converge na primeira etapa e a solución dada ten suficiente precisión (en termos de factibilidade respecto ao problema inicial) xa non sería necesario aplicar a segunda etapa, xa que a solución, dende o punto de vista teórico é igual de boa que a que nos pode proporcionar o algoritmo PSLP.

Vexamos o resultado que garante que se a sucesión xerada polo SLP-NTR converge se obtén unha solución KKT ao problema non linear orixinal. En caso contrario, non podemos garantir que os puntos de acumulación da sucesión sexan puntos KKT, do problema non linear orixinal, como era o caso do PSLP (ver Teorema 1.39). Denotamos por $S : \Pi \rightarrow P(\mathbb{R}^n)$ a aplicación que asigna a cada problema linear a súa rexión factible.

Teorema 2.1. *Sexa $\{\mathbf{x}^k\}$ a sucesión xerada polo algoritmo SLP-NTR. Supoñamos que $\{\mathbf{x}^k\}$ converge a un punto $\bar{\mathbf{x}}$ e que S é semicontinua inferiormente no problema $LP(\bar{\mathbf{x}})$. Entón, as seguintes afirmacións son certas,*

1. $\bar{\mathbf{x}}$ é solución óptima de $TNLP(\bar{\mathbf{x}})$.
2. $\bar{\mathbf{x}}$ é un punto factible para NLP.
3. $\bar{\mathbf{x}}$ é un punto KKT de NLP.

As principais desvantaxes da SLP-NTR, respecto a SLP, son que é menos estable en termos de converxencia, en algúns casos presenta oscilacións debido a eliminación da rexión de confianza. Por outra banda, a sucesión de puntos de acumulación dada polo SLP-NTR pode estar formada por non ser puntos KKT do problema orixinal e pode non converxer a puntos do interior do conxunto factible. Ilustramos isto no exemplo seguinte:

Exemplo 2.2. Consideremos o problema de optimización: $\min_{x \in [-1,1]} x^2$. Claramente se ve que o óptimo é cero. Imos ver que ocorre aplicando o algoritmo SLP-NTR.

1. Consideramos $k = 0$ e que $x^0 = -0.5$.
2. Resolvemos a linearización do problema NLP no punto x^0 , $TNLP(x^0, \infty)$: $\min_{x \in [-1,1]} -x$. A solución é 1.

3. Actualizamos $x^1=1$ e repetimos o paso anterior xa que non se se verifica o criterio de parada. A solución de $TNLP(x^1, \infty)$ sería -1.
4. Actualizamos $x^2 = -1$ e como non se verifica o criterio de parada, resolvemos $TNLP(x^2, \infty)$. Neste caso a solución sería 1.
5. Sucesivamente repetindo o anterior, iríamos en cada paso alternando do 1 ao -1.

No esquema anterior pode verse que a sucesión xerada polo algoritmo ten dous puntos de acumulación, os puntos 1 e -1 que non son puntos interiores do conxunto factible. Ademais observamos a inestabilidade na convexencia (ciclado). \square

Por outra banda, sábese que cando dous puntos da sucesión xerada polo algoritmo SLP-NTR están a distancia δ , entón o segundo punto é un punto casi-KKT. Un punto casi-KKT, intuitivamente, é un punto que está cerca dun punto factible \bar{x} , que case verifica as condicións de Karush-Khun-Tucker descritas no Teorema 1.18, é dicir, o ángulo formado polo gradiente $-\nabla f(\bar{x})$ e o cono convexo xerado polos gradientes das restricións activas en \bar{x} é moi pequeno (ou nulo). A formulación matemática destes resultados excede os contidos deste traballo polo que ese omiten, pero pódense consultar en [8].

Ao principio deste capítulo mencionamos que o algoritmo que se emprega para a resolución do problema determinista, introducido na sección anterior, era o algoritmo 2SLP, que consta de dúas etapas. A primeira etapa fai uso do algoritmo SLP-NTR (ver Figura 2.1), e proporciona unha configuración das variables binarias, que son fixas na segunda etapa do algoritmo 2SLP onde se emprega o PSLP, obtendo unha solución máis precisa para o problema non linear orixinal. O algoritmo PSLP, no 2SLP, emprega a solución obtida co SLP-NTR como punto inicial. A necesidade do emprego dunha segunda etapa está nos problemas de converxencia que presenta o SLP-NTR.

Empregar o algoritmo 2SLP para a resolución do problema introducido anteriormente é unha boa decisión, como se pode ver en [8]. Ademais, incluso en problemas sen variables binarias, obsérvase que o algoritmo 2SLP tende a conseguir mellores solucións en menor tempo que o algoritmo PSLP clásico.

Capítulo 3

Introducción á programación estocástica

Neste capítulo faremos unha introdución conceptual á programación estocástica. Centraremos principalmente no problema estocástico de dúas etapas con recurso. Na Sección 3.1 introducimos un dos problemas clásicos da programación estocástica, presentado como o problema do granxeiro en [4], ao que faremos referencia ao longo do capítulo.

Na maioría dos problemas do mundo real a miúdo se inclúe incerteza, non se coñece o valor dalgún parámetro pero si a súa distribución de probabilidade ou unha aproximación discreta desta. No contexto no que se traballa, que trataremos no Capítulo 4, veremos que o emprego da programación estocástica xorde na toma de decisións relativas á planificación da construción de novas infraestruturas en redes de gas existentes. A incerteza que se considera afecta á demanda de gas na rede, ao prezo por unidade de gas e ao custo por infraestrutura e intentase optimizar atendendo aos custos de construción e operación da rede, e o posible incumprimento da demanda.

3.1. Exemplo: o problema do granxeiro

Un granxeiro dispón de 500 hectáreas (ha) de terra e debe decidir durante o inverno canto terreo dedica a cada cultivo. Sendo os cultivos a sementar o trigo, o millo e a remolacha.

O granxeiro sabe que precisa polo menos 200 toneladas (T) de trigo e 240 de millo para alimentar ao seu gando durante un ano. No caso de que a colleita sexa inferior a esas cantidades deberá mercala e se as excede venderaa. Por outra banda, dispón dunha cota de venda de remolacha de 6000 T baixo as seguintes condicións: se a colleita de remolacha é inferior á cota poderá vender cada tonelada a 36€, se é superior só lle pagaran 10€/T. Baseándose no pasado, o granxeiro sabe que o rendemento medio da súa terra é 2.5 T, 3 T e 20 T por hectárea para o trigo, o millo e a remolacha, respectivamente. Na Táboa 3.1 recóllense os datos do problema.

Designamos as variables como seguen:

- x_1, x_2 e x_3 : hectáreas de terra dedicada ao trigo, ao millo e á remolacha, respectivamente.
- y_1 e y_2 : toneladas de trigo e millo a comprar, respectivamente.
- l_1 e l_2 : toneladas de trigo e millo a vender, respectivamente.

	Trigo	Millo	Remolacha
Rendemento (T/ha)	2.5	3	20
Custo plantación (€/ha)	150	230	260
Prezo de venda (€/T)	170	150	36 se a colleita é ≤ 6000 T 10 se a colleita é > 6000 T
Mínimo requirido (T)	200	240	-
Prezo da compra (€/ha)	238	210	-

Táboa 3.1: Información sobre os cultivos.

- l_3 e l_4 : toneladas de remolacha a vender por debaixo e por encima da cota, respectivamente.

Considerando os datos do problema recollidos na Táboa 3.1 e as variables anteriores, o problema fórmulase como segue:

$$\begin{aligned}
 \text{mín} \quad & 150x_1 + 230x_2 + 260x_3 + 238y_1 - 170l_1 + 210y_2 - 150l_2 - 36l_3 - 10l_4 \\
 \text{suxeito a} \quad & x_1 + x_2 + x_3 \leq 500 \\
 & 2.5x_1 + y_1 - l_1 \geq 200 \\
 & 3x_2 + y_2 - l_2 \geq 240 \\
 & l_3 + l_4 \leq 20x_3 \\
 & l_3 \leq 6000 \\
 & x_1, x_2, x_3, y_1, y_2, l_1, l_2, l_3, l_4 \geq 0.
 \end{aligned} \tag{3.1}$$

A solución do Problema 3.1, amosada baixo a columna con nome “Esperado” na Táboa 3.2, indica que se está dedicando suficientes hectáreas de terra para alcanzar a cota de 6000 T de remolacha e os requisitos mínimos de alimentación para os restantes cultivos. Ademais obteríase un excedente de trigo que pode ser vendido. Esta solución pode verse como que asigna a terra a cada cultivo en orde decrecente de beneficio por área.

Na formulación anterior estase a considerar un rendemento medio que se correspondería cunhas condicións meteorolóxicas normais. Está claro que se as condicións meteorolóxicas son boas ou malas, a produtividade da terra vai cambiar. Por isto, o granxeiro pregúntase se a solución óptima é sensible a variacións no rendemento. Considerando un rendemento o 20% maior e menor que o rendemento medio dado na Táboa 3.1 volveríase a formular o problema anterior. Por simplicidade, asúmese que as condicións meteorolóxicas e como consecuencia disto o rendemento da terra, non ten un impacto significativo no prezo.

Na Táboa 3.2 recóllense as solucións dos problemas mencionados anteriormente. Comprobase que a solución correspondíase coa proporcionada na referencia resolvendo os problemas en AMPL, o modelado do problema determinista pódese consultar no Apéndice A.

- Cando os rendementos son altos, precísanse menos hectáreas de terra para obter os requisitos mínimos de trigo e millo, e para alcanzar a cota de remolacha. Dédicase máis hectáreas ao trigo, cuxa produción extra é vendida.

- Cando o rendemento é menor ao esperado, precisaríanse grandes hectáreas de terra para obter os requisitos mínimos e alcanzar a cota de remolacha. Non se alcanzan as 240 T de millo necesarias para alimentar o gando e precísase comprar este cultivo.

Cultivos	Esperado			Por encima do esperado (+20%)			Por debaixo do esperado(-20%)		
	Trigo	Millo	Remolacha	Trigo	Millo	Remolacha	Trigo	Millo	Remolacha
Rendemento (T/ha)	2.5	3	20	3	3.6	24	2	2.4	16
Hectáreas a cultivar	120	80	300	183.33	66.67	250	100	25	375
Produtividade	300	240	6000	550	240	6000	200	60	6000
T a vender	100	-	6000	350	-	6000	-	-	6000
T a comprar	-	-	-	-	-	-	-	180	-
Beneficio total	118600€			167667€			59950€		

Táboa 3.2: Solución óptima baseada no rendemento esperado, por encima e por debaixo.

Está claro que a solución óptima do problema do granxeiro é sensible aos cambios de produtividade da terra e das condicións meteorolóxicas, e obviamente o granxeiro debe tomar a decisión de cantas hectáreas plantar de cada cultivo sen ter información sobre iso.

Non so temos que a solución é distinta nos distintos escenarios senón que é posible que a solución óptima nun escenario non sexa factible noutro. Por tanto, senón se modela adecuadamente un problema de programación baixo incerteza pódense obter solucións que resultan ser non factibles, o que dependendo de cada problema en cuestión pode ser máis ou menos crítico.

3.2. Formulación dun problema de programación estocástica

Nesta sección verase a formulación xeral dun problema estocástico e empregárase o exemplo introducido no apartado anterior para a súa ilustración.

A programación estocástica denota a optimización baixo incerteza. Na configuración do problema estocástico algúns parámetros do modelo tales como a matriz de restricións, os coeficientes da función obxectivo, os límites das variables e restricións poden tomar valores aleatorios.

Considérese o espazo de probabilidades (Ω, F, P) onde:

- Ω é o conxunto de todos os posibles resultados.
- F é unha σ -álgebra en π .
- $P : F \rightarrow [0, 1]$ é a medida de probabilidade.

A formulación xeral dun problema estocástico é:

$$\begin{aligned} \text{mín} \quad & f(\mathbf{x}, \xi) \\ \text{suxeito a} \quad & g_i(\mathbf{x}, \xi) \leq 0, \quad i = 1, \dots, n \\ & \mathbf{x} \in X \subset R^n, \end{aligned} \tag{3.2}$$

onde ξ é un vector de parámetros aleatorios definido no espazo de probabilidades (Ω, F, P) . Ademais, suponse que a medida de probabilidade P non depende de \mathbf{x} e que no momento da toma de decisión é coñecida a distribución da probabilidade.

O conxunto de decisións que deben ser tomadas sen ter información completa sobre os eventos aleatorios son as chamadas *decisións de primeira etapa*, son decisións nas que os valores dos parámetros son coñecidos (deterministas) e que se representarán mediante o vector \mathbf{x} . Posteriormente a toma destas decisións, recíbese información sobre a realización dun evento aleatorio ξ . Logo, son tomadas as *decisións de segunda etapa ou de recurso* que dependen do evento aleatorio e denotamos por \mathbf{y} . Denótase por ω o resultado aleatorio que caracteriza a incerteza, $\omega \in \Omega$. Nalgún caso empregárase a notación $\xi(\omega) = \xi_\omega$ ou $\mathbf{y}(s) = \mathbf{y}_s$ para facer explícita a dependencia destas decisións dos elementos ω e s . A sucesión de eventos e decisións nun problema estocástico resúmese en: $\mathbf{x} \rightarrow \xi(\omega) \rightarrow \mathbf{y}(\omega)$.

No problema do granxeiro tratado na sección anterior, vimos que a solución óptima era sensible aos cambios da produtividade da terra e das condicións meteorolóxicas. Polo tanto, o problema do granxeiro pode verse como un problema de programación estocástica onde este debe tomar a decisión de cantas ha plantar de cada cultivo, as decisións de primeira etapa que se corresponden coas variables (x_1, x_2, x_3) , sen coñecer cal vai ser o rendemento da terra. Así, as cantidades a comprar ou vender $(y_1, y_2, l_1, l_2, l_3, l_4)$ son as decisións de segunda etapa, dependen do rendemento.

O problema estocástico de dúas etapas

Por simplicidade na notación introdúcese a formulación do problema estocástico de dúas etapas lineais. Pero pode adaptarse ao caso no que a función obxectivo da primeira e segunda etapa, ou as restricións sexan non lineais.

O *problema estocástico de dúas etapas con recurso* formúlase como segue:

$$\begin{aligned} \text{mín} \quad & z = \mathbf{c}^\top \mathbf{x} + \mathbb{E}_\xi Q(\mathbf{x}, \omega) \\ \text{suxeito a} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{3.3}$$

onde \mathbf{x} é un vector de dimensión $n_1 \times 1$ que representa as decisións de primeira etapa. En correspondencia con \mathbf{x} están os vectores \mathbf{c} e \mathbf{b} , coñecidos, de dimensión $n_1 \times 1$ e $m_1 \times 1$, respectivamente, e a matriz \mathbf{A} de dimensión $m_1 \times n_1$. \mathbb{E}_ξ representa a esperanza matemática respecto a ξ . $Q(\mathbf{x}, \omega)$ é a función do *problema de recurso*:

$$\begin{aligned} \text{mín} \quad & \mathbf{q}(\omega)^\top \mathbf{y}(\omega) \\ \text{suxeito a} \quad & \mathbf{W}(\omega)\mathbf{y}(\omega) + \mathbf{T}(\omega)\mathbf{x} = \mathbf{h}(\omega), \\ & \mathbf{y}(\omega) \geq \mathbf{0} \end{aligned} \tag{3.4}$$

onde o vector \mathbf{y} representa a decisión de recurso ou de segunda etapa e $\omega \in \Omega$ representa un resultado aleatorio que é coñecido. Para unha realización ω , os datos do problema da segunda etapa $\mathbf{q}(\omega)$, $\mathbf{h}(\omega)$ e $\mathbf{T}(\omega)$, de dimensións $n_2 \times 1$, $m_2 \times 1$ e $m_2 \times n_1$, respectivamente; vólvense coñecidos. \mathbf{W} é unha matriz fixa, de dimensión $m_2 \times n_2$, que recibe o nome de matriz recurso. Cada compoñente de \mathbf{q} , \mathbf{T} e \mathbf{h} é unha posible variable aleatoria.

Denotando por $\mathbf{T}_i(\omega)$ a fila i -ésima de $\mathbf{T}(\omega)$ e xuntando as compoñentes estocásticas da segunda etapa obtemos o vector $\xi(\omega)^\top = (\mathbf{q}(\omega)^\top, \mathbf{h}(\omega)^\top, \mathbf{T}_1(\omega), \dots, \mathbf{T}_{m_2}(\omega))$ con $N = n_2 + m_2 + (m_2 \times n_1)$ compoñentes. Vemos que unha única realización de ω inflúe en varias variables aleatorias, aquí, en todas as compoñentes de ξ .

A función obxectivo do Problema 3.3 contén un termo determinista $\mathbf{c}^\top \mathbf{x}$ e a esperanza da función obxectivo da segunda etapa $\mathbf{q}(\omega)^\top \mathbf{y}(\omega)$, tomada sobre todas as realizacións do evento aleatorio. O termo da segunda etapa é máis difícil co primeiro, xa que, para cada ω , o valor $\mathbf{y}(\omega)$ é solución dun problema linear distinto. Para enfeitar este feito, emprégase a noción de *problema determinista equivalente*. Para unha realización dada ω , sexa

$$Q(\mathbf{x}, \xi(\omega)) = \min_{\mathbf{y}} \{ \mathbf{q}(\omega)^\top \mathbf{y} \mid \mathbf{W}(\omega) \mathbf{y} = \mathbf{h}(\omega) - \mathbf{T}(\omega) \mathbf{x}, \mathbf{y} \geq \mathbf{0} \} \quad (3.5)$$

o valor da función obxectivo do problema da segunda etapa. Logo defínese a *función valor esperado da segunda etapa* ou *función recurso* como

$$\mathfrak{L}(\mathbf{x}) = \mathbb{E}_\xi Q(\mathbf{x}, \xi(\omega)) \quad (3.6)$$

e o *problema determinista equivalente* (DEP) formúlase como:

$$\begin{aligned} \text{mín} \quad & z = \mathbf{c}^\top \mathbf{x} + \mathfrak{L}(\mathbf{x}) \\ \text{suxeito a} \quad & \mathbf{A} \mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (3.7)$$

A representación anterior ilustra a sucesión de eventos nun problema de recurso. As decisións de primeira etapa \mathbf{x} son tomadas en presenza de incerteza sobre realizacións futuras de ξ . Na segunda etapa, o valor de ξ é coñecido e as decisións de recurso \mathbf{y} poden ser tomadas. Os efectos futuros de ter tomado as decisións de primeira etapa son medidos pola función recurso, $\mathfrak{L}(\mathbf{x})$, xa que calcula o valor esperado de tomar a decisión \mathbf{x} .

Exemplo. Imos ilustrar o problema do granxeiro estocástico. Para isto, é útil unir as decisións de primeira e segunda etapa mencionadas anteriormente mediante un escenario, $s = 1, 2, 3$, correspondente a un rendemento por riba da media, ao rendemento medio e por debaixo da media, respectivamente. Isto, leva a considerar, na formulación do problema baixo incerteza, un novo conxunto de variables da forma l_{is} , $i = 1, 2, 3, 4$ e y_{js} , $j = 1, 2$, e suprimir as decisións iniciais da segunda etapa. Estas variables representan o que na segunda etapa deberá facer o granxeiro en cada escenario. Por exemplo, l_{41} representaría a cantidade de remolacha vendida a un prezo desfavorable se o rendemento está por encima da media. O conxunto de escenarios sería, neste caso, $\Omega = \{s_1 = \text{“tempo bo ou rendemento por riba da media”}, s_2 = \text{“tempo normal ou rendemento medio”}, s_3 = \text{“tempo malo ou rendemento por debaixo da media”}\}$. Ademais, coñécese a produtividade ou rendemento de cada cultivo en cada escenario $\xi(s)$ con $s \in \Omega$, $\xi(s_1) = (3, 3.6, 24)$, $\xi(s_2) = (2.5, 3, 20)$ e $\xi(s_3) = (2, 2.4, 16)$.

Supoñendo que quere maximizar os beneficios a longo prazo é razoable buscar a solución que maximice a utilidade esperada. Considerando que a probabilidade de cada escenario é p_1, p_2 e p_3 , o problema formularíase como segue:

$$\begin{aligned}
\text{mín } z = & 150x_1 + 230x_2 + 260x_3 \\
& -p_1(170l_{11} - 238y_{11} + 150l_{21} - 210y_{21} + 36l_{31} + 10l_{41}) \\
& -p_2(170l_{12} - 238y_{12} + 150l_{22} - 210y_{22} + 36l_{32} + 10l_{42}) \\
& -p_3(170l_{13} - 238y_{13} + 150l_{23} - 210y_{23} + 36l_{33} + 10l_{43})
\end{aligned}$$

$$\text{suxeito a } x_1 + x_2 + x_3 \leq 500$$

$$3x_1 + y_{11} - l_{11} \geq 200, \quad 2.5x_1 + y_{12} - l_{12} \geq 200, \quad 2x_1 + y_{13} - l_{13} \geq 200$$

$$3.6x_2 + y_{21} - l_{21} \geq 240, \quad 3x_2 + y_{22} - l_{22} \geq 240, \quad 2.4x_2 + y_{23} - l_{23} \geq 240$$

$$l_{31} + l_{41} \leq 24x_3, \quad l_{32} + l_{42} \leq 20x_3, \quad l_{33} + l_{43} \leq 16x_3$$

(3.8)

$$l_{31}, l_{32}, l_{33} \leq 6000$$

$$x_1, x_2, x_3 \geq 0$$

$$y_{11}, y_{12}, y_{13}, y_{21}, y_{22}, y_{23} \geq 0$$

$$l_{11}, l_{12}, l_{13}, l_{21}, l_{22}, l_{23}, l_{31}, l_{32}, l_{33}, l_{41}, l_{42}, l_{43} \geq 0$$

		Trigo	Millo	Remolacha
Primeira etapa	Ha a cultivar	170	80	250
s=1	Produtividade	510	288	6000
	T a vender	310	48	6000
	T a comprar	-	-	-
s=2	Produtividade	425	240	5000
	T a vender	225	-	5000
	T a comprar	-	-	-
s=3	Produtividade	340	192	4000
	T a vender	140	-	4000
	T a comprar	-	48	-
Beneficio total:108390€				

Táboa 3.3: Solución óptima baseada no modelo estocástico 3.8.

Na Táboa 3.3 recóllese a solución óptima para o modelo estocástico formulado en 3.8 considerando $p_1 = p_2 = p_3 = 1/3$. O modelado do problema estocástico para a resolución en AMPL pódese consultar no Apéndice A. Esta solución pode verse como que se asigna terra á remolacha evitando as vendas a prezo desfavorable, aínda que isto implique que non se alcance a cota cando os rendementos son os

esperados ou están por debaixo da media. Por outra banda, as ha dedicadas ao millo satisfán os requisitos de alimentación excepto cando os rendementos son baixos. Pola contra, as ha de terra dedicadas ao trigo son suficientes para satisfacer os requisitos mínimos necesarios e en todos os escenarios é posible vender certas toneladas. Esta solución ilustra que é imposible, baixo incerteza, encontrar unha solución ideal baixo todas as circunstancias. Baixo un prognóstico perfecto non tería lugar vender a remolacha a un prezo desfavorable ou deixar algunha ha sen usar. Esta decisión dáse no modelo estocástico porque as decisións deben ser cubertas fronte a diversos escenarios.

Neste caso, o problema de segunda etapa para un escenario particular s pode ser escrito como:

$$\begin{aligned}
 Q(x, s) &= \text{mín}\{238y_1 - 170l_1 + 210y_2 - 150l_2 - 36l_3 - 10l_4\} \\
 \text{suxeito a } \quad &t_1(s)x_1 + y_1 - l_1 \geq 200 \\
 &t_2(s)x_2 + y_2 - l_2 \geq 240 \\
 &l_3 + l_4 \leq t_3(s)x_3 \\
 &l_3 \leq 6000 \\
 &y, l \geq 0,
 \end{aligned} \tag{3.9}$$

onde $t_i(s)$ representa o rendimento do cultivo i baixo o escenario s . Para ver a relación entre a Formulación 3.3 e a 3.9 podemos dicir que o vector aleatorio $\xi = (t_1, t_2, t_3)$ está formado polos tres rendementos e que ξ pode tomar tres valores diferentes, $\xi_i = (t_1(i), t_2(i), t_3(i))$, $i = 1, 2, 3$. \square

Problema multietapa

Ata este momento estabamos a falar de problemas estocásticos bietapas pero na práctica os problemas de decisión soen involucrar unha sucesión de decisións que responden a resultados que evolucionan ao longo do tempo. Estes problemas coñécense como problemas estocásticos multietapa. Suponse que en cada etapa t as decisións das etapas anteriores e as realizacións dos eventos aleatorios son coñecidos.

A formulación do *problema estocástico multietapa* ven dada por:

$$\begin{aligned}
 \text{mín} \quad &z = \mathbf{c}^1 \mathbf{x}^1 + \mathbb{E}_{\xi^2}[\text{mín } \mathbf{c}^2(\omega)^\top \mathbf{x}^2(\omega^2)] + \dots + \mathbb{E}_{\xi^H}[\text{mín } \mathbf{c}^H(\omega)^\top \mathbf{x}^H(\omega^H)] \\
 \text{suxeito a } \quad &\mathbf{W}^1 \mathbf{x}^1 = \mathbf{h}^1, \\
 &\mathbf{T}^1(\omega^2) \mathbf{x}^1 + \mathbf{W}^2 \mathbf{x}^2(\omega^2) = \mathbf{h}^2(\omega^2), \\
 &\quad \vdots \quad \quad \quad \vdots \\
 &\mathbf{T}^{H-1}(\omega^H) \mathbf{x}^{H-1}(\omega^{H-1}) + \mathbf{W}^H \mathbf{x}^H(\omega^H) = \mathbf{h}^H(\omega^H), \\
 &\mathbf{x}^1 \geq \mathbf{0}; \mathbf{x}^t(\omega^t) \geq \mathbf{0}, \quad t \in \{2, \dots, H\};
 \end{aligned}$$

onde \mathbf{c}^1 e \mathbf{h}^1 son vectores coñecidos de dimensión $n_1 \times 1$ e $m_1 \times 1$, respectivamente. $\xi^t(\omega)^\top = (\mathbf{c}^t(\omega)^\top, \mathbf{h}^t(\omega)^\top, \mathbf{T}_1^{t-1}(\omega), \dots, \mathbf{T}_{m_t}^{t-1}(\omega))$ é un vector aleatorio de dimensión N_t en (Ω, F, P) (onde $F^t \subset F^{t+1}$) para todo $t = 2, \dots, H$, e cada \mathbf{W}^t é unha matriz coñecida de dimensión $m_t \times n_t$. A única iteración entre a etapa t e a etapa $t + 1$ é a través do termo $\mathbf{x}^t(\omega^t)$.

Imos ver a formulación DEM equivalente ao problema anterior. Dada a iteración entre etapas sucesivas, indicada polo termo $\mathbf{x}^t(\omega^t)$, podemos introducir unha formulación recursiva do problema anterior :

$$\begin{aligned} Q^H(\mathbf{x}^{H-1}, \xi^H(\omega)) &= \text{mín } \mathbf{c}^H(\omega)\mathbf{x}^H(\omega) \\ \text{suxeito a} \quad W^H\mathbf{x}^H(\omega) &= \mathbf{h}^H(\omega) - \mathbf{T}^{H-1}(\omega)\mathbf{x}^{H-1}, \\ \mathbf{x}^H(\omega) &\geq \mathbf{0}, \end{aligned}$$

para todo $t \in \{2, \dots, H-1\}$. A solución para outra etapa pode ser obtida mediante recursividade cara atrás. Considerando $\mathcal{L}^{t+1}(x^t) = \mathbb{E}_{\xi^{t+1}}[Q^{t+1}(\mathbf{x}^t, \xi^{t+1}(\omega))]$ para todo $t \in \{2, \dots, H-1\}$ resolveríase,

$$\begin{aligned} Q^t(\mathbf{x}^{t-1}, \xi^t(\omega)) &= \text{mín } \mathbf{c}^t(\omega)\mathbf{x}^t(\omega) + \mathcal{L}^{t+1}(\mathbf{x}^t) \\ \text{suxeito a} \quad \mathbf{W}^t\mathbf{x}^t(\omega) &= \mathbf{h}^t(\omega) - \mathbf{T}^{t-1}(\omega)\mathbf{x}^{t-1}, \\ \mathbf{x}^t(\omega) &\geq \mathbf{0}. \end{aligned}$$

O que se busca co anterior é resolver o problema:

$$\begin{aligned} \text{mín} \quad z &= \mathbf{c}^1\mathbf{x}^1 + \mathcal{L}^2(\mathbf{x}^1) \\ \text{suxeito a} \quad \mathbf{W}^1\mathbf{x}^1 &= \mathbf{h}^1, \\ \mathbf{x}^1 &\geq \mathbf{0}, \end{aligned}$$

o cal ten a mesma forma que o problema determinista equivalente de dúas etapas 3.7.

No caso de ter unha distribución finita dos parámetros aleatorios a estrutura do modelo pode representarse mediante unha árbore de decisión, como a da Figura 3.1. Antes de detallar como se constrúe a árbore, mencionar que un escenario s representa unha colección de resultados ω que teñen características comúns para un modelo específico. Ademais cada escenario terá asignada unha probabilidade $p(s)$. Cada arco representa a realización dun evento aleatorio entre dúas etapas. Cada nodo representa un subproblema correspondente a unha etapa dada e a secuencia de eventos aleatorios, determinado por un camiño dende o nodo orixe. Cada rama dende o nodo inicial a un nodo final representa un escenario particular, é dicir, unha posible realización da incerteza. Polo tanto, neste caso, temos 8 escenarios distintos. Ademais, asociado a cada nodo da árbore existen decisións que deben ser tomadas, é dicir, un grupo de variables que decidir en cada período ou etapa, t . Na árbore que estamos describindo temos $H = 3$ etapas, é dicir, t pode tomar os valores 1, 2 e 3 e dous niveis de incerteza por etapa. Nas etapas anteriores ($t < 3$), temos un menor número de posibles realizacións, e referímonos a eles como escenarios da etapa t . Os escenarios da etapa t terán un único escenario antecesor na etapa $(t-1)$ e, excepto na última etapa, varios escenarios descendentes na etapa $(t+1)$. Notar que os distintos escenarios da etapa t se corresponden coas mesmas realizacións e só se diferencian nos descendentes. Por exemplo, os escenarios 1 e 2 teñen os mesmos antecesores ata a etapa $t = 2$ e só se diferencian nos escenarios descendentes, na etapa $t = 3$.

Principio de non anticipación

Na programación estocástica, considérase que a solución obtida do modelo ata unha determinada etapa t , non depende da información que aínda non está dispoñible. O decisor non pode condicionar

as súas accións á información que aínda non ten, é dicir, non pode anticiparse ás accións. Isto coñécese como o “Principio de non anticipación”, que foi formulado por Wets y Rockafellar en [16] e di:

“Se dous escenarios distintos, son idénticos ata unha etapa determinada no horizonte de tempo, entón os valores das variables de decisión deben ser os mesmos ata esa etapa.”

Na Figura 3.2, represéntase a idea das decisións de non anticipatividade, é dicir, en cada etapa os rectángulos sombreados indican as decisións dos distintos escenarios que deben ser iguais.

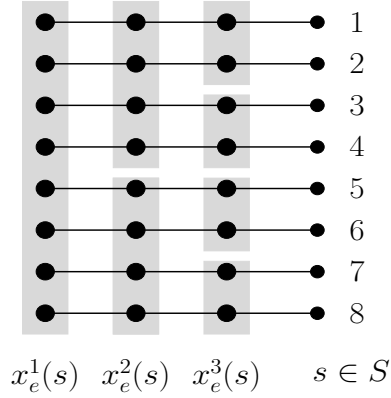
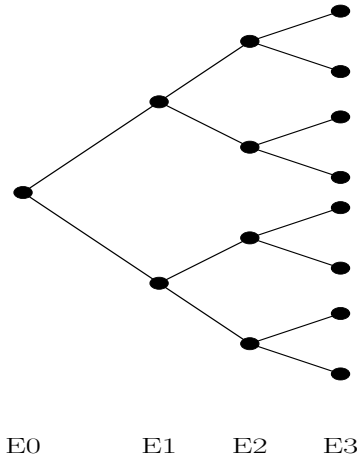


Figura 3.1: Árbore de decisión asociado a un problema de 3 etapas.

Figura 3.2: Condición de non anticipatividade.

Valor engadido á programación estocástica

A principal dificultade que presentan os problemas multietapa é que o número de variables aumenta moi rápido co número de escenarios, de aí que computacionalmente resulten difíciles de resolver. Isto leva a que na práctica en ocasións se resollen versións máis sinxelas destes, por exemplo, o problema resultado de substituír as variables aleatorias polos seus valores esperados ou varios problemas deterministas, cada un correspondente a un escenario en particular, e logo combínanse as diferentes solucións con algunha regra heurística. Debido a isto xorden os conceptos de *valor esperado da información perfecta* (EVPI) e *valor da solución estocástica* (VSS). O obxectivo destes é ver se a resolución dos problemas con este enfoque é axeitada. Vexamos isto con máis detalle.

O valor esperado da información perfecta (EVPI) é a máxima cantidade que un “decisor” estaría disposto a pagar a cambio da información completa e correcta sobre o futuro. É a suma ponderada da diferenza entre a solución do problema estocástico en cada escenario e a solución con información perfecta en dito escenario.

Supoñendo que a incerteza se pode modelar a través dun número de escenarios. Sexa ξ unha variable aleatoria cuxa realización se corresponde con varios escenarios. Definimos o problema asociado a un escenario particular ξ como:

$$\begin{aligned} \text{mín} \quad & z(\mathbf{x}, \xi) = \mathbf{c}^\top \mathbf{x} + \text{mín}\{\mathbf{q}^\top \mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{h} - \mathbf{T}\mathbf{x}, \mathbf{y} \geq \mathbf{0}\} \\ \text{suxeito a} \quad & \mathbf{A}\mathbf{x} = \mathbf{B}, \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{3.10}$$

onde $\xi(\omega)^\top = (\mathbf{q}(\omega)^\top, \mathbf{h}(\omega)^\top, \mathbf{T}_1(\omega), \dots, \mathbf{T}_{m_2}(\omega))$ e asúmense as consideracións feitas na formulación do

Problema 3.3.

Se $\bar{\mathbf{x}}(\xi)$ é solución do Problema 3.10 para cada un dos escenarios e $z(\bar{\mathbf{x}}(\xi), \xi)$ o valor óptimo da función obxectivo, pódese calcular o valor esperado da solución óptima, o cal se coñece na literatura como solución *wait and see* (WS),

$$WS = \mathbb{E}_\xi[\min_{\mathbf{x}} z(\mathbf{x}, \xi)] = \mathbb{E}_\xi z(\bar{\mathbf{x}}(\xi), \xi).$$

Esta solución sería a obtida se se dispón de información perfecta no escenario en cuestión. Por exemplo, no problema do granxeiro, se este fose adiviño tería información acerca das condicións meteorolóxicas e como consecuencia do rendemento da terra antes de decidir as ha de terra a cultivar, as cales decidiría en base ás solucións da Táboa 3.2. Ademais se se supón que os rendementos poden variar sendo cíclicos, isto é, un ano cun rendemento por encima do esperado é seguido por un cun rendemento medio e logo por un ano cun rendemento baixo, o beneficio medio dos tres anos sería a media dos beneficios totais recollidos na Táboa 3.2, é dicir, 115406€ por ano. Este beneficio sería o obtido baixo perfecta información.

A solución *wait and see* é comparable coa solución chamada *here and now* asociada ao problema de recurso que se pode escribir como:

$$RP = \min_{\mathbf{x}} \mathbb{E}_\xi z(\mathbf{x}, \xi), \quad (3.11)$$

con solución óptima dada por \mathbf{x}^* .

No problema do granxeiro esta solución é a que deixa ao agricultor cun beneficio esperado de 108390€, ver Táboa 3.3.

O valor esperado da información perfecta, é por definición, a diferenza entre a solución *wait and see* e a solución *here and now*, isto é:

$$EVPI = RP - WS.$$

Así, para o exemplo que se trata o valor esperado de información perfecta é 7016€. Esta cantidade representa a perda de beneficio debida a presenza de incerteza.

Encontrar a solución *wait and see* pode resultar custoso, incluso imposible se a información perfecta non está dispoñible a ningún prezo. Isto leva a que se considere a alternativa de resolver o problema resultante de substituír todas as variables aleatorias polos seus valores esperados. Este coñécese como *problema de valor esperado*, isto é:

$$EV = \min_{\mathbf{x}} z(\mathbf{x}, \bar{\xi}), \quad (3.12)$$

onde $\bar{\xi} = \mathbb{E}(\xi)$ denota a esperanza de ξ . Denotamos por $\bar{\mathbf{x}}(\bar{\xi})$ a solución óptima de 3.12, chamada *solución valor esperado*. Excepto que $\bar{\mathbf{x}}(\bar{\xi})$ sexa independente de ξ , non hai razón para crer que a solución valor esperado está cerca da solución do problema de recurso (RP).

No exemplo que estamos a tratar, temos que o agricultor pode supoñer que os rendementos sempre son os esperados, e asignar a superficie óptima de acordo con estes rendementos. Esta solución representa a solución valor esperado. Isto é común en optimización pero pode ter consecuencias desfavorables. Empregando a solución valor esperado resultaría un beneficio anual de 107240€. O modelado do problema en AMPL é similar ao determinista, pero neste caso as variables de primeira etapa (\mathbf{x}) son agora un parámetro do modelo, e toman o valor en base ao rendemento esperado (120, 80, 300). Tras esta consideración resólvese o problema para os tres escenarios e a solución valor esperado é a media das solucións obtidas.

O valor da solución estocástica (VSS) é o concepto que mide como de bo ou malo é a decisión $\bar{\mathbf{x}}(\bar{\xi})$ en termos do problema de recurso (RP). Antes de definir VSS introducimos *o resultado esperado de*

usar a solución EV como:

$$EEV = \mathbb{E}_\xi(z(\bar{\mathbf{x}}(\bar{\xi}), \xi)).$$

Esta cantidade mide como se comporta $\bar{\mathbf{x}}(\bar{\xi})$, permitindo que as decisións de segunda etapa sexan escollidas optimamente como funcións de $\bar{\mathbf{x}}(\bar{\xi})$ e ξ . Logo, o valor da solución estocástica (VSS) defínese como:

$$VSS = EEV - RP,$$

é dicir, o VSS mide como de boa ou mala é a decisión $\bar{\mathbf{x}}(\bar{\xi})$ en termos de 3.11.

Para o problema do granxeiro o valor da solución estocástica é $VSS=1150\text{€}$. Esta cantidade é o custo de ignorar a incerteza na elección dunha decisión. Isto sería a cantidade que como moito estaría disposto a pagar o agricultor para obter información sobre as condicións meteorolóxicas do próximo ano.

3.2.1. Métodos baseados na programación estocástica lagrangiana

O método de relaxación lagrangiana é unha das técnicas de resolución máis empregadas na optimización non linear. Emprégase cando existen restricións que complican a resolución do problema. Isto é, a resolución do problema sen esas ecuacións ten unha estrutura cuxa resolución é máis sinxela. Baséase na dualización das ecuacións difíciles e na formulación dun problema dual cuxa resolución proporciona unha aproximación do valor óptimo do problema orixinal mellor que a resolución da relaxación linear.

Consideramos a formulación do problema estocástico non linear sen restricións de igualdade (para simplificar a notación),

$$\begin{aligned} \text{ínf } z &= f^1(\mathbf{x}) + \mathcal{L}(\mathbf{x}) \\ \text{suxeito a } g_i^1(\mathbf{x}) &\leq \mathbf{0}, \quad i \in \{1, \dots, m_1\}, \end{aligned} \tag{3.13}$$

onde $\mathcal{L}(\mathbf{x}) = \mathbb{E}_\omega[Q(\mathbf{x}, \omega)]$ e

$$\begin{aligned} Q(\mathbf{x}, \omega) &= \text{ínf } f^2(\mathbf{x}, \mathbf{y}_\omega, \omega) \\ \text{suxeito a } g_j^2(\mathbf{x}, \mathbf{y}_\omega, \omega) &\leq 0 \quad j \in \{1, \dots, m_2\}. \end{aligned} \tag{3.14}$$

Este problema de programación non linear pódese resolver mediante métodos de descomposición, de gradiente e procedementos de penalización, pero nestes métodos asúmese que os gradientes de \mathcal{L} están dispoñibles ou son fáciles de obter. Isto non ocorre en problemas de programación estocástica porque cada avaliación pode implicar resolver varios problemas da forma 3.14. Os enfoques lagrangianos foron propostos para evitar este problema.

A idea básica detrás do enfoque lagrangiano é colocar as restricións que relacionan as decisións da primeira e segunda etapa (*linking constraint*) na función obxectivo evitando a optimización repetida de subproblemas para encontrar a solución. Para isto, escríbese o Problema 3.13 da seguinte forma:

$$\begin{aligned} \text{ínf } z &= f^1(\mathbf{x}) + \mathbb{E}_\omega[f^2(\mathbf{y}_\omega, \omega)] \\ \text{suxeito a } g_i^1(\mathbf{x}) &\leq 0, \quad i \in \{1, \dots, m_1\} \\ g_i^2(\mathbf{x}, \mathbf{y}_\omega, \omega) &\leq 0, \quad i \in \{1, \dots, m_2\}. \end{aligned} \tag{3.15}$$

Denotando por π o vector de multiplicadores de Lagrange asociados coas restricións, podemos escribir o dual do Problema 3.15 como:

$$\max_{\pi_{\omega} \geq 0} \omega = \theta(\pi), \quad (3.16)$$

onde,

$$\begin{aligned} \theta(\pi) &= \inf_{\mathbf{x}, \mathbf{y}} z = f^1 + \mathbb{E}_{\omega}[f^2(\mathbf{y}_{\omega}, \omega)] + \mathbb{E}_{\omega}[\sum_{i=1}^{m_2} \pi_{\omega_i}(g_i^2(\mathbf{x}, \mathbf{y}_{\omega}, \omega))] \\ \text{suxeito a } &g_i^1(\mathbf{x}) \leq 0, \quad i = 1, \dots, m_1. \end{aligned} \quad (3.17)$$

Consideremos agora o caso discreto. Supoñemos que $\Omega = \{1, \dots, K\}$ con $p_i = P\{\omega = i\}$ e $i \in \{1, \dots, K\}$. Temos que $\mathfrak{L}(\mathbf{x}) = \sum_{k=1}^K p_k Q(\mathbf{x}, k)$ e o dual lagrangiano fórmase como:

$$\begin{aligned} \text{maximizar } &\theta(\pi) = \inf\{f^1(\mathbf{x}) + \sum_{k=1}^K p_k (f^2(\mathbf{x}, \mathbf{y}_k, k) + \sum_{j=1}^{m_2} \pi_{k,j} g_j^2(\mathbf{x}, \mathbf{y}_k, k)) : \mathbf{x} \in X\} \\ \text{suxeito a } &\pi \geq 0, \end{aligned} \quad (3.18)$$

onde $X = \{\mathbf{x} : g^1(\mathbf{x}) \leq 0\}$.

Teorema 3.1. *Se todas as funcións son convexas e o problema ten un óptimo finito e un punto interior nas restricións. Entón, para todo $\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_K$ factible no problema P e π_1, \dots, π_K factible no problema dual, $z \geq \theta(\pi)$. Ademais, no óptimo coinciden.*

Imos ver agora o método de gradiente para o problema dual. Na Figura 3.3 recóllese o pseudocódigo deste método. Notar que dada a solución do dual π e unha solución $(\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_K)$ que resolve o problema $\theta(\pi)$, $g^2(\mathbf{x}, \mathbf{y}_k, k)$ é o gradiente asociado ao bloque k .

Basic Lagrangian Dual Ascent Method

Inicialización:

Fixar $\pi^0 \geq 0$, $v = 0$ e ir ao Paso 1.

Paso 1: Sexa $(\mathbf{x}^v, \mathbf{y}_1^v, \dots, \mathbf{y}_K^v)$ a solución de $\theta(\pi^v)$

- Definir $\Delta_{k,j} = \begin{cases} 0 & \text{se } \pi_{k,j}^v = 0 \text{ e } g_j^2(\mathbf{x}, \mathbf{y}_k, k) \leq 0 \\ g_j^2(\mathbf{x}, \mathbf{y}_k, k) & \text{noutro caso} \end{cases}$
- Se $\Delta = 0$, STOP.

Paso 2: Sexa λ^v solución de $\max\{\theta(\pi^v + \lambda\Delta) : \pi^v + \lambda\Delta \geq 0, \lambda \geq 0\}$.

- Definir $\pi^{v+1} = \pi^v + \lambda^v \Delta$, $v = v + 1$ e ir ao Paso 1.

Figura 3.3: Esquema do método de gradiente para o dual.

Resolver $\theta(\pi)$ é máis fácil que resolver o problema primal, xa que, en cada iteración do algoritmo resólvese un problema no que as *linking constraints* están na función obxectivo. Ademais o número de iteracións no dual é pequeno comparado co número de iteracións que se precisarían se se atacara directamente ao problema 3.13 e 3.14. Os principais problemas deste método son que se a solución de θ non é única hai que traballar con subgradients, e o basearse na *single-point linearization* de θ , a converxencia é lenta e pode dar lugar a comportamentos en zigzag cerca do óptimo. Estes problemas

levan a que se empreguen alternativas a este enfoque que usan información máis xeral ou de segunda orde.

Unha alternativa ao método de gradiente para o dual é incluír termos estritamente convexos na función obxectivo. Por exemplo, da forma $\|g_j^2(\mathbf{x}, \mathbf{y}_k, k)\|^2$, isto da lugar ao enfoque do lagrangiano aumentado (ver Sección 1.9) engadindo o termo anterior penalizado a $\theta(\pi)$. A vantaxe deste método é que se poden aplicar o algoritmo de Newton na resolución dado que temos unha matriz Hessiana non singular.

Unha alternativa máis eficaz é incluír na función obxectivo as restricións de non anticipación en lugar das restricións de segunda etapa. Se \mathbf{x} é unha solución de primeira etapa, considéranse solucións \mathbf{x}_k para as restricións de segunda etapa tales que $\mathbf{x}_k = \mathbf{x}$ para todo k . As restricións $\mathbf{x}_k = \mathbf{x}$ para todo k pásanse a función obxectivo ao dualizar. Desta maneira, θ convértese en

$$\begin{aligned} \theta'(\rho) = \inf z = f^1(\mathbf{x}) + \sum_{k=1}^K p_k f^2(\mathbf{x}_k, \mathbf{y}_k, k) + \sum_{k=1}^K [\rho_k^\top (\mathbf{x}_k - \mathbf{x}) + \frac{r}{2} \|\mathbf{x}_k - \mathbf{x}\|^2] \\ \text{sujeto a } g_i^1(\mathbf{x}) \leq 0 \quad i = 1, \dots, m_1 \\ g_i^2(\mathbf{x}_k, \mathbf{y}_k, k) \leq 0 \quad k \in \{1, \dots, K\} \end{aligned} \quad (3.19)$$

Na función obxectivo do problema 3.19 o termo $(\mathbf{x} - \mathbf{x}_k)$ é o único enlace entre as decisións de non anticipación x e as decisión do escenario k . O termo $\frac{r}{2} \|\mathbf{x}_k - \mathbf{x}\|^2$ pode verse como o análogo ao que se engade no lagrangiano aumentado.

O método anterior aseméllase o algoritmo *progressive hedging* proposto por Rockafellar and Wets [16]. Este método logra unha separación completa en escenarios individuais para cada iteración, e por tanto ten menos traballo en cada iteración, aínda que o número de iteracións pode ser maior. A clave deste método é que mantén a estrutura do subproblema ao longo do algoritmo. Ademais é facilmente paralelizable e dá solución a problemas de gran tamaño.

O método *progressive hedging* parte dunha solución non anticipativa $\hat{\mathbf{x}}^v$ e un multiplicador ρ^v . A solución non anticipativa (pero non necesariamente factible) é usada no lugar de \mathbf{x} en 3.19 excepto en f^1 e g^1 que aparecen as variables anticipativas \mathbf{x}_k . Isto é,

$$\begin{aligned} \inf z = \sum_{k=1}^K p_k (f^1(\mathbf{x}_k) + f^2(\mathbf{x}_k, \mathbf{y}_k, k) + (\rho_k^v)^\top (\mathbf{x}_k - \hat{\mathbf{x}}^v) + \frac{r}{2} \|\mathbf{x}_k - \hat{\mathbf{x}}^v\|^2) \\ \text{suxeito a } g^1(\mathbf{x}_k) \leq 0 \quad k \in \{1, \dots, K\} \\ g^2(\mathbf{x}_k, \mathbf{y}_k, k) \leq 0 \quad k \in \{1, \dots, K\}. \end{aligned} \quad (3.20)$$

Agora, 3.20 pode dividirse en subproblemas para cada k que poden ser tratados por separado.

Supoñendo que $(\mathbf{x}_k^{v+1}, \mathbf{y}_k^{v+1})$ é unha solución de 3.20, onde v denota a iteración. Obtense unha nova decisión non anticipativa tomando o valor esperado de \mathbf{x}^{v+1} como $\hat{\mathbf{x}}^{v+1}$ e un novo multiplicador ρ , $\rho^{v+1} = \rho^v + (\mathbf{x}^{v+1} - \hat{\mathbf{x}}^{v+1})$.

Na Figura 3.4 recóllese o pseudocódigo do algoritmo PHA. Mencionar que o parámetro r permite controlar a velocidade coa que se axustan os multiplicadores.

Para a elaboración desta sección baseámonos na Sección 5.8 do Capítulo 5 de [4], onde tamén se recolle información sobre a converxencia omitida aquí.

O algoritmo *progressive hedging* (PHA) adáptase facilmente ao caso multietapa. Por simplicidade, defínese a proxección, Π , como a proxección no espazo de solucións non anticipativas, definidas como

Algoritmo progressive hedging (PHA)Inicialización:

Partimos de \mathbf{x}^0 non anticipativa, multiplicador ρ^0 e $r > 0$. Tomamos $v = 0$ e imos ao Paso 1.

Paso 1: Dada $(\mathbf{x}_k^{v+1}, \mathbf{y}_k^{v+1})$ solución do subproblema 3.20. Sexa $\hat{\mathbf{x}}^{v+1} = (\hat{\mathbf{x}}^{v+1,1}, \dots, \hat{\mathbf{x}}^{v+1,K})$ onde $\hat{\mathbf{x}}^{v+1,k} = \sum_{l=1}^K p_l \mathbf{x}_l^{v+1} \forall k = 1, \dots, K$.

Paso 2: Defínese $\rho^{v+1} = \rho^v + r(\mathbf{x}_1^{v+1} - \hat{\mathbf{x}}^{v+1}, \dots, \mathbf{x}_K^{v+1} - \hat{\mathbf{x}}^{v+1})$.

- Se $\hat{\mathbf{x}}^{v+1} = \hat{\mathbf{x}}^v$ e $\rho^{v+1} = \rho^v$, STOP.
- Noutro caso, facer $v = v + 1$ e volver ao Paso 1.

Figura 3.4: Esquema do algoritmo *progressive hedging* (PHA).

a esperanza condicional de todas as solucións da etapa t que teñen os mesmos antecedentes ata a etapa t .

O principal subproblema para o caso de H etapas é unha extensión do Problema 3.20 como segue:

$$\begin{aligned} \inf z = & \sum_{k=1}^K p_k \left(f^0(\mathbf{x}_0, x_k^1) + \sum_{t=1}^H f^t(\mathbf{x}_k^t, \mathbf{x}_k^{t+1}, k) + (\rho^v)^\top (\mathbf{x}_k - \hat{\mathbf{x}}^v) + \frac{r}{2} \|\mathbf{x}_k - \hat{\mathbf{x}}^v\|^2 \right) \\ \text{suxeito a} \quad & g^0(\mathbf{x}_0, \mathbf{x}_k^1) \leq 0 \quad k \in \{1, \dots, K\} \\ & g^t(\mathbf{x}_k^t, \mathbf{x}_k^{t+1}, k) \leq 0 \quad t = 1, \dots, H, k \in \{1, \dots, K\}, \end{aligned} \quad (3.21)$$

onde \mathbf{x}_0 representa as condicións iniciais dadas.

A formulación anterior conduce ao PHA para problemas multietapa. Na Figura 3.5 recóllese o pseudocódigo do algoritmo no caso multietapa.

Algoritmo progressive hedging (PHA) multietapa.Inicialización:

Partimos de $\mathbf{x}^0 = (\mathbf{x}_k^t, k = 1, \dots, K; t = 1, \dots, H)$ non anticipativa, $\hat{\mathbf{x}}^0 = \mathbf{x}^0$, multiplicador inicial ρ^0 e $r > 0$. Tomamos $v = 0$ e imos ao Paso 1.

Paso 1: Sexa (\mathbf{x}_k^{v+1}) para $k = 1, \dots, K$ solución de 3.21. Sexa $\hat{\mathbf{x}}^{v+1} = \Pi(\mathbf{x}^{v+1})$, isto é, $\hat{\mathbf{x}}_k^{v+1}(i) = \hat{\mathbf{x}}_{k'}^{v+1}(i)$ en todas as compoñentes i correspondentes as decisións \mathbf{x}^t en tempo t mentres k e k' compartan o mesmo historial ata o tempo t .

Paso 2: Defínese $\rho^{v+1} = \rho^v + r(\mathbf{x}^{v+1,k} - \hat{\mathbf{x}}^{v+1})$.

- Se $\hat{\mathbf{x}}^{v+1} = \hat{\mathbf{x}}^v$ e $\rho^{v+1} = \rho^v$, STOP. $\hat{\mathbf{x}}^v$ e ρ^v son óptimos.
- Noutro caso, sexa $v = v + 1$ e volver ao Paso 1.

Figura 3.5: Esquema do algoritmo *progressive hedging* multietapa.

Exemplo: Imos aplicar o PHA ao problema do granxeiro que estivemos tratando ao longo deste capítulo. Recordamos que neste problema temos dúas etapas. Na primeira etapa baixo a incerteza das condicións meteorolóxicas que terán lugar, e como consecuencia disto do rendemento da terra; o granxeiro ten que tomar a decisión de cantas hectáreas de terra dedicar a cada un dos cultivos. Tras coñecer o rendemento da terra xorden as decisións de segunda etapa.

Para aplicar o PHA, o problema estocástico pode dividirse no seguinte subproblema, onde na

función obxectivo se inclúen as restricións de non anticipación.

$$\begin{aligned}
 \text{mín } z = & \quad 150x_1 + 230x_2 + 260x_3 + 238y_1 + 210y_2 - 170l_1 - 150l_2 - 36l_3 - 10l_4 \\
 & + \sum_{k=1}^3 \rho_k (\mathbf{x}_k - \hat{\mathbf{x}}^v) + \frac{r}{2} \|\mathbf{x}_k - \hat{\mathbf{x}}^v\|^2 \\
 \text{suxeito a } & \quad x_1 + x_2 + x_3 \leq 500 \\
 & \quad t_1(s)x_1 + y_1 - l_1 \geq 200 \\
 & \quad t_2(s)x_2 + y_2 - l_2 \geq 240 \\
 & \quad l_3 + l_4 \leq t_3(s)x_3 \\
 & \quad l_3 \leq 6000 \\
 & \quad x_1, x_2, x_3, y_1, y_2, l_1, l_2, l_3, l_4 \geq 0
 \end{aligned}$$

No Apéndice A tense a implementación deste subproblema e do algoritmo en código AMPL. Tomando $\rho^0 = 0$ e $r = 2$ e o valor inicial das variables igual a cero e empregando o solver gurobi, tense que en 8 iteracións se consegue resolver o problema estocástico empregando o PHA. Obtense que a solución do problema é -180390€. \square

Capítulo 4

Problema con incerteza no contexto das redes de gas

Neste capítulo imos introducir o problema con incerteza no contexto das redes de transporte de gas, como se modela e as principais alternativas de resolución coas que se afronta a resolución dende o ITMATI.

Mencionar que se conserva a notación empregada no Capítulo 2. Para a elaboración deste Capítulo fíxose uso de [13], informe elaborado dende o ITMATI polo equipo investigador do proxecto.

4.1. Problema de programación estocástica no contexto das redes de transporte de gas

Na modelización do problema de programación estocástica sobre as redes de gas, considéranse os seguintes elementos que non aparecían no modelo determinista:

- Infraestruturas novas. Consideramos o conxunto de infraestruturas $\{1, \dots, E\}$. Cada unha das infraestruturas $e \in \{1, \dots, E\}$ inclúe un conxunto de nodos e arestas que de ser construídos, o farán de forma conxunta.
- Etapas. Trátase de cada un dos espazos temporais a ter en conta. En particular, considéranse H etapas, podendo ser identificadas cada $t \in \{1, \dots, H\}$, por exemplo, cunha duración dun ano.
- Incerteza. Os seguintes elementos do problema están suxeitos a incerteza ξ , é dicir, non se coñece o seu valor con exactitude:
 - Consumo, $c_i^t(\xi)$, nos nodos N^c en cada etapa t .
 - Prezo unitario (por unidade de gas) do gas perdido polo autoconsumo na etapa t , $pr^t(\xi)$.
 - Custos de construción da infraestrutura e na etapa t , $cc_e^t(\xi)$.

Ademais dos elementos anteriores, os parámetros asociados a cada nodo e cada aresta que se mencionaron na descrición do problema determinista aparecen no modelado do problema con incerteza.

A incerteza nas diferentes etapas do problema será modelada mediante unha árbore de decisión. Así, se existen R posibles resultados da incerteza e consideramos H etapas no noso problema, cada

posibilidade na árbore de decisión quedará determinada por unha H -tupla, onde cada unha das súas compoñentes recolle a incerteza nesa etapa.

A solución do problema exposto permitirá coñecer, ademais dos valores habituais das presións nos nodos, os caudais nas arestas, os consumos nas arestas compresoras e as variables binarias para as válvulas, cando se debe construír unha determinada infraestrutura. Polo que, na formulación do modelo se introduce unha nova variable binaria x_e^t que representa a decisión de construír (1) ou non (0) a infraestrutura e na etapa t .

A resolución do problema de programación non linear e enteira empregando a programación estocástica busca minimizar, en media, os custos de construción e operación asociados a cada infraestrutura, o prezo do gas por autoconsumo, así como a imposición dunha penalización polo posible incumprimento da demanda nos nodos de consumo para cada posible escenario da árbore de decisión. É dicir, a función obxectivo do noso problema será:

$$\begin{aligned} \min \mathbb{E}_\xi \left[\sum_{t=1}^H \left\{ \sum_{e=1}^E \alpha_{cc} c_e^t (x_e^t(\xi^{t-1}) - x_e^{t-1}(\xi^{t-2})) + \frac{\alpha_{gc}}{b} pr^t(\xi^t) \sum_{k \in A^c} g_k^t \right\} + \right. \\ \left. + \alpha_{inc} \delta^{t-1} pr^t(\xi^t) \left(\sum_{i \in N^c} \left(d_i - \left(\sum_{k \in A_i^{ini}} q_k^t + \sum_{k \in A_i^{fin}} q_k^t \right) \right) \right) \right], \end{aligned} \quad (4.1)$$

onde α_{cc} , α_{gc} e α_{inc} son, os pesos impostos na función obxectivo aos custos de construción, ao gas consumido e aos posibles incumprimentos, respectivamente, nos que se incorra. O parámetro b é un factor de corrección que permite a estimación da demanda media anual a partir da demanda diaria dun escenario e δ determina o factor de ponderación temporal das etapas no problema de programación estocástica.

Á hora de considerar as restricións temos que distinguir se as infraestruturas están construídas dende un inicio ou se son candidatas a ser construídas. Para a modelización destas últimas hai que ter en conta un posible retardo r_e que fai referencia á etapa na que entra en funcionamento a nova infraestrutura. O tratamento da construción destas infraestruturas mediante variables binarias introduce unha condición de non anticipación, que di que se a infraestrutura foi construída na etapa t , nas seguintes debe manterse construída. Esta condición, exprésase como:

$$x_e^{t+1}(\xi^t) \geq x_e^t(\xi^{t-1}),$$

onde $x_e^t(\xi^{t-1})$ é a variable que indica a decisión de construír ou non a infraestrutura e na etapa t . Notar que, se na etapa t é necesaria unha certa infraestrutura e , a decisión debe tomarse r_e etapas antes, isto é, $x_e^{t-r_e} = 1$

Ademais das restricións que se consideraban no problema de optimización non linear tratado na sección 2.1, neste caso hai que considerar restricións adicionais que permitan a modelización da construción das novas infraestruturas. Así, para cada infraestrutura e , para cada etapa t e cada incerteza ξ^t considéranse as seguintes restricións (adaptacións das consideradas no problema determinista):

- Restricións de cota:

$$\underline{p}_i x_e^{t-r_e} \leq p_i \leq \bar{p}_i x_e^{t-r_e}, \quad \forall i \in N \cap e$$

$$\underline{q}_a x_e^{t-r_e} \leq q_a \leq \bar{q}_a x_e^{t-r_e}, \quad \forall a \in A \cap e.$$

- Restricións de conservación de fluxo:

$$\sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t = c_i^t(\xi^t), \quad \forall i \in N^c \cap e.$$

$$c_i^{min} x_e^{t-r_e} \leq \sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t \leq c_i^{max} x_e^{t-r_e}, \quad \forall i \in N^S \cap e.$$

- Restricións de perda de presión:

$$p_{in}^t - p_{out}^t \leq K_1 \frac{L_a}{D_a} |q_a| q_a + K_2 \frac{p_i^{sq} + p_j^{sq}}{2} (h_j - h_i) + (1 - x_e^{t-r_e}) M, \quad \forall k \in A^n.$$

$$p_{in}^t - p_{out}^t \geq K_1 \frac{L_a}{D_a} |q_a| q_a + K_2 \frac{p_i^{sq} + p_j^{sq}}{2} (h_j - h_i) - (1 - x_e^{t-r_e}) M, \quad \forall k \in A^n,$$

onde M denota unha constante de optimización suficientemente grande.

- Restricións de compresión:

$$g_{h,a}^t \leq K_3 \frac{\gamma}{\gamma-1} (\alpha_h^{\frac{\gamma-1}{\gamma}} - 1) |q_a^t|, \quad \forall h \in E^c \cap e, a \in A^c \cap e.$$

$$g_{h,a}^t \geq K_3 \frac{\gamma}{\gamma-1} (\alpha_h^{\frac{\gamma-1}{\gamma}} - 1) |q_a^t| - (1 - x_e^{t-r_e}) M, \quad \forall h \in E^c \cap e, a \in A^c \cap e.$$

onde α_h denomínase razón de compresión, $\alpha_h = \frac{p_j}{p_i}$ sendo $h = (i, j)$ aresta compresora da infraestrutura e . M é unha constante de optimización.

Por outra banda, as estacións de compresión permiten un incremento da presión na dirección do fluxo do gas, polo que,

$$\Delta_h^t = p_j^t - p_i^t, \quad \forall h \in E^c \cap e.$$

$$\Delta_h^t \geq -(1 - x_e^{t-r_e}) M, \quad \forall h \in E^c \cap e.$$

O problema estocástico resultante de considerar a función obxectivo 4.1 e as restricións asociadas tanto aos elementos deterministas como as infraestruturas planificables, é un problema de programación estocástico non linear cun número elevado de binarias. Isto débese a que ademais das variables existentes na optimización da rede de gas xa construída (determinista), hai que ter en conta a introdución de novas variables binarias indicadoras da construción de novas infraestruturas, cuxo número medra ao incrementar o número de etapas a estudar, aumentando así a complexidade do problema.

4.2. Resolución do problema estocástico

Nesta sección imos introducir algunha das posibilidades implementadas dende o ITMATI para resolución do problema estocástico.

A metodoloxía implementada dende o ITMATI recibe o nome de CSLP-estocástico e non é máis que o Algoritmo 2.2 aplicado ao problema estocástico. Lévese a cabo de maneira automática e consecutiva coas metodoloxías DEM e PHA implementadas en GANESO. Unha vez establecidas as variables binarias correspondentes coas decisións de planificación, coas estacións de compresión, coas válvulas de control de presión,... o algoritmo garante que a solución obtida sexa a mellor posible. Imos ver con máis detalle como se resolve o problema estocástico empregando a metodoloxía DEM e PHA.

Metodoloxía DEM

O método de resolución DEM (modelo determinista equivalente) ten en conta todas as posibilidades na árbore de decisión do problema, isto é, tantas como nodos conteña o diagrama de decisión aos que se asocia unha certa incerteza (variación da demanda, do prezo do gas consumido, do custo por construción). Por exemplo, para un problema de 3 etapas a árbore de decisión asociado sería 4.1.

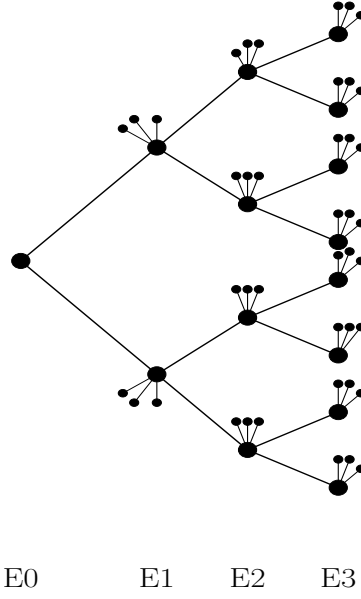


Figura 4.1: Árbore de decisión asociado a un problema de 3 etapas.

Neste caso, a función obxectivo 4.1 correspóndese coa media ponderada, polas probabilidades de cada escenario, dos custos de construción e do prezo do gas consumido asociados aos mesmos, así como da consideración do posible incumprimento da demanda nos distintos nodos de consumo do escenario. É dicir,

$$\begin{aligned} \min \sum_{S_H=1}^R \cdots \sum_{S_2=1}^R \sum_{S_1=1}^R \left[\sum_{t=1}^H p(s_1, \dots, s_t) \left\{ \sum_{t=1}^H \left\{ \sum_{e=1}^E \alpha_{cc} c_e^t (x_e^t(s_1, \dots, s_{t-1}) - x_e^{t-1}(s_1, \dots, s_{t-2})) + \right. \right. \right. \\ \left. \left. \frac{\alpha_{gc}}{b} pr^t(s_1, \dots, s_t) \sum_{a \in A^c} g_a^t \right\} + \alpha_{inc} \delta^{t-1} pr^t(\xi^t) \left(\sum_{i \in N^c} \left(d_i - \left(\sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t \right) \right) \right) \right\} \right]. \end{aligned}$$

Na expresión anterior estamos considerando $\xi^t = (s_1, \dots, s_t)$ para $t = 1, \dots, H$.

Por outra banda, as restricións asociadas ao problema estocástico quedan determinadas polas restricións asociadas ao problema determinista para infraestruturas xa construídas, polas asociadas ás infraestruturas a planificar e pola condición de non anticipación para cada unha das posibilidades en cada etapa. A condición de non anticipación exprésase como,

$$x_e^t(s_1, \dots, s_t) \geq x_e^{t-1}(s_1, \dots, s_{t-1}), \quad \forall e \in \{1, \dots, E\}, \quad \forall t \geq 2.$$

Omitimos as expresións das restantes restricións que son análogas as presentadas anteriormente.

Recordamos que H fai referencia ao número de etapas, R ao número de incertezas e $S = \{s_1, \dots, s_H\}$ é o espazo de todos os posibles escenarios. Entón, o problema resultante ten un total de $\sum_{t=1}^H R^{t-1} E$ situacións a considerar, e polo tanto, a mesma cantidade de variables binarias $x_e^t(s_1, \dots, s_{t-1})$ para cada infraestrutura $e \in \{1, \dots, E\}$.

O problema estocástico resultante de aplicar a metodoloxía determinista é resolto logo empregando o CSLP-estocástico. Veremos no capítulo seguinte, como se comporta esta metodoloxía na práctica.

Desde o ITMATI tamén implementaron un método coñecido como *expected value* (EV), o cal se pode considerar como unha simplificación da metodoloxía anterior. Neste caso, en lugar de considerar todos os posibles escenarios da árbore de decisión, considérase un escenario “promedio” por etapa que ten por demanda, a media das demandas da súa etapa e por prezo do gas, a media de prezos do gas en dita etapa. Así, denótanse por

$$\begin{aligned}\bar{c}_i^t &= \sum_{s_1, \dots, s_t} p(s_1, \dots, s_t) c_i^t(s_1, \dots, s_t) & \forall i \in N^c, \forall t \in \{1, \dots, H\}, \\ \bar{p}r^t &= \sum_{s_1, \dots, s_t} p(s_1, \dots, s_t) pr^t(s_1, \dots, s_t) & \forall t \in \{1, \dots, H\},\end{aligned}$$

a demanda media de consumo por etapa e o prezo do gas medio por etapa, respectivamente.

A árbore de decisión asociada a un problema de 3 etapas sería 4.2.

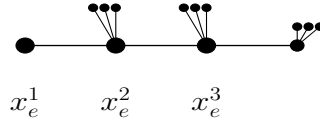


Figura 4.2: Arbore de decisión asociada a un problema de 3 etapas.

Con este método, o problema simplifícase, redúcese o número de variables binarias x_e^t a un total de $H \times E$, isto é, unha única variable binaria por cada etapa t e infraestrutura e . Así, a función obxectivo 4.1 fórmulase como segue:

$$\min \sum_{t=1}^H \left\{ \sum_{e=1}^E \alpha_{cc} c_e^t (x_e^t - x_e^{t-1}) + \frac{\alpha_{gc}}{b} \bar{p}r^t \sum_{a \in A^c} g_a^t + \alpha_{inc} \delta^{t-1} \bar{p}r^t \left(\sum_{i \in N^c} \left(d_i - \left(\sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t \right) \right) \right) \right\}$$

De novo, as restricións referentes a cada etapa quedan determinadas, ademais de polas restricións para infraestruturas xa construídas, polas referentes á infraestruturas a planificar (vese modificada a restrición asociada á conservación de fluxo nos nodos de demanda de consumo, que neste caso é igual á demanda media por etapa \bar{c}_i^t) e pola condición de non anticipación para variables binarias por infraestrutura asociada a cada etapa ($x_e^t \geq x_e^{t-1}$, $\forall e \in \{1, \dots, E\}$, $\forall t \geq 2$).

Metodoloxía PHA

Ademais da metodoloxía DEM, tamén está implementada a metodoloxía PHA de relaxación lagrangiana. Recordamos que neste método (ver Sección 3.2.1) se incorporaba á función obxectivo as restricións de non anticipación. Neste caso a restrición de non anticipación fai referencia á construción ou non dunha nova infraestrutura, polo que as variables de non anticipación son binarias. Na Figura 4.3 amósase a descomposición lagrangiana para o problema de 3 etapas.

Para a formulación do problema resultante baixo esta metodoloxía precisamos introducir novos parámetros que serven para introducir a condición de non anticipatividade na función obxectivo:

- λ_s^t , $s \in S = R^H$, $t \in \{1, \dots, H\}$.
- $\bar{x}_{e,s}^t$, $s \in S$, $t \in \{1, \dots, H\}$, $e \in \{1, \dots, E\}$.

tales que $\lambda_s^t = \lambda_{s'}^t$, $\bar{x}_{e,s}^t = \bar{x}_{e,s'}^t$ se e só se $(s_1, \dots, s_{t-1}) = (s'_1, \dots, s'_{t-1})$.

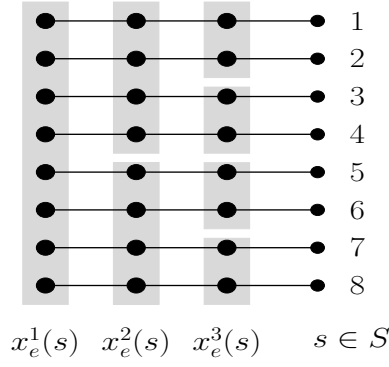


Figura 4.3: Descomposición lagrangiana.

Así, neste caso a función obxectivo 4.1 adopta a seguinte forma

$$\begin{aligned} \min \sum_{s=1}^S p(s) \left[\sum_{t=1}^H \left\{ \sum_{e=1}^E \alpha_{cc} c_e^t (x_e^t(s) - x_e^{t-1}(s)) + \frac{\alpha_{gc}}{b} pr(s)^t \sum_{a \in A^c} g_a^t + \right. \right. \\ \left. \left. + \sum_{e=1}^E \lambda_s^t (x_e^t(s) - \bar{x}_{e,s}^t) + \alpha_{inc} \delta^{t-1} pr(s)^t \left(\sum_{i \in N^c} \left(d_i - \left(\sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t \right) \right) \right) \right\} \right] \end{aligned}$$

Igual que no tratamento dos métodos anteriores as restricións asociadas ao problema de programación estocástica quedan determinadas, ademais de polas restricións asociadas a infraestruturas xa construídas, polas referentes á infraestruturas a planificar e pola condición de non anticipación ($x_e^t(s) \geq x_e^{t-1}(s)$, $\forall e \in \{1, \dots, E\} \forall t \geq 2$).

A solución do problema anterior obtense mediante a resolución do subproblema asociado a cada escenario posible, $s \in S$, cuxa función obxectivo é

$$\begin{aligned} \min \sum_{t=1}^H \left\{ \sum_{e=1}^E \alpha_{cc} c_e^t (x_e^t(s) - x_e^{t-1}(s)) + \frac{\alpha_{gc}}{b} pr(s)^t \sum_{a \in A^c} g_a^t + \right. \\ \left. + \sum_{e=1}^E \lambda_s^t (x_e^t(s) - \bar{x}_{e,s}^t) + \alpha_{inc} \delta^{t-1} pr(s)^t \left(\sum_{i \in N^c} \left(d_i - \left(\sum_{a \in A_i^{ini}} q_a^t - \sum_{a \in A_i^{fin}} q_a^t \right) \right) \right) \right\} \end{aligned}$$

e as restricións asociadas ao escenario s .

Na práctica, a resolución do problema estocástico empregando esta metodoloxía redúcese a considerar, nun único problema, as R^H ramas distintas da árbore de decisión, relacionando entre elas as variables binarias correspondentes á unha mesma etapa, cando toda a incerteza ata a mesma coincide.

No seguinte capítulo verase como se comportan estas metodoloxías na práctica, como xa apuntábase anteriormente.

Capítulo 5

Validación

Nos Capítulos 2 e 4 contextualizamos os problemas de optimización en redes de gas determinista e estocástico, respectivamente. Neste capítulo amósase parte do traballo levado a cabo nas prácticas realizadas nas instalacións do ITMATI dende Setembro de 2016 ata Febreiro de 2017, ambos meses incluídos. O principal obxectivo destas prácticas era validar a implementación do PHA levada a cabo polo equipo investigador do proxecto GANESO. O proxecto GANESO é un proxecto de investigación desenvolto conxuntamente pola empresa Regasificadora do Noroeste S.A. (Reganosa) e un grupo de investigación formado por persoal do ITMATI e da Universidade de Santiago de Compostela. Este proxecto consiste no desenvolvemento dunha ferramenta informática para a simulación e optimización de redes de gas, coñecida como GANESO. Esta permite ao usuario obter, de maneira interactiva, a distribución de fluxos óptima en base a distintos criterios, o cálculo de tarifas, a simulación en estados transitorios e a planificación de ampliacións da rede baixo incerteza. Na Sección 5.1 introdúcese brevemente esta ferramenta.

As prácticas vinculadas a este traballo, como se mencionou anteriormente, foron desenvolvidas nas instalacións do ITMATI. O Consorcio “Instituto Tecnológico de Matemática Industrial (ITMATI)” é un consorcio público no que participan as tres universidades galegas, a Universidade da Coruña (UDC), a Universidade de Vigo (UVIGO) e a Universidade de Santiago de Compostela (USC), que empezou as súas actividades no 2013. Ten a vocación de converterse nun centro de investigación tecnolóxica de referencia a nivel internacional no ámbito da Matemática Industrial. A súa principal misión é proporcionar solucións innovadoras aos sectores produtivos da sociedade, especialmente a empresas, industrias e administracións públicas en temas moi diversos como o control de calidade, a simulación e optimización de procesos, a planificación de recursos, etc.

A descrición do traballo e resultados obtidos pode verse con máis detalle en [5]. Este informe foi elaborado para amosar o labor desenvolvido e os resultados obtidos nas prácticas ao equipo investigador do proxecto GANESO e á empresa Reganosa. Na Sección 5.2 describimos brevemente en que consiste a validación e nas Seccións 5.3 e 5.4 amosamos os resultados obtidos para a rede de gas galega e da rede de gas española, respectivamente. Finalmente, na Sección 5.5 recóllense as conclusións obtidas e observacións para a continuación da labor desenvolvida nestas prácticas.

5.1. GANESO

GANESO (GAs Network Simulation and Optimization) é un software desenvolto por un equipo investigador da Universidade de Santiago de Compostela e o Instituto Tecnológico de Matemática Industrial (ITMATI), entre os que se atopan o titor e director deste traballo. Esta ferramenta, como xa se mencionou, foi desenvolto no marco dun proxecto de investigación coa empresa Regasificadora

do Noroeste S.A. que comezou a súa andaina no 2012 e segue vixente na actualidade.

GANESO está elaborada como unha aplicación software compatible cos actuais sistemas operativos Windows. Está formada por dúas compoñentes claramente diferenciadas: o núcleo de cálculo e a interface gráfica de usuario. O núcleo de cálculo está implementado principalmente en linguaxe Fortran 2003. A interface gráfica de usuario foi desenvolvida como un plug-in para o programa Quantum GIS, un sistema de información xeográfica de código aberto, en código Python. Na Figura 5.1 amósase a interface gráfica de GANESO cun escenario da rede de gas española cargado. Resaltado mediante un rectángulo amarelo as iconas da interface gráfica en cuestión.

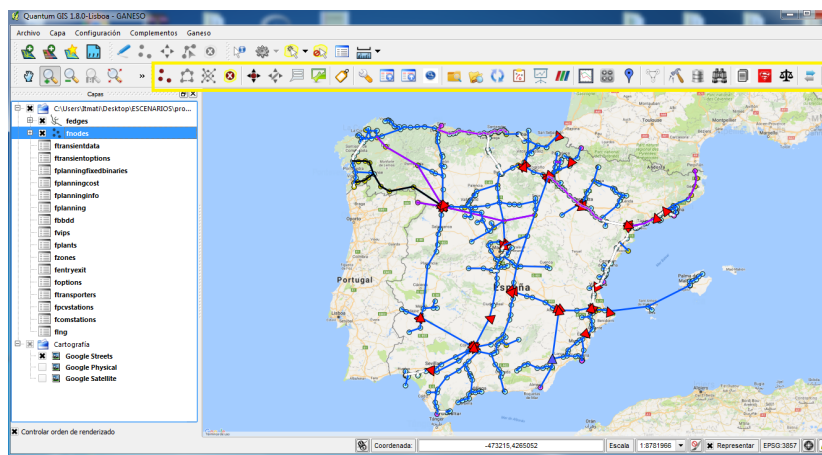


Figura 5.1: Interface gráfica de GANESO.

A entrada e saída de datos para a comunicación do núcleo de cálculo e a interface faise con arquivos XML compatibles con Microsoft Excel. Un escenario consiste nun ficheiro en extensión .xml onde cada folia de cálculo contén unha táboa, e o conxunto define todas as propiedades do problema. Na Figura 5.2 amósase a folia de opcións do escenario da rede galega en formato XML.

	A	B	C	D	E	F	G	H	I	J
15	plan_uncertainty	2								
16	plan_type	lagrange								
17	plan_stages	3								
18	plan_stage_trunk	0								
19	plan_stage_ev	0								
20	plan_retribution_tax	0,0509								
21	plan_penalization_bigm	10000								
22	plan_gas_price	0,036								
23	plan_exhaustive	0								
24	plan_deltat	1								
25	plan_alpha_inc	1000								
26	plan_alpha_gc	0,5								
27	plan_alpha_cc	0,5								
28	plan_additive	1								
29	pini_slp	60								
30	penalty_slp	0,001								
31	penalty_pcv	1								

Figura 5.2: Escenario da rede de gas galega en formato XML.

GANESO presenta diversos módulos que permiten, a través da interface: simular o estado dunha

rede de gas (en estado estacionario e transitorio), obter o punto óptimo de funcionamento da rede para minimizar custos de operación, determinar tarifas en función de diversas metodoloxías, determinar a potencia calorífica dos puntos de emisión a partir da calidade do gas introducido na rede e facer estudos sobre a planificación de novas infraestruturas baixo incerteza.

Os problemas no contexto das redes de gas cos que tratamos durante as prácticas son os introducidos no Capítulo 4, que se enmarcan no módulo de planificación de infraestruturas de transporte en base ás previsións da demanda de gas natural do sistema. Este módulo está baseado en técnicas de optimización baixo incerteza, en programación estocástica, e xurdiu para dar resposta á pregunta: ¿Como construír ou modificar a rede de transporte de gas natural da forma máis eficiente económica e tecnicamente posible, para unha demanda prevista?. A incerteza afecta á demanda de gas na rede, pero tamén ao prezo por unidade de gas e ao custo por infraestrutura e inténtase optimizar atendendo aos custos de construción e operación e ao posible incumprimento da demanda. É dicir, interesa reducir:

- Custos de operación: os autoconsumos das estacións de compresión.
- Custos de construción.
- Custos por incumprimento de demanda: págase un custo por cada unidade de demanda insatisfeita en puntos de consumo e por cada unidade que nos afastemos dos caudais impostos en puntos de subministración.

Considérase que a demanda de gas só pode aumentar ou diminuír, cunha certa probabilidade, respecto a etapa inmediatamente anterior. Isto permite a modelización da situación resultante mediante un diagrama de árbore no que cada aresta se corresponde cunha variación do consumo dunha etapa a outra, tendo en conta as oscilacións de demanda nas etapas xa pasadas, así como, a incorporación das infraestruturas candidatas a construír.

GANESO permite resolver estes problemas e o noso obxectivo é validar a implementación do algoritmo de resolución implementado polo equipo investigador do proxecto. Para isto, dispónse do escenario da rede de gas galega e española en formato XML.

5.2. Labor desenvolvido nas prácticas

As primeiras semanas de prácticas consisten nunha formación básica sobre a ferramenta de GANESO descrita na sección anterior. Para isto, revísase a bibliografía necesaria, apoiándose nos manuais que elaboran dentro do proxecto. Apréndese o necesario para definir unha rede de gas e como sería o procedemento para executar os diferentes módulos que compoñen o programa. Esta formación serve para localizar a información relevante para levar a cabo o obxectivo das prácticas. Traballamos cos arquivos XML que definen os escenarios da rede de gas galega e española e con scripts elaborados na linguaxe de programación R, que se poden ver no Apéndice B, sen necesidade de acceder a interface.

A continuación enumeramos os pasos que se levan a cabo para a validación do algoritmo implementado:

1. O primeiro que se precisa é unha batería de casos da que non se dispoñía. Polo que se implementou unha función que permite xerar automaticamente novos escenarios, aos que nos referimos como casos, modificando os parámetros de planificación dun escenario de partida. Ademais garda cada caso nunha carpeta independente de nome DEM e LAG, dado que se vai resolver o problema asociado a cada caso empregando as dúas metodoloxías.
2. Logo crease un batch que chama ao executable de ganeso para resolver o problema asociado a cada caso empregando o método de resolución DEM e o algoritmo *progressive hedging* (PHA) xunto co

algoritmo CSLP estocástico, empregando o solver gurobi. Para a resolución dos distintos casos, que se amosan nas seguintes seccións, empregouse o clúster Mariscal que se atopa na Facultade de Matemáticas da USC.

- Finalizada a resolución para cada caso, na carpeta correspondente temos unha serie de ficheiros, na Figura 5.3 amósanse os arquivos xerados tras a resolución correcta dun problema. Os ficheiros `1_slp_sto.xml`, `1_slp_sto.kml`, `planning.log` e `ganeso_output.txt` conteñen información relativa á resolución. Ademais dos mencionados anteriormente poden aparecer os arquivos `info_by_node.txt`, `NF_problem.xml`, `ISS_problem.ilp` e `ALL_problem.lp`. Os tres últimos xorden como consecuencia de que gurobi non é capaz de resolver o problema ou o problema presenta algunha infactibilidade.


 1_slp_sto.kml	16/01/2017 10:10	Archivo KML	3.607 KB
 1_slp_sto	16/01/2017 10:10	Documento XML	3.188 KB
 ganeso_output	16/01/2017 10:10	Documento de tex...	839 KB
 info_by_node	16/01/2017 10:10	Documento de tex...	7 KB
 planning	16/01/2017 10:10	Documento de tex...	132 KB
 prueba	16/01/2017 9:39	Documento XML	264 KB

Figura 5.3: Arquivos xerados tras a resolución dun problema.

Impleméntase unha función que permite extraer dos ficheiros anteriores información de interese para validar o algoritmo. A información correspondente a cada metodoloxía de resolución almacénase nunha táboa, cuxa cabeceira se amosa na Figura 5.1. Cada fila correspóndese cun caso da batería xerada a partir do XML que define o escenario de partida e as columnas que a forman son:

- $SumGcost_inc(gc,cc)$ fai referencia á suma dos custos de incumprimento acumulado (gas consumido nas estacións de compresión en funcionamento, custos de construción) ponderada polas probabilidades acumuladas de cada ganesito ¹ e polos pesos que acompañan a cada termo na función obxectivo.
- $Sumcost_inc(gc,cc)$ fai referencia á suma dos custos de incumprimento acumulado (gas consumido nas estacións de compresión en funcionamento, custos de construción) ponderada pola probabilidade acumulada dos distintos escenarios da árbore de decisión que presenta o escenario de partida.
- $Infact$ recolle o valor medio do incumprimento.
- δ_{SLP} e δ_{CSLP} recolle a diferenza entre iterantes sucesivos empregando o SLP e o CSLP. No caso de aplicar o PHA estes valores obtéñense multiplicando o δ_{slp} e δ_{cslp} pola probabilidade de cada rama que interveña no proceso iterativo.
- $Tipo\ error$ recolle a codificación do erro do solver (solve code)².
- $\exists _problem$ ³ e $\exists 1_slp_sto$ fan referencia a existencia (1) ou non (0) do arquivo con ese nome. A primeira columna fai referencia á existencia de arquivos xerados no caso de que o solver non é capaz de facer fronte á resolución, por exemplo pola dimensión do problema.

¹Chámase ganesito a cada nodo da árbore de decisión, o cal ten un escenario asociado.

²Algúns dos posibles códigos de estado de gurobi (solve code) son os seguintes: 1 (o modelo foi cargado pero non hai información da solución dispoñible), 3 (o modelo é infactible).

³Arquivo xerado por gurobi no caso de existir algunha infactibilidade.

α_{inc}	α_{gc}	α_{cc}	SumGcost_inc	SumGcost_gc	SumGcost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc ...
... Infact delta_SLP delta_CSLP Erro solver \exists _problem \exists 1_slp_sto								

Táboa 5.1: Cabeceira das táboas recollendo os resultados.

4. Elabórase un script para comparar os resultados obtidos para os distintos casos considerados a partir do escenario de partida. Para isto, impleméntase unha función que permite obter gráficas de radar e táboas, similares as que se poden facer na interface de GANESO, Figura 5.4, para ver:

- Uso de infraestruturas. Pódese coñecer o tanto por cen de uso de cada infraestrutura mediante a representación da gráfica de radar e a táboa coa información asociada. Amósanse os resultados referentes a última etapa.
- Construción por etapas. Pódese visualizar as diferentes infraestruturas consideradas e a primeira etapa na que aparecerían construídas.

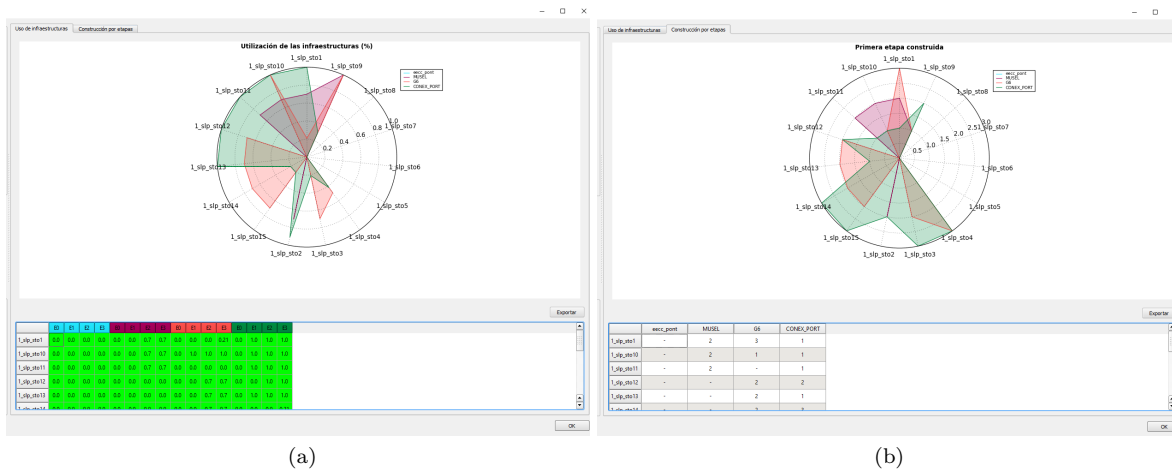


Figura 5.4: Gráficas de radar na interface de GANESO.

5. Tamén se recolle información sobre os tempos de resolución de cada caso empregando as dúas metodoloxías e o número de iteracións do PHA.

Chegar a obter a táboa descrita no punto 3 levou o seu tempo, xa que nos arquivos xerados non se escribía a información necesaria para completar as columnas $SumGcost_inc(gc,cc)$. Ademais tamén nos atopamos que había un erro na chamada ao solver gurobi, que levaba a que un mesmo escenario se resolvese en local pero non no clúster Mariscal. Isto levou a obter unha nova versión do módulo, modificando o código Fortran para que nos arquivos xerados se amosara esa información e ese erro non ocorrese. Mencionar que a extracción sobre os tempos de resolución foi unha tarefa secundaria que non se abordou desde un primeiro momento, puido levarse a cabo porque no Mariscal se xera un arquivo auxiliar con información referente a cada resolución, de feito se fixésemos a execución en local tal e como está implementado a creación do batch non disporíamos desta información, xa que, non se fai por defecto e se precisa indicar que se garde o transcurso da execución nun arquivo.

Unha descrición máis detallada da labor desenvolvida durante as prácticas pódese consultar en [5], como xa se mencionou na introdución deste capítulo.

5.3. Rede de gas galega

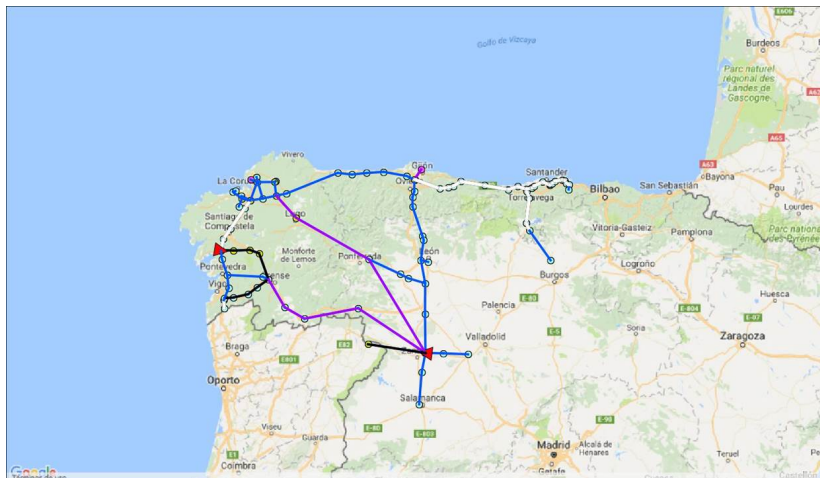


Figura 5.5: Representación da rede de gas galega na interface de GANESO.

Nesta sección imos amosar os resultados obtidos da resolución dunha batería de casos obtidos tras modificar os parámetros de planificación α_{inc} e α_{cc} dun escenario da rede de gas galega. Antes de iso, describimos brevemente en que consiste este escenario e o que o lector se vai atopar a continuación.

O escenario correspondente a rede de gas galega, representado na Figura 5.5 presenta 4 posibles infraestruturas planificables, coñecidas como: a estación de compresión de Pontevedra (eccc_pont), Musel, o G6 e o G6 Sur ou conexión con Portugal (CONEX_PORT). O número de etapas que se consideran na formulación do problema de programación estocástica é tres e a duración das etapas é dun ano por etapa. Ademais, presenta dous posibles resultados da incerteza e hai un incremento aditivo na distribución da demanda e das probabilidades.

Para este escenario lanzouse unha batería de casos da rede de gas galega anterior, resultante de modificar os parámetros de planificación α_{inc} entre 1000 e 3000 con paso 500 e α_{cc} entre 0.5 e 6.5 con paso 0.5. Con estas combinacións observouse que o algoritmo non era sensible á modificación do valor do parámetro α_{inc} e non se apreciou o patrón de cambio que se esperaba nas sumas de custos, omitimos a ilustración destes resultados, que se pode consultar en [5]. Isto levounos a un novo enfoque, considerar dous valores para α_{inc} (10 e 3000) e ver como cambiaba a suma de custos modificando o parámetro α_{cc} pero con valores máis “extremos” (0.10, 0.5, 10, 100, 500, 3000, 10000). Os resultados desta nova batería de casos amósanse na Subsección 5.3.1, baixo o nome “Resultados con escenario de partida galicia1”. Penseuse en que sería interesante ver como afectaba á suma de custos a modificación das probabilidades da árbore. Así, considerouse o escenario de partida da rede de gas galega, galicia1, e modificáronse as probabilidades de demanda, obténdose un escenario con igual probabilidade nas dúas ramas, que se denota como galicia2 e outro con probabilidades de demanda 0.9 na rama superior e 0.1 na inferior, galicia3. Nestes, lanzouse unha batería de casos considerando a mesma modificación dos parámetros de planificación que se considerara na Subsección 5.3.1. Os resultados con escenario de partida galicia2 e galicia3 recóllense nas Subseccións 5.3.2 e 5.3.3, respectivamente.

Para finalizar esta sección compararemos os resultados obtidos considerando os tres escenarios e as conclusión obtidas destes.

5.3.1. Resultados con escenario de partida galicial

O escenario ao que nos referimos como galicial recordamos que presenta 3 etapas e dous posibles resultados da incerteza, con probabilidades 0.7 e 0.3. A árbore de decisión, asociada a todos os casos obtidos a partir deste, represéntase na Figura 5.6. A batería de casos que se considera é resultado de combinar os valores dos parámetros de planificación $\alpha_{gc} = 0.5$, $\alpha_{inc} = 10, 3000$ e $\alpha_{cc} = 0.10, 0.5, 10, 100, 500, 3000, 10000$.

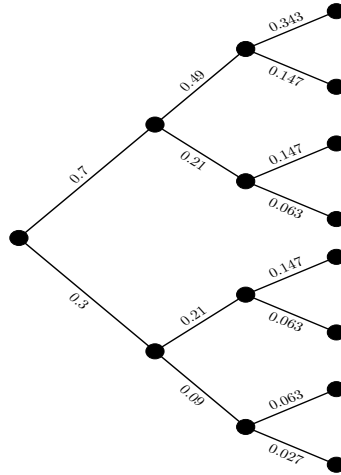


Figura 5.6: Árbore de decisión do escenario galicial1.

Na Táboa 5.2 recóllense os resultados de aplicar o método DEM na resolución do problema estocástico. Nas Táboas 5.3 e 5.4 podemos ver que infraestruturas son construídas en cada caso, o uso que se fai delas e en que etapa aparecen construídas. Podemos ver que:

- Os casos 5, 6, 7 e 8 teñen un valor de custos de construción igual a cero, dado que non se constrúe ningunha infraestrutura. Ademais o SLP non converge nestes casos pero si o CSLP.
- Para os casos 3 e 15 (resaltados en amarelo), que teñen α_{inc} igual a α_{cc} temos o mesmo valor de Sumcost_inc e Sumcost_cc. En ambos casos decídese construír o G6 e o G6-Sur e o uso destas infraestruturas é o mesmo.
- O valor de Sumcost_cc diminúe ao aumentar o valor de α_{cc} , excepto para o caso 13 resaltado en azul ($\alpha_{inc} = 3000$ e $\alpha_{cc} = 500$). Para $\alpha_{inc} = 3000$ fixo, temos o mesmo custo de incumprimento e construción cando α_{cc} é maior ou igual a metade do valor anterior.
- O valor de Sumcost_inc é similar en todos os casos, excepto para aqueles con $\alpha_{inc} = 10$ e $\alpha_{cc} \geq 100$ (casos 5, 6, 7 e 8), os cales se corresponden cos casos onde non se constrúe ningunha infraestrutura. Os restantes, teñen unha diferenza menor a unha unidade.

Omitimos a representación das gráficas de radar da información contida nas Táboas 5.3 e 5.4. Nestas, ademais do mencionado anteriormente podemos ver que a construción do G6 e G6-Sur van da man na maioría dos casos, sendo a porcentaxe de uso na etapa final maior no caso do G6-Sur. Só en dous casos (9 e 10) sae construída Musel.

	α_{inc}	α_{gc}	α_{cc}	SumGeost_inc	SumGeost_gc	SumGeost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSPL	Erro solver	\exists NF_problem	\exists 1_slp_sto
1	10	0.50	0.10	945.29983	0.00000	5.60189	94.52998	0.00000	56.01892	0.00000	3.25020e-05	2.87571e-12	0	0	1
2	10	0.50	0.50	945.29983	0.00000	30.63787	94.52998	0.00000	61.27573	0.00000	7.96484e-06	3.71877e-11	0	0	1
3	10	0.50	10	950.85930	0.00000	452.81012	95.08593	0.00000	45.28101	0.00000	6.55914e-06	3.81071e-11	0	0	1
4	10	0.50	100	1759.70954	0.00000	2822.17423	175.97095	0.00000	28.22174	0.00000	1.07394e-05	1.44519e-11	0	0	1
5	10	0.50	500	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	7.09450e+05	1.95985e-12	0	0	1
6	10	0.50	1500	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	4.02671e+04	7.99838e-13	0	0	1
7	10	0.50	3000	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	2.80303e+05	1.06786e-14	0	0	1
8	10	0.50	10000	6700.53559	0.00000	0.00000	670.05356	0.00000	0.00000	0.00023	1.15983e+05	5.79884e-05	0	0	1
9	3000	0.50	0.10	283589.94976	0.00000	13.01060	94.52998	0.00000	130.10601	0.00000	3.26953e-05	2.87620e-13	0	0	1
10	3000	0.50	0.50	283589.94977	0.00000	51.24013	94.52998	0.00000	102.48025	0.00000	5.32523e-05	6.79739e-11	0	0	1
11	3000	0.50	10	283589.94977	0.00000	612.75731	94.52998	0.00000	61.27573	0.00000	9.40916e-06	6.30389e-11	0	0	1
12	3000	0.50	100	283589.94976	0.00000	6127.57310	94.52998	0.00000	61.27573	0.00000	7.16842e-06	2.40998e-13	0	0	1
13	3000	0.50	500	285257.78861	0.00000	14425.22968	95.08593	0.00000	28.85046	0.00000	2.17655e-07	1.36473e-14	0	0	1
14	3000	0.50	1500	285257.78861	0.00000	67921.51750	95.08593	0.00000	45.28101	0.00000	1.09797e-01	2.32345e-13	0	0	1
15	3000	0.50	3000	285257.78861	0.00000	135843.03500	95.08593	0.00000	45.28101	0.00000	6.91466e+04	3.69666e-13	0	0	1
16	3000	0.50	10000	285259.01925	0.00000	452810.11667	95.08634	0.00000	45.28101	0.00033	3.53199e-05	3.44882e-05	0	0	1

Táboa 5.2: Resultados DEM asociados á batería de casos obtida do escenario galicia1.

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	1.00	0.00
1_slp_sto 2	0.00	0.00	0.70	1.00
1_slp_sto 3	0.00	0.00	0.70	0.91
1_slp_sto 4	0.00	0.00	0.49	0.91
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	1.00	0.00	0.00
1_slp_sto 10	0.00	0.70	0.00	1.00
1_slp_sto 11	0.00	0.00	0.70	1.00
1_slp_sto 12	0.00	0.00	0.70	1.00
1_slp_sto 13	0.00	0.00	0.70	0.21
1_slp_sto 14	0.00	0.00	0.70	0.91
1_slp_sto 15	0.00	0.00	0.70	0.91
1_slp_sto 16	0.00	0.00	0.70	0.91

Táboa 5.3: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia1 (DEM).

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	1.00	0.00
1_slp_sto 2	0.00	0.00	2.00	1.00
1_slp_sto 3	0.00	0.00	2.00	2.00
1_slp_sto 4	0.00	0.00	3.00	2.00
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	1.00	0.00	0.00
1_slp_sto 10	0.00	2.00	0.00	1.00
1_slp_sto 11	0.00	0.00	2.00	1.00
1_slp_sto 12	0.00	0.00	2.00	1.00
1_slp_sto 13	0.00	0.00	2.00	3.00
1_slp_sto 14	0.00	0.00	2.00	2.00
1_slp_sto 15	0.00	0.00	2.00	2.00
1_slp_sto 16	0.00	0.00	2.00	2.00

Táboa 5.4: Etapa de construción da infraestrutura para a batería de casos do escenario galicia1 (DEM).

	α_{inc}	α_{gc}	α_{cc}	SumGcost_inc	SumGcost_gc	SumGcost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSLP	Erro solver	\exists NF_problem	\exists 1_slp_sto
1	10	0.50	0.10	945.29983	0.00000	9.90698	94.52998	0.00000	99.06978	0.00000	3.04653e-05	2.94622e-11	0	0	1
2	10	0.50	0.50	950.85930	0.00000	39.54344	95.08593	0.00000	79.08688	0.00000	1.94446e-05	5.58892e-11	0	0	1
3	10	0.50	10	950.85930	0.00000	288.50459	95.08593	0.00000	28.85046	0.00000	2.22103e-05	6.93036e-11	0	0	1
4	10	0.50	100	2777.62153	0.00000	1427.66075	277.76215	0.00000	14.27661	0.00000	3.40541e+05	6.12431e-12	0	0	1
5	10	0.50	500	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	2.06681e+05	1.77196e-13	0	0	1
6	10	0.50	1500	6710.14776	0.00000	0.00000	671.01478	0.00000	0.00000	0.00000	2.06681e+05	1.76178e-05	0	0	1
7	10	0.50	3000	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	2.06681e+05	2.69031e-13	0	0	1
8	10	0.50	10000	6700.53106	0.00000	0.00000	670.05311	0.00000	0.00000	0.00000	2.06681e+05	4.43079e-11	0	0	1
9	3000	0.50	0.10	283590.07910	0.00000	16.69401	94.53003	0.00000	166.94009	0.00000	2.17204e-05	2.49041e-06	0	0	1
10	3000	0.50	0.50	283589.94976	0.00000	69.98016	94.52998	0.00000	139.96032	0.00000	1.67300e-05	1.39942e-12	0	0	1
11	3000	0.50	10	283589.94976	0.00000	950.81595	94.52998	0.00000	95.08160	0.00000	5.06080e-05	7.90835e-13	0	0	1
12	3000	0.50	100	285257.78861	0.00000	4983.72595	95.08593	0.00000	49.83726	0.00000	1.72759e-05	4.45618e-13	0	0	1
13	3000	0.50	500	283589.94976	0.00000	30637.86550	94.52998	0.00000	61.27573	0.00000	2.16573e-05	5.67221e-13	0	0	1
14	3000	0.50	1500	285257.78861	0.00000	43275.68905	95.08593	0.00000	28.85046	0.00000	2.85891e-05	7.77868e-15	0	0	1
15	3000	0.50	3000	285257.78861	0.00000	86551.37810	95.08593	0.00000	28.85046	0.00000	2.85891e-05	7.77868e-15	0	0	1
16	3000	0.50	10000	285259.43957	0.00000	288504.59366	95.08648	0.00000	28.85046				3	1	1

Táboa 5.5: Resultados PHA asociados á batería de casos obtida do escenario galicia1.

Na Táboa 5.5 recóllense os resultados de aplicar o PHA na resolución do problema, pódese ver que:

- Os casos 5, 6, 7 e 8 teñen un valor de custos de construción igual a cero (as columnas de nome *SumGcost_cc* e *Sumcost_cc* son cero) e un valor elevado de delta_SLP, non converge o SLP.
- O caso 16 resulta ser infactible, o solver que se emprega (gurobi) devolve o código 3. Analizando os arquivos que se xeran tense que o problema linealizado neste caso non ten solución. Obtéñense valores para as sumas de custos pero non son representativos.
- Observamos que a medida que aumenta o parámetro α_{cc} tamén o fai o valor do custo de construción *SumGcost_cc* mentres que *Sumcost_cc* diminúe, excepto para o caso 13 resaltado en azul. Podemos ver tamén, que efectivamente, o valor en *SumGcost_cc* é igual ao valor de *Sumcost_cc* multiplicado polo parámetro α_{cc} . Tamén podemos ver que estes valores son maiores cando α_{inc} é grande.
- Para os casos con *Sumcost_cc* igual a cero temos un valor similar de *Sumcost_inc* (670.05). Para os restantes o valor de *Sumcost_inc*, excepto para o escenario 4, é similar, non se chegan a apreciar un cambio dunha unidade, obténdose o mesmo valor (94.53) para os casos 1, 9, 10, 11 e 13 e para o 2, 3, 12, 14 e 15 (95.09). Estes teñen tamén o mesmo valor en *SumGcost_inc* cando se considera o mesmo parámetro α_{inc} .

Pasamos agora a comparar os casos anteriores. Para isto, fixémonos nas Táboas 5.6 e 5.7 ou nas gráficas de radar asociadas, Figura 5.7. Vemos que:

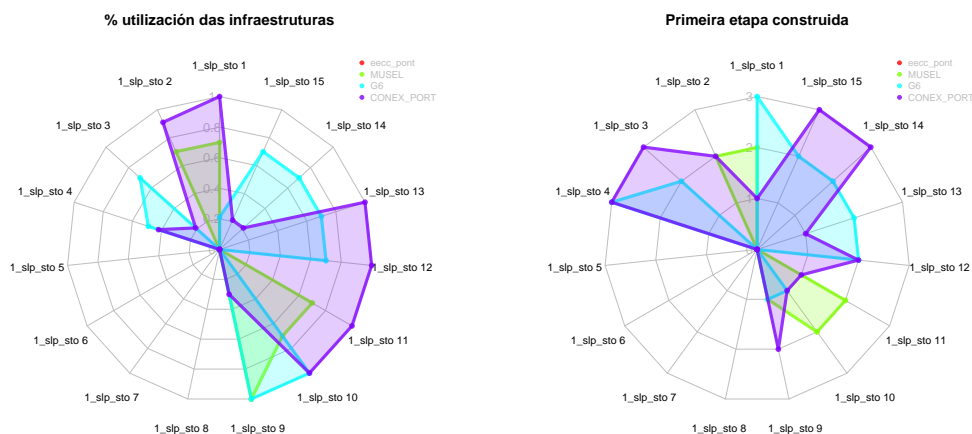
- Nos casos 5, 6, 7 e 8 non se constrúe ningunha infraestrutura de aí que o valor en *SumGcost_cc* e *Sumcost_cc* sexa cero.
- Os casos co mesmo valor en *Sumcost_cc*, por exemplo os casos 3 e 15, resaltados en amarelo, que se corresponden cos casos en que os parámetros de planificación α_{inc} e α_{cc} son iguais, fan uso das mesmas infraestruturas planificables, G6 e G6-Sur. O uso do G6-Sur é máis pequeno que o uso que resulta aplicando o método DEM na resolución, dado que esta infraestrutura aparece construída na segunda etapa aplicando o método DEM e na terceira etapa aplicando o PHA, e as probabilidades son menores neste último.

- Os escenarios con maior valor de $Sumcost_{cc}$, casos 1, 9 e 10, correspóndense con aqueles onde todas as infraestruturas planificables, excepto `eecc_pont` que non se constrúe en ningún caso, son construídas.

	eecc_pont	MUSEL	G6	CONEX_PORT		eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.70	0.21	1.00	1_slp_sto 1	0.00	2.00	3.00	1.00
1_slp_sto 2	0.00	0.70	0.00	0.91	1_slp_sto 2	0.00	2.00	0.00	2.00
1_slp_sto 3	0.00	0.00	0.70	0.21	1_slp_sto 3	0.00	0.00	2.00	3.00
1_slp_sto 4	0.00	0.00	0.49	0.42	1_slp_sto 4	0.00	0.00	3.00	3.00
1_slp_sto 5	0.00	0.00	0.00	0.00	1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00	1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00	1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00	1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	1.00	1.00	0.30	1_slp_sto 9	0.00	1.00	1.00	2.00
1_slp_sto 10	0.00	0.70	1.00	1.00	1_slp_sto 10	0.00	2.00	1.00	1.00
1_slp_sto 11	0.00	0.70	0.00	1.00	1_slp_sto 11	0.00	2.00	0.00	1.00
1_slp_sto 12	0.00	0.00	0.70	1.00	1_slp_sto 12	0.00	0.00	2.00	2.00
1_slp_sto 13	0.00	0.00	0.70	1.00	1_slp_sto 13	0.00	0.00	2.00	1.00
1_slp_sto 14	0.00	0.00	0.70	0.21	1_slp_sto 14	0.00	0.00	2.00	3.00
1_slp_sto 15	0.00	0.00	0.70	0.21	1_slp_sto 15	0.00	0.00	2.00	3.00

Táboa 5.6: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicial (PHA). Táboa 5.7: Etapa de construción da infraestrutura para a batería de casos do escenario galicial1 (PHA).

Na Táboa 5.8 recóllense os tempos de execución de aplicar o método DEM e o PHA ás configuracións de parámetros consideradas, e o número de iteracións que tardou en converxer o PHA en cada caso. Podemos ver que en varios casos tarda máis dunha hora a execución (tempos ≥ 3600). Observamos que nos casos con maior número de iteracións temos que o tempo de converxencia empregando o PHA é maior, por exemplo o caso 12 cuxo tempo de execución real pasa das 7 horas.



(a) Porcentaxe de uso de cada infraestrutura na etapa final.

(b) Primeira etapa construída.

Figura 5.7: Gráficas de radar asociadas á batería de casos a comparar obtida do escenario galicia1.

	CPU time		Real time		Iteracións
	CPU_DEM	CPU_PHA	Real_DEM	Real_PHA	PHA
1	253.16	3272.63	101.77	1675.58	27
2	245.18	8433.72	97.77	4260.91	66
3	278.10	1444.16	108.47	732.33	13
4	251.46	1635.12	107.61	815.14	12
5	181.88	59.40	81.89	41.60	2
6	86.80	54.34	50.27	39.51	2
7	64.72	33.72	46.15	25.56	2
8	83.80	110.07	67.88	81.09	2
9	133.80	3998.79	75.71	2175.24	34
10	168.37	7979.26	91.04	4572.62	131
11	280.02	9608.81	112.45	4890.87	66
12	318.47	60776.97	135.78	27011.77	322
13	310.87	4927.89	115.29	2373.55	43
14	295.72	4260.25	120.40	2018.68	31
15	293.00	3225.06	118.75	1550.02	24
16	438.50	11743.97	211.39	5302.94	86

Táboa 5.8: Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia1 e número de iteracións do PHA.

5.3.2. Resultados con escenario de partida galicia2

O escenario ao que nos referimos como galicia2 recordamos que presenta 3 etapas e dous posibles resultados da incerteza, coa mesma probabilidade. A árbore de decisión, asociada a todos os casos obtidos a partir deste, represéntase na Figura 5.8. As combinacións de parámetros empregadas son as mesmas que as consideradas no apartado anterior.

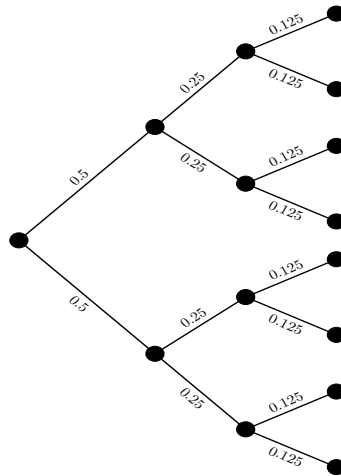


Figura 5.8: Árbore de decisión do escenario galicia2.

Na Táboa 5.9 recóllense os resultados de aplicar o método DEM na resolución do problema estocástico. Pódese ver que:

- Os casos 5, 6, 7 e 8 teñen un valor de custos de construción igual a cero, dado que non se constrúe ningunha infraestrutura planificable. Ademais o SLP non converxe nestes casos.
- Para os casos 3 e 15 (resaltados en amarelo), que teñen α_{inc} igual a α_{cc} temos o mesmo valor de $Sumcost_{inc}$ e $Sumcost_{cc}$. En ambos escenarios decídese construír o G6 e o G6-Sur e o uso destas infraestruturas é o mesmo. O mesmo ocorre noutros casos nos que os parámetros son distintos (casos 1, 2, 9, 10, 11, 12, 13 e 14).
- O valor de $SumGcost_{cc}$ aumenta ao aumentar o valor de α_{cc} . Nesta batería de casos ao aplicar o método DEM obsérvase que non se aprecia claramente que os custos da columna $Sumcost_{cc}$ diminúen ao aumentar o valor do parámetro α_{cc} , dado que nos casos 9 a 14 temos o mesmo valor de $Sumcost_{cc}$.
- O valor de $Sumcost_{inc}$ é similar en todos os casos, excepto para os escenarios cuxa suma de custos por construción é cero e para o escenario con $\alpha_{inc} = 10$ e $\alpha_{cc} = 100$. Nos restantes, a diferenza é menor a unha unidade.

Nas Táboas 5.10 e 5.11 pódese ver que infraestruturas son construídas, en que etapa e que uso se fai delas. A construción do G6 e G6-Sur vai da man en todos os casos, sendo a porcentaxe de uso na etapa final maior no caso do G6-Sur. En ningún caso se constrúe Musel.

	α_{inc}	α_{gc}	α_{cc}	SumGeost_inc	SumGeost_gc	SumGeost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSPL	Erro solver	\exists NF_problem	\exists 1_slp_sto
1	10	0.50	0.10	344.49702	0.00000	5.37429	34.44970	0.00000	53.74286	0.00000	6.17717e-05	4.09323e-11	0	0	1
2	10	0.50	0.50	344.49702	0.00000	26.87143	34.44970	0.00000	53.74286	0.00000	5.06571e-05	4.09192e-11	0	0	1
3	10	0.50	10	348.46807	0.00000	335.27112	34.84681	0.00000	33.52711	0.00000	5.06229e-05	3.90464e-11	0	0	1
4	10	0.50	100	1280.48962	0.00000	1066.55037	128.04896	0.00000	10.66550	0.00000	2.51996e+04	2.35208e-15	0	0	1
5	10	0.50	500	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	1.42238e+05	2.19896e-12	0	0	1
6	10	0.50	1500	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	7.25439e+04	3.58288e-13	0	0	1
7	10	0.50	3000	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	4.57400e+03	6.23189e-13	0	0	1
8	10	0.50	10000	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	1.39374e+05	6.30392e-11	0	0	1
9	3000	0.50	0.10	103349.10706	0.00000	5.37429	34.44970	0.00000	53.74286	0.00000	2.23849e-05	8.52998e-13	0	0	1
10	3000	0.50	0.50	103349.10706	0.00000	26.87143	34.44970	0.00000	53.74286	0.00000	6.17751e-05	7.16582e-13	0	0	1
11	3000	0.50	10	103349.10706	0.00000	537.42862	34.44970	0.00000	53.74286	0.00000	6.18448e-05	1.07588e-12	0	0	1
12	3000.00	0.50	100	103349.10706	0.00000	5374.28619	34.44970	0.00000	53.74286	0.00000	1.20073e-05	1.41036e-12	0	0	1
13	3000.00	0.50	500	103349.10706	0.00000	26871.43094	34.44970	0.00000	53.74286	0.00000	5.30570e-05	6.85223e-14	0	0	1
14	3000.00	0.50	1500	103349.87672	0.00000	80614.29281	34.44996	0.00000	53.74286	0.00022	1.84793e-05	9.75673e-06	0	0	1
15	3000	0.50	3000	104540.42052	0.00000	100581.33565	34.84681	0.00000	33.52711	0.00000	2.52004e+04	4.40885e-13	0	0	1
16	3000	0.50	10000	104540.42052	0.00000	247498.33400	34.84681	0.00000	24.74983	0.00000	6.91304e-05	6.30392e-11	0	0	1

Táboa 5.9: Resultados DEM asociados á batería de casos obtida do escenario galicia2.

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	0.50	1.00
1_slp_sto 2	0.00	0.00	0.50	1.00
1_slp_sto 3	0.00	0.00	0.50	0.75
1_slp_sto 4	0.00	0.00	0.25	0.50
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	0.00	0.50	1.00
1_slp_sto 10	0.00	0.00	0.50	1.00
1_slp_sto 11	0.00	0.00	0.50	1.00
1_slp_sto 12	0.00	0.00	0.50	1.00
1_slp_sto 13	0.00	0.00	0.50	1.00
1_slp_sto 14	0.00	0.00	0.50	1.00
1_slp_sto 15	0.00	0.00	0.50	0.75
1_slp_sto 16	0.00	0.00	0.50	0.50

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	2.00	1.00
1_slp_sto 2	0.00	0.00	2.00	1.00
1_slp_sto 3	0.00	0.00	2.00	2.00
1_slp_sto 4	0.00	0.00	3.00	3.00
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	0.00	2.00	1.00
1_slp_sto 10	0.00	0.00	2.00	1.00
1_slp_sto 11	0.00	0.00	2.00	1.00
1_slp_sto 12	0.00	0.00	2.00	1.00
1_slp_sto 13	0.00	0.00	2.00	1.00
1_slp_sto 14	0.00	0.00	2.00	1.00
1_slp_sto 15	0.00	0.00	2.00	2.00
1_slp_sto 16	0.00	0.00	2.00	3.00

Táboa 5.10: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia2 (DEM).

Táboa 5.11: Etapa de construción da infraestrutura para a batería de casos do escenario galicia2 (DEM).

A Táboa 5.12 contén os resultados de empregar o PHA na resolución e as Táboas 5.13 e 5.14 o uso e a etapa de construción das infraestruturas, respectivamente. Aquí, pódese ver:

- Os casos 5, 6, 7 e 8 teñen un valor de custos de construción igual a cero, dado que non se

constrúe ningunha infraestrutura planificable. Ademais presentan valores elevados de delta_SLP que indican que non converxe o SLP.

- O caso 16 resulta ser infactible, o solver que se emprega (gurobi) devolve o código 3. Analizando os arquivos que se xeran tense que o problema linealizado neste caso non ten solución. Obtéñense valores para as sumas de custos pero non son representativos.
- Obsérvase que a medida que aumenta o parámetro α_{cc} tamén o fai o valor do custo de construción SumGcost_cc mentres que Sumcost_cc diminúe, excepto no caso 13 resaltado en azul. Pódese ver que efectivamente, o valor de SumGcost_cc é igual ao valor de Sumcost_cc multiplicado polo parámetro α_{cc} . Tamén se aprecia que estes valores son maiores cando α_{inc} é grande.
- Para os casos con suma de custos de construción igual a cero tense un valor similar de Sumcost_inc . Para os restantes o valor de Sumcost_inc é similar, non se chega a apreciar unha diferenza no valor dunha unidade, obténdose o mesmo valor (34.4497) para os casos 1, 9, 10, 11, 12 e 13 e para o 2, 3, 14 e 15 (34.84681). Estes teñen tamén o mesmo valor en SumGcost_inc cando se considera o mesmo parámetro α_{inc} . A excepción dos casos anteriores é o escenario 4 ($\alpha_{inc} = 10$ e $\alpha_{cc} = 100$) que ten uns valores de incumprimento distintos dos demais.

α_{inc}	α_{gc}	α_{cc}	SumGcost_inc	SumGcost_gc	SumGcost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSLP	Erro solver	\exists NF_problem	\exists 1_slp_sto	
1	10	0.50	0.10	344.49702	0.00000	7.49381	34.44970	0.00000	74.93809	0.00000	2.63968e-05	5.81286e-11	0	0	1
2	10	0.50	0.50	348.46807	0.00000	16.76356	34.84681	0.00000	33.52711	0.00000	2.88907e-05	8.40950e-11	0	0	1
3	10	0.50	10	348.46807	0.00000	217.91003	34.84681	0.00000	21.79100	0.00000	2.67100e-05	6.06121e-11	0	0	1
4	10	0.50	100	1656.96031	0.00000	1173.61088	165.69603	0.00000	11.73611	0.00000	7.62742e+04	8.85730e-12	0	0	1
5	10	0.50	500	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	1.15089e+05	1.45585e-13	0	0	1
6	10	0.50	1500	3281.99901	0.00000	0.00000	328.19990	0.00000	0.00000	0.00000	1.15089e+05	6.42049e-06	0	0	1
7	10	0.50	3000	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	1.15089e+05	2.69045e-13	0	0	1
8	10	0.50	10000	3278.49439	0.00000	0.00000	327.84944	0.00000	0.00000	0.00000	1.15089e+05	6.83532e-11	0	0	1
9	3000	0.50	0.10	103349.10706	0.00000	13.08934	34.44970	0.00000	130.89337	0.00000	2.85733e-05	4.09042e-14	0	0	1
10	3000	0.50	0.50	103349.10706	0.00000	65.44668	34.44970	0.00000	130.89337	0.00000	1.47453e-05	9.96652e-15	0	0	1
11	3000	0.50	10	103349.10706	0.00000	537.42862	34.44970	0.00000	53.74286	0.00000	2.26400e-05	8.37792e-13	0	0	1
12	3000	0.50	100	103349.10706	0.00000	5374.28619	34.44970	0.00000	53.74286	0.00000	2.26400e-05	8.37792e-13	0	0	1
13	3000	0.50	500	103349.10706	0.00000	37469.04581	34.44970	0.00000	74.93809	0.00000	2.99554e-05	1.02311e-12	0	0	1
14	3000	0.50	1500	104540.42052	0.00000	32686.50465	34.84681	0.00000	21.79100	0.00000	3.03950e-05	8.01530e-15	0	0	1
15	3000	0.50	3000	104540.42052	0.00000	65373.00929	34.84681	0.00000	21.79100	0.00000	3.03950e-05	8.01530e-15	0	0	1
16	3000	0.50	10000	204006.39593	0.00000	188321.72794	68.00213	0.00000	18.83217				3	1	1

Táboa 5.12: Resultados PHA asociados á batería de casos obtida do escenario galicia2.

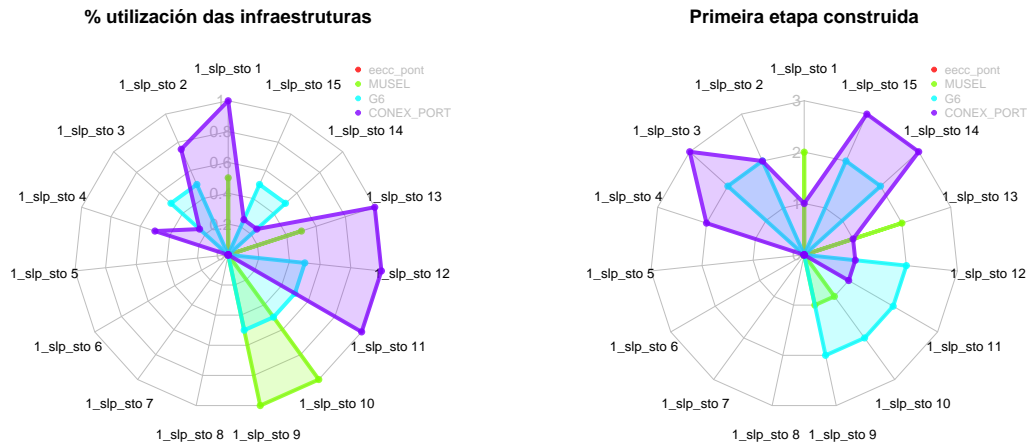
Na Figura 5.9 amósanse as gráficas de radar asociadas ao uso de cada infraestrutura na etapa final e a etapa de construción.

Na Táboa 5.15 recóllense os tempos de execución ao aplicar o método DEM e o PHA ás configuracións de parámetros tratadas anteriormente, e o número de iteracións que tardou en converxer o PHA en cada caso. Notar que no caso 16, que non ten solución o problema linearizado empregando o método PHA, estivo executándose máis de 2 días.

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.50	0.00	1.00
1_slp_sto 2	0.00	0.00	0.50	0.75
1_slp_sto 3	0.00	0.00	0.50	0.25
1_slp_sto 4	0.00	0.00	0.00	0.50
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	1.00	0.50	0.00
1_slp_sto 10	0.00	1.00	0.50	0.00
1_slp_sto 11	0.00	0.00	0.50	1.00
1_slp_sto 12	0.00	0.00	0.50	1.00
1_slp_sto 13	0.00	0.50	0.00	1.00
1_slp_sto 14	0.00	0.00	0.50	0.25
1_slp_sto 15	0.00	0.00	0.50	0.25

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	2.00	0.00	1.00
1_slp_sto 2	0.00	0.00	2.00	2.00
1_slp_sto 3	0.00	0.00	2.00	3.00
1_slp_sto 4	0.00	0.00	0.00	2.00
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	1.00	2.00	0.00
1_slp_sto 10	0.00	1.00	2.00	0.00
1_slp_sto 11	0.00	0.00	2.00	1.00
1_slp_sto 12	0.00	0.00	2.00	1.00
1_slp_sto 13	0.00	2.00	0.00	1.00
1_slp_sto 14	0.00	0.00	2.00	3.00
1_slp_sto 15	0.00	0.00	2.00	3.00

Táboa 5.13: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos do escenario galicia2 (PHA). Táboa 5.14: Etapa de construción da infraestrutura para a batería de casos do escenario galicia2 (PHA).



(a) Porcentaxe de uso de cada infraestrutura na etapa final.

(b) Primeira etapa construída.

Figura 5.9: Gráficas de radar asociadas á batería de casos a comparar obtida do escenario galicia2 (PHA).

	CPU time		Real time		Iteracións
	CPU_DEM	CPU_PHA	Real_DEM	Real_PHA	PHA
1	468.84	12583.55	156.41	7155.68	115
2	376.05	21035.14	143.40	11155.24	166
3	358.58	11907.03	140.73	5933.35	79
4	270.20	11954.35	111.99	5516.66	84
5	89.05	59.60	52.91	41.72	2
6	64.51	54.31	45.51	39.19	2
7	84.52	43.35	59.58	32.47	2
8	95.56	141.07	74.64	103.49	2
9	153.12	15137.45	94.13	9664.27	115
10	419.68	9927.89	153.87	5531.26	67
11	448.95	6201.47	154.78	3231.48	44
12	346.77	4931.62	117.51	2407.70	46
13	235.29	19775.71	104.09	8753.65	153
14	255.60	30819.61	109.91	14293.34	247
15	373.24	7212.65	155.60	3379.09	40
16	333.38	191050.83	161.00	80664.00	791

Táboa 5.15: Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia2 e número de iteracións do PHA.

5.3.3. Resultados con escenario de partida galicia3

O escenario ao que nos referimos como galicia3 recordamos que presenta 3 etapas e dous posibles resultados da incerteza, con probabilidades 0.9 e 0.1. A árbore de decisión, asociada a todos os casos obtidos a partir deste, represéntase na Figura 5.10. As combinacións de parámetros empregadas son as mesmas que as consideradas no apartado anterior.

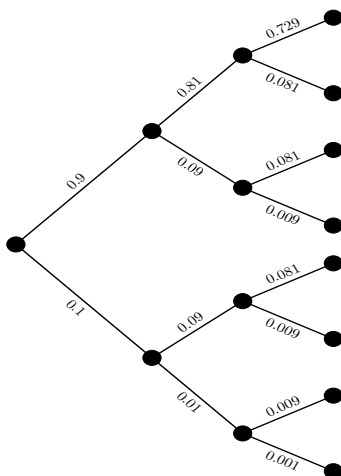


Figura 5.10: Árbore de decisión do escenario galicia3.

Na Táboa 5.16 recóllense os resultados de aplicar o método DEM na resolución do problema estocástico. Pódese ver que:

- Os casos 5, 6, 7 e 8 teñen un valor de custos de construción igual a cero, dado que non se constrúe

ningunha infraestrutura planificable. Ademais o SLP non converxe nestes casos.

- O caso 9 resulta ser infactible, o solver que se emprega (gurobi) devolve o código 3. Nas seccións anteriores, esta combinación de parámetros resolvíase. Aquí non se obteñen valores para as sumas de custos dado que non se xera o arquivo 1_slp_sto.xml.
- Os casos 3 e 15, resaltados en amarelo, que teñen α_{inc} igual a α_{cc} presentan o mesmo valor de *Sumcost_inc* e *Sumcost_cc*. En ambos escenarios decídese construír o G6 e o G6-Sur e o uso destas infraestruturas é o mesmo.
- O valor de *Sumcost_cc* diminúe ao aumentar o valor de α_{cc} e *SumGcost_cc* aumenta.
- O valor de *Sumcost_inc* é similar en todos os casos (excepto nos escenarios en que a suma de custos é cero e no escenario con valores $\alpha_{inc} = 10$ e $\alpha_{cc} = 100$), diferéncianse en apenas unha unidade.

	α_{inc}	α_{gc}	α_{cc}	SumGcost_inc	SumGcost_gc	SumGcost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSLP	Erro solver	$\exists_problem$	$\exists_1_slp_sto$
1	10	0.50	0.10	2009.10664	0.00000	5.60189	200.91066	0.00000	56.01892	0.00000	1.39391e-05	5.19573e-12	0	0	1
2	10	0.50	0.50	2009.10664	0.00000	28.00946	200.91066	0.00000	56.01892	0.00000	1.39266e-05	5.19668e-12	0	0	1
3	10	0.50	10	2016.25452	0.00000	349.63090	201.62545	0.00000	34.96309	0.00000	3.03884e-07	2.19717e-13	0	0	1
4	10	0.50	100	5250.84934	0.00000	1644.81907	525.08493	0.00000	16.44819	0.00000	6.42025e+05	6.07106e-14	0	0	1
5	10	0.50	500	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	1.10675e+00	1.51002e-12	0	0	1
6	10	0.50	1500	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	1.04830e+06	9.16970e-15	0	0	1
7	10	0.50	3000	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	4.09244e+05	3.90957e-14	0	0	1
8	10	0.50	10000	11532.09486	0.00000	0.00000	1153.20949	0.00000	0.00000	0.00000	6.42749e+05	4.59446e-05	0	0	1
9	3000	0.50	0.10										3	0	0
10	3000	0.50	0.50	602732.55241	0.00000	70.48085	200.91085	0.00000	140.96169	0.00024	5.15419e-01	2.82721e-05	0	0	1
11	3000	0.50	10	602731.99235	0.00000	688.08600	200.91066	0.00000	68.80860	0.00000	3.15658e-05	2.35167e-13	0	0	1
12	3000	0.50	100	602731.99235	0.00000	6880.86001	200.91066	0.00000	68.80860	0.00000	6.89199e-05	1.02205e-13	0	0	1
13	3000	0.50	500	604876.35658	0.00000	17481.54497	201.62545	0.00000	34.96309	0.00000	4.09015e-08	8.52489e-13	0	0	1
14	3000	0.50	1500	604876.35658	0.00000	52444.63491	201.62545	0.00000	34.96309	0.00000	3.35048e-05	1.13779e-13	0	0	1
15	3000	0.50	3000	604876.35658	0.00000	104889.26982	201.62545	0.00000	34.96309	0.00000	3.37861e-05	2.86295e-13	0	0	1
16	3000	0.50	10000	604876.69265	0.00000	349630.89938	201.62556	0.00000	34.96309	0.00000	4.27368e-05	2.91260e-04	0	0	1

Táboa 5.16: Resultados DEM asociados a batería de casos obtida do escenario galicia3.

Nas Táboas 5.17 e 5.18 podemos ver que infraestruturas son construídas, en que etapa e que uso se fai delas. A construción do G6 e G6-Sur vai da man naqueles casos en que sae construíble o G6-Sur. No caso en que se constrúe Musel (caso 10) non se constrúe ningunha outra infraestrutura.

A Táboa 5.19 recolle os resultados de aplicar o PHA na resolución do problema, e nas Táboas 5.21 e 5.20 o uso e etapa de construción das infraestruturas, respectivamente. Na Figura 5.11 amósanse as gráficas de radar asociadas a estas táboas. Pódese ver que:

- Os casos 5, 6, 7 e 8, igual ca nos casos anteriores, teñen valores de *SumGcost_cc* e *Sumcost_cc* igual a cero. Ademais teñen valores elevados de delta_SLP, polo que non converxe o SLP.
- O caso 16 resulta ser infactible, o solver que se emprega (gurobi) devolve o código 3. Analizando os arquivos que se xeran tense que o problema linealizado neste caso non ten solución. Obtéñense valores para as sumas de custos pero non son representativos.
- A medida que aumenta o parámetro α_{cc} tamén o fai o valor do custo de construción *SumGcost_cc* mentres que *Sumcost_cc* diminúe a excepción do caso 12 resaltado en azul. Pódese ver tamén,

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	1.00	0.00
1_slp_sto 2	0.00	0.00	1.00	0.00
1_slp_sto 3	0.00	0.00	0.90	0.09
1_slp_sto 4	0.00	0.00	0.81	0.09
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 10	0.00	1.00	0.00	0.00
1_slp_sto 11	0.00	0.00	0.90	1.00
1_slp_sto 12	0.00	0.00	0.90	1.00
1_slp_sto 13	0.00	0.00	0.90	0.09
1_slp_sto 14	0.00	0.00	0.90	0.09
1_slp_sto 15	0.00	0.00	0.90	0.09
1_slp_sto 16	0.00	0.00	0.90	0.09

Táboa 5.17: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida a partir do escenario galicia3 (DEM).

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.00	1.00	0.00
1_slp_sto 2	0.00	0.00	1.00	0.00
1_slp_sto 3	0.00	0.00	2.00	3.00
1_slp_sto 4	0.00	0.00	3.00	3.00
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 10	0.00	1.00	0.00	0.00
1_slp_sto 11	0.00	0.00	2.00	1.00
1_slp_sto 12	0.00	0.00	2.00	1.00
1_slp_sto 13	0.00	0.00	2.00	3.00
1_slp_sto 14	0.00	0.00	2.00	3.00
1_slp_sto 15	0.00	0.00	2.00	3.00
1_slp_sto 16	0.00	0.00	2.00	3.00

Táboa 5.18: Etapa de construción da infraestrutura para a batería de casos obtida a partir do escenario galicia3 (DEM).

que efectivamente, o valor de $SumGcost_cc$ é igual ao valor de $Sumcost_cc$ multiplicado polo parámetro α_{cc} . Tamén se pode ver que estes valores son maiores cando α_{inc} é grande.

- Os casos con custos igual a cero teñen valores similares de $Sumcost_inc$ e $SumGcost_inc$. Para os restantes o valor de $Sumcost_inc$ tamén é similar, non se chega a apreciar un cambio dunha unidade, obténdose o mesmo valor (200.91) para os casos 1, 9, 10 11 e 12 e para o 2, 13, 14 e 15 (201.63). Estes presentan o mesmo valor en $SumGcost_inc$ cando se considera o mesmo parámetro α_{inc} . Igual que nas seccións anteriores o escenario 4 ten un incumprimento diferente a calquera dos outros casos tratados.
- Os casos con α_{inc} igual a α_{cc} , que presentaban nos apartados anteriores o mesmo valor de $Sumcost_inc$ e $Sumcost_cc$, aquí teñen distinto valor de $Sumcost_cc$.

Na Táboa 5.22 recóllense os tempos de execución ao aplicar o método DEM e o PHA ás configuracións de parámetros tratadas anteriormente, e o número de iteracións que tardou en converxer o PHA en cada caso. Non se ten o tempo de execución no caso 9 ao aplicar o método DEM, neste caso non se xerou o arquivo 1_slp_sto.xml, a diferenza do caso 16 aplicando o PHA que tampouco se resolve pero si se obtén o tempo de execución.

	α_{inc}	α_{gc}	α_{cc}	SumGeost_inc	SumGeost_gc	SumGeost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CS LP	Erro solver	\exists _problem	\exists _1_slp_sto
1	10	0.50	0.10	2009.10664	0.00000	11.85633	200.91066	0.00000	118.56332	0.00000	2.42634e-05	1.32623e-11	0	0	1
2	10	0.50	0.50	2016.25452	0.00000	43.85346	201.62545	0.00000	87.70691	0.00000	3.14004e-05	1.59999e-11	0	0	1
3	10	0.50	10	2016.25452	0.00000	360.28269	201.62545	0.00000	36.02827	0.00000	2.60385e-05	4.29694e-11	0	0	1
4	10	0.50	100	5036.00434	0.00000	1751.33696	503.60043	0.00000	17.51337	0.00000	5.40621e+05	9.33015e-12	0	0	1
5	10	0.50	500	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	3.10423e+05	1.17021e-13	0	0	1
6	10	0.50	1500	11552.52713	0.00000	0.00000	1155.25271	0.00000	0.00000	0.00000	3.10423e+05	3.74443e-05	0	0	1
7	10	0.50	3000	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	3.10423e+05	2.21340e-13	0	0	1
8	10	0.50	10000	11532.08816	0.00000	0.00000	1153.20882	0.00000	0.00000	0.00000	3.10423e+05	1.72715e-11	0	0	1
9	3000	0.50	0.10	602731.99235	0.00000	11.85633	200.91066	0.00000	118.56332	0.00000	2.62382e-05	1.27656e-13	0	0	1
10	3000	0.50	0.50	602731.99235	0.00000	59.28166	200.91066	0.00000	118.56332	0.00000	2.55641e-05	2.35448e-13	0	0	1
11	3000	0.50	10	602731.99236	0.00000	560.18919	200.91066	0.00000	56.01892	0.00000	2.45658e-05	1.41201e-14	0	0	1
12	3000	0.50	100	602731.99235	0.00000	6880.86001	200.91066	0.00000	68.80860	0.00000	2.04359e-05	7.13444e-13	0	0	1
13	3000	0.50	500	604876.35658	0.00000	17481.54497	201.62545	0.00000	34.96309	0.00000	2.35479e-05	4.10980e-15	0	0	1
14	3000	0.50	1500	604876.35658	0.00000	52444.63491	201.62545	0.00000	34.96309	0.00000	2.35479e-05	4.10980e-15	0	0	1
15	3000	0.50	3000	604876.35658	0.00000	104889.26982	201.62545	0.00000	34.96309	0.00000	2.35479e-05	4.10980e-15	0	0	1
16	3000	0.50	10000	604878.90605	0.00000	349630.89938	201.62630	0.00000	34.96309	0.00000			3	1	1

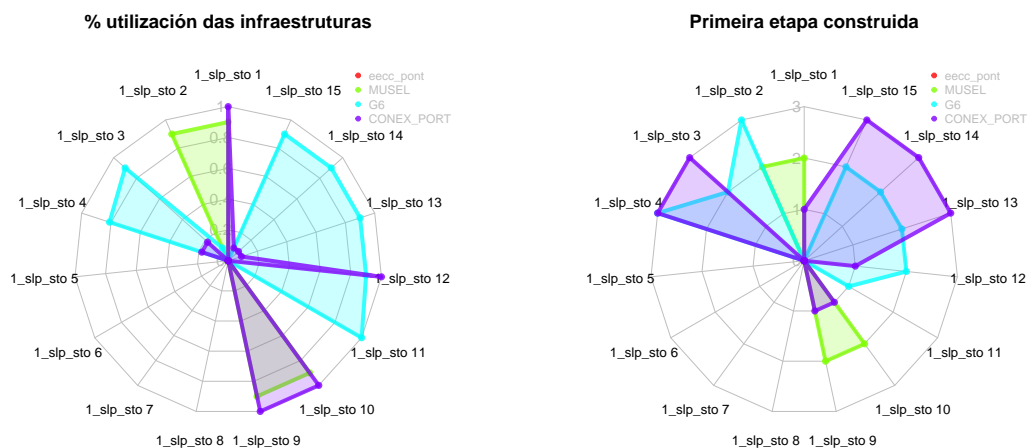
Táboa 5.19: Resultados PHA asociados á batería de casos obtidos do escenario galicia3.

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	0.90	0.00	1.00
1_slp_sto 2	0.00	0.90	0.09	0.00
1_slp_sto 3	0.00	0.00	0.90	0.18
1_slp_sto 4	0.00	0.00	0.81	0.18
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	0.90	0.00	1.00
1_slp_sto 10	0.00	0.90	0.00	1.00
1_slp_sto 11	0.00	0.00	1.00	0.00
1_slp_sto 12	0.00	0.00	0.90	1.00
1_slp_sto 13	0.00	0.00	0.90	0.09
1_slp_sto 14	0.00	0.00	0.90	0.09
1_slp_sto 15	0.00	0.00	0.90	0.09

	eecc_pont	MUSEL	G6	CONEX_PORT
1_slp_sto 1	0.00	2.00	0.00	1.00
1_slp_sto 2	0.00	2.00	3.00	0.00
1_slp_sto 3	0.00	0.00	2.00	3.00
1_slp_sto 4	0.00	0.00	3.00	3.00
1_slp_sto 5	0.00	0.00	0.00	0.00
1_slp_sto 6	0.00	0.00	0.00	0.00
1_slp_sto 7	0.00	0.00	0.00	0.00
1_slp_sto 8	0.00	0.00	0.00	0.00
1_slp_sto 9	0.00	2.00	0.00	1.00
1_slp_sto 10	0.00	2.00	0.00	1.00
1_slp_sto 11	0.00	0.00	1.00	0.00
1_slp_sto 12	0.00	0.00	2.00	1.00
1_slp_sto 13	0.00	0.00	2.00	3.00
1_slp_sto 14	0.00	0.00	2.00	3.00
1_slp_sto 15	0.00	0.00	2.00	3.00

Táboa 5.20: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida do escenario galicia3 (PHA).

Táboa 5.21: Etapa de construción da infraestrutura para a batería de casos obtida do escenario galicia3 (PHA).



(a) Porcentaxe de uso de cada infraestrutura na etapa final.

(b) Primeira etapa construída.

Figura 5.11: Gráficas de radar asociadas á batería de casos a comparar obtida do escenario galicia3.

	CPU time		Real time		Iteracións
	CPU_DEM	CPU_PHA	Real_DEM	Real_PHA	PHA
1	240.99	2978.14	103.16	1501.87	23
2	228.73	4026.80	97.13	2081.19	39
3	281.65	1301.16	104.54	630.93	10
4	442.96	2147.85	113.14	712.47	11
5	179.78	61.11	66.04	37.85	2
6	64.03	50.57	43.94	36.26	2
7	65.45	33.61	46.56	25.53	2
8	93.98	109.60	77.96	80.52	2
9	-	1037.95	-	548.70	10
10	187.32	4922.21	114.68	2694.85	49
11	221.41	11454.18	96.02	6224.11	112
12	380.15	11642.94	91.09	3529.52	46
13	551.21	4443.62	100.95	1352.15	22
14	582.28	2920.72	102.59	902.23	15
15	284.19	3294.02	105.99	1604.23	27
16	354.81	6694.15	180.26	3106.16	49

Táboa 5.22: Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario de partida galicia3 e número de iteracións do PHA.

5.3.4. Comparación

Observouse no estudo dos resultados das baterías de casos obtidos dos escenarios de partida galicia1, galicia2 e galicia3 un comportamento similar. As metodoloxías empregadas para a resolución proporcionan resultados coherentes co que se espera. A medida que se aumenta o parámetro α_{cc} aumenta o valor do custo de construción, e canto maior é este valor maior é o número de infraestruturas. Os casos

nos cales o valor do custo de construción é cero, é dicir, non se constrúe ningunha infraestrutura, os valores de incumprimento son maiores dado que é máis difícil satisfacer a demanda, ademais nestes casos o SLP non converge pero si o fai o CSLP. Ademais, nos tres escenarios tense que a combinación de parámetros $\alpha_{inc} = 3000$ e $\alpha_{cc} = 10000$ dá lugar a un problema infactible. Agora imos amosar as principais diferenzas, centrándonos nos resultados empregando o PHA.

Se comparamos os resultados da Táboa 5.5 cos da Táboa 5.12 observamos que a principal diferenza que se observa é que os valores de incumprimento e custos de construción teñen un valor menor cando as probabilidades das dúas ramas é a mesma. Por outra banda, se os comparamos cos resultados recollidos na Táboa 5.19, os valores de incumprimento e custo de construción teñen un valor maior que os recollidos na Táboa 5.5 na maioría dos casos, xa que nos escenarios 9,10 e 11, por exemplo, os custos de construción son menores.

Podemos apreciar como afecta o cambio nas probabilidades comparando as infraestruturas que saen a construír e o porcentaxe de uso destas. Se nos fixamos nas gráficas de radar da Figura 5.7 e da Figura 5.9 observamos que nesta última, correspondentes a batería de casos con igual probabilidade de demanda nas dúas ramas, o G6 sempre aparece construído na etapa 2 e en maior número de escenarios, mentres que Musel e G6-Sur aparecen en menos casos. Ademais como moito aparecen construídas dúas infraestruturas en cada caso. Observamos ademais, que nos casos en que α_{inc} é igual a α_{cc} , casos 3 e 15, constrúense as mesmas infraestruturas, G6 e Conex-Porto ou G6-Sur, pero o uso que se fai delas é distinto. O G6 ten unha maior porcentaxe de uso nos casos resultantes de modificar o escenario galicia1 e o G6-Sur maior porcentaxe de uso nos casos con igual probabilidade de demanda nas dúas ramas.

Se comparamos agora as gráficas de radar da Figura 5.7 e da Figura 5.11. Observamos que nesta última, correspondente a batería de casos obtida a partir do escenario con maior probabilidade de demanda na rama de arriba, que non hai ningún caso no que aparezan 3 infraestruturas construídas como si ocorre no outro caso a comparar. Por outra banda, se nos fixamos nos casos en que α_{inc} é igual a α_{cc} constrúense as mesmas infraestruturas pero o uso que se fai delas é distinto, a que aparece construída na última etapa, G6-Sur, ten unha menor porcentaxe de uso mentres co uso que se fai do G6 é maior. Ademais na batería de casos obtidos de galicia3 o uso que se fai do G6-Sur é menor. Por outro lado, nos casos que sae a construír o G6-Sur xunto co G6, esta infraestrutura aparece construída na última etapa.

5.4. Rede de gas española

O escenario correspondente a rede de gas española, representado na Figura 5.12 presenta 9 posibles infraestruturas planificables: Muruarte-Pte.Reina, Musel, Guitiriz_Zamora, Zamora_Barbolla, Triplicación, LLanera_Trero, 3IC, MidCat, Dup.Castelnou_VillarArnedo, a estación de compresión de Pontevedra (eccc_pont), Musel, o G6 e o G6 Sur ou conexión con Portugal (CONEX_PORT). Por outra banda, considéranse dous posibles resultados de incerteza, con probabilidades 0.7 e 0.3 e un incremento aditivo na distribución da demanda e das probabilidades. A árbore de decisión asociada a todos os casos obtidos a partir deste amósase na Figura 5.13.

O escenario proporcionado da rede de España presentaba cinco etapas pero houbo que modificar o número de etapas, xa que, a resolución do problema de optimización estocástico era moi lenta, non se obtían resultados nun tempo razoable para esta primeira etapa de validación. Na análise dos resultados que se imprimían na resolución observouse que o PHA non finalizaba, debido a que non se alcanzaban os criterios de parada. Por isto, modificouse o escenario considerando unicamente dúas etapas e deixando os restantes parámetros iguais. Considéranse dous posibles resultados da incerteza e a duración das etapas é de dous anos por etapa.

Imos presentar un análise dos resultados análogo ao feito na sección anterior, pero neste caso imos



Figura 5.12: Representación da rede de gas española na interface de GANESO.

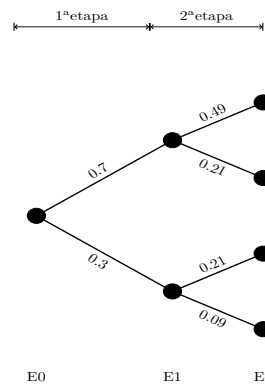


Figura 5.13: Árbore de decisión para o escenario da rede de gas española con dúas etapas e dúas incertezas.

facen máis fincapé nas infraestruturas planificables, dado que os patróns que se observaron na rede de gas galega non son tan claros aquí.

Na Táboa 5.23 recóllense os resultados de aplicar o método DEM na resolución do problema estocástico en cada caso. Obsérvase que o SLP non converge en ningún caso, os valores de delta_SLP altos e os valores de delta_CSLP son pequenos pero moito máis grandes que os que resultaban na resolución da batería de casos da rede galega. Nas Táboas 5.24 e 5.25 recóllese o uso que se fai das infraestruturas na última etapa e a etapa na que aparecen construídas, respectivamente. Na Figura 5.14 representáanse os gráficos de radar asociados aos resultados recollidos nestas. Coa combinación das tres táboas mencionadas anteriormente pódese ver que:

- O caso 16, ten a particularidade de que o parámetro de planificación $\alpha_{inc} = 0.001$ é moito máis pequeno que o parámetro $\alpha_{cc} = 0.5$, ao ter maior peso os custos de construción na función obxectivo, obtense que non se constrúe ningunha infraestrutura, $Sumcost_{cc}=0$, e os custos por incumprimento, $Sumcost_{inc}$, son elevados.

	α_{inc}	α_{ge}	α_{cc}	SumGeost_inc	SumGeost_ge	SumGeost_cc	Sumcost_inc	Sumcost_ge	Sumcost_cc	Infact	delta_SLP	delta_CSLLP	Erro solver	\exists _problem	\exists 1_slp_sto
1	0.001	0.50	0.001	0.00000	0.00559	0.07263	0.00332	0.01990	72.62766	0.00000	5.51130e+07	1.57643e-02	0	0	1
2	0.05	0.50	0.001	0.00010	0.00323	0.07263	0.00199	0.01149	72.62608	0.00000	4.91474e+07	8.55115e-03	0	0	1
3	0.10	0.50	0.001	0.00010	0.01751	0.08019	0.00099	0.06233	80.19402	0.00000	2.01894e+07	4.50694e-03	0	0	1
4	0.50	0.50	0.001	0.00055	0.00000	0.05971	0.00110	0.00000	59.70670	0.00024	9.65207e+07	1.61681e-05	0	0	1
5	1.00	0.50	0.001	0.00063	0.00772	0.08015	0.00063	0.02748	80.15078	0.00000	2.46917e+07	3.86985e-03	0	0	1
6	0.001	0.50	0.05	0.35089	0.18942	0.24949	350.88877	0.67434	4.98977	0.00000	1.82731e+07	2.66773e-02	0	0	1
7	0.05	0.50	0.05	0.00003	0.56736	0.24949	0.00062	2.01979	4.98977	0.00000	3.45642e+07	5.85402e-03	0	0	1
8	0.10	0.50	0.05	0.00006	0.59767	0.24949	0.00059	2.12772	4.98977	0.00000	1.31310e+07	7.40575e-04	0	0	1
9	0.50	0.50	0.05	37.50556	0.76585	0.24949	75.01113	2.72642	4.98977	0.00000	2.22310e+07	5.26526e-03	0	0	1
10	1.00	0.50	0.05	74.99014	0.77697	0.24949	74.99014	2.76602	4.98977	0.00040	2.23548e+07	8.15877e-05	0	0	1
11	0.001	0.50	0.10	0.48184	0.20440	0.49898	481.84037	0.72766	4.98977	0.00000	2.99665e+07	1.42797e-02	0	0	1
12	0.05	0.50	0.10	0.00007	1.27519	0.10394	0.00139	4.53967	1.03936	0.00000	2.49975e+07	5.73493e-03	0	0	1
13	0.10	0.50	0.10	0.00015	0.74546	0.10394	0.00150	2.65383	1.03936	0.00000	2.18209e+07	1.47342e-03	0	0	1
14	0.50	0.50	0.10	46.13819	5.94975	0.43173	92.27639	21.18110	4.31726	0.00032	2.64834e+07	7.53449e-03	0	0	1
15	1.00	0.50	0.10	0.00106	1.08822	0.10394	0.00106	3.87405	1.03936	0.00052	3.52196e+07	6.57584e-05	0	0	1
16	0.001	0.50	0.50	0.96272	0.45434	0.00000	962.72206	1.61745	0.00000	0.00065	1.10552e+04	6.37943e-05	0	0	1
17	0.05	0.50	0.50	5.76954	0.94430	0.18343	115.39083	3.36172	0.36685	0.00000	3.61101e+05	1.82232e-02	0	0	1
18	0.10	0.50	0.50	0.00017	1.23863	0.51968	0.00167	4.40951	1.03936	0.00000	3.95400e+07	1.34341e-02	0	0	1
19	0.50	0.50	0.50	0.00076	1.08826	0.51968	0.00152	3.87419	1.03936	0.00024	6.05356e+06	5.14899e-03	0	0	1
20	1.00	0.50	0.50	13.36457	0.83250	0.51968	13.36457	2.96372	1.03936	0.00013	2.03864e+07	2.98733e-04	0	0	1

Táboa 5.23: Resultados DEM asociados á batería de casos obtida do escenario da rede de gas española.

- Nos casos co mesmo parámetro α_{inc} , obsérvase que nalgún caso ao aumentar o parámetro α_{cc} diminúen os custos de construción, $Sumcost_cc$, dado que ao ter máis peso na función obxectivo, o número de infraestruturas que saen a construír é menor. Casos resaltados en azul e amarelo.
 Por exemplo, para $\alpha_{inc} = 0.5$, a medida que aumenta α_{cc} pásase de construír 4 infraestruturas no caso 4 a construír unha só no caso 19.
- Os resultados anteriores dános a idea de que considerando o parámetro $\alpha_{cc} > 0.1$ e $\alpha_{inc} > 0.1$ se vai obter que unicamente se constrúe Muruarte-Pte.Reina.
- Nos casos con $\alpha_{cc} = 0.001$, menor peso na función obxectivo, os custos de construción son máis altos. Constrúese maior número de infraestruturas, entre 4 e 6, o que leva a que os custos por incumprimento sexan cero, dado que se consegue satisfacer a demanda.
- En todos os casos, excepto para o caso 16, sae a construír Muruarte_Pte.Reina, polo que parece que esta infraestrutura é necesaria para o bo funcionamento da rede. Obsérvase tamén que nos casos 12 e 17 nos que se construíría unicamente esta infraestrutura e se faría na segunda etapa, os custos por incumprimento son elevados.

	Muruarte_Pte.Reina	Musel	Guitiriz_Zamora	Zamora_Barbolla	Triplicacion	Llanera_Treto	3IC	Midcat	Dup.Castelnou_VillarArnedo
1_slp_sto 1	1.00	0.70	0.70	0.70	0.00	0.00	1.00	0.00	0.00
1_slp_sto 2	1.00	0.70	0.70	0.70	0.00	0.00	1.00	0.00	0.00
1_slp_sto 3	1.00	0.70	0.70	0.70	0.00	0.70	1.00	0.00	0.00
1_slp_sto 4	1.00	0.70	0.70	0.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 5	1.00	0.70	0.70	0.70	0.00	0.70	1.00	0.00	0.00
1_slp_sto 6	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 7	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 8	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 9	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 10	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 11	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 12	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 13	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 14	0.70	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 15	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 17	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 18	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 19	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 20	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Táboa 5.24: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida da rede de gas española (DEM).

	Muruarte_Pte.Reina	Musel	Guitiriz_Zamora	Zamora_Barbolla	Triplicacion	Llanera_Treto	3IC	Midcat	Dup.Castelnou_VillarArnedo
1_slp_sto 1	1.00	2.00	2.00	2.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 2	1.00	2.00	2.00	2.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 3	1.00	2.00	2.00	2.00	0.00	2.00	1.00	0.00	0.00
1_slp_sto 4	1.00	2.00	2.00	0.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 5	1.00	2.00	2.00	2.00	0.00	2.00	1.00	0.00	0.00
1_slp_sto 6	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 7	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 8	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 9	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 10	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 11	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 12	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 13	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 14	2.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 15	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 17	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 18	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 19	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 20	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Táboa 5.25: Etapa de construción da infraestrutura para a batería de casos obtida da rede de gas española (DEM).

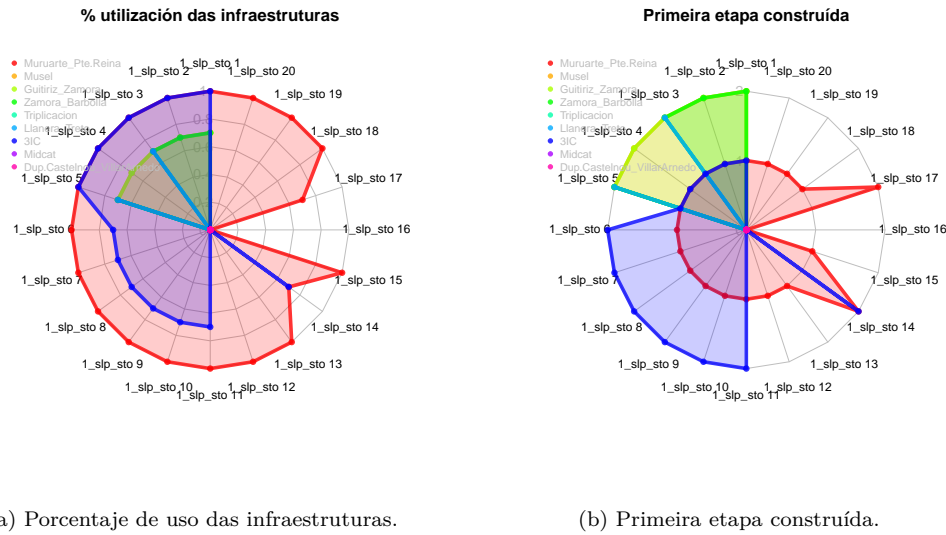


Figura 5.14: Gráficas de radar asociadas á batería de casos a comparar obtida da rede de gas española (DEM).

Na Táboa 5.26 recóllense os resultados de aplicar o PHA na resolución do problema estocástico en cada caso. Obsérvase que o SLP non converge en ningún caso, os valores de delta_SLP son altos e os valores de delta_CSPLP son pequenos pero moito máis grandes que os que resultaban na resolución da batería de casos da rede galega. Nas Táboas 5.27 e 5.28 recóllese o uso que se fai das infraestruturas na última etapa e a etapa na que aparece construída, respectivamente. Coa combinación das tres táboas mencionadas anteriormente pódese ver que:

	α_{inc}	α_{gc}	α_{cc}	SumGeost_inc	SumGeost_gc	SumGeost_cc	Sumcost_inc	Sumcost_gc	Sumcost_cc	Infact	delta_SLP	delta_CSPLP	Erro solver	\exists _problem	\exists 1_slp_sto
1	0.001	0.50	0.001	0.00000	0.06532	0.06537	0.00086	0.23253	65.36712	0.02031	5.66975e+07	5.30663e-03	0	0	1
2	0.05	0.50	0.001	0.00004	0.00784	0.07263	0.00078	0.02792	72.62516	0.00033	1.92691e+07	6.77015e-03	0	0	1
3	0.10	0.50	0.001	0.00008	0.00336	0.07263	0.00081	0.01195	72.62705	0.00054	6.29603e+07	2.19431e-05	0	0	1
4	0.50	0.50	0.001	0.00035	0.00204	0.08007	0.00071	0.00726	80.06770	0.00028	2.04846e+07	2.03237e-02	0	0	1
5	1.00	0.50	0.001	0.00168	0.00520	0.07263	0.00168	0.01851	72.62583	0.00022	2.29063e+07	2.85173e-03	0	0	1
6	0.001	0.50	0.05	0.22402	0.15917	0.24949	224.02432	0.56664	4.98977	0.00010	2.43037e+07	2.12547e-03	0	0	1
7	0.05	0.50	0.05	0.00006	1.49940	0.24949	0.00120	5.33788	4.98977	0.00014	2.65154e+07	8.22920e-03	0	0	1
8	0.10	0.50	0.05	7.65936	0.46577	0.24949	76.59362	1.65815	4.98977	0.00016	3.34000e+07	3.37456e-05	0	0	1
9	0.50	0.50	0.05	0.00025	1.52212	0.24949	0.00050	5.41873	4.98977	0.00015	5.54969e+07	2.51362e-03	0	0	1
10	1.00	0.50	0.05	0.00131	1.51300	0.24949	0.00131	5.38627	4.98977	0.00028	2.14191e+07	3.78537e-03	0	0	1
11	0.001	0.50	0.10	0.22402	0.15917	0.49898	224.02432	0.56664	4.98977	0.00010	2.43037e+07	2.12547e-03	0	0	1
12	0.05	0.50	0.10	0.00006	1.49940	0.49898	0.00115	5.33788	4.98977	0.00014	2.65154e+07	8.22030e-03	0	0	1
13	0.10	0.50	0.10	7.65936	0.46577	0.49898	76.59362	1.65815	4.98977	0.00016	3.34000e+07	3.37456e-05	0	0	1
14	0.50	0.50	0.10	0.00050	1.49940	0.49898	0.00101	5.33788	4.98977	0.00023	3.51128e+07	8.87585e-05	0	0	1
15	1.00	0.50	0.10	76.59343	0.73015	0.49898	76.59343	2.59933	4.98977	0.00013	2.14191e+07	3.77688e-03	0	0	1
16	0.001	0.50	0.50	0.80763	0.25494	0.00000	807.62524	0.90760	0.00000	0.00010	2.78340e+07	1.05872e-03	0	0	1
17	0.05	0.50	0.50	0.00002	0.92350	0.51968	0.00045	3.28765	1.03936	0.00014	4.33741e+06	8.17915e-03	0	0	1
18	0.10	0.50	0.50	10.06515	1.04876	0.26204	100.65154	3.73360	0.52407	0.00000	3.96126e+07	6.97696e-03	0	0	1
19	0.50	0.50	0.50	41.46042	4.38193	0.26204	82.92083	15.59965	0.52407	0.00025	1.99629e+07	2.49453e-03	0	0	1
20	1.00	0.50	0.50	82.92097	4.56228	0.26204	82.92097	16.24172	0.52407	0.00063	3.07766e+07	6.32546e-05	0	0	1

Táboa 5.26: Resultados PHA asociados á batería de casos obtida do escenario da rede de gas española.

- O caso 16, igual que ao aplicar o DEM, ten a particularidade de que o parámetro de plani-

ficación $\alpha_{inc} = 0.001$ é moito máis pequeno que o parámetro $\alpha_{cc} = 0.5$. Ao ter maior peso os custos de construción na función obxectivo, obtense que non se constrúe ningunha infraestrutura, $Sumcost_{cc}=0$, e os custos por incumprimento, $Sumcost_{inc}$, son elevados.

- Nos casos co mesmo parámetro α_{inc} , obsérvase que ao aumentar o parámetro α_{cc} diminúen os custos de construción, $Sumcost_{cc}$, dado que ao ter máis peso na función obxectivo, o número de infraestruturas que saen a construír é menor.

Por exemplo, consideremos $\alpha_{inc} = 0.001$, a medida que aumentamos α_{cc} observamos que pasamos de construír 5 infraestruturas no caso 1 a non construír ningunha no caso 16.

- A diferenza en empregar o PHA e a metodoloxía DEM na resolución está en que na maioría dos casos (excepto para o caso 11) sae a construír Murarte_Pte.Reina e o 3IC. Por outra banda o comportamento para os casos con $\alpha_{cc} = 0.05$ e $\alpha_{cc} = 0.1$ é similar.
- Observamos que nos casos con $\alpha_{cc} = 0.001$, menor peso na función obxectivo, os custos de construción son máis altos. Constrúese un maior número de infraestruturas, entre 5 e 6, o que leva a que os custos por incumprimento sexan cero, dado que se consegue satisfacer a demanda.
- En todos os casos, excepto o caso 16 comentado anteriormente, sae a construír Muruarte_Pte.Reina, polo que parece que esta infraestrutura é necesaria para o bo funcionamento da rede. Ademais, empregando a metodoloxía PHA na resolución aumenta o número de casos nos que sae a construír o 3IC.

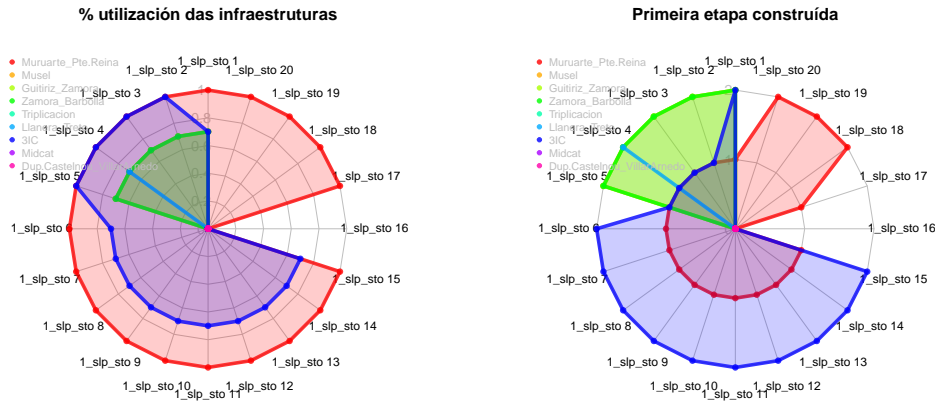
Observase que nos casos 11 e con $\alpha_{cc} = 0.5$, nos que se construíría unicamente Muruarte_Pte.Reina, os custos por incumprimento son elevados.

	Muruarte_Pte.Reina	Musel	Guitiriz_Zamora	Zamora_Barbolla	Triplicacion	Llanera_Treto	3IC	Midcat	Dup.Castelnou_VillarArnedo
1_slp_sto 1	1.00	0.70	0.70	0.70	0.00	0.00	0.70	0.00	0.00
1_slp_sto 2	1.00	0.70	0.70	0.70	0.00	0.00	1.00	0.00	0.00
1_slp_sto 3	1.00	0.70	0.70	0.70	0.00	0.00	1.00	0.00	0.00
1_slp_sto 4	1.00	0.70	0.70	0.70	0.00	0.70	1.00	0.00	0.00
1_slp_sto 5	1.00	0.70	0.70	0.70	0.00	0.00	1.00	0.00	0.00
1_slp_sto 6	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 7	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 8	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 9	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 10	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 11	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 12	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 13	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 14	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 15	1.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00
1_slp_sto 16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 17	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 18	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 19	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 20	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Táboa 5.27: Porcentaxe de uso da infraestrutura na etapa final para a batería de casos obtida da rede de gas española (PHA).

	Muruarte_Pte.Reina	Musel	Guitiriz_Zamora	Zamora_Barbolla	Triplificacion	Llanera_Treto	3IC	Midcat	Dup.Castelnou_VillarArnedo
1_slp_sto 1	1.00	2.00	2.00	2.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 2	1.00	2.00	2.00	2.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 3	1.00	2.00	2.00	2.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 4	1.00	2.00	2.00	2.00	0.00	2.00	1.00	0.00	0.00
1_slp_sto 5	1.00	2.00	2.00	2.00	0.00	0.00	1.00	0.00	0.00
1_slp_sto 6	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 7	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 8	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 9	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 10	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 11	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 12	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 13	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 14	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 15	1.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00
1_slp_sto 16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 17	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 18	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 19	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1_slp_sto 20	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Táboa 5.28: Etapa de construción da infraestrutura para a batería de casos obtida da rede de gas española (PHA).



(a) Porcentaxe de uso das infraestruturas.

(b) Primeira etapa construída.

Figura 5.15: Gráficas de radar asociadas á batería de casos a comparar obtida da rede de gas española (PHA).

Na Táboa 5.29 recóllense os tempos de execución ao aplicar o método DEM e o PHA ás configuracións de parámetros tratadas, e o número de iteracións que tardou en converxer o PHA en cada caso. Os “CPU time” na execución do PHA superan a hora de execución case en todos os casos.

	CPU time		Real time		Iteracións
	CPU_DEM	CPU_PHA	Real_DEM	Real_PHA	PHA
1	31260.41	12683.64	2843.46	4370.38	6
2	9250.79	7889.95	2333.89	4455.62	5
3	8153.30	7342.83	2145.92	4276.74	5
4	6497.99	8511.40	1574.25	4560.54	5
5	9434.34	7569.25	2549.28	4342.03	5
6	5053.74	4781.87	1297.00	2104.91	3
7	2736.44	4533.39	1179.72	2778.34	3
8	4064.46	4484.94	1613.84	2688.53	3
9	2890.19	4632.32	1194.29	2790.19	3
10	2840.43	4651.03	1118.79	2767.45	3
11	3515.24	4778.91	1199.22	2180.63	3
12	2376.77	4566.51	1123.70	2789.71	3
13	2487.52	4216.20	1121.78	2565.46	3
14	2994.89	4534.47	1200.12	2792.51	3
15	2641.63	4533.06	1083.51	2760.75	3
16	1916.13	2314.11	657.94	1140.38	2
17	2139.76	14184.20	1077.62	8783.39	11
18	1928.10	10600.62	1020.67	6175.39	8
19	2867.26	11142.91	1398.67	6531.56	8
20	1970.00	14920.41	1041.86	8661.81	11

Táboa 5.29: Tempos de execución (en segundos) dos métodos aplicados aos casos obtidos a partir do escenario da rede de gas española e número de iteracións do PHA.

5.5. Conclusións e observacións

Os resultados obtidos dan mostra dun bo comportamento do algoritmo na práctica a pesar de que a converxencia non é boa nalgúns casos. Ademais, as metodoloxías empregadas na resolución do problema estocástico dan solución á maioría dos casos tratados. Observamos tamén, como ao aumentar a dimensión do escenario, ao pasar da rede galega á rede española, os tempos de execución aumentaron en xeral e tamén os valores do parámetro de converxencia do algoritmo CSLP (δ_{CSLP}).

Por outra banda, o labor desenvolvido nestas prácticas considérase unha primeira fase no proceso de validación, xa que, non se fixo fincapé nos tempos de resolución e actualmente o equipo investigador segue a traballar no módulo de planificación de GANESO. Suxeriuse ao equipo investigador que aumente o número de etapas na árbore de decisión da rede de gas española con un menor número de infraestruturas a construír, xa que, o escenario proporcionado durante as prácticas presenta un gran número o que complica a resolución xunto co número de etapas, debido ao tamaño do problema. Pode verse nos resultados, que se amosan, que algúns casos leva varios días resolver.

Consideramos que o equipo investigador pode conseguir unha versión mellorada do módulo de planificación de GANESO, que facilite a validación. Para a obtención dos resultados que se amosan, fixéronse modificacións nos arquivos de FORTRAN, onde están implementados os algoritmos de resolución, para facilitar a extracción de información referente á converxencia. Ademais observouse que a información que se recolle no arquivo `planning.log` empregando o método DEM correspóndese coa información do arquivo `ganeso_output.txt` empregando o PHA, pero non se cambiou no código de FORTRAN, simplemente se accedeu na lectura de resultados en R ao arquivo de interese en cada caso. Tamén se suxeriu que pode ser interesante incorporar un resumo final no arquivo `ganeso_output.txt`, cando se emprega o PHA, coa información relativa á execución que amosamos nas táboas de resultado (os valores de delta), que aquí é obtida como unha media ponderada polas probabilidades de cada

rama; e co número de iteracións do PHA, que non aparece como tal en ningún dos arquivos xerados. Outra información interesante, que pode ser omitida en caso de que non se garde o arquivo de como avanza a execución, son os tempos de execución, polo que tamén se pode pensar en facer automática a xeración deste arquivo ou incorporar esta información nun dos arquivos que se xeran por defecto se o modelo se pode resolver.

Para rematar mencionar que a realización destas prácticas e deste traballo fin de mestrado, supuxo unha introdución no mundo da programación matemática, en concreto na optimización e programación estocástica que durante o mestrado non abordara, completando así a formación académica obtida.

Apéndice A

Código AMPL

Neste Apéndice recóllese a implementación do problema do granxeiro introducido no Capítulo 3 en AMPL de maneira que pode ser resolto por un solver. A implementación do problema en AMPL faise da forma máis xeral posible, de maneira que cando se cambien os datos non sexa necesario modificar o modelo, e só sexa necesario modificar o arquivo .dat onde se recollen os datos do problema.

A forma máis habitual de traballar con AMPL é definir tres ficheiros de extensións .mod, .dat e .run. No primeiro tense modelado o problema que se quere resolver, no .dat indícanse os datos que se van empregar e no .run definimos a información que queremos que se nos amose e como se vai resolver o problema.

A continuación amosamos estes ficheiros para a versión determinista e estocástica do problema do granxeiro.

Ficheiro .dat para o problema determinista cando o rendemento da terra é normal.	Ficheiro .dat para o problema estocástico.
<code>param n := 3;</code>	<code>param n := 3;</code>
<code>param m := 4;</code>	<code>param m := 4;</code>
<code>param r := 2;</code>	<code>param r := 2;</code>
<code>param custo :=</code>	<code>param custo:=</code>
<code>1 150</code>	<code>1 150</code>
<code>2 230</code>	<code>2 230</code>
<code>3 260;</code>	<code>3 260;</code>
<code>param venda :=</code>	<code>param venda :=</code>
<code>1 170</code>	<code>1 170</code>
<code>2 150</code>	<code>2 150</code>
<code>3 36</code>	<code>3 36</code>
<code>4 10;</code>	<code>4 10;</code>
<code>param compra :=</code>	<code>param compra :=</code>
<code>1 238</code>	<code>1 238</code>
<code>2 210;</code>	<code>2 210;</code>
<code>param rendemento :=</code>	<code>param rendemento: 1 2 3 :=</code>
<code>1 2.5</code>	<code>1 3 2.5 2</code>
<code>2 3</code>	<code>2 3.6 3 2.4</code>
<code>3 20;</code>	<code>3 24 20 16;</code>

```

# Ficheiro .run para o modelo determinista e estocástico.
solve;
display z;
display x;
display y;
display w;

```

```

# Parámetros do modelo
param n>=0, integer; # número de cultivos a plantar 3
param m>=0, integer; # número de cultivos que se poden vender 4
param r>=0, integer; # número de cultivos que se poden comprar 2

# Conxunto de índices
set CULTIVOS := 1..n;
set VENDA := 1..m;
set COMPRA := 1..r;

# Outros parámetros do modelo
param custo {i in CULTIVOS};
param venda {j in VENDA};
param compra {l in COMPRA};
param rendemento { i in CULTIVOS};

# Variables e restricións de non negatividade
var x {i in CULTIVOS}>=0;
var w {j in VENDA}>=0;
var y {l in COMPRA}>=0;

# Función obxectivo:
minimize z: sum {i in CULTIVOS} custo[i]*x[i]+ sum {l in COMPRA} compra[l]*y[l]-
            sum {j in VENDA} venda[j]*w[j];

# Restricións do modelo:
subject to restricion1: sum{i in CULTIVOS} x[i]<=500;
subject to restricion2: rendemento[1]*x[1]+y[1]-w[1]>=200;
subject to restricion3: rendemento[2]*x[2]+y[2]-w[2]>=240;
subject to restricion4: w[3]+w[4]<=rendemento[3]*x[3];
subject to restricion5: w[3]<=6000;

```

Figura A.1: Formulación en AMPL do versión determinista do problema do granxeiro.

```

# Parámetros do modelo
param n, >= 0, integer;
param m, >= 0, integer;
param r, >= 0, integer;
param s, >= 0, integer;

# Conxunto de índices
set CULTIVOS:=1..n;
set VENDA:=1..m;
set COMPRA:=1..r;
set ESCENARIO:=1..s;

# Outros parámetros do modelo
param custo {i in CULTIVOS};
param venda {j in VENDA };
param compra {l in COMPRA};
param rendimento {(i, t) in {CULTIVOS,ESCENARIO}};
param prob {t in ESCENARIO} := 1/s;

# Variables e restricións de non negatividade
var x {i in CULTIVOS}>=0;
var w {(j,t) in {VENDA,ESCENARIO}}>=0;
var y {(l,t) in {COMPRA,ESCENARIO}}>=0;

# Función obxectivo:
minimize: sum {i in CULTIVOS} custo[i]*x[i]- sum {t in ESCENARIO} prob[t]*(venda[1]*w[1,t]-
compra[1]*y[1,t]+ venda[2]*w[2,t]-compra[2]*y[2,t]+venda[3]*w[3,t]+venda[4]*w[4,t]);
# Restricións do modelo:
subject to restricion1: sum{i in CULTIVOS} x[i]<=500;
subject to restricion2 {t in ESCENARIO}: rendimento[1,t]*x[1]+y[1,t]-w[1,t]>=200;
subject to restricion3 {t in ESCENARIO}: rendimento[2,t]*x[2]+y[2,t]-w[2,t]>=240;
subject to restricion4 {t in ESCENARIO}: w[3,t]+w[4,t]<=rendimento[3,t]*x[3];
subject to restricion5 {t in ESCENARIO}: w[3,t]<=6000;

```

Figura A.2: Formulación en AMPL da versión estocástica do problema do granxeiro.

```

# Parámetros do modelo
param n, >=0, integer;
param m, >=0, integer;
param r, >=0, integer;
param s, >=0, integer;

# Conxunto de índices
set CULTIVOS:=1..n;
set VENDA:=1..m;
set COMPRA:=1..r;
set ESCENARIO:=1..s;
set ESCEN;

# Outros parámetros do modelo
param custo {i in CULTIVOS};
param venda {j in VENDA};
param compra {l in COMPRA};
param rendimento {(i, t) in {CULTIVOS,ESCENARIO}};
param prob{t in ESCENARIO} := 1/s;

# Parámetros PHA
param Rho {CULTIVOS, ESCENARIO} default 0;
param R default 2;
param x_barra {CULTIVOS} default 0;

# Variables e restricións de non negatividade
var x {i in CULTIVOS}>=0;
var w {j in VENDA}>=0;
var y {l in COMPRA}>=0;

var custos_totales = sum {i in CULTIVOS} custo[i]*x[i] - sum{j in VENDA} venda[j]*w[j] +
sum{l in COMPRA} compra[l]*y[l];
var dual = sum{i in CULTIVOS, t in ESCEN} Rho[i,t]*(x[i]-x_barra[i]);
var regularizacion = sum{i in CULTIVOS} R/2*(x[i]-x_barra[i])^2;

# Función obxectivo
minimize z: custos_totales + dual + regularizacion;
# Restricións do modelo
subject to restricion1: sum{i in CULTIVOS} x[i] <= 500;
subject to restricion2 {t in ESCEN}: rendimento[1,t]*x[1]+y[1]-w[1] >= 200;
subject to restricion3 {t in ESCEN}: rendimento[2,t]*x[2]+y[2]-w[2] >= 240;
subject to restricion4 {t in ESCEN}: w[3]+w[4] <= rendimento[3,t]*x[3];
subject to restricion5: w[3] <= 6000;

```

Figura A.3: Formulación en AMPL do subproblema a resolver empregando o PHA no problema do granxeiro.

```

param act_x {CULTIVOS, ESCENARIO} default 0;
param act_y {COMPRA, ESCENARIO} default 0;
param act_w {VENDA, ESCENARIO} default 0;
param act_obj {ESCENARIO} default 0;
param prev_Rho {CULTIVOS, ESCENARIO} default Infinity;
param prev_x_barra {CULTIVOS} default Infinity;
param crit_Rho;
param crit_x_barra;
param tol := 0.01;
param maxiter := 1000;
let crit_Rho := sum{i in CULTIVOS, t in ESCENARIO: abs(Rho[i,t]-prev_Rho[i,t]) >= 0.001} 1;
let let crit_x_barra := sum{i in CULTIVOS: abs(x_barra[i]-prev_x_barra[i]) >= 0.001} 1;
for {iter in 1..maxiter}{
for {t in ESCENARIO}{
let ESCEN := {t};
solve;
let {i in CULTIVOS} act_x[i, t] := x[i];
let {l in COMPRA} act_y[l, t] := y[l];
let {j in VENDA} act_w[j, t] := w[j];
let act_obj[t] := custos_totales;
};
let {i in CULTIVOS} prev_x_barra[i] := x_barra[i];
let {i in CULTIVOS} x_barra[i] := sum{t in ESCENARIO} prob[t]*act_x[i, t];
let {i in CULTIVOS, t in ESCENARIO} prev_Rho[i,t] := Rho[i,t];
let {i in CULTIVOS, t in ESCENARIO} Rho[i,t] := prev_Rho[i,t] + R * (act_x[i,t]-x_barra[i]);
let crit_Rho := sum{i in CULTIVOS, t in ESCENARIO: abs(Rho[i,t]-prev_Rho[i,t]) >= 0.001} 1;
let crit_x_barra := sum{i in CULTIVOS: abs(x_barra[i]-prev_x_barra[i]) >= 0.001} 1;
if crit_Rho + crit_x_barra == 0 then {
break;
};
};
};
display sum{t in ESCENARIO} prob[t]*act_obj[t];

```

Figura A.4: Ficheiro de comandos para a execução do algoritmo PHA.

Apéndice B

Código R

Neste Apéndice descríbense brevemente os scripts e funcións elaboradas para a validación do algoritmo de resolución do problema estocástico no contexto das redes de gas, introducido no Capítulo 4. Recórdase que en 5 se amosan os resultados da validación, para a obtención dos mesmos fíxose uso de código implementado na linguaxe de programación R (ver [15]) que se amosa a continuación.

`funcions_auxiliares.R`

Este script recolle a implementación de diversas funcións. A continuación describimos brevemente para que se emprega cada unha delas:

- `combinacions()`: permite construír un `data.frame`, coas combinacións dos parámetros de planificación, a partir do cal se vai crear unha batería de casos.
- `construe_casos()`: a partir dunha matriz ou `data.frame` coas combinacións dos parámetros de planificación que nos interesa e do ficheiro XML do escenario de partida, esta función crea tantas carpetas de nome “prueba_i” como combinacións se consideran, e no seu interior as subcarpetas DEM e LAG onde se recollen os novos casos coas opcións de planificación modificadas acorde ao que indica na matriz de combinacións. A carpeta DEM fai referencia a que na resolución do problema estocástico se emprega o método DEM e a LAG o algoritmo *progressive hedging*. Recordamos os parámetros de planificación que aparecen na formulación do problema e que se modifican de acorde á matriz de combinacións que se considera:
 - `plan_alpha_inc` (α_{inc}): fai referencia ao peso imposto sobre o posible incumprimento da demanda. Un valor moi elevado penalizará un posible incumprimento da demanda esixida nos nodos de consumo fronte aos custos de construción e operación na función obxectivo. Polo contrario, un valor pequeno incrementa a importancia destes últimos.
 - `plan_alpha_gc` (α_{gc}): fai referencia ao peso imposto sobre o gas consumido. Se toma un valor moi alto penalizarase o gas consumido polas estacións de compresión no escenario. Se pola contra, toma un valor pequeno, dámoslle menos peso na función obxectivo. Nas probas que se amosan, mantense en tódolos casos igual a 0.5.
 - `plan_alpha_cc` (α_{cc}): fai referencia ao peso imposto sobre os custos de construción das diferentes infraestruturas fronte a outros custos (por gas consumido) asociados. Se toma un valor pequeno estariamos a darlle maior peso á posible construción na función obxectivo.

- `crea_batch()`: esta función crea un archivo con extensión `.bat` facendo uso dos directorios onde se atopan o executable `ganeso.exe` e o escenario de partida (o ficheiro XML). Para que o directorio se escriba na sintaxe axeitada un dos argumentos desta función define o sistema operativo que se emprega.
- `probs_ac()`: permite extraer as probabilidades acumuladas do escenario de partida, as cales se gardan nun arquivo en formato `TXT`, necesarias para calcular os parámetros de converxencia do método lagrangiano. Isto é necesario porque os arquivos resultantes de aplicar o PHA non conteñen esta información. Para toda a batería de casos obtida do mesmo escenario de partida serán as mesmas.
- `le_resultados_DEM()` e `le_resultados_LAG()` como o seu nome indican, serven para a lectura dos resultados. En ambas accédese o arquivo `1_slp_sto.xml`, para recoller a información referente aos custos de incumprimento, gas consumido e construción. Por outro lado, na primeira rastrexase o arquivo `planning.log` e na segunda o arquivo `ganeso_output.txt` para a obtención dos parámetros de converxencia. Os resultados recóllense nunha matriz (`resum`) que se garda en formatos `CSV` e `TXT` no directorio de partida (`ruta_escenario`).
- `busca_erros()`: función que busca na matriz `resum` cos resultados obtidos, que se obtén das funcións anteriores, se hai algún caso que da erro na execución. Devolve un vector `fallos` cos índices das fila de `resum` que dán erro.
- `crea_batch_fallos_DEM()` e `crea_batch_fallos_LAG()`: con estas funcións créanse un batch coa ruta do executable e do caso que dea fallo.

```
#####
# Combinaciones posibles de los parámetros #
#####
combinaciones<-function(lista){
mod=lista$mod
valin=lista$valin
paso=lista$paso
ncasos=lista$ncasos
l<-list()
for(i in 1:length(valin)){
param<-numeric(ncasos[i]+1)
if(mod[i]==1){
for(j in 1:(ncasos[i]+1)){
param[j]=valin[i]+(j-1)*paso[i]
}
}else {param<-valin[i]}
l[[i]]<-param
}
comb<-expand.grid(l)
return(comb)
}

#####
# Construye casos #
#####
construe_casos<-function(comb,escenario){
for(i in 1:nrow(comb)){
#Creación de los distintos escenarios
setwd(ruta_escenario)
fileName <- escenario
con <- file(fileName,open="r")
doc <-readLines(con)

#Buscamos "plan_alpha_gc"
ind <- grep("plan_alpha_gc",doc)
new_line <- sprintf('<Cell><Data ss:Type="Number">%5.3f</Data></Cell>\' ,comb[i,1])
```



```

doc[ind+1]<-new_line

#Buscamos "plan_alpha_inc"
ind <- grep("plan_alpha_inc",doc)
new_line <- sprintf('<Cell><Data ss:Type="Number">%5.3f</Data></Cell>',comb[i,2])
doc[ind+1]<-new_line

#Buscamos "plan_alpha_cc"
ind <- grep("plan_alpha_cc",doc)
new_line <- sprintf('<Cell><Data ss:Type="Number">%5.3f</Data></Cell>',comb[i,3])
doc[ind+1]<-new_line

#Creación de carpetas
nombre_carpeta<- sprintf('prueba_%d',i)
dir.create(nombre_carpeta, showWarnings = TRUE, recursive = FALSE)
setwd(nombre_carpeta)
dir.create(DEM<-sprintf('DEM'), showWarnings = TRUE, recursive = FALSE)
dir.create(LAG<-sprintf('LAG'), showWarnings = TRUE, recursive = FALSE)
setwd(LAG)
ind <- grep("plan_type",doc)
new_line <- sprintf('<Cell><Data ss:Type="String">%09s</Data></Cell>', 'lagrange')
doc[ind+1]<-new_line
write(doc, file = "prueba.xml")

setwd(ruta_escenario)
setwd(nombre_carpeta)
setwd(DEM)
ind <- grep("plan_type",doc)
new_line <- sprintf('<Cell><Data ss:Type="String">%09s</Data></Cell>', 'DEM')
doc[ind+1]<-new_line
write(doc, file = "prueba.xml")

setwd(ruta_escenario)
close(con)
}
sprintf(' Hay un total de %d casos',nrow(comb))
return()
}

#####
#                Crea batch                #
#####
crea_batch<-function(n,ruta_executable,ruta_escenario,so){
write('', "batch.bat")
fileName <- "batch.bat"
con <- file(fileName,open="r")
doc <-readLines(con)
j<-1
for (i in 1:n){
nombre_carpeta<- sprintf('prueba_%d',i)
setwd(nombre_carpeta)
setwd(DEM<-sprintf('DEM'))
ruta_esc1<-getwd()
ruta_esc1<-paste(ruta_esc1,'/prueba.xml',sep='')
ruta_esc1<-paste('', ruta_esc1, '', sep='')
ruta_ex1<-sprintf('%s %s %s', ruta_executable, ruta_esc1, "-sto")
if (so == 0){ruta_ex1<-gsub('/', '\\\\', ruta_ex1, perl=TRUE)}
else{ruta_ex1<-gsub('\\\\', '/', ruta_ex1, perl=TRUE)}
doc[j]<-ruta_ex1
j<-j+1
setwd(ruta_escenario)
write(doc, file = "batch.bat")
setwd(nombre_carpeta)
setwd(LAG<-sprintf('LAG'))
ruta_esc2<-getwd()

```

```

ruta_esc2<-paste(ruta_esc2,'/prueba.xml',sep='')
ruta_esc2<-paste(' ',ruta_esc2,' ',sep='')
ruta_ex2<-sprintf('%s %s %s',ruta_executable,ruta_esc2,"-sto")
if (so == 0){ruta_ex2<-gsub('/', '\\\\', ruta_ex2,perl=TRUE)}
else{ruta_ex2<-gsub('\\\\\\', '/', ruta_ex2,perl=TRUE)}
doc[j]<-ruta_ex2
j<-j+1
setwd(ruta_escenario)
write(doc, file = "batch.bat")
}
close(con)
return()
}

#####
# Lee resultados DEM #
#####
le_resultados_DEM<-function(ruta_escenario,n,comb,escenario){
setwd(ruta_escenario)
#Inicializamos a matriz resumen
resum <- matrix(nrow = n, ncol = 15)
colnames(resum)<-c("alpha_inc", "alpha_gc", "alpha_cc", "SumGcost_inc",
"SumGcost_gc", "SumGcost_cc", "Sumcost_inc", "Sumcost_gc", "Sumcost_cc",
"Infact", "delta_SLP", "delta_CSLP", "Erro solver", "exists NF_problem",
" exists 1_slp_sto")
for(i in 1:n){
setwd(ruta_escenario)
name_file<- sprintf('prueba_%d',i)
setwd(name_file)
setwd(DEM<-sprintf('DEM'))
file_res <- "1_slp_sto.xml"
if (file.exists(file_res)){
result <- xmlParse(file = file_res)
rootnode <- xmlRoot(result)
# Búsqueda do índice da folla do 1_slp_sto.xml que nos interesa
ind_res= 0
for (r in 1: length(xmlChildren(rootnode))) {
if (! is.null(xmlAttrs(rootnode[[r]])[["Name"]])){
if (xmlAttrs(rootnode[[r]])[["Name"]] == "planning_obj_result"){
ind_res = r}
}
}
# Poñémonos na folla do excel "planning_obj_result":
sol2 <- rootnode[[ind_res]][[1]]
# Extraemos tódolos valores da tabla en forma de matriz
sol2_mat <- xmlSApply(sol2, function(x) xmlSApply(x, xmlValue))
sol2_df <- data.frame(t(sol2_mat),row.names=NULL)
# Recollemos custos
resum[i,4:6]<-as.matrix(sol2_df[nrow(sol2_df)-1,3:5])
resum[i,7:9]<-as.matrix(sol2_df[nrow(sol2_df),3:5])
}else{
sprintf('Non se crearon os resultados do caso %d',i)
}
#Sacamos os valores dos parámetros sen ler o 1_slp_sto
values<-as.matrix(comb[i,c(2,1,3)])
resum[i,1:3]<-values
#Rastrexamos planning.log:
plan<-readLines(con<-file("planning.log"))
slp<-grep( "delta slp " ,plan,value=F)
delta_slp<-ifelse(length(slp)==0, "problem", as.numeric(str_extract_all(plan[max(slp)],
"[0-9].+"))))
cslp<-grep( "delta_cslp_ST" ,plan,value=F)
delta_cslp<-ifelse(length(cslp)==0, "problem", as.numeric(str_extract_all(plan[max(cslp)],
"[0-9].+"))))
infact<-grep("Constraint", plan, value=T)
}
}

```

```

infact<-str_extract_all(str_extract_all(infact,"C.*"),"[0-9].+")
solve<-grep("solve code", plan, value=TRUE)
solve<-str_extract_all(str_extract_all(solve,"code.*"),"[0-9]")
resum[i,10]=ifelse(length(infact)==0, 0,sum(as.numeric(infact)))
resum[i,13]<-ifelse(length(solve)==0,0,as.numeric(solve))
resum[i,14]=ifelse(file.exists("NF_problem.xml")|file.exists("ISS_problem.ilp")|
file.exists("ALL_problem.lp"), 1, 0)
resum[i,11]<-ifelse(file.exists("1_slp_sto.xml")& resum[i,14]==0,delta_slp, "problem")
resum[i,12]<-ifelse(file.exists("1_slp_sto.xml")& resum[i,14]==0,delta_cslp,"problem")
resum[i,15]=ifelse(file.exists("1_slp_sto.xml"),1,0)
close(con)
}
#Escritura de resultados
setwd(ruta_escenario)
write.csv(x = resum, file = "resultadosDEM.csv", row.names = FALSE)
resum_DEMm<-apply(resum, 2, as.double)
print(xtable(resum_DEMm, caption=paste("Resultados DEM asociados ao escenario",
escenario, "."),digits=c(rep(2,4),rep(5,12)), display=c(rep("f",11),rep("e",2),
rep("d",3))), table.placement = getOption("xtable.table.placement", "h!"),
caption.placement = getOption("xtable.caption.placement", "bottom"), scalebox =
getOption("xtable.scalebox", 0.4), include.rownames=T,file="resultadosDEMlatex.csv")
return(resum)
}

#####
# Creacion txt coas probabilidades acumuladas #
#####
probs_ac<-function(ruta_escenario){
setwd(ruta_escenario)
name_file<- sprintf('prueba_%d',1)
setwd(name_file)
setwd(DEM<-sprintf('DEM'))
plan<-readLines(con<-file("planning.log"))
ini<-grep(" Xi_AC", plan, value=F)
fin<-grep(" > ITERA: 1", plan, value=F)
probs<-plan[ini:(fin-3)]
close(con)
setwd(ruta_escenario)
write(probs,"prob_ac.txt")
return(probs)
}

#####
# Lee resultados LAG #
#####
le_resultados_LAG<-function(ruta_escenario,n,comb,netapas,escenario){
setwd(ruta_escenario)
#Inicializar matriz resumen
resum <- matrix(nrow = n, ncol = 15)
colnames(resum)<-c("alpha_inc", "alpha_gc", "alpha_cc", "SumGcost_inc", "SumGcost_gc",
"SumGcost_cc", "Sumcost_inc", "Sumcost_gc", "Sumcost_cc", "Infact", "delta_SLP",
"delta_CSLP", "Erro solver", "exists NF_problem", "exists 1_slp_sto")
for(i in 1:n){
setwd(ruta_escenario)
name_file <- sprintf('prueba_%d',i)
setwd(name_file)
setwd(LAG<-sprintf('LAG'))
file_res <- "1_slp_sto.xml"
if (file.exists(file_res)){
result <- xmlParse(file = file_res)
rootnode <- xmlRoot(result)
# Búsqueda do índice da folla do 1_slp_sto.xml que nos interesa
ind_res= 0
for (r in 1: length(xmlChildren(rootnode))) {
if (! is.null(xmlAttrs(rootnode[[r]][["Name"]]))){

```

```

if (xmlAttrs(rootnode[[r]])[["Name"]] == "planning_obj_result"){
  ind_res = r}
}
}
# Poñémonos na folla do excel "planning_obj_result":
sol2 <- rootnode[[ind_res]][[1]]
# Extraemos tódolos valores da tabla en forma de matriz
sol2_mat <- xmlSApply(sol2, function(x) xmlSApply(x, xmlValue))
sol2_df <- data.frame(t(sol2_mat),row.names=NULL)
# Recollemos custos
resum[i,4:6]<-as.matrix(sol2_df[nrow(sol2_df)-1,3:5])
resum[i,7:9]<-as.matrix(sol2_df[nrow(sol2_df),3:5])
}else{sprintf('Non se crearon os resultados do caso %d',i)}
#Sacamos os valores dos parámetros de planificación sen ler o 1_slp_sto
values<-as.matrix(comb[i,c(2,1,3)])
resum[i,1:3]<-values
#-----Rastreamos o txt-----#
file_ganeso<-readLines(con <- file("ganeso_output.txt"))
inicio <- grep(" Applying CSLP to Lagrangian relaxation planning", file_ganeso, value=F)
j <- grep(" delta slp it_final " , file_ganeso, value=F)
a<-j>inicio
a<-j[a==TRUE]
delta_slp <- file_ganeso[a] #valores de delta_slp
delta_slp <- as.numeric(str_extract_all(delta_slp,"[0-9].+"))
c <- grep(" Iteracion final cSLP", file_ganeso, value=F)
delta_cslp <- file_ganeso[c+1] #valores de delta_cslp_ST
delta_cslp <- as.numeric(str_extract_all(delta_cslp, "[0-9].+"))
ramas<-grep("Checking",file_ganeso,value=F)
ramas<-file_ganeso[ramas[ramas>inicio]]
#Eliminamos espacios en blanco
rama<-gsub(pattern = " ", replacement = "", x = ramas, ignore.case = F)
#ramas do proceso iterativo
rama_sel<-str_extract(rama,gsub(" ","",paste("[:digit:]{",netapas,"}")))
infact<-grep("Total incumplimiento =", file_ganeso, value=T)
infact1<-as.numeric(str_extract_all(infact,"[0-9].+"))
solve<-grep("solve code", file_ganeso, value=TRUE)
solve<-str_extract_all(str_extract_all(solve,"code.*"),"[0-9]")
resum[i,10]<- sum(infact1)
resum[i,13]<-ifelse(length(solve)==0,0,as.numeric(solve))
resum[i,14]<-ifelse(file.exists("NF_problem.xml")|file.exists("ISS_problem.ilp")|
file.exists("ALL_problem.lp"),1, 0)
resum[i,15]<-ifelse(file.exists("1_slp_sto.xml"),1,0)
####Ramas do árbol: obtención das probabilidades acumuladas
setwd(ruta_escenario)
probs<-readLines(con1<-file("prob_ac.txt"))
ramas<-str_extract(probs[3:length(probs)],gsub(" ","",paste("[:digit:]{",netapas,"}")))
p_ac<-as.numeric(str_extract(probs[3:length(probs)],"[:graph:]{6}"))
#Cálculo de delta_slp e delta_cslp total
g<-intersect(ramas, rama_sel)
delta_slp_fin<-0
delta_cslp_fin<-0
for(k in 1:length(g)){
delta_slp_fin<- p_ac[ramas==g[k]]*delta_slp[k]+delta_slp_fin
delta_cslp_fin<- p_ac[ramas==g[k]]*delta_cslp[k]+delta_cslp_fin
}
resum[i,11] <- ifelse(length(delta_slp)==0 |resum[i,14]==1, "problem", delta_slp_fin)
resum[i,12] <- ifelse(length(delta_cslp)==0 |resum[i,14]==1, "problem", delta_cslp_fin)
close(con1)
close(con)
}
#Escritura de resultados
setwd(ruta_escenario)
write.csv(x = resum, file = "resultadosLAG.csv", row.names = FALSE)
resum_LAGm<-apply(resum, 2, as.double)
print(xtable(resum_LAGm, caption=paste("Resultados LAG asociados ao escenario",

```

```

escenario, "."),digits=c(rep(2,4), rep(5,12)), display=c(rep("f",11),rep("e",2),
rep("d",3))), table.placement = getOption("xtable.table.placement", "h!"),
caption.placement = getOption("xtable.caption.placement", "bottom"), scalebox =
getOption("xtable.scalebox", 0.4),include.rownames=T,file="resultadosLAGlatex.csv")
return(resum)
}

```

```

#####
# Busca erros de ejecución #
#####

```

```

busca_erros<-function(resum,n){
fallos<-0
j<-1
for( i in 1:n){
if(length(which(is.na(resum[i,]))>=1){
fallos[j]<-i
j<-j+1
}
}
for(k in 1:j){
sprintf('No ejecutó correctamente el caso %f',fallos[k])
}
return(fallos)
}

```

```

#####
# Crea batch fallos DEM #
#####

```

```

crea_batch_fallos_DEM<-function(ruta_executable,ruta_escenario,fallos){
setwd(ruta_escenario)
write('',"batch_fallos_DEM.bat")
fileName <- "batch_fallos_DEM.bat"
con <- file(fileName,open="r")
doc <-readLines(con)
j<-1
for (i in 1:length(fallos)){
nombre_carpeta<- sprintf('prueba_%d',fallos[i])
setwd(nombre_carpeta)
setwd(DEM<-sprintf('DEM'))
ruta_esc1<-getwd()
ruta_esc1<-paste(ruta_esc1,'/prueba.xml',sep='')
ruta_esc1<-paste('',ruta_esc1,'',sep='')
ruta_ex1<-sprintf('%s %s %s',ruta_executable,ruta_esc1,"-sto")
if (so == 0){ruta_ex1<-gsub('/', '\\\\', ruta_ex1,perl=TRUE)}
else{ruta_ex1<-gsub('\\\\\\', '/', ruta_ex1,perl=TRUE)}
doc[j]<-ruta_ex1
setwd(ruta_escenario)
write(doc, file = "batch_fallos_DEM.bat")
j<-j+1
}
close(con)
return()
}

```

```

#####
# Crea batch fallos LAG #
#####

```

```

crea_batch_fallos_LAG<-function(ruta_executable,ruta_escenario,fallos){
setwd(ruta_escenario)
write('',"batch_fallos_LAG.bat")
fileName <- "batch_fallos_LAG.bat"
con <- file(fileName,open="r")
doc <-readLines(con)
j<-1
for (i in 1:length(fallos)){

```

```

nombre_carpeta<- sprintf('prueba_%d',fallos[i])
setwd(nombre_carpeta)
setwd(LAG<-sprintf('LAG'))
ruta_esc2<-getwd()
ruta_esc2<-paste(ruta_esc2,'/prueba.xml',sep='')
ruta_esc2<-paste('',ruta_esc2,'',sep='')
ruta_ex2<-sprintf('%s %s %s',ruta_executable,ruta_esc2,"-sto")
if (so == 0){ruta_ex2<-gsub('/', '\\\\', ruta_ex2,perl=TRUE)}
else{ruta_ex2<-gsub('\\\\', '/', ruta_ex2,perl=TRUE)}
doc[j]<-ruta_ex2
setwd(ruta_escenario)
write(doc, file = "batch_fallos_LAG.bat")
j<-j+1
}
close(con)
return()
}

```

tempos.R

Este script permítenos obter os tempos de execución de cada caso, obtidos a partir dun mesmo escenario de partida.

```

library(stringr)
library(xtable)
# Lectura do arquivo auxiliar que se xenera e que contén os tempos.
salida<-readLines(arq<-file("valida.Rout"))
# Busca dos tempos:
CPU<-grep(" CPU time ",salida, value=T)
CPUtime<-as.numeric(gsub("seconds.", "",str_extract_all(str_extract_all(CPU, "C.*"),
"[0-9].+"))))

REAL<-grep(" Real time ",salida, value=T)
Realtme<-as.numeric(gsub("seconds.", "",str_extract_all(str_extract_all(REAL, "R.*"),
"[0-9].+"))))

# Construción da matriz de tempos de execución
r<-length(Realtme)
time<-matrix(0, nrow=r/2, ncol=4)
colnames(time)<-c("CPU DEM", "CPU PHA", "Real DEM", "Real PHA")
time[,1]<-CPUtime[seq(1,r-1,2)]
time[,2]<-CPUtime[seq(2,r,2)]
time[,3]<-Realtme[seq(1,r-1,2)]
time[,4]<-Realtme[seq(2,r,2)]

# Gardar a matriz en formato latex nun arquivo .csv.
print(xtable(time),file="time.csv")

```

lectura_lambda_xes.R

Este script permite obter os parámetros de converxencia da implementación do PHA realizada polo equipo investigador, a representación destes e a obtención do número de iteracións realizadas empregando o PHA.

```

#####
#           Parámetros de converxencia do PHA           #
#           e número de iteracións                       #
#####
library(stringr)

```

```

library(xtable)
n=20 # n° de casos
ruta_escenario<-getwd()

PHA<-function(n, ruta_escenario){
  setwd(ruta_escenario)
  # Creamos a carpeta parámetros onde se gardaran as representacións
  # dos parámetros de converxencia do PHA.
  dir.create(parametros<-sprintf('parametros'), showWarnings = TRUE, recursive = FALSE)
  # Matriz nula onde se almacenarán o número de iteracións
  iter<-matrix(0, nrow=n, ncol=2)
  # Recorremos as carpetas coas saídas do algoritmo PHA e rastrexamos
  #o arquivo planning.log.
  for(i in 1:n){
    setwd(ruta_escenario)
    name_file<- sprintf('prueba_%d',i)
    setwd(name_file)
    setwd(LAG<-sprintf('LAG'))
    file_planning<-readLines(con<-file("planning.log"))
    lambda<-as.numeric(str_extract_all(grep(" Convergence lambda ", file_planning,value=T),
    "[0-9].+"))
    xes<-as.numeric(str_extract_all(grep(" Convergence xes ", file_planning, value=T),
    "[0-9].+"))
    iter[i,1]<-length(lambda)
    iter[i,2]<-length(xes)
    setwd(ruta_escenario)
    setwd(parametros)
    pdf(gsub(" ", "", paste("parametros", i, ".pdf")))
    par(mfrow=c(1,2))
    plot(lambda,xlab=" ", type="b")
    abline(h=0, col="blue")
    plot(xes,xlab=" ",type="b")
    abline(h=0, col="blue")
    dev.off()
    close(con)
  }
  print(xtable(iter), file="iter.csv")
  return()
}

```

script_lectura_execucion.R

Este script contén as funcións `execucion()` e `lectura()` que son comúns para calquera escenario de partida.

- `execucion()`: encárgase da execución do `.bat` para a resolución do problema, engloba as funcións `combinacions()`, `construe_casos()` e `crea_batch()`.
- `lectura()`: emprégase para a lectura dos resultados obtidos e a busca de erros na execución. Engloba as restantes funcións auxiliares que non se mencionaron na función `execucion()`.

```

execucion<-function(comb, escenario, ruta_executable, ruta_escenario, n, so){
  casos<-construe_casos(comb,escenario)
  # Creación batch
  crea_batch(n,ruta_executable,ruta_escenario,so)
  # Execución batch
  if(so==0){shell('batch.bat')} #en local
  else{system(' chmod +x batch.bat') #no cluster
  system('./batch.bat')}
}
return()

```

```

}

lectura<-function(ruta_escenario, escenario, netapas, n, comb, ruta_executable){
# Lectura e escritura de resultados
resum_DEM <- le_resultados_DEM(ruta_escenario, n, comb, escenario)
prob_ac <- probs_ac(ruta_escenario) #función auxiliar para lectura de datos en el LAG
resum_LAG <- le_resultados_LAG(ruta_escenario, n, comb, netapas, escenario)

# Busca erros de ejecución
fallos_DEM <- busca_erros(resum_DEM, n)
fallos_LAG <- busca_erros(resum_LAG, n)

# Crea batch con ejecuciones erróneas
if(fallos_DEM[1]!=0){
crea_batch_fallos_DEM(ruta_executable, ruta_escenario, fallos_DEM)}
if(fallos_LAG[1]!=0){
crea_batch_fallos_LAG(ruta_executable, ruta_escenario, fallos_LAG)}
return()
}

```

valida.R

Mediante este script invócase a ejecución do módulo de planificación de GANESO e a lectura dos resultados obtidos facendo uso das funcións implementadas (recollidas nos scripts `funciones_auxiliares` e `script_lectura_execucion`). Ademais, cárganse as as librerías necesarias: `XML`, `XLConnectJars`, `XLConnect` e `xlsx` que se empregan para o tratamento dos arquivos en formato XML (modificación, lectura e escritura de resultados), a librería `stringr` para a extracción de resultados que precisan do uso de expresións regulares e a librería `xtable` para a escritura dos resultados en código LaTeX. Tamén se definen os datos de entrada necesarios para a ejecución das funcións implementadas:

- `escenario`: nome do escenario da rede de gas a partir do cal se obtén a batería de casos.
- `ruta_executable`: directorio do executable `ganeso.exe`.
- `ruta_escenario`: directorio do ficheiro XML que define o escenario cuxo nome se indica en `escenario`.
- `comb`: matriz ou `data.frame` coas combinacións dos parámetros de planificación que se empregan na construción da batería de casos.
- `so`: definición do sistema operativo, 0:Windows y 1:Linux.

```

library("XML")
library("XLConnectJars")
library("XLConnect")
library("xlsx")
library("xtable")
library("stringr")

source("funciones_auxiliares.R")
source("script_lectura_execucion.R")

# Nome do escenario de partida
escenario<-"spain3.xml"

# Sistema operativo (0 => Windows, 1=> Linux)
so<-0

```



```

# Cambios nos parámetros
#lista<-list(mod=c(0,1,1), valin=c(0.5,1000,0.5), paso=c(1,500,1.5), ncasos=c(3,4,4))
#comb<-combinaciones(lista)
# Nos vectores mod, valin, paso e ncasos as compoñentes refírense a alpha_gc,
#alpha_inc, alpha_cc nesa orde.
#mod: parámetro nesa posición modifícase (1) ou non (0)
#valin: valor inicial do parámetro
#paso: cambio que sufrirá o parámetro no caso de ser modificado
#ncasos: nº de modificaciónes

ruta_escenario<-getwd()

#Matriz de combinación dos parámetros
comb<-matrix(c(0.5,0.001,0.001, 0.5,0.05,0.001,
0.5,0.1,0.001, 0.5,0.5,0.001,
0.5,1,0.001, 0.5,0.001,0.05,
0.5,0.05,0.05, 0.5,0.1,0.05,
0.5,0.5,0.05, 0.5,1,0.05,
0.5,0.001,0.1, 0.5,0.05,0.1,
0.5,0.1,0.1, 0.5,0.5,0.1,
0.5,1,0.1, 0.5,0.001,0.5,
0.5,0.05,0.5, 0.5,0.1,0.5,
0.5,0.5,0.5, 0.5,1,0.5), ncol=3, byrow=T)
n=dim(comb)[1]

#Etapas da arbore
setwd(ruta_escenario)
esc<-readLines(escenario)
plan_stage<-grep("plan_stages", esc)
netapas<-as.numeric(str_extract(esc[plan_stage+1], "[:digit:]" ))

#Definición das rutas a usar
if(so==0){
ruta_executable<- "C:/Program Files (x86)/Quantum GIS Lisboa/Kernel/5.0.1_gurobi/ganeso.exe"
} else{ruta_executable<- "/home/ana.buide/ganesoKernel/bin/x86_64_gurobi/ganeso.exe"}
ruta_executable

# Construír novos casos de execución
crea_casos_batch<-ejecucion(comb, escenario, ruta_executable, ruta_escenario, n, so)

#Lectura de resultados
resultados<-lectura(ruta_escenario, escenario, netapas, n, comb, ruta_executable)

```

comparar.R

Este script serve para comparar os resultados de distintos casos, algo similar ao que se pode facer na interface de GANESO, pero externamente, sen ter que recorrer a interface. Neste script recóllese a función `comparativa()` que permite obter gráficas de radar e táboas para ver:

- Uso de infraestruturas. Podemos coñecer o tanto por cen de uso de cada infraestruturas mediante a representación da gráfica de radar e a táboa coa información asociada. Amosaremos os resultados referentes a última etapa.
- Construción por etapas. Podemos visualizar as diferentes infraestruturas consideradas e a primeira etapa na que aparecerían construídas.

Para a representación das gráficas de radar, tamén coñecidas como diagramas de araña, fíxose uso da función `radarchart` da librería `fmsb`.

```
escenario<-"spain3.xml"
```

```

escenario1<-"spain3.xlsx"
ruta_escenario<-getwd()

##Librerías que emplearemos:
library("stringr")
library("XML")
library("xlsx")
library(fmsb)

##Datos
A=read.csv("resultadosLAG.csv",sep=",") #data.frame con los resultados obtenidos de ganeso
A1=A[, 1:9] #escenarios que queremos comparar (modificar sólo las filas)
A=A[,1:4] #distintos casos que hemos considerado en el escenario
esc_comp<-as.numeric(intersect(rownames(A),rownames(A1)))
carpeta<-"LAG"

comparativa<-function(ruta_escenario,escenario, escenario1, esc_comp, carpeta){
setwd(ruta_escenario)
dir.create(comparativa <- sprintf('Comparativa'), showWarnings = TRUE, recursive = FALSE)
setwd(comparativa)
dir.create(carpeta <- sprintf(carpeta), showWarnings = TRUE, recursive = FALSE)
#Etapas del arbol
setwd(ruta_escenario)
esc <- readLines(escenario)
plan_stage <- grep("plan_stages", esc)
netapas <- as.numeric(str_extract(esc[plan_stage+1], "[:digit:]" ))
planning <- read.xlsx(escenario1, sheetIndex = "planning")
ninfra <- dim(planning)[1]-1
infra <- as.matrix(planning[-1,2])
uso_infra <- matrix(nrow=length(esc_comp), ncol=ninfra*(netapas+1))
colnames(uso_infra) <- rep(paste("E", 0:netapas), ninfra)
rownames(uso_infra) <- paste("1_slp_sto", esc_comp)
cons_etapa <- matrix(nrow=length(esc_comp), ncol=ninfra)
colnames(cons_etapa) <- c(infra)
rownames(cons_etapa) <- paste("1_slp_sto", esc_comp)

for(i in 1:length(esc_comp)){
setwd(ruta_escenario)
name_file <- sprintf('prueba_%d',esc_comp[i])
setwd(name_file)
setwd(carpeta <- sprintf(carpeta))
file_res <- "1_slp_sto.xml"
doc <- readLines(con<-file(file_res,open="r"))
if (file.exists(file_res)){
result <- xmlParse(file = file_res)
rootnode <- xmlRoot(result)
# Poñémonos na folla do excel "planning_infra_result":
plan_infra_result <- rootnode[[17]][[1]]
# Extraemos tódolos valores da tabla en forma de matriz:
plan_infra_result_mat <- xmlSApply( plan_infra_result , function(x) xmlSApply(x, xmlValue))
plan_infra_result_df <- data.frame(t( plan_infra_result_mat[-(1:2)]),row.names=NULL)
E<-plan_infra_result_df[,1]
levels(E)=infra
t<-plan_infra_result_df[,2]
econs<-plan_infra_result_df[,3]
euso<-plan_infra_result_df[,4]
uso_infra[i,]<-as.vector(euso)
for(k in 1:length(E[econs==1])){
cons_etapa[i, as.numeric(E[econs==1])[k]]<-as.numeric(t[econs==1][k])-1
}
}
setwd(ruta_escenario)
setwd(comparativa)
setwd(carpeta)
write(doc, file =gsub(" ", "", paste("1_slp_sto",esc_comp[i],".xml")))
}
}

```

```

close(con)
}
uso_infra<-apply(uso_infra,2,as.numeric)
rownames(uso_infra)<-paste("1_slp_sto",esc_comp)
write.csv(uso_infra, "%_uso_infra.csv")
cons_etapa[is.na(cons_etapa)] <- 0
write.csv(cons_etapa, "1_etapa_cons.csv")

#Radarchart aplicado á matriz cons_etapa
cons_etapa1<-as.data.frame(t(cons_etapa))
pdf("1_etapa_construida.pdf")
radarchart(rbind(rep(netapas,length(esc_comp)), rep(0,length(esc_comp)), cons_etapa1),
axistype=1, seg=netapas, #custom polygon
pcol= rainbow(ninfra,alpha=0.8), pfcpl= rainbow(ninfra,alpha=0.2), plwd=4, plty=1, #custom the grid
cglcol="grey", cglty=1, axislabcol="grey",caxislabels=seq(0,netapas,1), cglwd=0.8,
centerzero=TRUE, vlce=0.8, title="Primeira etapa construida", vlabels=rownames(cons_etapa))
legend("topleft", legend = infra, bty = "n", pch=20, col=rainbow(ninfra, alpha=0.8),
text.col = "grey", cex=0.75,pt.cex=1)
dev.off()

#Radarchart aplicado á matriz uso_infra
windows()
uso_infra2<-as.data.frame(uso_infra)
uso_infra3<-as.data.frame(uso_infra2[names(uso_infra2)==paste("E",netapas)])
data<-as.data.frame(t(cbind(rep(1,length(esc_comp)), rep(0,length(esc_comp)), uso_infra3)))
pdf("uso_infraestructuras.pdf")
radarchart(data,axistype=1,seg=5, #custom polygon
pcol= rainbow(ninfra,alpha=0.8), pfcpl= rainbow(ninfra,alpha=0.2), plwd=4, plty=1, #custom the grid
cglcol="grey", cglty=1,axislabcol="grey",caxislabels=seq(0,1,0.2), cglwd=0.8,
centerzero=TRUE, vlce=0.8, title="% utilización das infraestructuras", vlabels=rownames(uso_infra))
legend("topleft", legend = infra, bty = "n", pch=20, col=rainbow(ninfra, alpha=0.8),
text.col = "grey", cex=0.75,pt.cex=1)
dev.off()
return()
}

comparativa(ruta_escenario, escenario, escenario1, esc_comp, carpeta)
setwd(ruta_escenario)

```


Bibliografía

- [1] Bazaraa M.S., Sherali H.D, and Shetty C.M. (2006) *Nonlinear programming: Theory and algorithms*. John Wiley & Sons.
- [2] Beale, E.M.L. (1995) *On minimizing a convex function subject to linear inequalities*. Royal Statistical Society, 17, 173–184.
- [3] Bermúdez A., González Díaz J., González-Diéguez F.J., González Rueda A.M and Fernández de Córdoba M.P. (2015) *Simulation and optimization models of steady-state gas transmission networks*. Energy Procedia, 64, 130–139.
- [4] Birge, J. R. and Louveaux, F. (2011) *Introduction to Stochastic Programming*. New York City: Springer.
- [5] Buide Carballosa, A.B. (2017) *Informe de validación del módulo de planificación*. ITMATI, Santiago de Compostela.
- [6] Ciarlet,P.G. (1999) *Introducción á análise numérica matricial e á optimización*. Servicio de Publicacións da USC.
- [7] Dantzig, G. (1955) *Linear Programming Under Uncertainty*. Management Science, 1, 197–206.
- [8] Fernández de Córdoba, M.P. (2015) *Optimización no linear: un pequeño giro a la programación linear sucesiva*. Trabajo Fin de Máster. Universidade de Santiago de Compostela.
- [9] Fourer, R., Gay, D. and Kernighan, B. (2002). *AMPL: A Modeling Language for Mathematical Programming* (Second Edition). Duxbury Press, Brooks/Cole Publishing Company.
- [10] González Díaz J. (2016). *Apuntes de la materia de Programación Matemática*. Universidade de Santiago de Compostela, Santiago de Compostela. Máster en Técnicas Estadísticas.
- [11] González Diéguez, F. J. (2017) *Modeling, simulation and optimization of gas transport networks*. Ph.D. thesis. Departamento de Matemática Aplicada, Departamento de Estadística, Análise Matemática e Optimización. Universidade de Santiago de Compostela.
- [12] González Díaz, J., González Rueda, A.M and Fernández de Córdoba, M.P (2016) *A two step sequential linear programming algorithm for MINLP problems: An application to gas transmission networks*. Mimeo.
- [13] ITMATI (2016) *Simulación numérica y optimización de la red española de gas natural. Informe 14: Programación estocástica*. Santiago de Compostela.
- [14] Palacios-Gómez,F., Lasdon L. and Engquist M. (1982) *Non linear optimization by Succesive Linear Programming Source*. Management Science, 10, 1106–1120.

- [15] R Core Team (2016). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.r-project.org/>
- [16] Rockafellar R.T and Wets R.J-B Wets (1991) *Scenarios and policy aggregation in optimization under uncertainty*. Mathematics of Operations Research, 16, 119–147.