

(MEMORIA DEL PROYECTO)

versión: 24/01/13 01:03:37

Autores: Enzo Coppes, Federico Barreto, Carlos Tutté, Fernanda Maciel, Marcelo Forets, Eliana Cornalino, Milena Gurín Añasco, María Cristina Álvarez, Felipe Palacios, Daniel Cohn.y Ruben

Chaer*

(*) Responsable científico del proyecto.

Instituto de Ingeniería Eléctrica.

Noviembre2012

Institución: IIE – Montevideo – Uruguay.



Prefacio.

La vida es una trayectoria que exige permanentemente tomar decisiones con información incompleta. La intuición es la que nos ayuda a completar la falta de información y la intuición sólo es posible con el ejercicio que nos permite el probar y aprender los comportamientos y reacciones de las diferentes partes del universo. Un simulador de un sistema es nada más ni nada menos que una herramienta para ejercitar y lograr el conocimiento y capacidad de intuición que de otra forma sería muy costosa.

SimSEE es una plataforma de Simulación de Sistemas de Energía Eléctrica desarrollada con el propósito de facilitar la construcción de simuladores de sistemas de energía eléctrica. Los simuladores creados son herramientas de análisis de diferentes formas de operación del sistema mediante simulaciones de las diferentes escenarios, que permiten calcular los costos futuros de seguir por cada una de las trayectorias posibles y experimentar las consecuencias de las diferentes decisiones. SimSEE fue desarrollado en el Instituto de Ingeniería Eléctrica (IIE) de la Facultad de Ingeniería (FING) en el marco del proyecto PDT 47/12 durante los años 2006 y 2007.

Esta memoria, documenta el conjunto de mejoras realizadas en el marco del proyecto ANII-FSE-2009-18 financiado por el Fondo Sectorial de Energía de la ANII durante los años 2011 y 2012. El proyecto se llevó a cabo en el Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de la República Oriental del Uruguay.

Las principales mejoras realizadas son:

- Cálculo de la Política Óptima de Operación con Aversión al Riesgo.
- Modelado de la red eléctrica para el despacho de la Generación Distribuida.
- Desarrollo de una plataforma de Optimización Distribuida de Funciones de Alto Costo de Valuación.
- Implementación de la versión distribuida de SimSEE para cálculo distribuido en red de computadoras.
- Módulo de cálculo de emisiones de CO₂ del sistema eléctrico.

Ruben Chaer / Director proyecto SimSEE.

Índice de contenido

Capítulo I.SimSEE+Flucar.....	6
A.Nomenclatura.....	6
B.Introducción.....	6
C.Adaptación del modelo de red a la plataforma.....	7
i.Planteo general del problema de flujo de cargas, estudio y adaptación	7
ii.Incorporación de los TAP's variables de los transformadores a la formulación del problema.....	7
a)Nodos tipo k (o genéricos).....	8
b)Nodos tipo regulado (o tipo h).....	9
c)Nodos tipo reguladores (o tipo m).....	9
d)Matriz Jacobiana.....	10
iii.Desarrollo de los algoritmos de lectura de datos de los archivos de datos de red del PSSE.....	11
iv.Estudio de diferentes formas de incorporar la red al SimSEE e implementación de la solución elegida.....	12
v.Solución implementada.....	12
vi.Pruebas y validación de los resultados obtenidos.....	14
a)Caso de prueba del flujo de cargas.....	14
b)Caso de prueba SimSEE.....	14
Capítulo II.SimSEE+Miner2012.....	15
A.Introducción.....	15
B.Nociones básicas de optimización.....	16
i.Supremo de un problema.....	16
ii.Convexidad y búsqueda local.....	17
iii.Relajación de un problema.....	19
iv. Optimización sin restricciones.....	20
v. Optimización con restricciones – Relajaciones paramétricas.....	21
vi.MINER2012 – Implementación en SimSEE.....	22
vii.Detalles de la implementación MINER2012.....	24
viii.Consluciones.....	25
Capítulo III.SimSEE + VaR.....	26
A.Introduction.....	26
B.Stochastic dynamic programming with histogram.....	28
C.VALUE AT RISK VaR.....	30
D.risk averse objective function.....	31
E. case of application.....	32
F.conclusions.....	33
G.References.....	33
H.Biographies.....	33
Capítulo IV.SimSEE+RIEE.....	34
A.Introducción.....	34
B.Planteo del problema – Definición de una Política de Operación – PO.....	35
i.La idea original.....	35
C.Reducción de la dimensión de fuentes de ruido con dinámica – CEGH.....	37
Capítulo V.SimSEE+NetTopos.....	38
A.Objetivo.....	38
B.Introducción.....	38

C.Problema.....	38
i.Planteamiento del problema	38
ii.Formulación del problema	39
iii.Resolución del problema	39
iv.Distribuidor de Tareas.....	40
D.Diagrama de Clases.....	40
E.Jerarquía de clases.....	41
F.Optimizador distribuido.....	42
G.Simulador distribuido.....	43
i.Resultados.....	44
Capítulo VI.OddFace–Teoría.....	45
A.Capa de comunicación entre los agentes.....	47
B.Implementación Algoritmos Genéticos en OddFace v1.0.....	51
i. Definiciones básicas para manejo de cadenas de ADN.....	51
a)ADN-> Genotipo y Fenotipo.....	52
b)Descriptores de Genotipos.....	53
ii.Definición de Clases básicas para descripción del problemas OddFace.....	54
a)TIndividuo.....	55
iii.Defectos de la implementación OddFace v1.0.....	57
a)Muestreos de Monte Carlo.....	57
b)Residuo de bits en la cadena de ADN.....	58
C.OddFace_prepare – Manual de Usuario.	61
i.Introducción.....	61
ii.Listado de problemas.....	62
iii.Edición de los parámetros de un OddFace.....	64
iv.Consultas sobre el historial de evaluaciones.....	69
Capítulo VII.OddFace-PAM1.....	73
A.Introducción.....	73
i.Versiones Oddface:.....	74
B.Casos de estudio (con Aglutinar resultados y versión oddface_prepare_Aglutinante_v2):... 74	
i.Parámetros generales para todos los casos (no son modificados):.....	74
ii.CASO 1.....	76
a)Ídem al anterior pero con version OddFace_v11_GEA_binarios.....	78
iii.CASO 2.....	82
a)Ídem al 117 pero con version OddFace_v11_GEA_binarios.....	84
iv.CASO 3.....	88
v.CASO 4.....	91
vi.CASO 5.....	93
vii.CASO 6.....	97
viii.CASO 7.....	99
ix.CASO 8.....	103
x.CASO 9.....	106
xi.Resumen de los casos:.....	108
C.CONCLUSIONES.....	110
D.ANEXO: “Bugs” detectados y mejoras propuestas por el equipo de trabajo que fueron implementadas durante la etapa de trabajo.....	111
E. Anexo: Propuestas de mejora al programa hacia el futuro.....	117
Capítulo VIII.OddFace+PAM (2).....	118
A. Definición del problema.....	118
B. Caso de estudio analizado.....	122

i. ANALISIS DE LA MEJOR SOLUCION.....	124
ii. ANALISIS DEL CONJUNTO DE SOLUCIONES FACTIBLES.....	131
a) Análisis de la solución optima.....	131
b) Análisis del resto de las soluciones factibles.....	133
c) CONCLUSIONES	139
iii. ANALISIS DE SOLUCIONES VARIANDO VALOR ESPERADO, AVERSION AL RIESGO Y DIAS POR ETAPAS.....	140
a) CASO BASE con $ro_VE=0.85$ y $ro_VaR=0.15$ –prob. 130 (Versión original).....	140
b) CASO BASE-prob. 146 (Versión nueva).....	142
c) CASO BASE con $ro_VE=0.85$ y $ro_CVaR=0.15$ –prob. 141 (Nueva versión).....	144
d) CASO BASE con $ro_VE=0.2$ y $ro_CVaR=0.8$ – prob. 156 (Versión nueva).....	145
e) CASO BASE con 15 días/etapa-prob. 139 (Versión nueva).....	146
f) CASO BASE con 10 días/etapa-prob. 140 (Versión nueva).....	148
g) ANÁLISIS Y CONCLUSIONES.....	149
AVERSION AL RIESGO.....	149
C. CONCLUSIONES GENERALES.....	151
Capítulo IX. OddFace+PIG.....	153
A. Planteo del Problema:.....	153
B. Resolución del Problema:	154
C. Consulta y análisis de la solución:.....	155
D. Caso de estudio.....	156
i. Planteo:.....	156
ii. Resolución:.....	157
iii. Análisis de resultados:.....	159
E. Errores encontrados.....	160
i. Sugerencias.....	160
F. Conclusiones.....	161
G. Referencias.....	161
Capítulo X. SimSEE+CO2.....	162
A. Introducción.....	162
B. Definiciones.....	162
i. Factor de emisión de una unidad/planta generadora.....	163
a) Ejemplo de cálculo.....	165
b) Obtención de datos para los cálculos.....	165
ii. Factor de emisión del sistema eléctrico.....	165
C. Metodología de cálculo.....	167
i. Factor de emisión OM.....	167
a) OM simple.....	167
b) OM simple ajustado.....	168
c) OM promedio.....	171
ii. Factor de emisión BM.....	171
iii. Factor de emisión CM.....	173
iv. Aplicabilidad y recomendaciones ó Conclusión y recomendaciones.....	173

Capítulo I. SimSEE+Flucar.

Implementación de módulo de flujo de cargas a la plataforma SimSEE y su aplicación al análisis de restricciones de operación del sistema uruguayo con alta penetración de energía eólica.

Autores: E. Coppes, M. Forets, A. Rondoni, R. Escuder y Ruben Chaer.

Resumen—Este trabajo presenta el resultado de una parte del Proyecto ANII FSE2009-18 "Mejoras a la plataforma de Simulación de Sistemas de Energía Eléctrica - SimSEE". SimSEE es una plataforma de cálculo para la coordinación hidrotérmica desarrollado en la FING-IIE-UDELAR. Actualmente SimSEE permite la definición de nodos y arcos que posibilitan tener una primera aproximación al problema reflejando en forma aproximada los límites de potencia por las líneas de transmisión y las pérdidas en las mismas, pero ignorando por completo posibles problemas de tensión de las barras. Al pensarse en una expansión en base a energía eólica y/o centrales de biomasa distribuidas geográficamente e interconectadas al sistema por líneas radiales es claro que será necesario considerar las afectaciones que sobre la tensión tendrán estas instalaciones. Debido al porte de la generación distribuida que va a ser incorporada al SIN, es necesario tener en cuenta las limitaciones que imponen las líneas y transformadores en cuanto a su capacidad de transporte. Por esta razón se incorporo al SimSEE un modelado más exacto del sistema de red, como es un flujo de cargas AC, que permita incluir las restricciones sobre las tensiones de los nodos y los límites de capacidad de las canalizaciones.

Index Terms—flujo de cargas AC, generación distribuida, coordinación hidrotérmica.

A. Nomenclatura

ANII: Agencia Nacional de Investigación e Innovación

IIE: Instituto de Ingeniería Eléctrica.

FING: Facultad de Ingeniería.

PSSE: Programa comercial de SIEMENS de flujo de cargas.

SimSEE: Simulador de Sistemas Energía Eléctrica.

SIN: Sistema Interconectado Nacional.

UDELAR: Universidad de la República Oriental del Uruguay.

B. Introducción

PARA la incorporación de la red eléctrica al programa SimSEE se considero conveniente utilizar, un programa de flujo de cargas desarrollado en el IIE-FING llamado FLUCAR que estaba implementado en lenguaje Pascal, el mismo con el que esta desarrollado SimSEE. Esto permitió integrar la capacidad de hacer las ecuaciones de flujo de carga directamente sobre SimSEE, evitando de esta forma afectaciones en los tiempos de cálculo e integrando todas las funcionalidades en un solo entorno para facilidad del usuario final. Se hizo una reingeniería de este flujo de cargas incorporando un resolvidor de ecuaciones en variable complejas que utiliza el método de Newton-Raphson. Utilizando estos algoritmos se incorporo también, dentro del algoritmo de optimización, un tratamiento especial para los TAP's de los transformadores, para lo cual se considero otra variable compleja, aparte de las de voltaje y potencia en cada nudo.

Un punto importante que se tuvo en cuenta, fue la lectura de los datos de RED. Para ello se evaluaron los programas de flujo de carga conocidos y se decidió que, por su facilidad y amplitud de uso, el formato de flujo de cargas de PSSE (Power System Simulator for Engineering tool) de la empresa SIEMENS, en texto plano o RAW, proporciona una forma de entrada de los datos cuasi estandarizada y ampliamente usada tanto en Uruguay como en el MERCOSUR.

Otro aspecto considerado fue la incorporación del flujo de cargas en el algoritmo de optimización. Para ello se analizaron las diferentes alternativas usadas en la literatura especializada. Este aspecto del problema se resolvió hacerlo en dos etapas en el cual la primera consiste en resolver el problema del despacho hidrotérmico a nivel de balance energético en pasos de tiempo divididos en bandas horarias para representar las solicitudes de diferentes potencias a diferentes horas del día, luego de lo cual se hace una corrida de flujo de cargas para ver si el

problema es factible desde el punto de vista de la red y si los flujos de potencia circulan por los corredores de transportes resultante del despacho energético. En caso de que el problema de red no sea factible o exista un apartamiento importante de los flujos, se itera haciendo las correcciones necesarias en el despacho energético hasta lograr una convergencia entre la representación energética y la representación detallada de red en cada uno de los postes del paso de tiempo de simulación en SimSEE.

Se hicieron pruebas con un sistema sencillo y luego con una versión simplificada del sistema eléctrico uruguayo.

C. Adaptación del modelo de red a la plataforma

Para adaptar el modelo de red eléctrica a la plataforma SimSEE se planteo el siguiente plan de trabajo:

- Planteo general del problema de flujo de cargas, estudio y adaptación.
- Incorporación de los TAP's variables de los transformadores a la formulación del problema.
- Desarrollo de los algoritmos de lectura de datos de los archivos de datos de red del PSSE.
- Estudio de diferentes formas de incorporar la red al SimSEE e implementación de la solución elegida.
- Pruebas y validación de los resultados obtenidos.

i. Planteo general del problema de flujo de cargas, estudio y adaptación

Se planteo la resolución del problema de flujo de cargas por medio de la metodología orientada a objetos donde cada elemento del sistema eléctrico representa un actor que tiene la posibilidad de interactuar con los demás en lo que se denomina una “sala de juegos”, la que organiza el intercambio de información y resuelve el problema en variables complejas mediante un resolutor complejo que hace uso del método de Newton-Raphson modificado. Se considero en esta parte también, el tratamiento que se le hace a los transformadores con TAP's variables de los transformadores.

ii. Incorporación de los TAP's variables de los transformadores a la formulación del problema.

Esta parte aborda, por un lado, la descripción del problema del flujo de carga y cómo queda la matriz jacobiana para ingresarla a un resolutor de variables complejas no lineal, que implementa el método de Newton-Raphson para un sistema de ecuaciones complejas. Por otro lado, se describe la forma en que son entradas o inscriptas en el resolutor complejo, las funciones de flujo de cargas en cada nodo del sistema y sus derivadas.

Se debe resaltar que este esfuerzo es sustancialmente nuevo respecto a la bibliografía existente, en cuanto el problema se re-enfoca como un sistema de ecuaciones de variable compleja para todas las variables involucradas, y por tanto las funciones y sus derivadas se deben expresar como funciones complejas en sus componentes (ya sean cartesianas o polares). Por otra parte, este enfoque es necesario porque la implementación computacional fue diseñada para requerir como input las ecuaciones que se deducen en este apartado.

En este problema se considera que la red consiste en n nodos o barras. El sistema de ecuaciones que se debe satisfacer (ec. 1) consiste en n ecuaciones, una por cada nodo, de valores complejos a valores complejos y que refleja el balance de las as potencias aparentes extraídas e inyectas de los nodos.

$$\tilde{f}_i: \tilde{V}_i \tilde{I}^* - \tilde{S}_i = 0, \quad i=1, \dots, n \quad (\text{ec. 1}).$$

En la implementación se diferencian tres tipos de nodos: los nodos tipo (k) o *genérico*, los nodos tipo *regulado* (h) y los

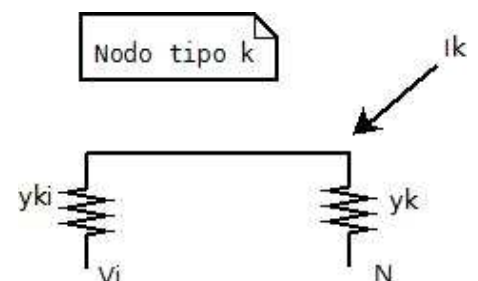


Fig. 1: Representación de un nodo genérico del sistema.

nodos tipo *regulador* (m). En las figuras 1 y 2 se muestran diagramas esquemáticos de los tipos de nodos. En caso de transformadores con tabs de regulación, se tendrá que la salida del trafo vincula un nodo del tipo (h) con un nodo del tipo (m) como se muestra en la fig. 2.

A continuación se deducen las ecuaciones de corriente en cada uno de los tipos de nodo, que surgen de aplicar las ecuaciones de nodos y mallas y eventualmente las ecuaciones del transformador ideal.

Observar que si bien se presenta la matriz como si sólo existiera un trafo en toda la red, que se hace por simplicidad, no conlleva pérdida de generalidad cuando se tiene más de uno, ya que quedará evidente que se necesitan modificar tan sólo dos líneas de dicha matriz Jacobiana, correspondientes a los nodos extremos del transformador.

a) Nodos tipo k (o genéricos)

$$\begin{aligned}\bar{I}_k &= \bar{V}_k \bar{y}_k + \sum_{i=1, \dots, n, i \neq k} (\bar{V}_k - \bar{V}_i) \bar{y}_{ki} \\ &= \bar{V}_k \left(\bar{y}_k + \sum_{i=1, \dots, n, i \neq k} \bar{y}_{ki} \right) + \sum_{i=1, \dots, n, i \neq k} \bar{V}_i (-\bar{y}_{ki}) \\ &= \bar{V}_k \bar{Y}_{kk} + \sum_{i=1, \dots, n, i \neq k} \bar{V}_i \bar{Y}_{ki} \\ &= \sum_{i=1, \dots, n} \bar{V}_i \bar{Y}_{ki}\end{aligned}$$

Donde se definió convenientemente:

$$\begin{aligned}\bar{Y}_{kk} &= \bar{y}_k + \sum_{i=1, \dots, n, i \neq k} \bar{y}_{ki} \\ \bar{Y}_{ki} &= -\bar{y}_{ki}\end{aligned}$$

b) Nodos tipo regulado (o tipo h)

$$\begin{aligned}\bar{I}_h &= \bar{V}_h \bar{y}_h + \sum_{i=1, \dots, n, i \neq h, i \neq m} (\bar{V}_h - \bar{V}_i) \bar{y}_{hi} - (\bar{V}_h - t_m \bar{V}_m) \bar{y}_{hm} \\ &= \bar{V}_h \left(\bar{y}_h + \sum_{i=1, \dots, n, i \neq h} \bar{y}_{hi} \right) + \sum_{i=1, \dots, n, i \neq h} \bar{V}_i (-\bar{y}_{hi}) + t_m \bar{V}_m (-\bar{y}_{hm}) \\ &= \sum_{i=1, \dots, n, i \neq m} \bar{V}_i \bar{Y}_{hi} + t_m \bar{V}_m \bar{Y}_{hm}\end{aligned}$$

Donde se definió convenientemente:

$$\begin{aligned}\bar{Y}_{hh} &= \bar{y}_h + \sum_{i=1, \dots, n, i \neq h} \bar{y}_{hi} \\ \bar{Y}_{hi} &= -\bar{y}_{hi} \\ \bar{Y}_{hm} &= -\bar{y}_{hm}\end{aligned}$$

c) Nodos tipo reguladores (o tipo m)

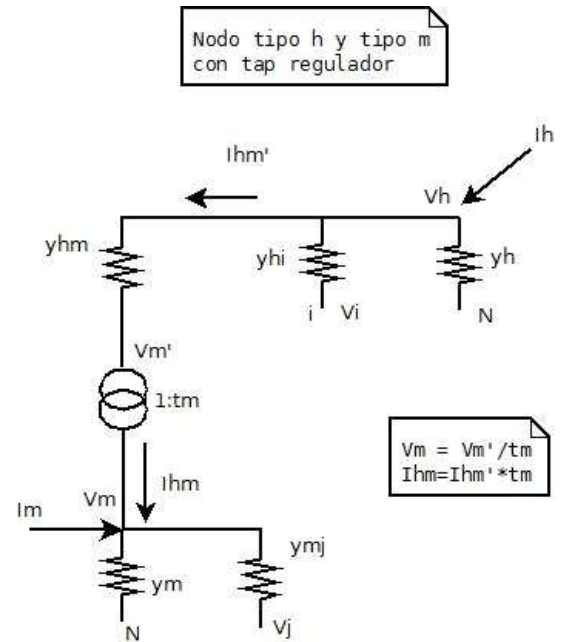


Fig. 2: Representación de nodos en presencia de un transformador con tap regulador.

$$\begin{aligned}
 \tilde{I}_m &= \tilde{V}_m \tilde{y}_m + \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} (\tilde{V}_m - \tilde{V}_i) \tilde{y}_{mi} + t_m (t_m \tilde{V}_m - \tilde{V}_h) \tilde{y}_{mh} \\
 &= \tilde{V}_m \left(\tilde{y}_m + \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} \tilde{y}_{mi} + t_m^2 \tilde{y}_{mh} \right) + \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} \tilde{V}_i (-\tilde{y}_{mi}) - \tilde{V}_h \tilde{y}_{mh} t_m \\
 &= \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} \tilde{V}_i \tilde{Y}_{mi} + \tilde{V}_m \tilde{Y}'_{mm} + t_m \tilde{V}_h \tilde{Y}_{mh} - t_m^2 \tilde{V}_m \tilde{Y}_{mh}
 \end{aligned}$$

Donde se definió convenientemente:

$$\tilde{Y}'_{mm} = \tilde{y}_m + \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} \tilde{y}_{mi}$$

$$\tilde{Y}_{mh} = -\tilde{y}_{mh}$$

$$\tilde{Y}_{mi} = -\tilde{y}_{mi}$$

Observar que:

$$\tilde{Y}'_{mm} = \tilde{Y}_{mm} - \tilde{y}_{mh} = \tilde{Y}_{mm} + \tilde{Y}_{mh}.$$

Por lo que este valor se puede obtener a partir de la matriz admitancia de barras generada de la misma forma que con los otros tipos.

d) **Matriz Jacobiana.**

Resumiendo los resultados anteriores se tiene que:

→ para nodos *genéricos* :

$$\tilde{I}_k = \sum_{i=1,\dots,n} \tilde{V}_i \tilde{Y}_{ki}$$

→ para nodos tipo *regulado* (*h*) :

$$\tilde{I}_h = \sum_{\substack{i=1,\dots,n \\ i \neq m}} \tilde{V}_i \tilde{Y}_{hi} + t_m \tilde{V}_m \tilde{Y}_{hm}$$

→ para nodos tipo *regulador* (*m*) :

$$\tilde{I}_m = \sum_{\substack{i=1,\dots,n \\ i \neq m, i \neq h}} \tilde{V}_i \tilde{Y}_{mi} + \tilde{V}_m \tilde{Y}'_{mm} + t_m \tilde{V}_h \tilde{Y}_{mh} - t_m^2 \tilde{V}_m \tilde{Y}_{mh}$$

Será conveniente expresar la matriz jacobiana del sistema de la siguiente manera:

$$\begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_m \\ \vdots \\ \tilde{f}_h \\ \vdots \\ \tilde{f}_{n-1} \\ \tilde{f}_n \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & \dots & A_{1n} & B_{11} & \dots & \dots & \dots & \dots & B_{1n} & 0 \\ A_{21} & A_{22} & \dots & \dots & A_{2n} & B_{21} & \dots & \dots & \dots & \dots & B_{2n} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{m1} & A_{m2} & \dots & C_{m,m} & \dots & C_{m,h} & \dots & A_{m,n} & B_{m1} & \dots & B_{m,n} & D_{m,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{h1} & A_{h2} & \dots & E_{h,m} & \dots & E_{h,h} & \dots & A_{h,n} & B_{h1} & \dots & B_{h,n} & D_{h,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n-1,1} & A_{n-1,2} & \dots & \dots & A_{n-1,n} & B_{n-1,1} & \dots & \dots & \dots & \dots & B_{n-1,n} & 0 \\ A_{n1} & A_{n2} & \dots & \dots & A_{nn} & B_{n1} & \dots & \dots & \dots & \dots & B_{nn} & 0 \end{pmatrix} \begin{pmatrix} d\tilde{V}_1 \\ d\tilde{V}_2 \\ \vdots \\ d\tilde{V}_m \\ \vdots \\ d\tilde{V}_h \\ \vdots \\ d\tilde{V}_{n-1} \\ d\tilde{V}_n \\ d\tilde{S}_1 \\ \vdots \\ d\tilde{S}_n \\ dt_m \end{pmatrix}$$

A continuación se especifica cada uno de los elementos de dicha matriz:

- $A_{ij} = \frac{\partial}{\partial \tilde{V}_j} \tilde{f}_i$, para nodos tipo *k* (o *genéricos*)

→ Caso $i \neq j$:

$$\frac{\partial}{\partial |\tilde{V}_j|} : \tilde{V}_i \left(e^{j \arg(\tilde{V}_j)} \tilde{Y}_{ij} \right)^* \quad (2)$$

$$\frac{\partial}{\partial \arg \tilde{V}_j} : -j \tilde{V}_i \tilde{V}_j^* \tilde{Y}_{ij}^* \quad (3)$$

→ Caso $i = j$:

$$\frac{\partial}{\partial |\tilde{V}_j|} : e^{j \arg(\tilde{V}_j)} \tilde{I}_j^* + |\tilde{V}_j| \tilde{Y}_{jj}^* \quad (4)$$

$$\frac{\partial}{\partial \arg \tilde{V}_j} : j \tilde{V}_j \tilde{I}_j^* - j \tilde{V}_j \tilde{V}_j^* \tilde{Y}_{jj}^* \quad (5)$$

- $B_{ij} = \frac{\partial}{\partial \tilde{S}_j} \tilde{f}_i$, para todos los nodos, $i = 1, \dots, n$.

$$\rightarrow \frac{\partial}{\partial P_j} \tilde{f}_i: -1 \text{ si } i = j \text{ y } 0 \text{ si } i \neq j.$$

$$\rightarrow \frac{\partial}{\partial Q_j} \tilde{f}_i: -j \text{ si } i = j \text{ y } 0 \text{ si } i \neq j.$$

$$\rightarrow \frac{\partial}{\partial |\tilde{S}_j|} \tilde{f}_i: -\frac{\tilde{S}_i}{|\tilde{S}_i|} \text{ si } i = j \text{ y } 0 \text{ si } i \neq j.$$

$$\rightarrow \frac{\partial}{\partial \arg \tilde{S}_j} \tilde{f}_i: -j \tilde{S}_j \text{ si } i = j \text{ y } 0 \text{ si } i \neq j.$$

- $C_{mj} = \frac{\partial}{\partial \tilde{V}_j} \tilde{f}_m$, para $j = 1, \dots, n$ (o reguladores)

→ Caso $j = h$: (regulado)

$$\frac{\partial}{\partial |\tilde{V}_h|} : \tilde{V}_m t_m \left(e^{j \arg(\tilde{V}_h)} \tilde{Y}_{mh} \right)^* \quad (6)$$

$$\frac{\partial}{\partial \arg \tilde{V}_h} : -j \tilde{V}_m t_m \tilde{V}_h^* \tilde{Y}_{mh}^* \quad (7)$$

→ Caso $j \neq h$, $j \neq m$:

$$\frac{\partial}{\partial |\tilde{V}_j|} : \tilde{V}_m \left(e^{j \arg(\tilde{V}_j)} \tilde{Y}_{mj} \right)^* \quad (8)$$

$$\frac{\partial}{\partial \arg \tilde{V}_j} : -j \tilde{V}_m \tilde{V}_j^* \tilde{Y}_{mj}^* \quad (9)$$

Observar que $C_{mj} = A_{mj}$ si $j = 1, \dots, n$ con $j \neq m$, $j \neq h$.

→ Caso $j = m$:

$$\frac{\partial}{\partial |\tilde{V}_m|} : e^{j \arg(\tilde{V}_m)} \sum_{\substack{l=1, \dots, n \\ l \neq m, l \neq h}} \tilde{V}_l^* \tilde{Y}_{ml}^* + 2 |\tilde{V}_m| \tilde{Y}_{mm}^* (')$$

$$+ e^{j \arg(\tilde{V}_m)} t_m \tilde{V}_h^* \tilde{Y}_{mh}^* - 2 |\tilde{V}_m| t_m^2 \tilde{Y}_{mh}^*$$

$$\frac{\partial}{\partial \arg \tilde{V}_m} : j \tilde{V}_m \sum_{\substack{l=1, \dots, n \\ l \neq m, l \neq h}} \tilde{V}_l^* \tilde{Y}_{ml}^* + j \tilde{V}_m t_m \tilde{V}_h^* \tilde{Y}_{mh}^*$$

Observar que $\sum_{\substack{l=1, \dots, n \\ l \neq m, l \neq h}} \tilde{V}_l^* \tilde{Y}_{ml}^* = \tilde{I}_m^* - \tilde{V}_m^* \tilde{Y}_{mm}^* - \tilde{V}_h^* \tilde{Y}_{mh}^*$.

- $D_{mm} = \frac{\partial}{\partial t_m} \tilde{f}_m = \tilde{V}_m \tilde{V}_h^* \tilde{Y}_{mh}^* - 2 \tilde{V}_m t_m \tilde{V}_h^* \tilde{Y}_{mh}^*$

- $E_{hj} = \frac{\partial}{\partial \tilde{V}_j} \tilde{f}_h$, para $j = 1, \dots, n$ (o regulados)

→ Caso $j = m$ ($m \neq h$): (regulador)

$$\frac{\partial}{\partial |\tilde{V}_m|} : t_m \tilde{V}_h \left(e^{j \arg(\tilde{V}_m)} \tilde{Y}_{hm} \right)^* \quad (10)$$

$$\frac{\partial}{\partial \arg \tilde{V}_m} : -j t_m \tilde{V}_h \tilde{V}_m^* \tilde{Y}_{hm}^* \quad (11)$$

→ Caso $j \neq m$, $j \neq h$:

$$\frac{\partial}{\partial |\tilde{V}_j|} : \tilde{V}_h \left(e^{j \arg(\tilde{V}_j)} \tilde{Y}_{hj} \right)^* \quad (12)$$

$$\frac{\partial}{\partial \arg \tilde{V}_j} : -j \tilde{V}_h \tilde{V}_j^* \tilde{Y}_{hj}^* \quad (13)$$

→ Caso $j = h$:

$$\frac{\partial}{\partial |\tilde{V}_h|} : e^{j\arg(\tilde{V}_h)} \sum_{\substack{l=1,\dots,n \\ l \neq m}} \tilde{V}_l^* \tilde{Y}_{hl}^* \quad (14)$$

$$+ \tilde{V}_h \left(e^{j\arg(\tilde{V}_h)} \tilde{Y}_{hh} \right)^* + t_m e^{j\arg(\tilde{V}_h)} \tilde{V}_m^* \tilde{Y}_{hm}^* \quad (15)$$

$$\frac{\partial}{\partial \arg \tilde{V}_h} : j \tilde{V}_h \sum_{\substack{l=1,\dots,n \\ l \neq m}} \tilde{V}_l^* \tilde{Y}_{hl}^* - j \tilde{V}_h \tilde{V}_h^* \tilde{Y}_{hh}^* \quad (16)$$

$$+ j t_m \tilde{V}_h \tilde{V}_m^* \tilde{Y}_{hm}^* \quad (17)$$

Observar que $\sum_{\substack{l=1,\dots,n \\ l \neq m}} \tilde{V}_l^* \tilde{Y}_{hl}^* = \tilde{I}_h^* - \tilde{V}_m^* \tilde{Y}_{hm}^*$.

$$\bullet D_{hm} = \frac{\partial}{\partial t_m} \tilde{f}_h = \tilde{V}_h \tilde{V}_m^* \tilde{Y}_{hm}^*$$

Luego del desarrollo analítico anterior se procedió a implementarlo computacionalmente. El procedimiento de inscripción de funciones complejas al sistema de ecuaciones es directo; esencialmente se debe indicar las variables que intervienen y para las derivadas con respecto a qué variable se deriva especificando si se trata de la parte radial o la parte angular.

iii. Desarrollo de los algoritmos de lectura de datos de los archivos de datos de red del PSSE.

Durante esta etapa se procedió a la implementación de módulos en Pascal para la lectura y creación de objetos de flujo de carga desde archivos tipo RAW de PSSE. Se encontró que los archivos de las versiones mas usadas son la versión 26 y 32 de este programa. Se implemento un modulo de lectura de ambos tipos de versiones.

Para discriminar los actores que están o no en servicio fue necesario desarrollar algoritmos de conectividad de red.

iv. Estudio de diferentes formas de incorporar la red al SimSEE e implementación de la solución elegida.

Hay diferentes formas de aproximarse el tratamiento del “Unit Commitment” junto con el comportamiento de la red. Se vieron varios estudios en los que la red a los efectos del voltaje se modela solo en DC, en AC y otros hacen el tratamiento mediante OPF.

Trabajar con flujo de cargas DC tiene la ventaja de representar en forma muy simplificada la red disminuyendo el tiempo de calculo, pero tiene la contra de que no se representa el comportamiento de la reactiva, por lo que si tengo elementos de compensación de reactiva o en su defecto el control de voltaje por las maquinas generadoras, no resulta un modelo adecuado a los objetivos propuestos. En cuanto a los modelos AC, son los de flujo de cargas tradicionales.

Se vio también la conveniencia de avanzar en la implementación de un OPF ya que este modelado representaría no solo la red sino que seria una forma de implementar un Resolvedor no lineal para el problema de despacho completo, introduciendo también características más complejas de los actores del SimSEE que podrían llegar a ser representadas mediante funciones no lineales.

Como forma de aproximarnos al problema consideramos tratar en un principio el flujo de cargas AC en forma separada del despacho hidrotérmico, como dos subproblemas. O sea se resuelve un paso de optimización del sistema hidrotérmico que tiene como resultado un determinado despacho de maquinas, el cual es tomado por el flujo de cargas para verificar la validez del caso en cuanto a los flujos por las canalizaciones. En caso de que el problema de red no sea factible, se itera haciendo las correcciones necesarias en el despacho hidrotérmico hasta que el sistema hidrotérmico-red tenga una solución factible.

v. Solución implementada.

En el programa SimSEE de simulación hidrotérmica se agregó una rutina que carga un caso dado de flujo de cargas en memoria para ser utilizado en cada paso de optimización.

El programa SimSEE tiene la posibilidad de definir nudos con sus correspondientes demandas y generaciones. Además se permite la posibilidad de modelar Arcos entre los nudos con ciertos parámetros para simular en forma aproximada el comportamiento no lineal de las canalizaciones. Estos parámetros son: “Rendimiento”, “Peaje” y “Potencia Máxima”. Estos parámetros imponen restricciones en el despacho hidrotérmico por lo cual su modificación le dará al SimSEE las señales económicas para modificar el despacho económico si fuera necesario.

En cada paso de optimización se cargan las demandas y las generaciones de cada maquina en el caso de flujo de cargas. Se corre el flujo de cargas, se analizan las potencias intercambiadas entre las diferentes zonas y en base a estos resultados se cambian los parámetros de los arcos de SimSEE en caso de ser necesario, procediendo a iterar nuevamente el despacho económico si dichos valores fueron alterados.



Fig. 3. Diagrama de flujos de la interacción entre el programa de coordinación hidrotérmica SimSEE y el flujo de cargas.

Para establecer la equivalencia entre los actores que intervienen en el SimSEE y los actores que intervienen en el flujo de cargas se definió que:

- Cada nodo del SimSEE tiene un equivalente en una Zona del flujo de cargas, para lo cual se agregó un parámetro en la definición de los nodos que representa la zona correspondiente.
- Cada Zona del flujo de cargas esta compuesto por una serie de cargas y generadores.
- Cada carga en una determinada Zona del flujo de cargas tiene un parámetro que representa el porcentaje de la carga con respecto a la demanda total de la Zona.
- Cada nodo del SimSEE aporta una demanda a una zona del flujo de cargas la cual es distribuida de acuerdo a los porcentajes definidos anteriormente.
- Cada Generador de SimSEE tiene su correspondiente generador en el flujo de cargas, agregando en este caso un par de parámetros para establecer dicha correspondencia.
- Cada Arco de SimSEE se corresponde de esta forma con todas las canalizaciones entre las diferentes zonas del flujo de cargas.

En base a estas consideraciones se implementaron en Pascal las rutinas necesarias para lograr estas correspondencias.

vi. Pruebas y validación de los resultados obtenidos.

Para validar la metodología implementada se armaron dos casos de prueba: un caso de flujo de cargas y un caso de SimSEE con sus correspondencias asociadas.

a) Caso de prueba del flujo de cargas.

Se creo un caso de flujo de cargas en formato PSEE (archivo *.RAW) que representa el sistema Eléctrico Uruguayo en el año 2015 con alta penetración de energía renovable (1200 MW de generación eólica mas 200 MW de generación de Biomasa).

Para armar este caso se estudiaron las diferentes zonas a ser implementadas a fin de estudiar posibles transferencias de potencia críticas entre las mismas. Luego de un estudio minucioso del Sistema eléctrico Uruguayo se identificaron 7 zonas.

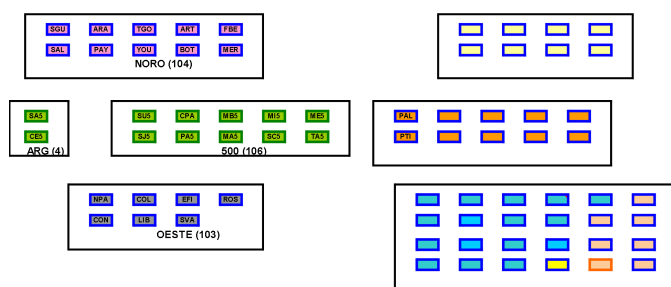


Fig. 4. Esquema de zonas en las que se dividió el Sistema Eléctrico Uruguayo a fin de armar el caso de flujo de cargas y el caso de SimSEE.

A efectos de simplificar los cálculos se hizo un equivalente del Sistema Argentino el cual se modeló con su equivalente de Thévenin en el nudo de Salto Grande Argentina.

También se tomó dicha barra como la barra Slack o flotante del flujo de cargas ya que esto permite que dicha barra no interfiera en el modelado hidrotérmico del Sistema Uruguayo.

b) Caso de prueba SimSEE.

Una vez definidas las zonas se procedió a armar un caso SimSEE con 7 nudos correspondientes a cada una de ellas.

Los archivos necesarios para ejecutar este caso de prueba se encuentran en el sitio web

http://iie.fing.edu.uy/simsee/proyecto_fse_2009_18

Capítulo II. SimSEE+Miner2012

Optimizador No Lineal aplicable a la resolución del Despacho Óptimo de Energía de un sistema Hidrotérmico.

Ruben Chaer.
Diciembre 2012
Montevideo – Uruguay.

A. Introducción.

La resolución del despacho óptimo de un sistema hidrotérmico se plantea generalmente como un problema de optimización con restricciones. Es común que dicho problema sea planteado como un Problema Lineal, simplificando para ello el conjunto de ecuaciones que intervienen para lograr expresarlas en forma aproximada por un conjunto de restricciones lineales y una función de costo lineal.

Las variables de optimización suelen ser las potencias despachadas por las diferentes unidades de generación y las principales restricciones son las conocidas como “restricciones de nodo” que imponen el equilibrio de potencias inyectadas en un nodo. En cada nodo del sistema, la suma de potencias inyectadas debe ser CERO. Como función de costo se utiliza la suma de los costos de combustible en las unidades de generación térmicas, más el costo de las importaciones menos los ingresos por eventuales exportaciones más el costo de “falla” (o racionamiento) atribuido a eventuales demandas insatisfechas por falta de generación. Suponiendo que los costos de producción de las unidades generadoras son proporcionales a la energía generada, el costo se plantea como una combinación lineal de las variables de optimización (las potencias). En la práctica, las unidades generadoras no tienen rendimiento constante por lo que aparecen así ya las primeras no-linealidades simplificadas al suponer rendimientos constantes. En la plataforma SimSEE se pueden utilizar modelos de centrales térmicas con un rendimiento no constante y con mínimo técnico de producción (esto es una potencia mínima para estar en servicio), y el costo todavía se puede expresar como una dependencia lineal con la potencia producida.

Otras fuentes de no-linealidad son la dependencia del rendimiento de las centrales hidráulicas con el caudal erogado (por pérdida de altura efectiva ocasionada por el caudal erogado) y la dependencia cuadrática de las pérdidas de las líneas de transmisión con la potencia transmitida.

En el caso del coeficiente variable de las hidráulicas, en el planteo lineal, se realiza una estimación del caudal erogado, antes de resolver el problema (por ejemplo considerando como primer estimación el caudal erogado en el paso anterior) y se resuelve el problema con esa estimación. Si luego de la resolución, el caudal erogado resultante difiere mucho del estimado se repite la resolución modificando la estimación original. Este método es el utilizado en SimSEE para resolver una sucesión de problemas lineales que representen la no-linealidad.

En este trabajo se documenta el desarrollo de un optimizador no-lineal para ser utilizado en lugar del método Simplex clásico y poder representar así las no linealidades en SimSEE.

Además de poder representar no-linealidades, el planteo de otra forma de resolución del problema de optimización que no implique la manipulación del problema (esto es que no implique la operación sobre las ecuaciones del problema) representa un atractivo importante dado que los problemas que se analizan son representables por ecuaciones “ralas” (sparsas) que involucran pocas variables, pero que por la cantidad de elementos del sistema pueden contener muchas variables y muchas restricciones. Si la solución se basa en la manipulación de las ecuaciones (como

es el caso del método Simplex) la estructura “rala” (sparsa) se pierde y el tiempo de cálculo se ve comprometido. Por el contrario, un método como la búsqueda por dirección no implica la manipulación de las ecuaciones, sólo su evaluación y mientras que se mantenga la estructura rala (sparsa) la evaluación se mantiene reducida en tiempo de cómputo.

B. Nociones básicas de optimización.

En forma genérica un problema de optimización se puede plantear como:

$$\begin{cases} \min_X f(X) \\ @ \quad X \in \Omega \end{cases} \quad (\text{ec. 2})$$

Siendo X el vector de “variables de optimización” y Ω el conjunto al que debe pertenecer el vector de variables para obtener una solución “factible” del problema. Al conjunto Ω se le llama “conjunto factible” o “dominio” del problema. La función a minimizar $f(X)$ es denominada “función objetivo” o “función de costo” y la supondremos definida de Ω en el campo de los reales R . El problema planteado por la ec. 2 se debe leer como “encontrar el valor del vector X que minimiza $f(X)$ sujeto a que X pertenezca al conjunto Ω ”.

En la formulación del problema se suele expresar al conjunto Ω por un conjunto de restricciones del tipo $r_i(X) \leq 0$ (restricciones de desigualdad) y del tipo $h_i(X) = 0$ (restricciones de igualdad) que determinan cada una un conjunto siendo Ω determinado por la intersección de todos los conjuntos fijados por las restricciones. Con esta notación el planteo del problema se escribe como:

$$\begin{cases} \min_X f(X) \\ @ \quad \begin{aligned} &h_j(X) = 0; j = 1 \dots n_h \\ &r_i(X); i = 1 \dots n_r \leq 0 \end{aligned} \end{cases} \quad (\text{ec. 3})$$

i. Supremo de un problema.

Se conoce como “supremo de $f(X)$ sobre Ω ” o simplemente “supremo del problema” al conjunto de los puntos (X, z) tal que $X \in \Omega$ y $z \geq f(X)$.

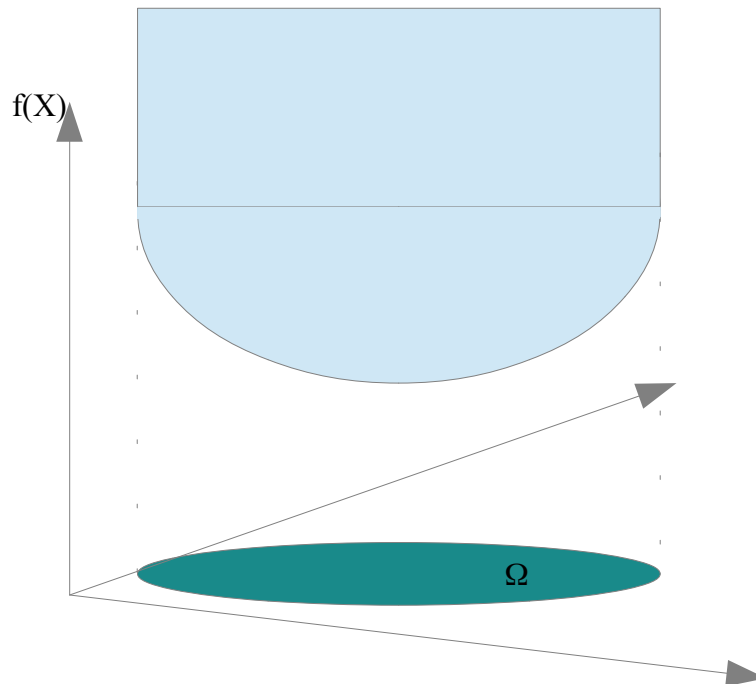


Fig. 3: Supremo de un problema.

Observar que toda la información del problema queda contenida en su supremo. Dar el supremo equivale a dar la especificación completa del problema.

ii. Convexidad y búsqueda local.

La fig. 3 muestra un ejemplo de un problema con un dominio en R^2 con su supremo. La importancia de introducir el concepto de supremo, es que el mismo permite tener una visión geométrica del problema de optimización y facilita el entendimiento. Dado el supremo, encontrar la solución implica encontrar el punto más bajo de ese “cuerpo”. Si el supremo es un conjunto convexo (es decir que dados cualquier par de puntos, el segmento que los une pertenece al conjunto), por definición, cualquier punto del supremo (en particular los de la superficie limitada por $f(X)$) tiene “visión directa” con el mínimo. Es decir que si hacemos la ficción de que estamos parados en un punto cualquiera del supremo, existe un rayo que une dicho punto con el mínimo. Esto habilita a afirmar, que cualquier algoritmo de búsqueda local del mínimo (esto es búsqueda en un entorno de un punto dado) conduce hacia la solución del problema. Esta es la situación de la Fig. 4. Por el contrario, si el supremo no es convexo se pueden dar situaciones donde una búsqueda local lleve a “falsas soluciones” como se puede ver en la Fig. 5 en que la no convexidad está dada por una “lomada” en $f(X)$ y en la Fig. 6 en que la no convexidad está dada por el carácter no-convexo de Ω

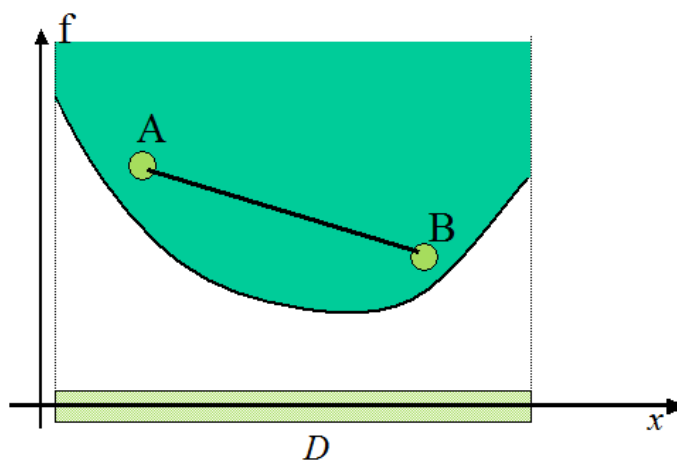


Fig. 4: Supremo convexo.

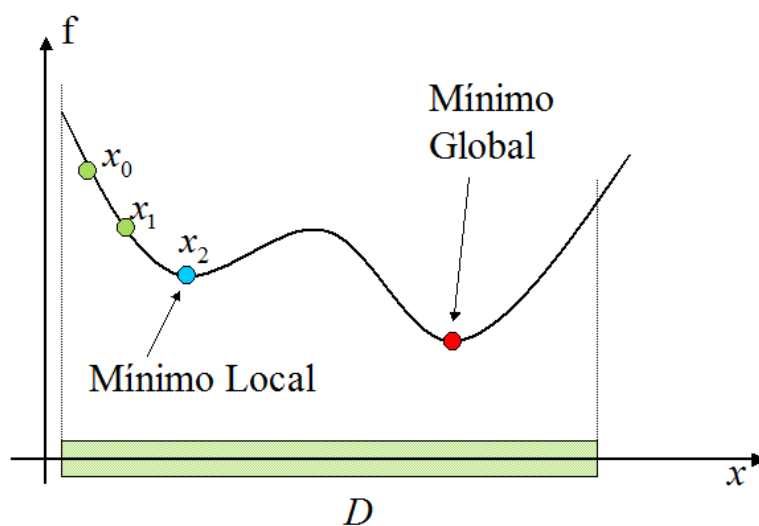


Fig. 5: Función no convexa.

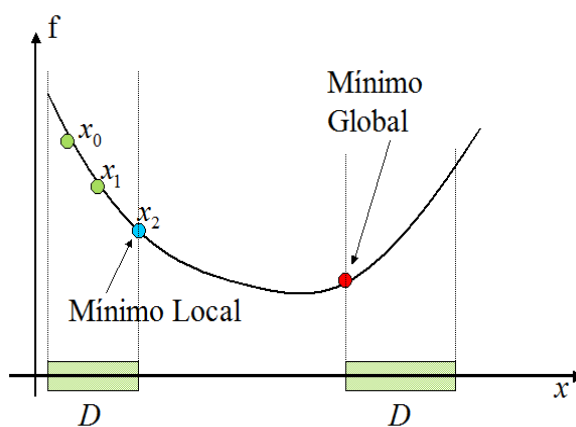


Fig. 6: Dominio no convexo.

En las aplicaciones sobre la plataforma SimSEE se supondrá o que los problemas son convexos o en el caso de no serlos, se conoce la forma de describir el problema como una colección de sub-problemas convexos. Cuando es posible construir dicha colección, para obtener la solución del problema es necesario solucionar la colección de problemas para poder seleccionar el menor de los óptimos de la colección.

iii. Relajación de un problema.

La definición del supremo de un problema facilita la interpretación del concepto de “relajación de un problema”. Dado dos problemas “A” y “B” con supremos A y B diremos que el problema B es una relajación del problema A si $A \subset B$. La Fig. 7 representa gráficamente la situación para un ejemplo y permitirá mostrar intuitivamente la relación entre la solución de un problema A y un problema B que sea una relajación de A.

En la figura, la solución al problema B es el punto “celeste”, como se puede apreciar, al ser una solución del problema relajado, es inferior a la solución del problema A y por lo tanto es una cota inferior de la solución de A. Por otro lado, si como es el caso de la figura, la solución del problema relajado está dentro del dominio del problema original (o no relajado) entonces podemos calcular la función objetivo del problema no relajado y necesariamente ese valor es una cota superior del mínimo buscado del problema no relajado. En la figura, el punto marcado con Amarillo, es la cota superior obtenida entonces al resolver el problema B.

Podemos decir entonces, que el valor de la función objetivo del problema A en el óptimo (punto Rojo de la figura) está entre el Celeste (mínimo del problema B) y el Amarillo (valor de la función objetivo de A evaluada para el mismo punto que el punto Celeste).

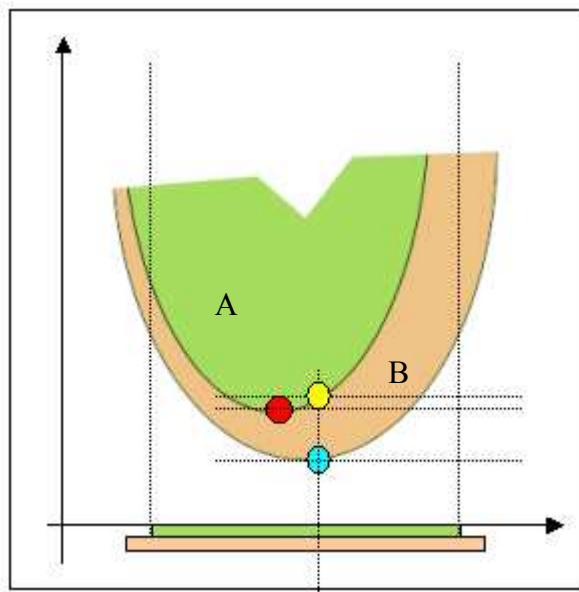


Fig. 7: Relajación de un problema.

Las técnicas de relajación se utilizan como herramienta para encontrar la solución al problema no-relajado. La estrategia es “ablandar” (o relajar) el problema A de forma que resulte más sencillo de solucionar y luego ir ajustando la relajación para obtener una solución cada vez más cercana a la del problema A. Por esta razón, al problema no-relajado se le denomina “problema original”. Dado que la resolución de un problema relajado de otro nos permite encontrar seguro una cota inferior y en caso de pertenecer la solución al dominio del problema no relajado una superior, la relajación consiste en crear una sucesión de relajaciones, donde cada problema es más ajustado que el anterior tendiendo hacia el problema original. De esa forma se van obteniendo una sucesión de aproximaciones del resultado del problema original. Si cada problema es una “desrelajación” del anterior, la nueva solución es mayor o igual que la anterior (pues el problema anterior es una relajación del actual). Entonces, de la sucesión de problemas se va obteniendo la cota inferior y en caso de ser factible la solución en el dominio del problema original una cota superior.

Es fácil concluir, que como $A \subset B$, seleccionada una dirección cualquiera del espacio, el punto más lejano de B es esa dirección siempre será más lejano que el punto más lejano de A en esa misma dirección. En particular, si se considera la dirección sobre la que está definida $f(X)$ en

el sentido de $-\infty$ se tendrá que la solución del problema relajado es necesariamente menor o igual que la solución del problema no-relajado.

Formalmente, podemos escribir que si $X^B \in B$ es la solución de B y $X^A \in A$ es la solución de A entonces, se tiene que $f(X^A) \geq f(X^B)$. Entonces una vez resuelto el problema relajado, se tendrá una cota inferior de la solución del problema original.

Al resolver el problema B se pueden dar dos situaciones.

- 1) La solución de B corresponde a un punto factible de A . Sea $X^B \in B$ la solución de B si además $X^B \in A$ es posible evaluar la función objetivo del problema A y por lo tanto ubicar el punto $(X^B, f_A(X^B))$ en B (ver la figura). Y como por definición la solución de A es el mínimo de $f_A(X)$ sujeto a $X \in A$ se tiene que $f_A(X^B)$ es una cota superior de la solución de A.
- 2) La solución de B no corresponde a un punto factible de A . En este caso, no es posible calcular $f_A(X^B)$ dado que $X^B \notin A$ y por lo tanto no es posible calcular una cota superior del problema original mediante la evaluación de la solución del problema relajado en la función objetivo del problema original como se hizo en 1).

iv. Optimización sin restricciones.

Cuando el conjunto Ω abarca todo el espacio al que pertenece, decimos que el problema es optimización “sin restricciones”. En este caso, al no tener fronteras, la solución al problema (ec. 2) deberá cumplir con que el gradiente de la función objetivo de anule en la solución. De no anularse, siempre es posible encontrar en el entorno del punto propuesto como solución otro punto en el que la función objetivo tenga menor valor. Una condición necesaria entonces para obtener un punto X solución, es que $\nabla_x f(X) = 0$.

El gradiente de la función fija la dirección de máxima variación de la función. En la dirección del gradiente, la función tiene su máximo crecimiento y en el sentido opuesto al gradiente tiene su máximo decrecimiento. A la dirección opuesta al gradiente es a la que se conoce como dirección de máximo descenso y es utilizada en la mayoría de los algoritmos clásicos de optimización para buscar un punto X_s a partir de un X dado suponiendo que en el punto X dado se conoce

el gradiente $\nabla_x f(X)$ haciendo $X_s = X + \lambda \cdot \vec{v}$ siendo $\vec{v} = -\frac{\nabla_x f(X)}{\|\nabla_x f(X)\|}$ la dirección de

máximo descenso y λ el largo de paso de búsqueda. Para un paso suficientemente pequeño, necesariamente se encontrará un valor para el cual $f(X_s) < f(X)$.

Este método de búsqueda en el entorno de un punto (esto es lo que se conoce como búsqueda local) en una dirección y con un paso ajustable tiene diferentes variantes. Se puede aplicar utilizando como dirección la de máximo descenso (como se describió en el párrafo anterior) o otras direcciones que tenga proyección sobre la de máximo descenso. También es posible buscar en la dirección de máximo descenso hasta encontrar el mejor punto en esa dirección (esto es probando con diferentes valores del paso) o encontrado un mejor punto X_s volver a calcular el gradiente y la dirección de máximo descenso en ese nuevo punto y buscar a partir de ahí una mejora. Dependiendo de cuán costoso sea en tiempo de cálculo la estimación del gradiente se da preferencia a los métodos que calculan más o menos veces el gradiente.

Cualquiera de las variantes de búsqueda local conduce a un mínimo local. Si el problema es convexo, ese mínimo es la solución del problema. Si el problema no es convexo, podrá haber otras

zonas “no exploradas” del espacio en que exista un mínimo local a la zona mejor que el encontrado.

v. Optimización con restricciones – Relajaciones paramétricas.

El problema

$$\begin{cases} \min_X f(X) \\ h_j(X)=0; j=1 \dots n_h \\ @ \\ r_i(X) \leq 0; i=1 \dots n_r \end{cases} \quad (\text{ec. 4) Problema Original.}$$

Los algoritmos para resolución de optimización con restricciones, en forma explícita o implícita (métodos de Lagrange o de Barrera) enfocan la resolución del problema mediante la creación de un problema relajado del problema original en el que las restricciones son eliminadas. La forma más usual de lleva cabo esta formulación es mediante la adición de las restricciones a la función objetivo con un juego de parámetros

$$\min_X \{ f(X) + \sum_{j=1}^{j=n_h} \beta_j \Phi(h_j(X)) + \sum_{i=1}^{i=n_r} \lambda_i \Psi(r_i(X)) \} \quad (\text{ec. 5) Relajación Paramétrica}$$

Siendo $\Phi(\xi): R \rightarrow R$ tal que $\Phi(\xi) > 0 \forall \xi \neq 0$ y $\Phi(0) = 0$

y $\Psi(\xi): R \rightarrow R$ tal que $\Psi(\xi) > 0 \forall \xi > 0$ y $\Psi(\xi) = 0 \forall \xi \leq 0$

Observar que en el dominio del problema original se cumple que $h_j(X) = 0$ con lo cual se anula la primera de las sumatorias de la (ec. 5) y $r_i(X) \leq 0$ con lo cual se anula la segunda de las sumatorias de la (ec. 5) con lo cual el supremo del problema (ec. 5) coincide sobre el dominio original con el supremo del problema original y por lo tanto lo contiene.

Dado un juego de parámetros y β_j y λ_i el problema (ec. 5) se puede resolver como un problema de optimización sin restricciones. El valor solución será entonces una cota inferior del problema original. Si además el valor de X cumple con todas las restricciones del problema original (es decir que es un punto factible) el valor encontrado es también una cota superior del problema original. Es obvio que al ser el valor cota inferior y superior es entonces la solución.

En la práctica el desafío está en cómo lograr el juego de parámetros β_j y λ_i lo suficientemente “blandos” al principio como para guiar obtener una resolución del problema relajado y la secuencia de “endurecimiento” del problema que permita ir ajustando la solución relajada a la del problema original. Los diferentes algoritmos utilizan diferentes estrategias y por lo tanto son más o menos eficaces según el tipo de problema. Es decir que los algoritmos iteran sobre el juego de parámetros de la relajación para ir “ajustando” la misma al problema original. Por simple observación del (ec. 5), si los parámetros de la relajación son positivos, cuanto mayor sean, mayor será el valor de la función (ec. 5) fuera del dominio del problema original. Las funciones $\Phi(\xi)$ y $\Psi(\xi)$ actúan como “barreras” (o paredes) que en la medida que los parámetros β_j y λ_i se hacen crecer (suponiendo que son positivos) van siendo cada vez más “verticales” y forzando entonces a la solución del problema (ec. 5) hacia la zona factible.

En el límite si $\beta_j \rightarrow +\infty$ y $\lambda_i \rightarrow +\infty$ la relajación paramétrica (ec. 5) es equivalente al problema original (ec. 4) y por supuesto igualmente difícil de resolver.

Las funciones $\Phi(\xi)$ y $\Psi(\xi)$ juegan un papel importante en cuanto a la rapidez de convergencia inicial y en cuanto al ajuste final al óptimo. En particular, si se utilizan funciones cuya derivada entorno a cero se anulen, la barrera será “débil” al acercarnos a la frontera de la zona factible, facilitando la convergencia inicial por no imponer discontinuidades en los gradientes pero dificultando el ajuste al óptimo en las iteraciones del juego de parámetros de la relajación.

Ejercicio:

La (ec. 6) muestra un ejemplo sencillo en que la función a minimizar es un paraboloide definido de R^2 en R con mínimo en $(x, y) = (5.0, 5.0)$.

La zona factible es el círculo de radio 4 y centro $(x, y) = (0.0, 0.0)$

$$\begin{cases} \min_{(x, y)} (x-5)^2 + (y-5)^2 \\ @ \quad x^2 + y^2 - 16 = 0 \end{cases} \quad (\text{ec. 6}) \text{ Ej. Problema original.}$$

Una relación paramétrica del problema puede escribirse como:

$$\min_{(x, y)} (x-5)^2 + (y-5)^2 + \lambda (x^2 + y^2 - 16)^2 \quad (\text{ec. 7}) \text{ Ej. Relajación paramétrica.}$$

Siendo la función $\Phi(\xi)$ de la formulación de la (ec. 5) seleccionada como: $\Phi(\xi) = \xi^2$

Para la resolución del problema (ec. 7) en función del parámetro λ aplicando el algoritmo de máximo descenso con “line search”, se parte de un punto inicial (x, y) , se calcula la dirección de máximo descenso y se busca en esa dirección el mínimo. Encontrado el mínimo en esa dirección se procede a considerarlo como nuevo punto inicial y se repite el procedimiento.

$$\vec{g} = \begin{bmatrix} 2(x-5) + \lambda 2(x^2 + y^2 - 16) 2x \\ 2(y-5) + \lambda 2(x^2 + y^2 - 16) 2y \end{bmatrix}$$

La dirección de máximo descenso se calcula entonces como:

$$\vec{v} = -\frac{\vec{g}}{\|\vec{g}\|}$$

En la búsqueda lineal en la dirección $\vec{v}$ se continuará descendiendo mientras que los puntos (x', y') sean una mejora en cuanto al cálculo de la función objetivo del problema relajado. El punto óptimo sobre la dirección de búsqueda $\vec{v}$ necesariamente cumple que el gradiente de la función objetivo calculada en ese punto es ortogonal a $\vec{v}$.

$$0 = (\vec{g} \cdot \vec{g}') = ((x-5) + \lambda(x^2 + y^2 - 16) 2x)((x'-5) + \lambda(x'^2 + y'^2 - 16) 2x') \\ ((y-5) + \lambda(x^2 + y^2 - 16) 2y)((y'-5) + \lambda(x'^2 + y'^2 - 16) 2y')$$

vi. MINER2012 – Implementación en SimSEE.

El algoritmo implementado en SimSEE lo hemos llamado MINER2012 y consiste en la resolución de dos problemas anidados. Creada la relajación paramétrica como en la (ec. 5), la resolución pasa por encontrar los valores de β_j y λ_i que maximizan el valor del mínimo solución de la (ec. 5).

Formalmente se puede escribir:

$$\max_{\lambda_i, \beta_j} \left\{ \min_X \left\{ f(X) + \sum_{j=1}^{j=n_h} \beta_j \Phi(h_j(X)) + \sum_{i=1}^{i=n_r} \lambda_i \Psi(r_i(X)) \right\} \right\} \quad (\text{ec. 8}) \text{ MaxMin.}$$

Tanto la maximización como la minimización se realiza por métodos de búsqueda lineal (line search) en la dirección del gradiente de la función objetivo en el caso de la maximización y en sentido opuesto al gradiente en el caso de la minimización.

Como ejemplo de test, se utilizó el problema (ec. 6) que aunque es formalmente sencillo, tiene la complejidad que impone la restricción de igualdad en forma de círculo. Esto crea un dominio claramente no-convexo lo cual hace que sea un desafío para la mayoría de los algoritmos de optimización. Se implementó con objetivo académico, un programa con interface gráfica que ubica un punto al azar de partida. En la Fig. 8 se muestra una copia de la pantalla de una resolución del caso de test. Se han colocados marcas 1, 2, 3, 4 y 5. La marca “1” corresponde al punto inicial seleccionado al azar como punto de partida del algoritmo. La marca “2” corresponde a la solución del primer problema de minimización con el primer juego de multiplicadores β_j y λ_i . La marca “3” corresponde al resultado de la minimización con el segundo juego de multiplicadores y así, hasta la marca “5” que corresponde al óptimo del problema (ec. 6). En la figura, las flechas Rojas indican pasos exitosos durante la búsqueda lineal en la minimización y las flechas Negras muestran intentos fallidos durante el la búsqueda lineal de la minimización.

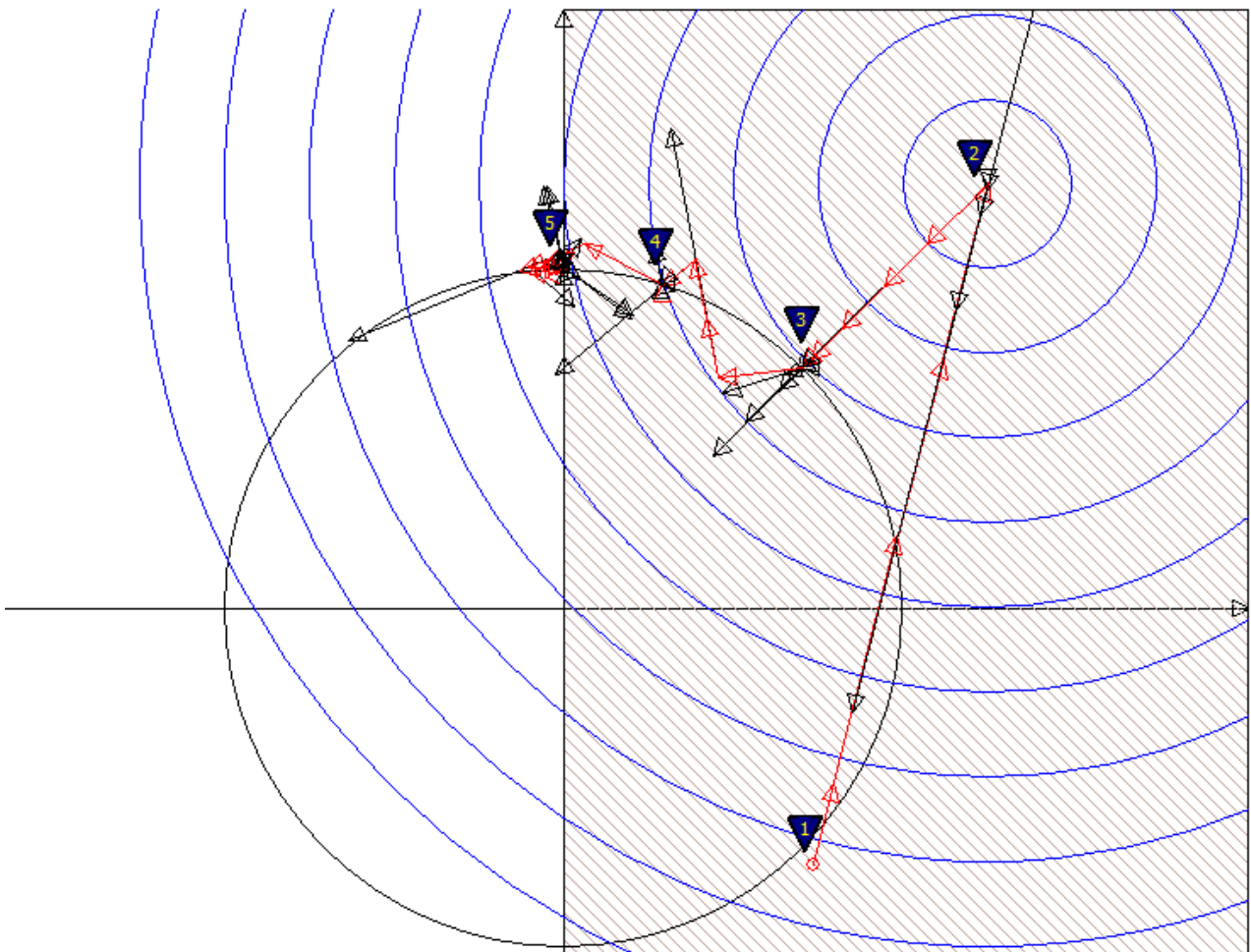


Fig. 8: Miner2012 - Caso de test.

Para el cálculo del primer juego de multiplicadores se consideran nulos los multiplicadores de las restricciones de desigualdad que no estén activas. Sobre las restricciones activas, se calculan las proyecciones del gradiente de la restricción y el gradiente de la función objetivo. Si ambos gradientes “van en el mismo sentido” (o sea su producto escalar es positivo) entonces el multiplicador se considera NULO (pues la búsqueda del mínimo de la función objetivo ya va en el mismo sentido de satisfacer la restricción). Del conjunto de restricciones activas cuyos gradientes tienen producto escalar negativo con el gradiente de la función objetivo, se busca el conjunto de multiplicadores de forma tal que la combinación lineal de los gradientes de las restricciones sumado al gradiente de la función objetivo sea lo más próxima a cero posible.

vii. Detalles de la implementación MINER2012.

En la implementación se buscó crear una librería que brindara los mismos servicios ya existentes para resolución de problemas de optimización en SimSEE. Esto es necesario, para no tener que re-escribir todos los modelos de los Actores.

Se creó una clase TProblema_01 que tiene los mismos métodos que la clase TSimplex utilizada por los actores para el armado del problema de despacho de un paso. TSimplex resuelve problemas lineales por método Simplex. TProblema_01, resuelve problemas no-lineales por el método explicado en la sección anterior.

Para facilitar la escritura de modelos, se creó también la clase Tfx que permite definir una función de $f: R^n \rightarrow R$ y su gradiente $g: R^n \rightarrow R^n$. Esta clase es útil para escribir tanto la función objetivo como las restricciones.

Para dar soporte en forma natural a los problemas lineales (y mantener compatibilidad con lo ya desarrollado) se creó la clase Tfx_lineal_x (refinamiento de Tfx).

Para facilitar el manejo de las restricciones tanto durante la definición de un problema como en los algoritmos de resolución se definen en la unidad uresfxgx algunos tipos y clases de los cuales vale la pena comentar algunos:

Estas constantes son usadas para determinar el tipo de una restricción:

```
TTipoRestriccion = (  
    TR_Libre, // no es restricción  
    TR_Igualdad, // f(x) = f_min = f_max  
    TR_Mayor, // f_min <= f(x)  
    TR_Menor, // f(x) <= f_max  
    TR_Entre); // f_min <= f(x) <= f_max
```

Estas constantes son usadas para clasificar el estado de una restricción durante la resolución.

```
TRestriccionActiva = (  
    INDEFINIDA, // no se calculó todavía  
    NINGUNA, // no hay ninguna restricción activa  
    INFERIOR, // está activa el borde inferior. f(x) = f_min  
    SUPERIOR); // está activo el borde superior. f(x) = f_max
```

Definición de la clase TResfx usada para soportar restricciones:

```
TResfx = class  
    fx: Tfx;  
    tipo: TTipoRestriccion;  
    f_min: NReal;  
    f_max: NReal;  
    r_activa: TRestriccionActiva;  
    ultimo_f: NReal; // ultimo valor calculado de f
```

```
ultimo_g: TVectR; // ultimo valor del gradiente calculado.

distancia_frontera_f: NReal;
// es abs( ultimo_f - f_umbral ) ; el f_umbral es el límite violado
// si no hay límite violado, el valor es - min( abs( ultimo_f - f_min), abs( ultimo_f -
f_max) )
// si la restricción es de igualdad siempre es >= 0 la distancia

// Multiplicador de la restricción para formar la Lagrangiana.
// Suponemos lambda positivo y aplicado sobre f(x) <= f_max
// si está activa la otra restricción, en la formación de la Lagrangiana se debe
// utilizar -lambda.
lambda: NReal;

constructor Create(tipo_: TTipoRestriccion; f_min_, f_max_: NReal;
  const fx_: Tfx; dim_x: integer);
function eval_f(const X: TVectR): NReal;
// calcula fx(X) y de paso actualiza r_activa, ultimo_f y distancia_frontera
procedure eval_g(const X: TVectR);
procedure Free;
end;
```

viii. Conclusiones.

La implementación contiene una formulación de la relajación del problema mediante el cálculo de la Lagrangiana (y su gradiente), pero es fácilmente extendible a utilizar otros criterios de formulación de relajaciones paramétricas como puede ser el de la Lagrangiana aumentada o también métodos de barreras (no necesariamente relajaciones) como los métodos de Punto Interior.

La implementación lograda es lo suficientemente clara como para servir de plataforma educativa de los diferentes métodos de optimización con restricciones.

Capítulo III. SimSEE + VaR

Differences between the operations of the generation power system of Uruguay operated minimizing the Expected Value vs. minimizing the Value at Risk of the future operating costs.

Larrosa, E. Coppes, G. Casaravilla and R. Chaer, *Senior Member IEEE*

Abstract – Hydrothermal systems optimal operation includes a step of optimizing the resources that are valued system storable. This optimization is performed traditionally by a stochastic dynamic programming in which the objective function to minimize is the expected value of future cost of operation, also known as Bellman function. While in theory, minimize the expected value of future cost of operation is "objective" in practice there are many reasons why the actual operation includes additional precautions, sometimes actually taken by operators who are the ones which have responsibility for the consequences of the operation or sometimes made based on safety considerations were not introduced in the optimization of the operation. This work shows the implementation on the platform Simulation of Electric Power Systems of stochastic dynamic programming algorithm for specifying the objective function cost reduction future with a certain probability of exceedance. This work was performed as part of the draft platform enhancements SimSEE with funding from the Energy Sector ANII. The paper presents the results of the operation to minimize the expected value of future cost and minimizing the risk value of 5% of being exceeded. Both operations are compared both costs achieved as in the qualitative aspects. The results allow evaluating the cost of being introduced by risk averse and also identify situations where there are major differences. It also discusses the impact on the marginal cost of system operation with a slogan risk averse. This value is relevant because it is the basis for calculating the Uruguayan market spot price. These scenarios correspond to the operation in 2017 with high penetration of wind energy in the system.

A. Introduction

The optimal operation of a dynamic system involves the calculation of a function $y(t)$ from knowledge of the system state $x(t)$ and its inputs $r(t)$ determine the value of the control variables $u(t)$ system leading to the optimal path.

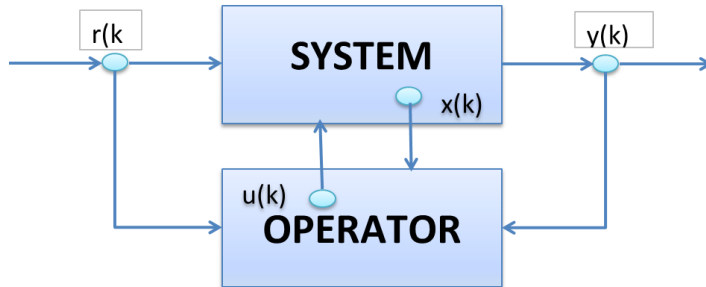


Fig.1. Block diagram of the relationship between the System and the Operator

For dynamical system we mean a system in which "the past matters." Assuming that past information is representable in a vector of state variables, system dynamics can be modeled by an evolution equation as shown in (1):

$$x' = x_{k+1} = f(x_k, u_k, r_k, k) \quad (1)$$

Where we have assumed a discretization of time and the vector x_k is the state vector at the beginning of step k by definition contains all the relevant information system of the past needed to calculate their future paths from the input series $\{u_j\}$, $\{r_j\}$, $\{j\}$ for all $j > k$.

The series $\{r_j\}$ are the inputs "uncontrolled" usually associated with stochastic processes. By way of example, the wind speed from which energy is generated in the wind farm is part of this set of entries not controlled. R_k will call the "noise vector" that "attacks" the system at stage k .

U_k parameter in (1) is the control input of the system. The system operator must select at each stage the "best" value for that vector control in the space of possible values. The series of vectors $\{u\}$ is the number of control vectors.

The Operator, considering the system state information and tickets uncontrolled fixed at all times the control vector u_k to guide the system for the optimal path. Explicitly or implicitly, to do this the operator has a policy function or operation as shown in (2):

$$u_k = PO(x_k, r_k) \quad (2)$$

Given the evolution equation of state of the system (1) Operation and Operator Policy (2), it is possible to simulate the trajectories of the system from a known initial state x_0 if you know the possible series of entries $\{r_k\}$ controlled for all $k > 0$. Since the vector r_k corresponds to the realizations of stochastic processes, from an initial state given the state evolve through different paths. In each of these possible paths is possible to calculate the cost incurred during the operation of the system and indicators of exposure experienced by the system to go through them. Whether an Operation Policy is better than another depends on what the merit function is used for comparison. The simplest and most commonly used is to minimize the expected value of future cost of operation. The aim of this work is to develop alternatives to the expected value of future cost of operation that allows including the concept of risk aversion in determining Operating Policy.

At this point, it should clarify the meaning of the word "cost" used in context "cost function" and beyond. The term "cost function" refers to the objective function minimization problem. Outside that context, the cost (e.g. Future Operating Cost System) refers to the cost, expressed in money (usually in constant currency of a given date) and coincides with the meaning that most people assigned to the word "cost". Strictly speaking, these two assignments of meanings for the word "cost" are not divergent but on the contrary, because usually the "cost function" is intended to represent the sum of all costs, both those incurred in a direct and quantifiable in cash and other less direct costs and are able to quantify and add in the cost function.

When the system is generating electricity, the optimal path is usually one that minimizes the expected value of future cost (FC) of operation. This CF is the sum of expenditure on fuel imports and costs assigned to the energy not supplied (costs of shortages or rationing) less revenue in your system from energy sales to other countries (exports).

Theoretically, when the cost function represents all the real components of the cost, minimizing the expected value of said function is the aim of optimization for excellence. In practice there are situations not always well captured in the cost function and lead to you prefer to be "more conservative" or risk averse. One of the reasons for that risk aversion is not to fall into situations which by their low probability weigh little in the expected value of the cost of operation but in the event of a disorder means you may have economic consequences rather than on system administrators or the economy. Of course you can always discuss whether these rare but catastrophic events should not be included reflecting that "catastrophic cost" in a cost function "well formed".

To fix ideas, if the system in question is the power systems of a country like Uruguay, strongly interconnected with Argentina and Brazil that are between two and 70 times larger than Uruguay. Optimizing the Operation Policy with a goal of reducing the expected value of future cost of operation could lead to at some point Uruguay sell all the energy of the lakes at a great price (compared to a prolonged drought of the neighbors who considerably raise their prices) and it was a good deal in expected value, but there is a chance (albeit very low) that it rain not in the short term in Uruguay and therefore have to make cuts power to the country's domestic demand. One could argue that if the costs allocated to power rationing (values of failure costs) reflect the actual cost to the country that the optimizer would not happen "selling water", but this statement is guilty of two errors (or large budgets) . The first of the errors is that the precision of the tools and models

used are only approximations to reality, thus determining an optimal policy via a cost reduction in expected value is a mechanism that takes accuracy over what happens in situations of very low probability and very high costs as described. This is because the models and the data available for calibration is based on what happens to most likely and unlikely are well represented "tails of probability." The second assumption is that errors or information you have on the evolution of the variables of interest (e.g. fuel costs) is perfect and what is not and again this effect while impacting on all possible paths in those in which the system goes through situations very high costs, any error in the prediction of cost values is amplified.

This paper documents an implementation on the platform on Simulation of Electric Power Systems - SimSEE for calculating the optimal operating policy of a hydro-thermal generation with a criterion of Risk Aversion.

This work is done in the context of research project ANI_FSE_18/2009 of Faculty of Engineering of Uruguay with funding from the Energy Sector of the ANII (Agencia Nacional de Investigación e Innovación) of Uruguay.

B. Stochastic dynamic programming with histogram.

This section shows how to modify the recursive algorithm of the stochastic dynamic programming so that instead of calculating the expected value of future operating cost (Bellman function), calculated in each state of the system, at every point of time, probability distribution function of the future cost of operation. This function is denoted by a vector of equi-probable samples arranged in decreasing order. This representation can be thought of as a histogram particularly where the discretization is selected such that all boxes are an example.

In (1) shows the evolution equation of the system state, where x is the vector of state variables and as such captures all necessary information from the past to calculate the evolution of the system from knowledge of x entries system from a given instant. Tickets r and u are the vectors representing the set of inputs on which we have no control and vector control or to which the operator can impose the values (complying with the restrictions in your system for it) to drive the system by the optimal path. Then (1) can then calculate the state x' at the end of the stage (or time step) k from the knowledge x and state of uncontrollable inputs r at the beginning of stage k .

In Fig 2 shows in schematic form the evolution of the system state from the position x at the beginning of stage k to state x' at the end of the stage under the influence of uncontrolled inputs r and the control inputs u .

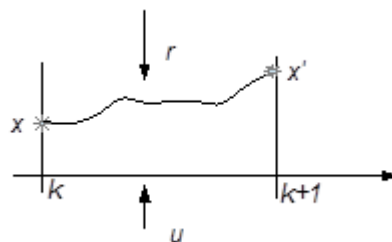


Fig. 2. Evolution of the system state.

In the kind of dynamic system under consideration we assume that the input vector while r 'is not controllable, "is known at the start of the stage and is used as input for the calculation of the control vector u better be applied to the stage during .

We assume that the cost of operating the system is an integral function of the costs incurred at each stage. For example, in the system of power generation (our purpose) consists of expenditure on fuel + imports - exports + rationing costs at each stage.

Given a set of inputs:

$$R_k = \{r_j\}; j = k, k+1, \dots$$

And a series of control:

$$U_k = \{u_j\}; j = k, k+1, \dots$$

Both from the stage k known system state at the beginning of this stage, the succession of states:

$$X_k = \{x_j\}; j = k, k+1, \dots$$

is calculable using the evolution equation (1).

If we have:

$$ce_k(x_k, u_k, r_k, k)$$

The cost of stage k. Future Cost from state x_k can be calculated as:

$$CF(x_k, R_k, U_k) = \sum_{j=k}^{j=\infty} ce_j(x_j, u_j, r_j, j)$$

Sum which can be written recursively as:

$$CF(x_k, R_k, U_k) = ce_k(x_k, u_k, r_k, k) + CF(x_{k+1}, R_{k+1}, U_{k+1})$$

If the system operator has a policy of the form of Operation (2) can eliminate the dependence of the series in the U_k previous recursive equation being:

$$CF(x_k, R_k) = ce_k(x_k, u_k, r_k, k) + CF(x_{k+1}, R_{k+1}) \quad (3)$$

We are assuming that the operator knows (x_k, r_k) to determine what the control vector u_k that used in stage k that completely ignores the future behavior of uncontrollable inputs R_{k+1} . Thus, the value of future cost in the state of arrival (at the end of step k) is a random variable. The classical method of optimization is to consider the expected value of that variable in the method we are proposing we want to have a representation of the distribution of the variable in order to minimize such things as the value given risk of absence rather than the expected value. To get a representation of:

$$CF(x_{k+1}, R_{k+1})$$

Will store a sample of the distribution function of the random variable CF samples keeping a number of equi-probable. We define a parameter that is the amount of samples stored in this representation will name: N_{mCF} (Number of samples Cost Future).

The optimization algorithm uses Monte Carlo draws in each stage to generate a set of r_k^h with

$$h = 1, 2 \dots N_{Nsop}$$

possible realizations of the uncontrollable variables. With each of these values, the operator must decide what will be the control vector u_k will use the knowledge that the system will evolve according to (1) and the cost associated with their decision will be that arising from (3). Thus, in the resolution of the step from a known state and for each value x_k and r_k^h obtain a value of costs incurred in stage

$$ce_k(x_k, u_k, r_k, k)$$

and a state value at the end of step by applying the equation of state evolution r_k^h . In turn, as the future cost in the state of arrival, we have represented by equi-probable N_{mCF} samples and depend only on the future (i.e. are conditioned only state which is reached at the end of the stage and future information not by the particular values (x_k, r_k, u_k) future cost will be the beginning of stage k combinations N_{mCF} future costs in each state of arrival may x_k^h . Thus there are, $N_{mCF} * N_{Nsop}$ equi-probable values for CF (x_k, r_k) . To keep the representation of CF (x_k, r_k) with equi-probable N_{mCF} samples, the algorithm proceeds to sample $N_{mCF} * N_{Nsop}$ them for a reduced set of N_{mCF} and to continue the recursive calculation of CF representations. In implementing the sampling switch to N_{mCF} samples from $N_{mCF} * N_{Nsop}$, is done by ordering samples roughly $N_{mCF} * N_{Nsop}$ in

decreasing order and then selecting samples N_{mCF} , N_{Nsop} apart each beginning with the first as close as possible to half the interval of the first N_{Nsop} as shown in Fig. 2:

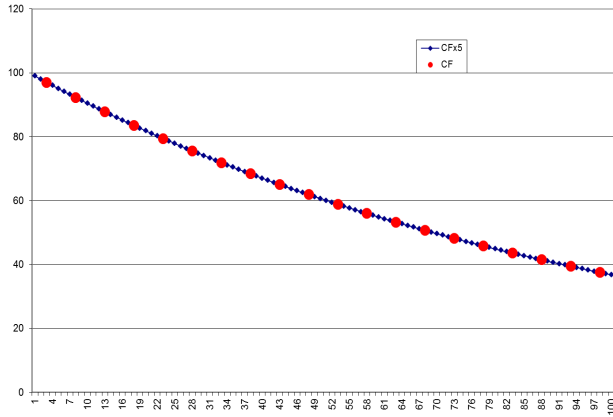


Fig. 2: Resampling from histograms of CF.

In the case of Fig. 2 the amount of points representing the histograms Future Cost is $N_{mCF} = 20$ and the number of draws of Monte Carlo optimization step is $N_{Nsop} = 5$. The figure shows for a given starting state, in blue, the equiprobable $N_{mCF} * N_{Nsop} = 100$ points achieved in the solution of step sorted in decreasing order and the red circles, the $N_{mCF} = 20$ are selected as representative of the distribution future cost CF (x_k , r_k) to continue the optimization algorithm.

C. VALUE AT RISK VaR.

To give generality to the measure of risk, introduce two measures of risk can be calculated on a random variable which is the value that is exceeded with a given probability of exceedance and the Value at Risk with a certain probability of Leave.

Given a random variable:

$$y \in D_y \subset R^1$$

probability density function $p(y)$

and fixed probability value of risk:

$$u \in [0, 1]$$

define the VeR (Value and is exceeded with probability u) and VaR (Value at Risk and with probability u) as:

$$VeR = Y \in D_y / P(y > Y) = u$$

(4)

$$VaR = \int_{y=VeR}^{y=+\infty} y \cdot p_y \cdot dy \quad (5)$$

The expected value of y is by definition:

$$VE = \int_{y=-\infty}^{y=+\infty} y \cdot p_y \cdot dy$$

Fig. 3 shows an example. Corresponds to 100 equi probable samples arranged in order of decreasing cost. Fixed as a probability of absence for the purposes of the risk values 5%, in Fig vertical line was drawn separating the 5% of the highest values of the rest. The value at risk of 5%

is exceeded $VeR = 80$ MU\$S and 5% value at risk is the average value greater than VeR and $VaR=88$ MU\$S.

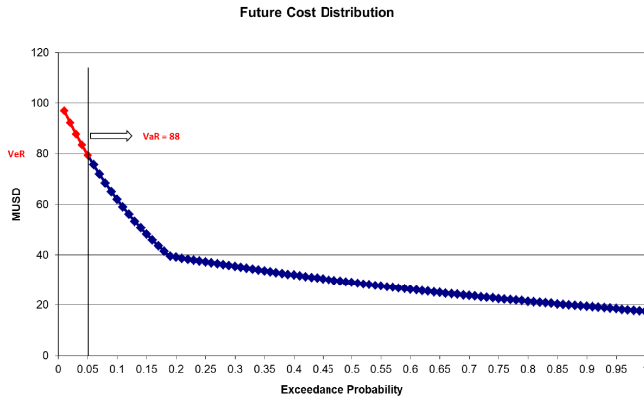


Fig. 3: Example of VeR and VaR with risk 5%.

D. risk averse objective function.

In Section III showed how to construct recursively the histograms of the future cost function. To do this, it was assumed that the system operator has a policy that allows operation to calculate the vector control as a function of state at the beginning of the stage, the realization of uncontrollable input vector and time (step k).

$$u_k = po(x_k, r_k, k)$$

Different functions $u_k = po(x_k, r_k, k)$ lead to different system operations. These operations can be classified into more or less "successful" if it has a measure of credit for the operation. The classical merit function is the expected value of costs incurred in the operation and as mentioned in the introduction the purpose of this paper is to propose an alternative merit function that allows us to take into account the risk aversion likely to have the operator.

In the end, an operator who is VERY risk averse will not look the expected value of future operating costs but will try to minimize the maximum value of that cost. Instead of minimizing the average values of the histogram (which is an estimate of the expected value) will attempt to minimize the maximum values of the histograms.

Fixed an exceedance probability for risk measurement (usually 5%) say that the operator is 100% risk-averse if the objective of optimization is to minimize the value at risk VaR and say that is 0% risk-averse when operating minimizing expected value VE .

Defining a Coefficient of Risk Aversion:

$$CAR \in [0, 1]$$

we can define the objective function of the operation with that level of risk aversion:

$$J = CAR \cdot VaR + (1 - CAR) \cdot VE \quad (6)$$

Now we show the application of this cost function with risk aversion on the stochastic dynamic programming algorithm for optimal operating policy of a dynamic system minimizing the future cost of operation with a coefficient of risk aversion given.

Having examined the state at the beginning of a time step, and a realization of uncontrolled input vector and applying the procedure described in Section III can be calculated recursively a

representation of the distribution function of the future cost for each system state. This representation allows us to evaluate the objective function (6) in the state of arrival

$$x' = f(x, r, u, k)$$

And gain control vector u that achieves minimize the cost of step

$$ce_k(x_k, u_k, r_k, k)$$

more objective point of arrival

$$J(x', k+1)$$

Observe that given a realization r_k minimize the sum

$$ce_k(x_k, u_k, r_k, k) + J(x', k+1)$$

is to minimize the cost of the equation (6).

Knowing the couple (x_k, r_k) and for each value of u_k is calculated:

$$ce_k(x_k, u_k, r_k, k)$$

and

$$x' = f(x_k, u_k, r_k, k)$$

E. case of application.

For the purpose of verifying the operation of the deployment is executed for optimizing the operating policy corner of Lake Bonete for the last week of November 2017.

This room was chosen as one SimSEE room in which there is an abundance of wind power. This room was calculated operating policy, No Risk aversion and two values of the probability of exceedance $PE = 0.05$ and 0.10 for the risk measure, we performed the same optimization $CAR = 0.00, 0.25, 0.50, 0.75$ and 1.00 .

The number of points for representation of the histograms was 400.

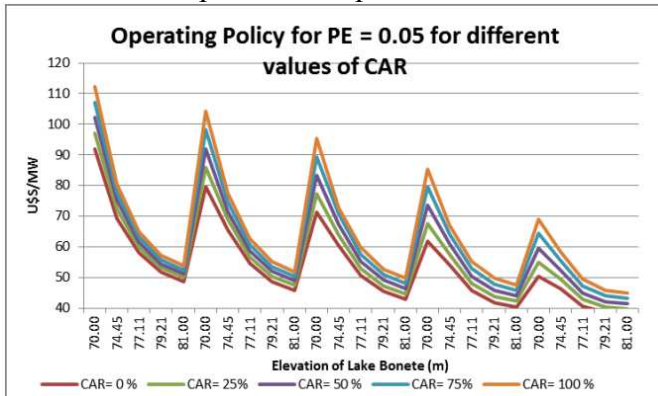


Fig. 4: Politics of operation for PE = 5% for different values of CAR

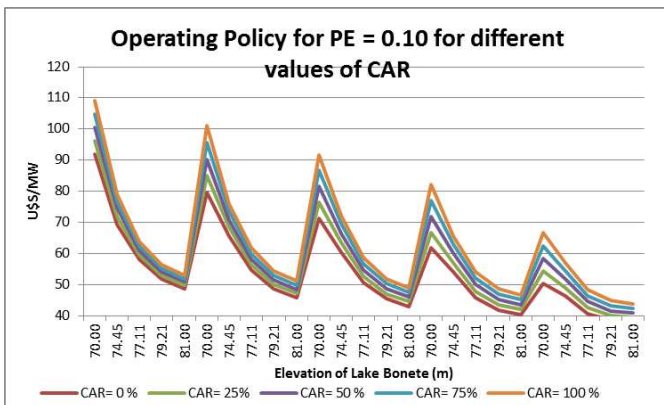


Fig. 5: Politics of operation for PE = 10% for different values of CAR

As you can see the value of the power plant's Rincon del Bonete grows inversely proportional to PE and directly proportional to the CAR as expected, since the operator to be more risk averse is expected to assume greater costs.

It also presents the average level evolves plant Rincon del Bonete for CAR = 100% for different values of PE over 2017.

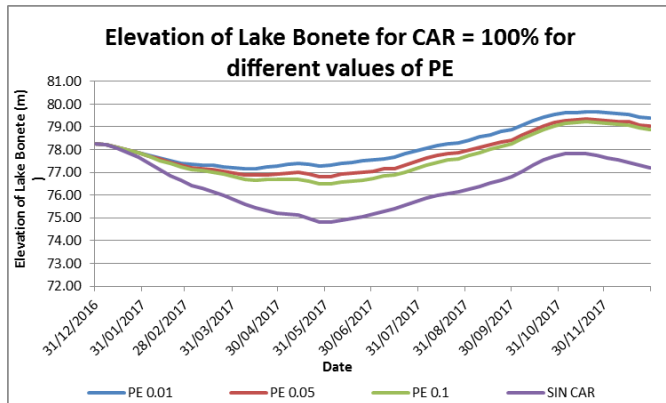


Fig. 6: Politics of operation for PE = 10% for different values of CAR

We interpret this graph that the lower PE, the higher the average level of the plant, since the operator to be more adverse to the tail of probability using other resources (usually more expensive) before using the water from this plant that has seasonal capacity.

F. conclusions.

The implementation SimSEE optimize the possibility of an electrical system being risk averse, it works correctly in all cases analyzed.

It is easy to see the increased costs associated in operating policies as well as conservative in the use of short-term lake system is the more risk averse operation becomes.

G. References

- [1] EFECTO DE LA CONSIDERACION DEL RIESGO EN LA PLANIFICACIÓN DE LA EXPANSIÓN PARA LA GENERACIÓN DE ENERGIA ELÉCTRICA, Marisa León, Nicolás Castromán, Ruben Chaer. 8º Encuentro de especialistas en Energía Potencia Instrumentación y Medidas de la IEEE-Uruguay, Montevideo-Nov.2010. <http://iee.fing.edu.uy/epim2010/myreview/myPapers/p93.pdf>
- [2] SimSEE: Simulador de Sistemas de Energía Eléctrica. Proyecto PDT 47/12, Gonzalo Casaravilla, Ruben Chaer, Pablo Alfaro. Technic Report 7, Universidad de la República (Uruguay). Facultad de Ingeniería. Instituto de Ingeniería Eléctrica, Number 7 - dec 2008. <http://www.fing.edu.uy/ie/publicaciones/2008/CCA08a/CCA08a.pdf>
- [3] Modelo de series correlacionadas CEGH. R. Chaer-Octubre 2005. http://iee.fing.edu.uy/simsee/curso2010/Cap9_sintetizadorCEGH.pdf

H. Biographies

Enzo Coppes was born in Paysandú, Uruguay in 1969. He received the Electrical Engineer degree in 2003. He is currently Grade 2 teacher from the Faculty of Engineering of Uruguay. He is currently Chief Execution of Operation on energy power company UTE in Uruguay.

Ruben Chaer was born in Tacuarembó - Uruguay, in 1962. He received the Engineering degree electrician in 1990 and Master's Degree in Electrical Engineering in 2009 from the University of the Republic of Uruguay. Currently associate professor of Institute of Electrical and presidential adviser on energy power company UTE in Uruguay.

Casaravilla Gonzalo was born in Montevideo, Uruguay, in 1963. He received the Electrical Engineer degree in 1990, the Master's Degree in Electrical Engineering in 2000 and the degree of Doctor of Electrical Engineering in 2003 from the Faculty of Engineering of Uruguay. He is currently president of the utility UTE and Associate Professor at the Institute of Electrical Engineering.

Capítulo IV. SimSEE+RIEE

SimSEE – Reducción Inteligente del Espacio de Estados en la Programación Dinámica Estocástica.

Ruben Chaer.

Montevideo – Julio 2012.

A. Introducción.

La plataforma SimSEE es un simulador de un sistema dinámico y para lograr simular la operación óptima del sistema debe llevar a cabo el cálculo de lo que se conoce como “La Política Óptima de Operación”. Dado un sistema dinámico (supondremos que se ha discretizado el tiempo) modelado por su ecuación de transición de estados como: $x_{k+1} = f(x_k, u_k, r_k, k)$ siendo x_k el vector de estado del sistema en el instante asociado al paso de discretización k y los vectores u_k y r_k las entradas de control y no controladas respectivamente al sistema en el mismo paso k .

Por definición, el estado x_k , captura toda la información del pasado del sistema que agregada con la información de las entradas a partir del instante k , nos permiten calcular la evolución del sistema (es decir los sucesivos valores del estado).

La operación óptima de sistemas dinámicos implica el cálculo de una Política de Operación. Dicha PO es una función que a partir de la información conocida en cada instante permite calcular el conjunto de valores que deben tomar las variables de control del sistema para guiar al mismo por la trayectoria óptima. La información conocida es el vector de estado del sistema x y el vector de entradas no controlables r . En el caso en que la función de costo se puede plantear como una integral sobre la trayectoria, el algoritmo clásico de cálculo de la PO es el de la Programación Dinámica Estocástica PDE [1]. Este algoritmo implica un calculo iterativo del costo futuro de operación esperado $CF(x, k)$ para cada valor del estado x y en cada etapa de tiempo k . La iteración supone conocida la función $CF(x, k)$ al inicio de la etapa de tiempo k y a partir de dicho valor calcula la función en $k-1$. Esta recursión supone en cada etapa de tiempo realizar un barrido del espacio de estados para considerar los puntos definidos por el producto cartesiano de las discretizaciones que se hayan realizado sobre cada una de las dimensiones del vector de estado x .

Si $x^T = [x_1, x_2, \dots, x_n]$ es el vector de estado, y si suponemos que las discretizaciones sobre cada dimensión son $N_1, N_2, \dots, N_n$, la cantidad de puntos de cálculo para cada etapa k será $N_1 * N_2 * \dots * N_n$. Esta representación discreta del espacio de estado sufre del inconveniente de que cada nueva variable de estado que se considere, agrega una dimensión al espacio de estados y por lo tanto una discretización del rango posible de dicha variable y multiplica la cantidad de puntos de cálculo necesario. Este aumento multiplicativo de la cantidad de puntos de cálculo lleva a que el algoritmo de la PDE sea inaplicable al tener que considerar muchas dimensiones y es lo que se conoce como la Maldición de la Dimensionalidad de Bellman (MDB) [1].

En este trabajo analizamos dos formas de luchar contra la MDB. Ambos mecanismos llevan a una representación simplificada de $CF(x, k)$ y por lo tanto a una pérdida de parte de la información contenida en las derivadas direccionales de dicha función que determinan la PO y por lo tanto se tendrá un PO que no es sub-óptima. En la lucha contra la MDB, el mecanismo que

logre reducir el esfuerzo de cómputo y alejando lo menos posible a la PO que se obtenga de la óptima será el mejor.

B. Planteo del problema – Definición de una Política de Operación – PO.

En la lucha contra la maldición de la dimensionalidad de Bellman, nos planteamos llegar a políticas de operación sub-óptimas, que utilicen una representación de la función de costo futuro menos pesada que la que surge del producto cartesiano de la discretizaciones de las diferentes variables de estado.

El algoritmo que buscamos es uno que simplifique en la medida de lo necesario para que el cálculo sea posible. Esto puede parecer “tosco” pues estamos asumiendo liza y llanamente conformarnos con un su-óptimo. En la práctica esto es lo que hacemos en forma cotidiana y es lo que hacemos también cuando resolvemos problemas con “todas las dimensiones” pues en ese “todas las dimensiones” ya hay una simplificación, nunca representamos “todas las dimensiones”. Siempre, en forma conciente o no hacemos simplificaciones y la frase “solo se que no se nada” no es otra cosa que el reconocimiento de esto.

i. La idea original

Cuando se comenzó con este trabajo de investigación se tenía la idea de, realizar un cálculo completo para un paso de tiempo k de la función $CF(x, k)$ durante la iteración del algoritmo de la PDE y en base al análisis de los resultados tomar una decisión en cuanto a no continuar explorando toda la discretización del espacio de estado en el cálculo de $CF(x, k-1)$. La idea original, era calcular la reducción en base al análisis de cuadros (frames) completos y en base al mismo análisis, calcular “puntos” para evaluar en los frames siguientes y también “puntos de control” que permitan evaluar si la reducción perdió precisión, en cuyo caso se procedería a calcular nuevamente un frame completo, una nueva reducción y puntos de cálculo y control y se proseguiría con el mismo mecanismo. Si bien NO se implementó este mecanismo en SimSEE, se considera que vale la pena explicar el mismo en base a un ejemplo.

La Fig. 9 Muestra, las “franjas iso-nivel” del resultado de la función $CF(x, k)$ para un paso k dado. Llamamos “frame k ” al conjunto de valores que toma CF sobre la discretización del vector de estado x . En este caso, el vector de estado x es de dimensión 2. En absisas está la discretización de la primera variable de estado que es la cota del lago de Rincón de Bonete (principal embalse de central hidroeléctrica ubicada en el Río Negro – Uruguay). La segunda variable de estado (discretización en ordenadas de la Fig. 9) corresponde a lo que se conoce como “La variable de estado Hidrológico” en el sistema uruguayo. Esta variable intenta captar el estado del sistema hidrológico y condiciona los aportes esperados a los tres embalses del Uruguay. Las franjas de colores, representan zonas de igual Costo Futuro esperado de operación CF. Como es de esperar, a menor agua almacenada (menor cota = lado izquierdo de las absisas) mayor es el CF. Igualmente, como un mayor valor de la variable de estado hidrológico significa mayor probabilidad de lluvias, el CF disminuye al aumentar H (ordenadas crecientes) y aumenta al disminuir H . Por simple observación de la Fig. 9, se puede apreciar que como las franjas iso-nivel son cuasi-paralelas, es posible hacer un cambio de variables de forma tal que bastaría una variable de estado para describir los valores $CF(x, k)$. La flecha verde en la figura muestra cuál debiera ser el nuevo eje (ortogonal a las iso-nivel) que mejor describe la función si se ignora la información en el otro eje.

En este ejemplo, es bastante sencillo identificar “la dirección de máxima variación” la cual es la dirección candidata a ser uno de los ejes del nuevo espacio de estados. El procedimiento para encontrar la reducción sería; Buscar la dirección de máxima variación global, elegirla como primer eje del nuevo espacio buscar la nueva dirección de máxima variación, pero en un espacio que excluye la dimensión (ortogonal a la dirección) ya seleccionada y encontrar así la segunda dirección más relevante y así sucesivamente. Luego según la cantidad de direcciones que se quiera “ignorar” simplemente se ignoran las últimas de las direcciones seleccionadas de acuerdo al procedimiento antes descrito. En el caso de la figura, la primer dirección será la de la flecha verde y la segunda una ortogonal a ella. En este caso, si se decide hacer una reducción a una sola variable de estado, la dirección a ignorar sería la ortogonal a la flecha verde. De la misma figura surgen que los puntos marcados con círculos rojos podrían ser buenos candidatos para el cálculo de la variación de CF en los frames sucesivos y fueron seleccionados de forma de cubrir todo el rectángulo y bien diferenciados en la dirección de la flecha verde. Como punto de control podría calcularse adicionalmente el punto marcado con el círculo celeste. El resultado del cálculo del punto celeste, se compararía con el valor estimado a partir de los puntos rojos para la misma coordenada sobre la dirección de la flecha verde y si la diferencia supera un umbral, se procedería a calcular un nuevo conjunto de Reducción, Puntos de Cálculo y de Control.

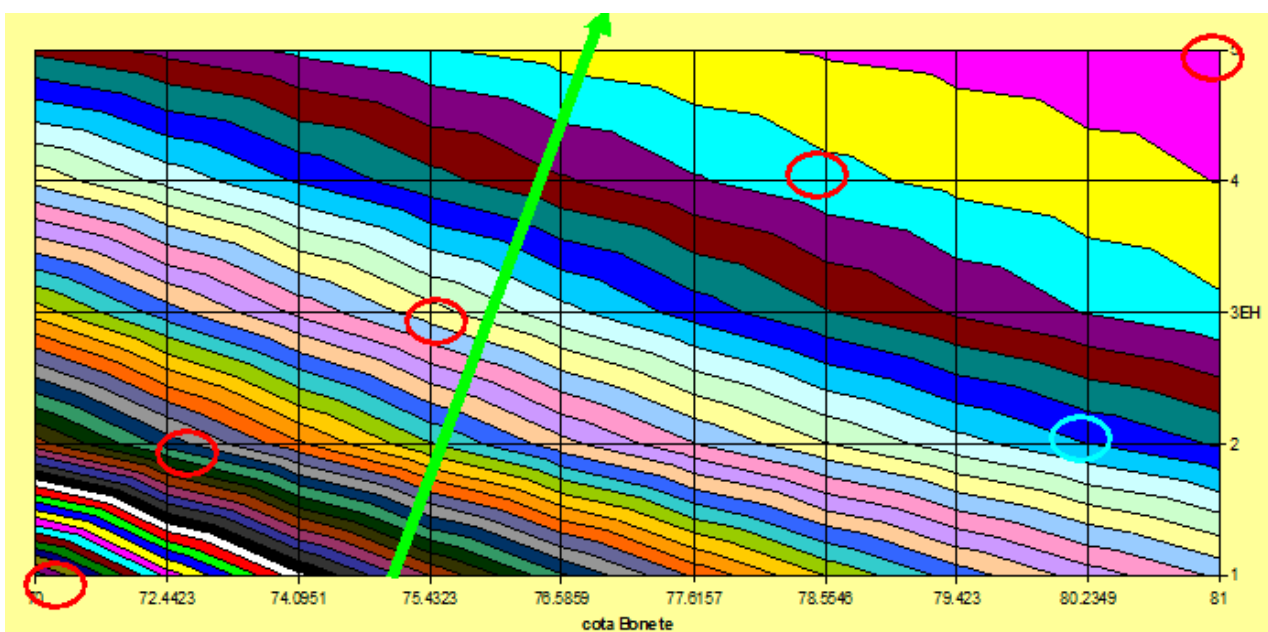


Fig. 9: Frame de $CF(x,k)$

Como ya se mencionó, no se llegó a una implementación utilizable del mecanismo de reducción de estados antes descrito en el marco de este proyecto. La razón principal es que al comenzar con la implementación surgieron elementos a analizar que no habían sido previstos como el que se describe en la siguiente sección que motivaron dedicar horas de trabajo a su investigación y en parte a agregar mejoras a la plataforma en sentidos no-previstos. No se descarta retomar este tema en futuros trabajos y lograr una implementación, pues en la práctica, para el uso de la SimSEE en tareas tales como la optimización del abastecimiento de la demanda semanal, se asumen reducciones de las variables de estados realizadas en base a la experiencia de los usuarios sin mediar de una cuantificación del error cometido por la información que se pierde en las reducciones.

C. Reducción de la dimensión de fuentes de ruido con dinámica – CEGH.

*** COMO SE MENCIONÓ, ESTE OBJETIVO SE CONSIDERA INCONCLUSO **

Las siguientes anotaciones son solo a los efectos de documentar por dónde continuar el trabajo:

Dado un proceso $x_{k+1} = A x_k + f_k B r_k + s_k$ en donde

Donde:

- $x_k \in R^n$ es el vector de estados del proceso.
- $r_k \in R^n$ Es el vector de ruidos blancos gaussianos que alimentan el filtro lineal.
- Las matrices A y B son las correspondientes al filtro lineal del modelo CEGH.
- Los valores f_k y s_k corresponden a las series de valores de atenuación del ruido y sesgo respectivamente, que pueden utilizarse para imponer un pronóstico en el espacio gaussiano en los modelos CEGH. La serie de atenuadores de ruido f_k es una serie de matrices diagonales que aplican atenuadores para cada coordenada del estado. Normalmente conocido el pronóstico, los atenuadores van de 0 (cero) en el instante inicial donde el pronóstico es casi certeza y tienen a 1 (uno) en la medida en que el pronóstico deja de dar información. Los valores s_k son vectores que suman un sesgo en cada variable de estado y que tienen a 0 (Cero) en la medida en que el pronóstico no agrega información. Los sesgos desvían la trayectoria del estado con probabilidad 50% en el espacio gaussiano.

Sea M_r una transformación “reductora” del espacio R^n en un espacio de menor dimensión R^m con $m < n$.

$$z = M_r x$$

Supondremos que M_r es unívoca, en el sentido en que dado un valor del estado $x \in R^n$ queda determinado el valor $z = M_r x$ con $x \in R^n$.

Dado que se está realizando una reducción del espacio, necesariamente habrá varios valores del espacio no reducido que mapean en el mismo punto del espacio reducido.

La transformación “expansora” podrá tener la forma:

$$x = M_a z + M_b w + M_c s$$

Dónde w denota un vector de ruidos independientes, que tienen el efecto de recomponer la aleatoriedad original de X .

Para mantener las características estadísticas de X imponemos que la matriz de covarianzas $M_c = (I - M_a M_r \text{Inv}(I - A))$ para recomponer el valor esperado del sesgo.

Atención, en el cálculo de M_b entra el “f”.

***** TRABAJO SIN TERMINAR *****

Capítulo V. SimSEE+NetTopos

Federico Barreto y Ruben Chaer.

A. Objetivo

Desarrollar una plataforma robusta y eficiente que permita la gestión de nodos de cálculo para la distribución de procesos con alto costo de cómputo. Particularmente la Optimización y la Simulación.

B. Introducción

Una forma de reducir el tiempo de cálculo es la posibilidad de distribuir el cálculo entre varias computadoras o nodos.

SimSEE cuenta con una biblioteca de comunicación entre procesos, locales o remotos, previamente implementada, además cuenta con una aplicación experimental y de uso interno del equipo de investigación para el cálculo de la Optimización de forma distribuida.

Los resultados obtenidos por esta aplicación en cuanto a la reducción del tiempo de cómputo mostraron la viabilidad de distribuir el cálculo.

Si bien la biblioteca permite interconectar nodos vía Internet las pruebas realizadas indican que para que tenga sentido la distribución con el fin de reducir el tiempo cómputo es necesaria una velocidad de comunicación importante.

Uno de los objetivos propuestos en este proyecto es mejorar la implementación disponible para que sea utilizable por el público en general integrándola a la aplicación que instalan los usuarios en sus PCs. De manera que cualquier grupo de PCs conectados en red que tengan instalados el SimSEE puedan ser usado como nodos de cálculo, uno será el servidor del cálculo y el resto en la medida que tengan activa esta facilidad podrán ser usados como clientes.

C. Problema

i. Planteamiento del problema

Para comprender la necesidad de desarrollar una plataforma para la distribución de cálculos para SimSEE es necesario mostrar la complejidad del problema que se intenta resolver.

SimSEE permite, entre otras, evaluar el costo de operación de un sistema de energía eléctrica para diferentes escenarios.

El algoritmo utilizado es recursivo, se selecciona una ventana de tiempo y se discretiza en pasos. La iteración comienza en el último paso y va retrocediendo en el tiempo paso a paso hasta alcanzar el primero. A su vez en cada paso de tiempo se discretizan las variables de estado del

sistema obteniendo así un mapeo discreto de todos los estados del sistema, para los cuales se calcula el costo de operación y se le suma el costo futuro que se incurrirá al operar el sistema.

Ejemplo: dado un sistema con 2 variables de estado cada una de ellas discretizada en n_1 y n_2 puntos respectivamente y una ventana de tiempo con k pasos, la dimensión del mapeo de las variables de estado en el tiempo es $n_1 \times n_2 \times k$, esto muestra la cantidad total de puntos a calcular.

Para sistemas complejos la cantidad de variables de estado es mayor y se necesita tener una buena resolución en la discretización del espacio de estados, esto hace que la cantidad de puntos se incremente y con ello el tiempo que insume el mapeo.

Una vez resuelta la operación óptima del sistema esta puede ser evaluada en distintos escenarios independientes para observar su comportamiento, lo que permite verificar cuan optima fue la solución obtenida y hacer ajustes sobre el modelo propuesto, etc. Por lo tanto aumentar la cantidad de escenarios donde probar la operación óptima del sistema permitirá sacar mejores conclusiones.

La implementación distribuida de la plataforma SimSEE implicará que se pueda agregar detalle al modelado del sistema lo que es fundamental para los objetivos de investigación de la misma. Además poder analizar los resultados de un mejor modelo nos permite realizar simplificaciones para la operación cotidiana del sistema que de otra forma son realizadas con aproximaciones rústicas.

ii. Formulación del problema

Para cada proceso a distribuir se debe analizar la forma de paralelizarlo, es decir que hay que identificar las partes independientes, que llamaremos tareas. Por ejemplo si el proceso fuera elevar al cuadrado las entradas de un vector de largo n , para este proceso el cálculo de cada entrada es independiente. Una solución sería partir el vector en m partes, o sea m tareas, distribuirlas entre nodos disponibles que realicen el cálculo y recolectar los resultados.

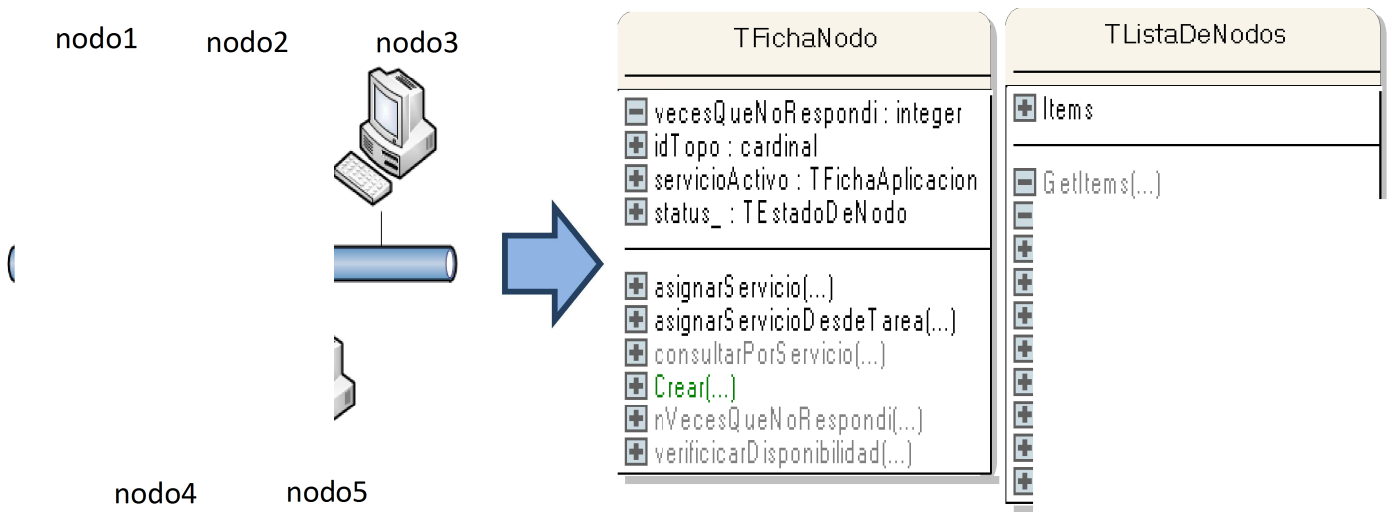
Teniendo en cuenta el ejemplo se propuso implementar una plataforma capaz de gestionar nodos conectados en una red de datos y administrar de forma eficiente la distribución de las tareas, verificando que todas sean realizadas y tomando acción en caso contrario.

iii. Resolución del problema

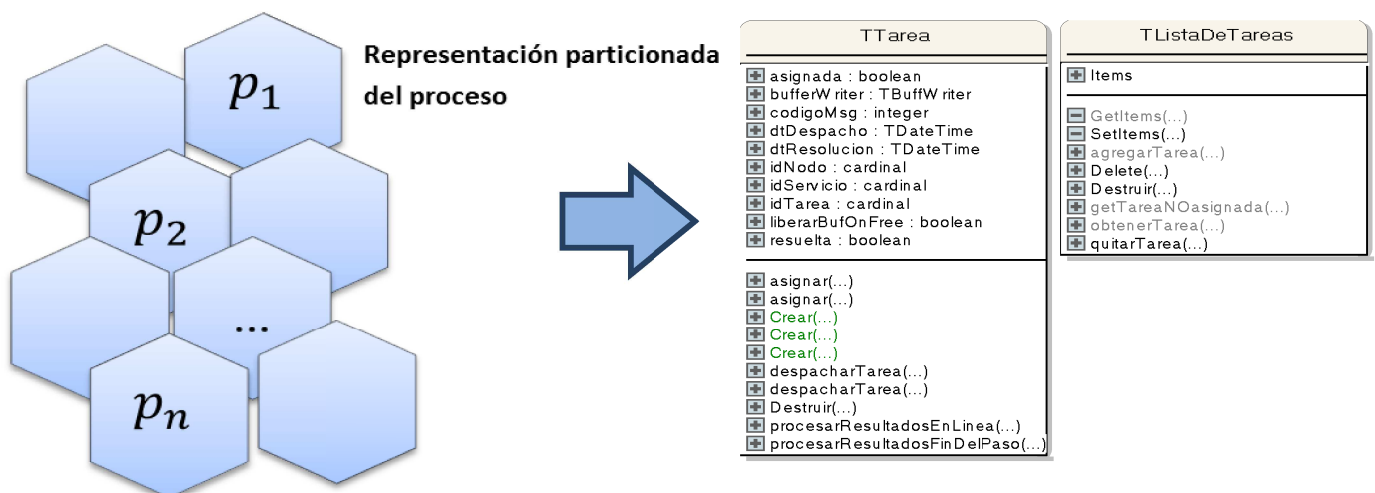
La plataforma se desarrollo manteniendo la filosofía de programación orientada por los objetos de SimSEE intentado representar cada de entidad de forma simple e incremental. Esto facilita la corrección de errores y la actualización de la plataforma, entre otros.

Se identificaron 2 entidades básicas, la red de nodos de cálculo y una lista de tareas a distribuir.

La red de nodos de cálculo es modelada por una Lista de Fichas de Nodos, en el diagrama se representa una porción de una red, como la que podemos encontrar en las oficinas habitualmente y a la derecha las clases implementadas para la representación de la red en la plataforma.



de importancia es la lista de tareas, ésta nta
 ibuir. Las distintas porciones del proces rep
 TTarea y ordenadas en una lista.

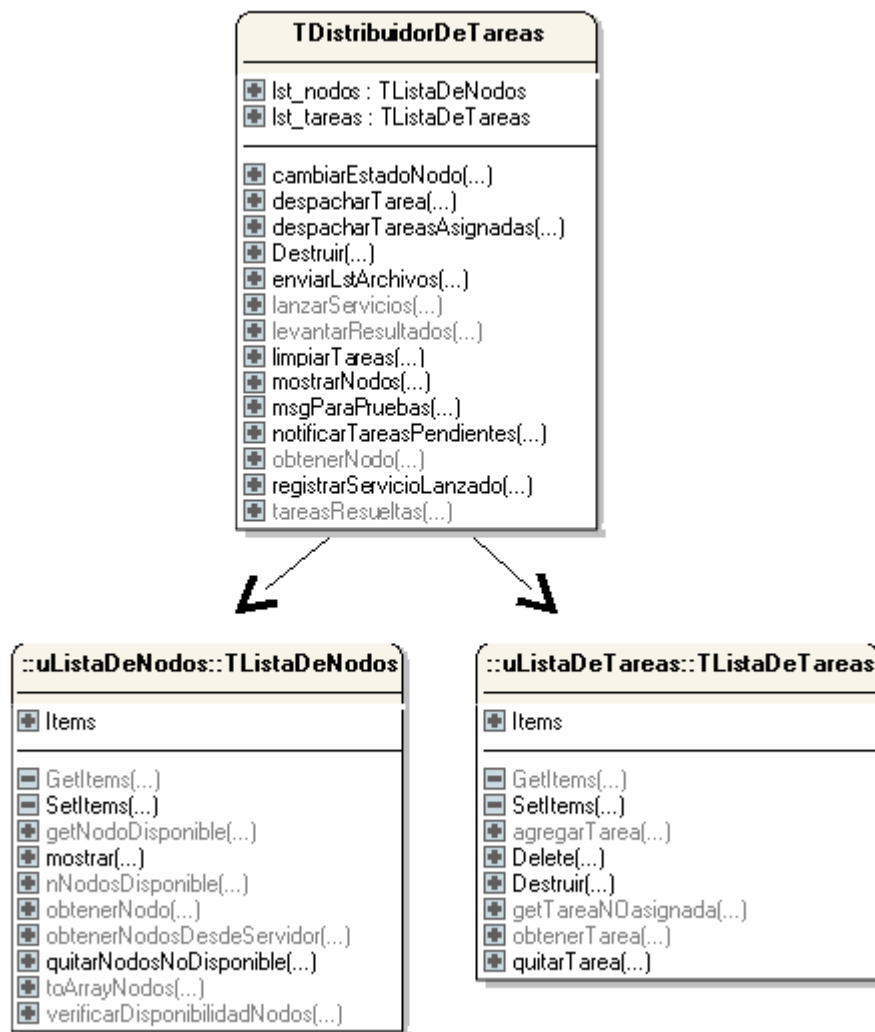


iv. Distribuido

Luego de implementa
 entre quienes, se pro
 gestionar la lista de n
 distribuir éste debe se
 distribuida.

as del problema, o sea lo que va a ser distribuido y
 distribuidor de tareas. Este objeto se encarga de
 o que se desee
 suelto de forma

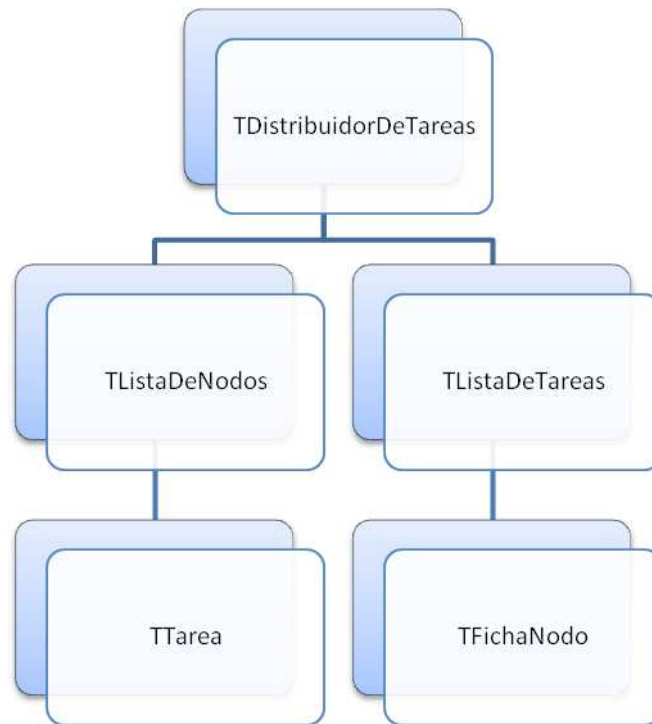
D. Diagrama de Clases



Vale la pena destacar dos funciones propias del distribuidor implementadas mediante distribución de tareas:

- **Lanzar aplicaciones:** Esta función recibe como parámetro el nombre de la aplicación cliente capaz de resolver el problema que se pretende distribuir. Las tareas creadas son asignadas a cada uno de los nodos disponibles. Una vez despachadas el distribuidor queda esperando por la confirmación de que la aplicación fue lanzada en cada nodo, si algunos de los nodos no confirma o no contesta entonces ese nodo es quitado de la lista de nodos.
- **Enviar archivos:** Para la resolución de algunos problemas puede ser necesario la transmisión de archivos entre los distintos nodos, esta función no es obligación ejecutarla como el lanzamiento de las aplicaciones cliente pero se decidió integrarla en el distribuidor más básico ya que será de uso común.

E. Jerarquía de clases



F. Optimizador distribuido.

La versión experimental del Optimizador distribuido utilizaba una sala de juegos especial y con las sucesivas mejoras realizadas al SimSEE quedo obsoleta, por lo tanto se optó por desarrollar un nuevo optimizador que utilice la misma sala de juegos que el optimizador habitual. De esta manera el código se mantiene actualizado automáticamente.

Dada la capacidad de almacenar recursos de los sistemas de energía eléctrica no es posible obtener la operación óptima de forma independiente en cada paso de tiempo, es decir es necesario haber calculado los pasos siguientes para poder calcular el paso actual. Por lo tanto la paralelización de este proceso se debió realizar por pasos de tiempo ya que el costo de cada punto de la discretización es independiente.

El nuevo optimizador fue implementado como una máquina de estados donde a cada estado se le asocia una lista de tareas para realizar y la condición para avanzar de estado es haber resuelto todas las tareas.

A cada estado se le asocia un “time out” y una acción correctiva para resolver las tareas pendientes y continuar con el proceso.

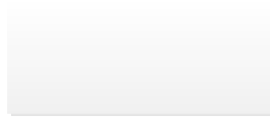
El siguiente diagrama de flujo ilustra los procesos distribuidos necesarios para realizar una Optimización distribuida:



G. Simulador distribuido.

Evaluar en distintos escenarios la operación óptima del sistema obtenida es ejecutar el algoritmo de simulación de la sala de juegos fijando una semilla aleatoria distinta en cada caso. La distribución de este proceso es sencilla ya que no depende de los pasos de tiempo, solo depende de la cantidad de crónicas y de la semilla aleatoria.

El siguiente diagrama de flujo ilustra los procesos distribuidos necesarios para realizar una Optimización distribuida:



i. Resultados

Para verificar la utilidad de la plataforma se seleccionó una sala de juegos para ser optimizada de forma distribuida entre 3 PCs y de forma habitual para comparar los tiempos en cada caso.

Sala: Psem2011_27_1emb.es

Tasa de actualización: 0.12

Número de crónicas: 100

Semilla: 31

Máximo Número de Iteraciones Opt: 0

Con sorteos: True

Estabilizar: False

La optimización de esta sala de forma habitual requirió de 1068s para ser completada mientras que la ejecución distribuida demora 635s.

Capítulo VI. OddFace–Teoría.

En este documento se presenta el módulo para Optimización Distribuida De Funciones de Alto Costo de Evaluación (ODDFACE) desarrollado en el marco del proyecto ANII FSE 18-2009 "Mejoras de la plataforma SimSEE".

La plataforma SimSEE permite realizar simulaciones del sistema de generación eléctrica de una región o país. En la simulación es posible representar tanto las centrales de generación en base a combustibles fósiles (fuel oil – gasoil, etc.) como centrales de generación en base a hidroelectricidad o fuentes renovables como la eólica o la solar. También es posible representar en las simulaciones las interconexiones con otros sistemas eléctricos (por ejemplo Uruguay con Argentina y Brasil). Esta breve descripción se realiza para mostrar claramente que la simulación de un sistema tiene asociada la representación de una realidad llena de detalles e incertidumbres. Estas simulaciones se realizan mediante la simulación de muchas "crónicas" o "historias posibles" o "realización de los procesos estocásticos involucrados". Normalmente, como resultado principal de la simulación se obtiene el costo de abastecimiento de la demanda que es la integral de los costos incurridos en el horizonte de tiempo considerado. Cuanto mayor sea el número crónicas (o historias posibles) simuladas se obtendrá un resultado mejor caracterizado desde el punto de vista estadístico.

El primer punto a destacar es que nuestra función de costo es evaluada mediante simulaciones de Monte Carlo y en consecuencia lo que se obtiene es una estimación de su valor que mejora con la cantidad de sorteos utilizados. También es interesante que cada sorteo de Monte Carlo está asociado a una evaluación de la función que es independiente del resto y esto tiene dos implicancias directas que son el permitir la evaluación distribuida de los diferentes sorteos y también poder ir teniendo estimadores intermedios que van mejorando con la cantidad de evaluaciones que se van realizando.

En términos formales, la función de costo a evaluar, es del tipo:

$F(X, r)$ donde $X \in D$ es el vector de parámetros de optimización del problema que puede tomar valores en el "Dominio del Problema" D y r identifica una realización posible del conjunto de procesos estocásticos (puede considerarse que es un ordinal que identifica el número de sorteo de Monte Carlo). La función $F(X, r)$ es "el costo" de operar el sistema si los parámetros se fijan en X y la simulación se realiza con la crónica (suerte, o realización de los procesos estocásticos) r .

Estamos interesados en resolver el problema de encontrar "el mejor" juego de parámetros X , entendiendo por "el mejor" aquél que minimiza una función de costo (o función objetivo) que veremos se puede construir a partir de las evaluaciones $F(X, r)$ en el conjunto de las r . A modo de ejemplo (y seguramente el objetivo de mayor uso) es buscar minimizar el valor esperado de $F(X, r)$ en el ensamble (conjunto de realizaciones posibles) de las r .

En el caso del valor esperado, la función objetivo a minimizar es:

$$f(X) = \langle F(X, r) \rangle_r$$

Se supone en este trabajo que la función $F(X, r)$ es de "alto costo de evaluación" y por consiguiente se intentará evaluar la menor cantidad de veces posible. Generalmente un alto costo de evaluación significa un uso de una potencia de cálculo importante durante un tiempo prolongado.

En el ejemplo, considerando el sistema Uruguayo, una simulación del tipo realizada para propósitos de planificación de inversiones, con paso semanal de un horizonte de 20 años lleva del orden de 10 minutos de cálculo en un PC de escritorio.

El algoritmo desarrollado hace una exploración del dominio D utilizando diferentes "agentes de exploración" que se pueden comportar de forma diferente. Todos los agentes van alimentando los resultados a una base de datos compartida. Los diferentes agentes, se ejecuta en diferentes nodos de una red de cálculo logrando así paralelizar la exploración.

En cuanto a la evaluación de un punto $X \in D$, la implementación permite indicar un subconjunto del ensamble $s \in [r]$. Esto implica que el mismo punto puede ser calculado más de una vez, con diferentes subconjuntos. Cada nueva información es utilizada así para mejorar la representación de la función objetivo. Si el mismo punto es evaluado en los subconjuntos de realizaciones $s_1, s_2, \dots, s_k$, la estimación de la función objetivo irá mejorando en una serie que tiende al valor verdadero si cada nueva evaluación incluye la información de las anteriores,

$$f_1(X), f_2(X), \dots, f_k(X) \rightarrow f(X)$$

La exploración del dominio D se realiza entonces en forma distribuida, por varios agentes utilizando la información de todas las evaluaciones compartida en forma permanente entre los agentes. Cada agente, utilizando un algoritmo de estimación de un nuevo punto, genera una propuesta de nuevo punto de evaluación y genera una evaluación con un conjunto de realizaciones. Se obtiene así una "estimación" del objetivo en dicho punto. Si el punto ya había sido calculado, se integra la nueva información mejorando la estimación existente.

Las principales componentes del algoritmo son:

- 1) Capa de comunicación para permitir el cálculo distribuido.
- 2) Agentes de cálculo de difentes tipos.
- 3) Traductor de parámetros a configuración de los archivos de simulación (salas) de SimSEE para poder ejecutar el simulador y obtener los resultados.
- 4) Mejora incremental de las estimaciones.

A. Capa de comunicación entre los agentes.

Como ya se mencionó, OddFace es una plataforma que permite la búsqueda distribuida del óptimo de un problema de optimización mediante un conjunto de Agentes Exploradores, que comparten la información de las zonas exploradas del Dominio del Problema, por medio de una base de datos central. La fig. Error: No se encuentra la fuente de referencia muestra en forma esquemática la situación.

Entre las diferentes alternativas de comunicación se priorizó aquella que asegure una conectividad sencilla y robusta. Los tiempo de comunicación son despreciables en comparación con el tiempo involucrado en cada evaluación.

Para permitir que los agentes puedan ejecutarse en cualquier tipo de computadora que tenga conexión a Internet, se optó por centralizar la información de las evaluaciones en un servidor de base de datos con acceso vía web y los agentes se comunican con dicho servidor mediante consultas simples vía protocolo http: usando una librería de comunicación sencilla.

El primer paso para iniciar la solución de un problema es obtener el NID (número identificador) de la solución. Par ello, se utilizará una aplicación desarrollada llamada "OddFace_prepare" que permite "CREAR UN NUEVO PROBLEMA" y suministra el NID. Esta misma aplicación permite especificar parámetros de ejecución del problema, monitorear su ejecución, detener, rearrancar y terminar la ejecución de un problema.

Al crear el problema se debe indicar el tipo de problema y según el tipo se podrá especificar los datos característicos del problema. Cada tipo de problema tiene asociado una descripción de parámetros (con lo cual queda definido un largo de cadena de ADN) y una forma específica de decodificación del ADN en los parámetros reales que permiten la evaluación del problema.

En particular, como ejemplo de aplicación se implementó un Optimizador de Inversiones de Generación sobre SimSEE (OddFace_PIG) y un Optimizador del Plan Anual de Mantenimientos (OddFace_PAM). Es así que en el formulario de de definición de problemas de OddFace_Pprepare se puede elegir inicialmente entre estas aplicaciones (o tipos de problemas).

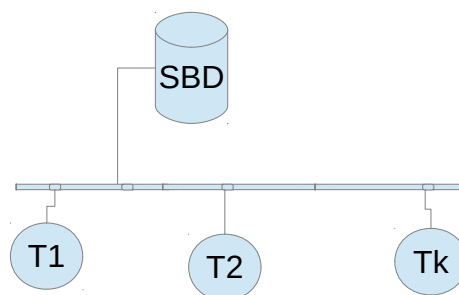


Fig. 10 Servidor de Base de Datos y Trabajadores.

El problema es definido en un Servidor de Base de Datos (SDB) con visibilidad desde todos los nodos de cálculo. En el SDB se define un almacen de datos (una tabla en la base de datos) que contiene un record para cada punto evaluado. Todo punto $X \in D$ evaluado, es almacenado con la información asociada al resultado. La información almacenada incluye una codificación del punto

(el ADN) que identifica en forma unívoca el punto y los resultados de la evaluación. Como resultados de la evaluación se incluye el valor esperado (en el conjunto crónicas consideradas) "f_VE", valor en riesgo "f_VaR", valor en riesgo condicionado "f_CVaR" y un histograma con las evaluaciones correspondiente a cada una de las crónicas consideradas. Como el mismo punto puede ser evaluado más de una vez (sobre un conjunto diferente de crónicas) el registro incluye la cantidad de veces que fue sujeto de evaluación el punto. Para los puntos con más de una evaluación (cada evaluación se realiza con la misma cantidad de crónicas) los valores almacenados corresponden al promedio de los valores correspondientes a cada evaluación.

La fig .. muestra el ciclo de funcionamiento de cada aplicación

La tabla de problemas en el servidor tiene la siguiente definición:

```
CREATE TABLE `ofe_problemas` (  
  `nid` int(11) NOT NULL,  
  `usuario` int(11) NOT NULL,  
  `llave` int(11) NOT NULL,  
  `dt_creacion` datetime NOT NULL,  
  `ro_VE` double NOT NULL,  
  `ro_VaR` double NOT NULL,  
  `ro_CVaR` double NOT NULL,  
  `N_CronicasPorVez` int(11) NOT NULL,  
  `N_DiscretizacionesHistograma` int(11) NOT NULL,  
  `semilla_madre` int(11) NOT NULL,  
  `ro_GA` double NOT NULL,  
  `ro_EG` double NOT NULL,  
  `ro_MJ` double NOT NULL,  
  `GA_prob_premio_exito` double NOT NULL,  
  `GA_prob_mutacion` double NOT NULL,  
  `descripcion` text NOT NULL,  
  `tipo` int(11) NOT NULL,  
  `estado` int(11) NOT NULL,  
  PRIMARY KEY (`nid`) )  
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

```
CREATE TABLE `ofe_tipos_problema` (  
  `nid` int(11),  
  `nombre` varchar(8) NOT NULL,  
  `descripcion` text NOT NULL,  
  PRIMARY KEY (`nid`) )  
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

```
INSERT INTO ofe_tipos_problema ( nid, nombre, descripcion ) VALUES ( 0, 'base', '-- sin definir  
--' );
```

```
INSERT INTO ofe_tipos_problema ( nid, nombre, descripcion ) VALUES ( 1, 'PIGSimSEE',  
'Planificación de Inversiones de Generación sobre SimSEE' );
```

```
INSERT INTO ofe_tipos_problema ( nid, nombre, descripcion ) VALUES ( 2, 'PAMSimSEE',  
'Programa Anual de Mantenimientos de unidades de generación sobre SimSEE' );
```

```
CREATE TABLE `ofe_estados_problema` (  
  `nid` int(11),  
  `nombre` varchar(8) NOT NULL,  
  `descripcion` text NOT NULL,  
  PRIMARY KEY (`nid`) )  
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

```
INSERT INTO ofe_estados_problema ( nid, nombre, descripcion ) VALUES ( 0, 'Inactivo',  
'Inactivo' );
```

```
INSERT INTO ofe_estados_problema ( nid, nombre, descripcion ) VALUES ( 1, 'Activo', 'Activo'  
);
```

Funcion Objetivo

Valor Esperado y Aversión al Riesgo.

Construcción de Histogramas.

Individuo

Poblacion

Solución

B. Implementación Algoritmos Genéticos en OddFace v1.0

En esta sección se describe la implementación de los Algoritmos Genéticos en OddFace (versión 1.0 Noviembre 2012) al nivel suficiente como para poder entender y realizar modificaciones del código.

La implementación está repartida en dos Units (Módulos Pascal) que son “utipos_ga” en la que se definen los tipos básicos de datos para la representación de las cadenas de ADN y de los Genotipos y “uoddface” en la que se definen las clases bases para la descripción de problemas de optimización OddFace.

i. Definiciones básicas para manejo de cadenas de ADN.

La unidad Pascal "utipos_ga" contiene las definiciones básicas para el manejo de Algoritmos Genéticos. En esta sección se usará “*Negrita cursiva*” para escribir definiciones textuales del código fuente Pascal.

Lo primero a resaltar, es que en la implementación las cadenas de ADN son representadas por vectores de BITS organizados en vectores de datos de 16 bits (tipo WORD de Pascal). O sea que toda cadena de ADN está almacenada como vectores de palabras de 16 bits.

En la Interface, se define el tipo:

TCadenaADN = packed array of word;

Así definida, una instancia de TCadenaADN será un vector de longitud a definir de datos del tipo *Word* (16 bits Pascal). La palabra “packed” le indica al compilador que ubique en forma “contigua” todos los bytes, previendo que podremos acceder a los mismos en forma directa. Igualmente en la implementación se buscó acceder siempre por medio de la estructura del vector con lo cual no debiera ser relevante el que los bytes estén contiguos en memoria.

Con esta definición de ADN, si se quiere acceder al bit k de una cadena, hay que acceder al dato ($k \div 16$) del vector y dentro de ese dato al bit ($k \bmod 16$). Para acceder al bit ($k \bmod 16$) hay que construir una máscara de bit, que tenga 0 en todos los bits salvo en la posición ($k \bmod 16$) para lograr así mediante una operación binaria AND aislar el bit al que se quiere acceder.

Se definen las constantes:

- ***BIT_MAS_SIGNIFICATIVO = \$8000;*** Esta constante sirve de máscara para aislar el bit más significativo en un dato de 16 bits.
- ***BIT_MENOS_SIGNIFICATIVO = \$0001;*** Esta constante sirve de máscara para aislar el bit menos significativo en un dato de 16 bits.

Entonces, dado un vector de datos ADN para leer el bit “k” (suponiendo $k: 0..NBits-1$) tendríamos que hacer:

mascara_de_bit = BIT_MENOS_SIGNIFICATIVO shl (k mod 16);

jw:= k div 16;

$Bitk := ADN[jw] \text{ and } mascara_de_bit;$

Donde: el operador "*shl*" es estándar de Pascal (se lee Shift Left) y tiene el efecto de hacer un desplazamiento binario hacia la izquierda del primer parámetro ($k \bmod 16$) posiciones.

El operador "*div*" es estándar de Pascal calcula la parte entera de la división.

El operador "*mod*" es estándar de Pascal y calcula el residuo de la división.

Para poner a UNO el Bit *k* de la cadena ADN la operación sería:

$ADN[jw] := ADN[jw] \text{ or } mascara_de_bit;$

Para poner a CERO el bit *k* de la cadena ADN la operación sería:

$ADN[jw] := ADN[jw] \text{ and } NOT(mascara_de_bit);$

Para invertir el bit *k* de la cadena ADN la operación sería:

$ADN[jw] := ADN[jw] \text{ xor } mascara_de_bit;$

a) ADN-> Genotipo y Fenotipo.

Toda la información contenida en los cromosomas se conoce como genotipo, sin embargo dicha información puede o no manifestarse en el individuo. El fenotipo se refiere a la expresión del genotipo más la influencia del medio.

La Cadena de ADN es una representación del Genotipo y caracteriza totalmente al individuo desde el punto de vista genético. La experiencia posterior de cada individuo en su ambiente podrá llevar a diferenciar individuos que desde el punto de vista genético son idénticos. Por ejemplo, dos gemelos con igual información genética, luego son diferenciables por rasgos externos.

Los términos "genotipo" y "fenotipo" fueron creados por Wilhelm Johannsen en 1911. El genotipo es la información hereditaria completa de un organismo, incluso si no se expresa. El fenotipo es una propiedad observada en el organismo, como la morfología, el desarrollo, o el comportamiento.

Observando la cadena de ADN se puede conocer el genotipo, observando la apariencia y desempeño del organismo en su ambiente se puede conocer el fenotipo.

Las propiedades físicas de un organismo son las que determinan directamente sus posibilidades de supervivencia y reproducción, mientras que la herencia de las propiedades físicas sólo se produce como una consecuencia secundaria de la herencia de genes.

El mapeo de un conjunto de genotipos a una serie de fenotipos a veces se denomina mapa genotipo-fenotipo. En el caso de aplicación en OddFace sobre la plataforma SimSEE, el genotipo de un individuo determina todo lo necesario para poder simular y evaluar la performance de ese individuo. Por ejemplo fija todos los parámetros de una Sala SimSEE para hacer la simulación y obtener el costo futuro de operación del sistema como un índice de desempeño. Al realizar una simulación, se seleccionará un conjunto de crónicas (muestreos de Monte Carlo) y es así que dos evaluaciones del mismo individuo con dos conjuntos de crónicas diferentes puede llevar a dos evaluaciones diferentes de la performance del mismo individuo. Es así que en el caso de aplicación, que involucra una plataforma de simulación que simula "el ambiente" en el que el "individuo" (caracterizado por su genotipo) tiene la oportunidad de desempeñarse y mostrar así su Fenotipo.

Esta representación, ajustada a la realidad, en que un mismo genotipo puede correr con suertes diferentes durante la evaluación, puede considerarse como un comportamiento "robuzto" de

la naturaleza, en que no se define un “líder absoluto” sino que existen un conjunto de individuos que son “los mejores” pero que dependiendo de la suerte de vida de cada uno, hay algunas genéticas que resultan mejores, peor en otra circunstancia pueden invertirse la condición de ser mejor. Esto de alguna forma impide que quede una genética UNICA como ganadora y es importante para que no se pierda la capacidad de adaptación de los organismos. Pero pensando en el objetivo de encontrar el individuo óptimo como la solución a un problema concreto que tiene una especificación clara (no condiciones externas que van cambiando) esta influencia de “la suerte” del individuo durante la evaluación es más bien un aspecto negativo de los AG. En esta primera implementación de OddFace v1.0 se ha implementado que “no hay control” sobre la suerte del individuo y por lo tanto, el mismo genotipo, evaluado dos veces puede tener diferenciación. En el análisis de los casos de test realizados para esta implementación más adelante se profundiza sobre este tema y se propone una implementación alternativa, que aunque no sea ajustada a la realidad, se considera mejor para la resolución del tipo de problemas de test realizados.

b) Descriptores de Genotipos.

En la Unidad “utipos_ga” se define la clase:

```
TDescriptorGenotipo = class  
nombre: string; // identificador del parámetro  
nbits: integer;  
constructor Create( nombre_: string; nbits_: integer );  
procedure codificar_ADN( var adn: TCadenaADN; var offset: integer; var mask: word; var Genotipo ); virtual;  
function decodificar_ADN( var Genotipo; var adn: TCadenaADN; var offset: integer; var mask: word ): boolean; virtual;  
end;
```

Esa clase se utiliza para describir un Genotipo, o parámetro del individuo. Por ejemplo, en el caso del PAM, dada una Orden de mantenimiento de una unidad (Por Ej.: “Sacar la 6ta Unidad de Central Batlle durante 15 días para mantenimiento entre 1/1/2012 y 1/3/2013) el genotipo podría representar para esa orden la fecha de inicio del mantenimiento.

En el descriptor del Genotipo:

- La propiedad “nombre” nos permite identificar claramente a qué parámetro se refiere (por ejemplo podría ser el número de orden de mantenimiento).
- La propiedad “nbits” indica la cantidad de bits necesarios para codificar el rango posible de variación del parámetro.
- El constructor nos permite crear una instancia del descriptor.
- El procedimiento “codificar_ADN”, recibe como parámetro una cadena de ADN y el offset al casillero en el que comienza la codificación del Genotipo. El parámetro “mask” es una máscara binaria de 16 bits, todos en cero, salvo en la posición del primer bit del Genotipo en la posición ADN[offset]. Al codificar el genotipo, se copia la representación binaria del parámetro Genotipo, a partir del bit determinado por “mask” dentro de ADN[offset] y se devuelven los parámetros “offset” y “mask” de forma de dejarlos apuntando al bit siguiente al último usado dentro de la cadena para codificar el Genotipo. Fijando inicialmente offset=0, mask = 1, y llamando al procedimiento codificar_ADN sobre el vector de todos los Genotipos, se obtiene en ADN la codificación total del individuo.

- La función “decodificar_ADN”, permite leer el genotipo a partir de una cadena de ADN. Los significados de los parámetros “offset” y “mask” son los mismos que los explicados en el párrafo anterior y permiten leer de la cadena de bits, los correspondientes al Genotipo particular y modificar los parámetros para prepararlos para que el próximo descriptor de genotipo pueda hacer su trabajo. El resultado de la función es TRUE si la decodificación binaria no precisó ser ajustada para cubrir el rango especificado para el parámetro y FALSE si hubo que hacer un ajuste. Para entender mejor el sentido de TRUE o FALSE como resultado leer la descripción del manejo de Genotipos Enteros a continuación.

Como se puede apreciar, en esta clase genérica, nada se dice sobre el tipo de dato del parámetro “Genotipo” solo se supone que en el lugar de memoria apuntado por dicho parámetro se pueden leer o escribir nbits.

Se definen clases refinadas de TDescriptorGenotipo, para facilitar el manejo de parámetros Booleanos, Reales y Enteros.

TDescriptorGenotipoBooleano. En este caso simplemente usa la clase genérica fijando nbits=1.

TdescriptorGenotipoEntero. En esta clase se agregan los parámetros “k_min y k_max” que deben ser pasados en el constructor y que fijan el rango admitido para el parámetro. La cantidad de bits se calcula en el constructor y se fija de forma tal que nbits sea el menor entero tal que $(k_{\max} - k_{\min}) \leq 2^{\text{nbits}}$. Supongamos que se trata de un parámetro entero llamado “Camino” que puede tomar los valores 1, 2 o 3. Entonces, al crear la instancia del descriptor de genotipo llamaríamos al constructor así: camino:= TdescriptotrGenotipoEntero.Create('Camino', 1, 3); Como consecuencia de dicha llamada, se creará un descriptor, en el que nbits = 2. Esto lleva a que en las cadenas de ADN existan 2 bits que están para representar este genotipo. Como 2 bits pueden representar 4 valores y sólo se necesitan 3 en este caso, hay más de una codificación binaria que terminará dando el mismo genotipo. Es aquí donde interviene el resultado de la función “decodificar_genotipo”. En la implementación se decidió realizar directamente la decodificación binaria del “tramo de bits” del ADN a un número entero obteniendo un número entre $2^{\text{nbits}} - 1$. Es valor así obtenido se suma a k_min para obtener el valor del genotipo, si el resultado es mayor que k_max, se ajusta a k_max y se retorna FALSE para indicar que hubo que hacer un ajuste. Si el resultado es TRUE es que no fue necesario hacer ningún ajuste de rango.

TDescriptorGenotipoReal. Esta clase refinada de TdescriptorGenotipo es útil para el manejo de parámetros Reales (punto flotante). El constructor permite pasar como parámetros, el nombre del genotipo, el valor mínimo y máximo del parámetro y un parámetro “nbits” que determina la fineza con que se discretizará el rango ($x_{\max} - x_{\min}$) en la representación. El rango ($x_{\max} - x_{\min}$) será representado por 2^{nbits} puntos siendo por tanto la distancia entre los puntos de la discretización $dx = (x_{\max} - x_{\min}) / (2^{\text{nbits}} - 1)$. Como se puede apreciar, el parámetro real es representado por una discretización numerable por lo que en esencia es tratado como si fuera un parámetro entero.

ii. Definición de Clases básicas para descripción del problemas OddFace.

La unidad Pascal “uoddface” contiene las definiciones de las Clases Pascal útiles para la definición de problemas OddFace.

Las clases básicas son:

- **TIndividuo.** Esta Clase define la base de la cual derivar clases que puedan describir los individuos de cada problema concreto. Un individuo se diferencia de otro en esencia por su Genotipo, representado en su cadena de ADN. El Individuo está asociado a un Problema y es en el marco de ese problema que será evaluado y dependiendo de su desempeño será clasificado como más o menos apto.
- **TProblema.** Esta Clase define la base de la cual derivar clases para problemas concretos. El Problema contiene la descripción que permite poder evaluar el desempeño de los individuos asociados al problema. Por ejemplo, en un problema PIG (Planificación de Inversiones de Generación) el Problema contiene la información como para generar la Sala SimSEE a partir de un plan de inversiones dado (es decir un individuo) y para evaluar el costo resultante de ese plan de inversiones.
- **TExplorador.** Esta Clase generaliza el mecanismo de búsqueda de individuos. Como método principal tiene el de “proponer un nuevo individuo”. De esta clase básica es posible derivar exploradores refinados con diferentes estrategias para proponer nuevos individuos.
- **TExploradorGenetico.** Esta clase es un refinamiento de TExplorador. El mecanismo para proponer un nuevo individuo es en esta clase la del Cruzamiento de dos individuos seleccionados del conjunto de individuos ya evaluados del problema de acuerdo a un índice de performance para obtener así el ADN del nuevo individuo. El ADN resultante del cruzamiento puede a su vez sufrir cambios por la operación de Mutación.
- **TExploradorMejorador.** Esta clase es un refinamiento de TExplorador. El mecanismo para proponer un nuevo individuo es seleccionar uno de los mejores ya evaluados y simplemente repetir la evaluación con “otra suerte” para mejorar así la evaluación del individuo.

La solución de un problema pasa por activar exploradores que vayan buscando los mejores individuos (aquellos que minimizan la función objetivo de la optimización).

a) TIndividuo.

La clase TIndividuo tiene las propiedades:

- **Problema:** *TProblema*; Esta variable almacena una referencia al problema al que pertenece el individuo.
- **tipo_COD:** *integer*; Esta propiedad determina el tipo de codificación en que se almacena la representación binaria para las operaciones de Cruzamiento y Mutación. Los valores pueden ser: 0, 1 o 2 según que la codificación sea: BINARY, GRAY o UNARY.
- **nid:** *integer*; Ese es un identificador único del Individuo en la tabla de individuos evaluados asociados al problema al que pertenece el individuo. Si el valor es -1 (menos uno) significa que aún no se ha asignado un identificador único al individuo. Los identificadores únicos deben ser solicitados al servidor de base de datos.
- **XR:** *TVectR*; Vector de genotipos reales. Contiene el conjunto de valores reales que representan los parámetros reales del individuo.

- *XE: TVectE*; Vector de genotipos enteros. Contiene el conjunto de valores enteros que representan los parámetros enteros del individuo.
- *ADN: TCadenaADN*; Representación del Genotipo (*XR* y *XE*) por una cadena de bits.
- *f_VE, f_VaR, f_CVaR, f_MIN, f_MAX, f_objetivo: NReal*; Estas variables almacenan las estimaciones realizadas en base a conjunto de crónicas (muestreos de Monte Carlo) del valor esperado de la función de costo, del que es excedido con probabilidad 5%, del valor esperado del conjunto del 5% más alto de costos, el valor mínimo del conjunto muestreado, el valor máximo del conjunto muestreado y el valor de la función objetivo que se busca minimizar respectivamente. El valor objetivo se compone como una combinación lineal de *f_VE, f_VaR* y *f_CVaR* según los pesos especificados en la definición del problema. Estos estimadores son obtenidos para cada evaluación (realizada con un número de muestreos de Monte Carlo) y si el individuo es evaluado más de una vez los promedios de los estimadores obtenidos en cada evaluación son almacenados en las variables *f_VE, f_VaR, f_CVaR, f_objetivo* y en las variables *f_MIN, f_MAX* se almacenan el mínimo y el máximo de los valores estimados en cada evaluación.
- *cnt_evaluaciones: integer*; Cantidad de veces que fue evaluado este individuo. Cada evaluación consisten en la realización de un número de muestreos de la función de costo definido en la especificación del problema. El mismo individuo pudo haber sido evaluado más de una vez y el valor *cnt_evaluaciones* da cuenta de eso.
- *f_histo: TVectR*; Esta propiedad es del tipo vector de reales y almacena directamente el valor de la función de costo obtenida por cada simulación de Monte Carlo realizada durante una evaluación. Por ejemplo, si en la especificación del problema se estableció que cada evaluación se realice sobre 100 crónicas (muestreos) entonces el vector *f_histo* tendrá los 100 valores obtenidos durante la simulación. En el caso en que un individuo haya sido evaluado más de una vez, el vector *f_histo* almacena el promedio de los valores *f_histo* que tendría cada evaluación en forma independiente si fuera la única.

```
// crea un nuevo individuo "limpio".
```

```
constructor CreateNew(problema_: TProblema);
```

```
// crea un individuo con un ADN dado poniendo a cero todos los demás parámetros.
```

```
// es útil para decodificar la cadena.
```

```
constructor CreateFromADN_HexStr( problema_: TProblema; ADN_HexStr:string );
```

```
// crea un individuo desde un record de la DB
```

```
constructor CreateFromRec(problema_: TProblema; r: TDataRecord);
```

```
// comunica el resultado al la DB
```

```
procedure ComunicarResultado;
```



```
// convierte el ADN a codificación GRAY  
procedure toGray;
```

```
// convierte el ADN a codificación BINARY  
procedure toBinary;
```

```
function ADN_AsBinaryStr: string;  
procedure Free; virtual;  
end;
```

iii. Defectos de la implementación OddFace v1.0.

La versión 1.0 de OddFace es la primer implementación se realizó y se testeó con casos de resolución de problemas de Planificación de Inversiones en Generación (PIG) y de optimización del Plan Anual de Mantenimientos (PAM).

En los casos resueltos se observa una convergencia lenta hacia una solución y una dispersión importante en los valores de la función objetivo. Seguramente la dispersión de los valores de la función objetivo son en parte causa de la convergencia lenta, pero también se detectó un problema en la forma de codificar el ADN que colabora en la convergencia lenta.

a) *Muestreos de Monte Carlo.*

La primer observación es que al tratarse de la búsqueda de un mínimo de una función, que es evaluada mediante un conjunto de sorteos de Monte Carlo por simulación y dado que la búsqueda del mínimo es en definitiva la comparación entre la evaluación en diferentes puntos del espacio de búsqueda, las técnicas de reducción de varianza por muestreo sincronizado valen para este caso. En otras palabras sería deseable comparar los valores para el mismo conjunto de realizaciones de los procesos estocásticos, o por lo menos que en lo posible, es decir en aquellos aspectos que los procesos estocásticos involucrados no dependen del punto de evaluación se mantenga para ambas evaluaciones.

A modo de ejemplo, en el caso del PAM, la rotura fortuita de las máquinas debiera mantenerse incambiada al igual que otros procesos como ser las velocidades de viento o los caudales de aportes a las centrales hidroeléctricas para que los mismos no agreguen diferencias a la comparación.

Para lograr esta mejora, bastaría con introducir un orden en las semillas aleatorias con que se evalúan los puntos. En la implementación actual, cada nodo de cálculo está seleccionando semillas aleatorias para realizar las simulaciones.

Si el problema se resolviera admitiendo una sola evaluación de NCronicas para cada individuo, resulta evidente que la mejor solución sería usar la misma semilla aleatoria para todas las evaluaciones. Tal como está implementada la generación de números aleatorios en SimSEE a partir de la versión 2.63 “Anarquía” la semilla aleatoria determina la generación de números aleatorios de cada componente de la simulación en forma independiente con lo cual al evaluar dos individuos, que en definitiva significa realizar la optimización y simulación de dos Salas SimSEE, si se utiliza la misma semilla aleatoria, la generación de aleatoriedad en cada componente que permanezca incambiado en ambas Salas será idéntica.

Ahora bien, ¿cómo se podría implementar un perfeccionamiento que permita mantener un muestreo de Monte Carlo coherente junto con la posibilidad de ir mejorando la resolución por aumento de la cantidad de evaluaciones?. Una implementación posible sería fijar que el algoritmo genético funcione siempre con individuos que hayan sido evaluado la misma cantidad de veces. Es decir, cuando se buscan los progenitores para proponer un nuevo individuo, se realiza la búsqueda sobre los individuos con igual cantidad de evaluaciones.

Primero se impondría que la semilla aleatoria para una evaluación sea un valor base más el número de evaluación de forma de garantizar que la primer evaluación se realiza para todos los individuos con la misma semilla (por ejemplo 31) y que la segunda se realiza para todos con otra semilla diferente a la primera pero iguales para todos (por ejemplo 32) y de igual forma para las evaluaciones de orden superior.

b) Residuo de bits en la cadena de ADN.

En la i

1) Coherencia de los SORTEOS.

=====

Milena me preguntó hace unos días si al poner la probabilidad de mejora $ro_MJ = 0$ nunca se repetían evaluaciones y si todas las evaluaciones en ese caso se realizaban sobre el mismo conjunto de crónicas.

La respuesta es NO y NO.

NO1, porque, aunque $ro_MJ = 0$, por la sola acción del algoritmo genético (cruzamiento + mutación) se pueden llegar a repetir ADN's y por lo tanto se pueden repetir evaluaciones.

NO2, porque, luego de obtenido el ADN (sea por el método que sea) antes de hacer la corrida para evaluar la $f_Objetivo$ se elige una semilla totalmente aleatoria.

mmmm..... esto es parte del tema por el que hay tanta varianza en la comparación de la $f_Objetivo$. Seguro que es MUCHO MEJOR, mantener un conjunto de sorteos Coherentes.

Por ejemplo, que la primer evaluación de cualquier punto sea siempre con la semilla 31, la segunda siempre con 32 y así sucesivamente.

De esa forma, considerando puntos con igual cantidad de evaluaciones sabemos que sufrieron las mismas suertes y que las diferencias que vemos entre ellos no está tan afectada por la suerte que tuvieron (o por lo menos que para la misma suerte las diferencias son por su diferente posición en el espacio de búsqueda).

2) Redundancia en la codificación.

=====

De las corridas de María Cristina y Felipe, hay una que es muy sencilla por tener solo dos variables y cada una de las variables puede tomar solo 13 valores. Esto lleva a un espacio de búsqueda muy reducido 13 x 13 valores. Sin embargo la tabla de Individuos evaluados tiene MUCHOS MAS. Analizando los resultados se ve que hay MUCHOS ADN's que al decodificarlos dan las mismas salas.

Lo que sucede es que las cadenas de ADN se implementaron como un tramo continuo de memoria en múltiplos de 16 bits. En el caso de este problema para codificar 13 valores es suficiente con 4 bits, por lo que en total el largo de la cadena de ADN es de 8 bits. Pero en la codificación se utiliza 16 bits. Al operar en los cruzamientos y mutaciones opera sobre toda la cadena (los 16 bits) cuando decodifica el ADN a los parámetros reales, solo importan los primeros 8 bits.

Estos primeros 8 bits son capaces de representar 16×16 salas, pero los números a representar son 13×13 .

El algoritmo implementado, primero decodifica el número entre 0 y 15 y lo mapea entre 0 y 12 asignando a 12 los valores 14 y 15.

De esta forma, el extremo superior de un parámetro está privilegiado en cuanto que hay más combinaciones que lo representan.

Entonces, para este ejemplo, los 8 primeros bits del ADN tomados en dos grupos de 4 bits y representan todos los puntos de exploración (13×13) con una redundancia sobre los valores del borde superior de cada una de las dos variables que tiene 3 representaciones en lugar de 1.

Pero los 8 bits siguientes pueden tomar cualquier número siendo que los 8 bits primeros son los que importan en la decodificación. Entonces cada sala (fijada por los primeros 8 bits) tiene $2^8 = 256$ representaciones posibles.

Esto NO es necesariamente un ERROR. Cuando lo implementé era consciente de que estaba creando redundancia (por lo menos la del rango de cada parámetro) pero no me percaté de que podría ser tan violenta como los 256 por cada una.

Llevado al extremo de una codificación que solo use un bit de los últimos 16 del ADN, sobran 15 bits al final que crean redundancia y eso significa una multiplicidad de $2^{15} = 32768$ valores.

El algoritmo tal como está tendría que ser capaz igual de encontrar los óptimos. Hay muchos clones sueltos nada más.

Pero creo que amerita una mejora.

Y implementé que la redundancia al final de la cadena de ADN se ponga toda a CERO y que no pueda tomar cualquier valor.

Con respecto a la redundancia de final de los intervalos de los parámetros, talvés una solución sea eliminar el individuo por INVIALE. O sea, si por el cruzamiento+mutación se obtiene un individuo fuera del rango en algunas de las variables, en lugar de obligarlo al extremo superior, directamente marcamos el individuo como INVIALE y no lo evaluamos.

Ambas mejoras me parecen MUY RELEVANTES, pero que no son un impedimento para documentar lo actual tal como está.

Por lo que NO VOY a subir las mejoras al servidor por ahora.

Versión 1.2-GEA.

Se implementó la independencia de los sorteos en las evaluaciones asignando la semilla a partir de la semilla inicial más el número de evaluación. Para que esta implementación sea válida, solo un nodo puede estar calculando la siguiente evaluación de un problema. En la implementación se verifica antes de subir el resultado de la evaluación, si la cantidad de evaluaciones aumentó durante el cálculo de la presente evaluación. Si aumentó significa que otro nodo estaba calculando la siguiente evaluación. Esto crea una ineficiencia pues varios nodos pueden comenzar la siguiente evaluación de un individuo, pero solo uno de los nodos logrará cargar el resultado. Otra implementación posible, sería, que el nodo “reservara” la evaluación antes de comenzar a calcular y por lo tanto los demás nodos directamente adelantan trabajo reservando las sucesivas evaluaciones del mismo individuo. Esta segunda implementación sería más eficiente, en el sentido de que no se desperdicia trabajo de los nodos, pero tiene el inconveniente de dejar la base de datos, en un estado que dependen de la contestación de los nodos de cálculo. Para una futura implementación habría que diseñar un mecanismo de representación de las reservas y de liberación de las mismas si pasado un tiempo no se recibió la respuesta.

C. OddFace_prepare – Manual de Usuario.

Optimización distribuida de Funciones de alto costo de evaluación.

Proyecto ANII-FSE-18-2009 "Mejoras plataforma SimSEE".

Milena Gurín Añasco, Daniel Cohn y Ruben Chaer

IIE-FING-UDELAR

Montevideo – Uruguay - Setiembre 2012

i. Introducción.

Este documento describe el conjunto de utilidades desarrolladas para la resolución de problemas de "Optimización distribuida de Funciones de alto costo de evaluación" – OddFace desde el punto de vista de un potencial usuario. Una descripción más detallada de los algoritmos utilizados se realiza en el documento "OddFace – Fundamentos" de este mismo proyecto.

Un problema de optimización puede plantearse como la búsqueda del mejor valor de un conjunto de parámetros de optimización $\mathcal{X}$ que puede tomar valores en un conjunto D (Dominio del problema en la jerga de optimización). Por "mejor solución" se entiende aquella que minimiza una función $f_{\text{Objetivo}}: D \rightarrow R$ denominada función de costo (o función objetivo). Para buscar el mejor $\mathcal{X}$ dependiendo de la estructura del dominio y de la función objetivo hay diferentes técnicas. En los problemas OddFace partimos de la base que desconocemos una formulación explícita de la función objetivo y que la misma es estimada dado un punto $\mathcal{X}$ mediante simulaciones de Monte Carlo y que las mismas implican un tiempo de cálculo que se considera "alto".

La estrategia de solución adoptada implica ir explorando el dominio, obteniendo estimaciones de la función de costo que se irán mejorando en aquellas zonas del dominio donde "parece estar" el óptimo. La exploración del dominio, se realizará por diferentes "Exploradores" en forma simultánea, distribuidos en una red de computadoras. Los exploradores se comunican entre ellos para colaborar en la búsqueda de la solución. Para ello existe un "Servidor de Base de Datos" (SBD) que es visible por todos los Exploradores y que utilizan para conocer los problemas OddFace a resolver, para ir guardando los resultados que van obteniendo y para compartir dicha información entre los exploradores. Se supone que el tiempo de comunicación es despreciable comparado con el tiempo de evaluar una estimación de $f_{\text{Objetivo}}(x)$.

El conjunto de aplicaciones desarrolladas consiste, por una lado en una aplicación denominada "OddFace_Prepate", que es la necesaria para la administración del SDB, la definición de problemas, el arranque y la parada del proceso de exploración distribuido y el monitoreo de soluciones en curso y por otro lado, los exploradores específicos para cada tipo de problema. Para la programación de los exploradores se desarrolló una librería de clases Pascal que facilita la escritura del explorador. En particular, se suministra una clase que facilita la evaluación de $f_{\text{Objetivo}}(x)$ por simulación con SimSEE de una sala base pasada como parámetro de definición

del problema, sobre la cual el explorador va cambiando las fichas de unidades que definen las inversiones en una tecnología de generación o unidades que representan el mantenimiento programado de las centrales de generación, según cuál sea el tipo de problema que estamos resolviendo. Sobre esta base es relativamente sencillo implementar otros tipos de optimizaciones que impliquen la alteración de una sala de SimSEE y su simulación como forma de evaluación de $f_{Objetivo}(x)$.

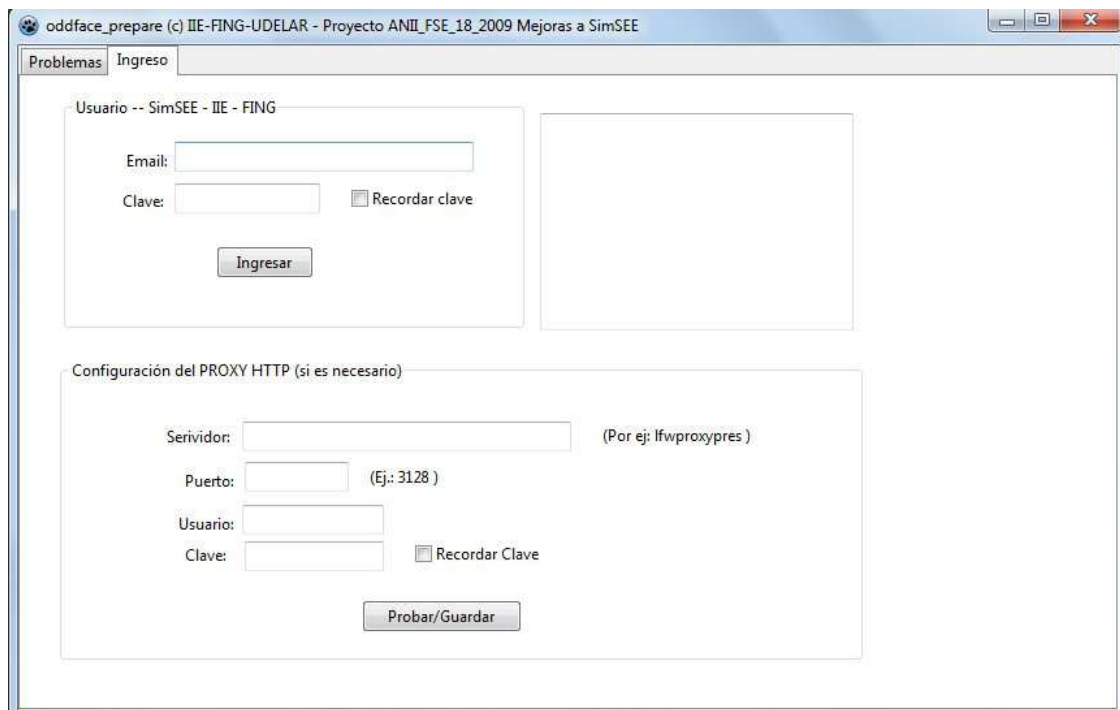
La rutina del Explorador, es un bucle que consiste en: "Informarse" desde el SBD cuál es el conjunto de evaluaciones ya realizado y en base a dicha información "Proponer" un nuevo punto x del espacio de exploración D , "Evaluar" la $f_{Objetivo}(x)$ e "Informar" al SBD sobre los resultados.

Como ejemplo de aplicación de OddFace se implentó un Planificador de Inversiones de Generación (PIG) y un optimizador del Plan Anual de Mantenimiento (PAM). Ambos agentes actúan sobre una sala SimSEE cambiando las unidades disponibles de las centrales de generación marcadas para optimización. Los agentes exploradores de estas dos aplicaciones ejemplo (y que son objetivos específicos del proyecto de investigación) son "OddFace_PIG" y "OddFace_PAM".

La aplicación "OddFace_Prepare" permite editar problemas para su solución con "OddFace_PIG" y con "OddFace_PAM" y está diseñada para que pueda servir de base para la edición de otros tipos de problemas en que se trate de optimizar sobre una sala de SimSEE.

ii. Listado de problemas.

Esta aplicación se conecta al SBD y en la primera pantalla (Figura 1) se da la opción de especificar un servidor proxy en caso de ser necesario para la conexión o simplemente "Ingresar" sin ninguna especificación.



The screenshot shows a web application window titled "oddface_prepare (c) IIE-FING-UDELAR - Proyecto ANII_FSE_18_2009 Mejoras a SimSEE". The window has two tabs: "Problemas" and "Ingreso", with "Ingreso" currently selected. The "Ingreso" tab contains two main sections. The first section, titled "Usuario -- SimSEE - IIE - FING", includes input fields for "Email:" and "Clave:", a checkbox for "Recordar clave", and an "Ingresar" button. The second section, titled "Configuración del PROXY HTTP (si es necesario)", includes input fields for "Servidor:", "Puerto:", "Usuario:", and "Clave:", checkboxes for "Recordar Clave", and a "Probar/Guardar" button. There are also example text strings: "(Por ej: lfwproxypres)" next to the "Servidor:" field and "(Ej.: 3128)" next to the "Puerto:" field.

Figura 1

Luego de Ingresar se muestra un listado de los problemas OddFace definidos. En la figura 2 se muestra dicha pantalla. En este caso, como se puede apreciar, hay varios problemas definidos. La

penúltima columna del listado tiene un "semáforo" que indica el estado del problema.

Si está verde  (como el de la primera fila) el problema está actualmente "en ejecución". Si el semáforo está en rojo  (como el de la segunda fila) el problema está "detenido".

Cuando se desea modificar algún parámetro en un problema existente es recomendable detener la ejecución y esperar un tiempo (alrededor de 20 minutos) para asegurarse de que todos los exploradores del clúster se hayan detenido. De esta forma, cuando se inicie nuevamente la ejecución, los exploradores comenzarán a evaluar con los mismos datos.

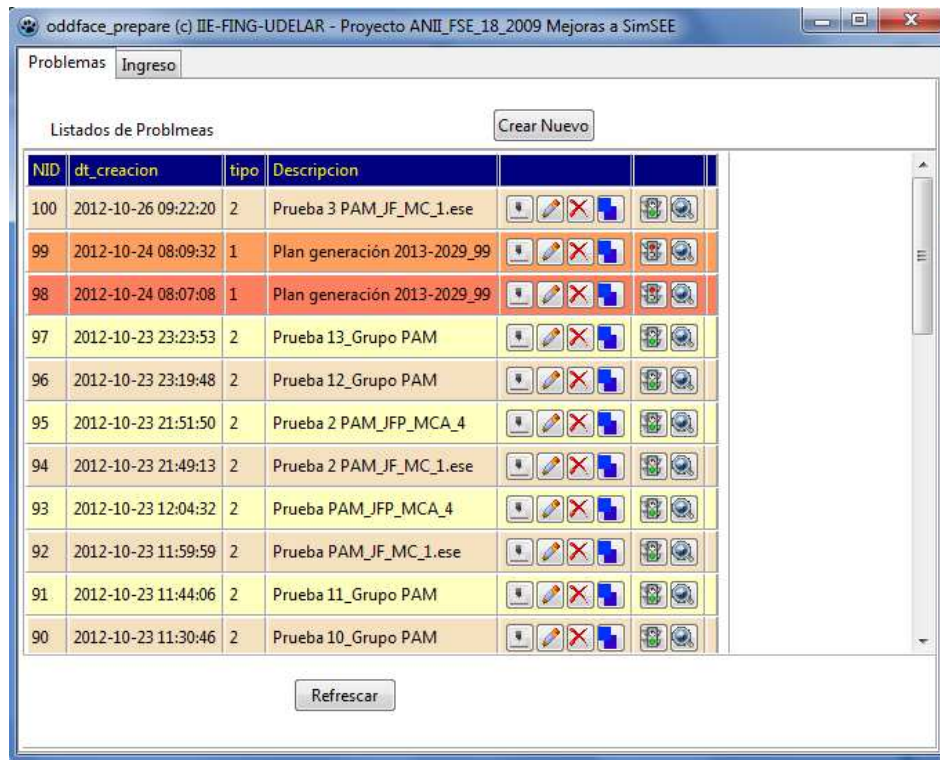


Figura 2

Presionando sobre el semáforo, se cambia su estado y por consiguiente se detiene o se reanuda la ejecución del conjunto de exploradores.

En la primera columna aparece el NID del problema que es un número de identificación que se asigna automáticamente cuando se crea un problema.

En la segunda columna aparece el "dt_creación" que es la fecha y hora en la cual se creó el problema.

En la tercera columna se muestra el tipo de problema, siendo el 1 un "OddFace_PIG" y el 2 un "OddFace_PAM".

En la cuarta columna aparece una breve descripción del problema que se escribe cuando se crea el mismo.

Por último se tiene una serie de botones cuyas funciones son las siguientes:

El botón  permite eliminar el problema por completo del SBD.

El botón  permite editar los parámetros del problema OddFace.

El botón  permite descargar la sala "limpia" de SimSEE y sus archivos asociados. Esto es útil para modificar dicha sala y/o para ejecutar la misma en forma manual para realizar pruebas. Los exploradores que buscan la solución del problema, bajan dicha sala para utilizarla en la

exploración.

La descarga se guarda en la carpeta donde se encuentra instalado el SimSEE, en la subcarpeta “ofe_NID” (siendo “NID” es el número de problema correspondiente).

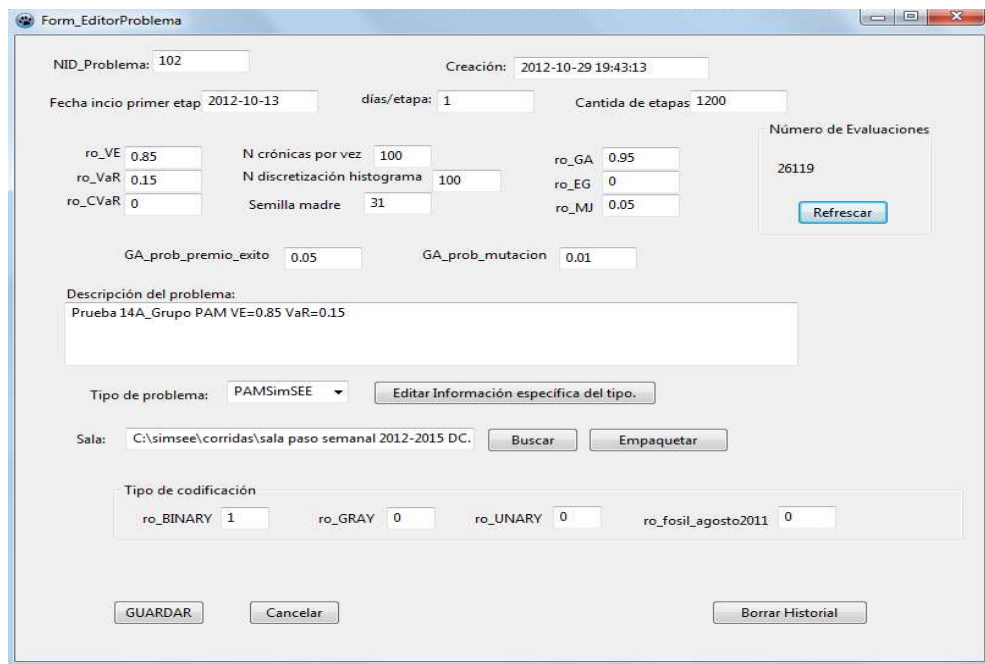
El botón  permite realizar consultas sobre el historial de evaluaciones del problema.

El botón  permite clonar un problema ya existente.

iii. Edición de los parámetros de un OddFace.

Si bien los algoritmos de exploración podrían aplicarse a problemas en los que no interviene el tiempo, es muy común, que la optimización involucre el hecho de encontrar el mejor juego de parámetros en un conjunto de etapas temporales. Por esta razón, y orientando la solución hacia su integración con la plataforma SimSEE se incluye, entre los parámetros, la descripción de un conjunto de "etapas" que deben ser consideradas como "pasos de decisión". El vector de parámetros $\mathcal{X}$ está compuesto por un conjunto de tramos que describen los parámetros posibles de cada etapa. Si el problema OddFace que se está resolviendo no implica etapas temporales, bastará con considerar que hay una única etapa.

La figura 3 muestra el formulario de edición de los parámetros generales de un OddFace.



The screenshot shows a window titled 'Form_EditorProblema' with the following fields and controls:

- NID_Problema:** 102 (read-only)
- Creación:** 2012-10-29 19:43:13 (read-only)
- Fecha inicio primer etap:** 2012-10-13
- días/etapa:** 1
- Cantida de etapas:** 1200
- ro_VE:** 0.85
- ro_VaR:** 0.15
- ro_CVaR:** 0
- N crónicas por vez:** 100
- N discretización histograma:** 100
- Semilla madre:** 31
- ro_GA:** 0.95
- ro_EG:** 0
- ro_MJ:** 0.05
- GA_prob_premio_exito:** 0.05
- GA_prob_mutacion:** 0.01
- Descripción del problema:** Prueba 14A_Grupo PAM VE=0.85 VaR=0.15
- Tipo de problema:** PAMSimSEE (dropdown menu)
- Sala:** C:\simsee\corridas\sala paso semanal 2012-2015 DC.
- Tipo de codificación:**
 - ro_BINARY:** 1
 - ro_GRAY:** 0
 - ro_UNARY:** 0
 - ro_fosil_agosto2011:** 0
- Número de Evaluaciones:** 26119 (with a 'Refrescar' button)
- Buttons:** GUARDAR, Cancelar, Borrar Historial, Buscar, Empaquetar, Editar Información específica del tipo.

Figura 3

Comenzando desde la parte superior del formulario se tiene:

- **NID_Problema:** Este campo no es editable. Es el identificador único asignado al problema. Es útil para referenciar el problema y para el depurado de la aplicación.
- **Creación:** Es la fecha y hora de creación del problema. Este campo no es editable.
- **Fecha inicio primera etapa:** El formato de este campo es Año-mes-día (aaaa-mm-dd) usando cuatro dígitos para el año y dos dígitos para el mes y el día (formato ISO). Especifica la fecha de inicio de la primera etapa de decisión.
- **Días/etapa:** Este campo debe tener el número de días por etapa, es decir el número de días entre decisiones. Esto es, a partir de la “Fecha inicio primera etapa”, cada cuántos días

después puede elegir comenzar con los mantenimientos de cualquiera de ellas en el período.

- Cantidad de etapas: Este campo indica la cantidad total de etapas de decisión.
- ro_VE , ro_VaR , ro_CVaR son los factores que se utilizarán para calcular la función objetivo del problema:

$$f_Objetivo = ro_VE \cdot f_VE + ro_VaR \cdot f_VaR + ro_CVaR \cdot f_CVaR,$$

Siendo “ f_VE ”, “ f_VaR ” y “ f_CVaR ” los valores estimados del Valor Esperado, del Value at Risk y del Conditional Value at Risk del costo por la simulación de la sala modificada.

De la simulación de cada crónica j se obtiene el costo total (ct) como:

$$ct(j) = \sum_{i=0}^{N-1} q^i \times cdp(i) + CF(X_{final}) \times q^N$$

Siendo $i = 0 \dots N-1$ los pasos de tiempo de simulación, q un factor de actualización del dinero y $cdp(i)$ el costo de cada paso de tiempo.

Un Operador "no adverso al riesgo" minimizará el valor esperado del costo total y por lo

tanto su función objetivo será $VE = \frac{\sum_{i=0}^{NCrónicas} ct(i)}{NCrónicas}$ que es el estimador disponible del valor

esperado del costo total.

Un Operador que sea "totalmente adverso al riesgo" mirará más que el valor esperado, los valores máximos de $ct[j]$. Como medida de estos valores máximos se calculan dos: “ VaR ” y “ $CVaR$ ”.

“ VaR ” (Value At Risk) es el valor que es excedido por el 5% de los valores $ct[j]$ y “ $CVaR$ ” (Conditional Value At Risk) es el promedio de los valores de $ct[j]$ que exceden el valor “ VaR ”. El umbral de probabilidad de 5% se fijó en la programación de OddFace por ser el valor habitual en la consideración del riesgo.

La función Objetivo a minimizar de un problema OddFace se obtiene entonces como:

$$f_Objetivo = ro_VE \cdot VE(ct) + ro_VaR \cdot VaR(ct) + ro_CVaR \cdot CVaR(ct)$$

Los factores “ ro_VE ”, “ ro_VaR ” y “ ro_CVaR ”, permiten ser más o menos adversos al riesgo según que “ ro_VE ” sea más cercano a CERO o a UNO respectivamente. La suma “ ro_VaR ” + “ ro_CVaR ” debe ser igual a $(1 - ro_VE)$ para mantener la medida de aversión al riesgo entre CERO y UNO. Cuánto del valor $(1 - ro_VE)$ se asigna a “ ro_VaR ” y cuánto a “ ro_CVaR ”, depende del criterio de medida del riesgo que se quiera tener.

Generalmente la medida del riesgo por “ $CVaR$ ” es más estable y representa mejor los peores casos que la mediada por “ VaR ”, por lo cual parece razonable considerar “ ro_CVaR ” = $(1 - ro_VE)$ y “ ro_VaR ” = 0, pero se deja la posibilidad de realizar diferentes combinaciones por tratarse de una plataforma con un fuerte enfoque en la investigación.

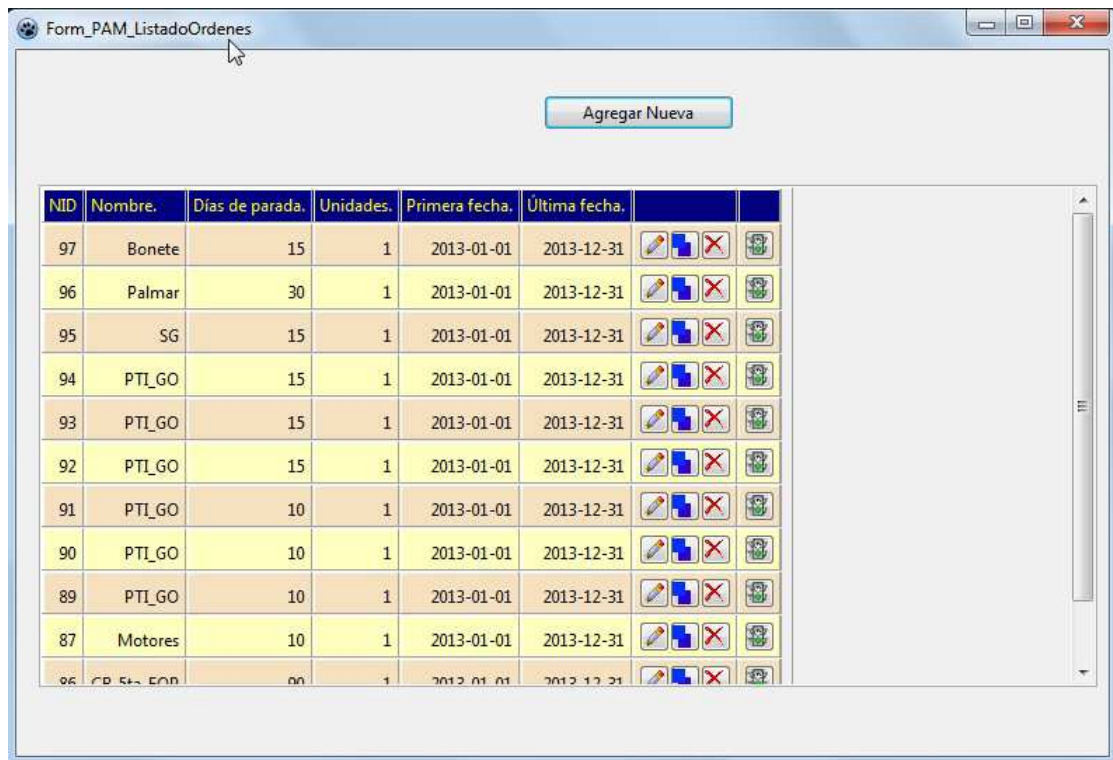
- N crónicas por vez: Es la cantidad de crónicas utilizadas en cada simulación.
- N discretización del histograma: Es la cantidad de puntos utilizados para representar el histograma de $f(x)$. (En esta primera versión este número debe ser igual a “N crónicas”). La idea es que pueda ser diferente, pero por simplicidad de esta primera implementación se

fijan iguales.

- Semilla madre: Es la semilla aleatoria, que se utilizará para inicializar todo el problema. La semilla aleatoria se impone con un valor igual a 30 + cantidad de evaluaciones; de esta forma ocurrirá siempre que la primera evaluación tiene semilla 31, la segunda tiene 32 y así sucesivamente.
- ro_GA, ro_EG y ro_MJ: Son las probabilidades de que actúe el mecanismo de propuesta de nuevo punto de exploración por el algoritmo genético (GA), por estimación del gradiente (EG) o que se repita la evaluación sobre uno de los puntos mejores ya evaluados con el fin de mejorar su estimación (MJ). En esta primera implementación están disponibles solamente las posibilidades GA y MJ. La implementación de EG está sin terminar. Por ejemplo, en la Figura 3 con probabilidad 95% se sugiere el siguiente punto a explorar usando el algoritmo genético y con probabilidad 5% se mejorará la evaluación de uno de los puntos seleccionados "como mejores". Para el criterio de selección de "los mejores" se utiliza el mecanismo de selección del algoritmo genético.
- GA_premio_exito: Este parámetro regula el comportamiento de selección del algoritmo genético. En cada paso, el explorador selecciona dos puntos como los progenitores para cruzarlos y proponer un nuevo punto. Para la selección de los progenitores parte de una lista ordenada de todos los puntos evaluados, ordenados por orden decreciente de mérito (creciente del valor de "f_Objetivo"). Con probabilidad "GA_premio_exito", se selecciona al primer individuo, y con (1-GA_premio_exito) no. Si no se selecciona el primero, se repite el mismo procedimiento con la lista quitando el primero y así sucesivamente. Esto lleva a que se seleccionen los mejores con una probabilidad máxima para el más exitoso y que decae exponencialmente en la medida que se recorre la lista del más al menos exitoso.
- GA_prob_mutación: Luego de seleccionados los progenitores y cruzados por simple combinación azarosa de sus genes, se aplica el mecanismo de "mutación" que invierte bits de la cadena binaria que representa el ADN del individuo. "GA_prob_mutación", es la probabilidad de mutación de un bit aislado. En la medida que un problema tenga una cadena de ADN más larga, la probabilidad de que ocurra alguna mutación es mayor para igual parámetro "GA_prob_mutación".
- Descripción del problema: Este cuadro de texto se utiliza sólo para insertar una descripción del problema que facilite entender la información del SBD.
- Tipo de Problema: Este combo de selección permite determinar el "tipo de problema" y posibilita entonces editar con el botón "Información específica del tipo" el resto de la información del problema. En esta primera implementación, los tipos posibles son "base", "PIGSimSEE" y "PAMSimSEE". El tipo "base" requiere más información y se incluye sólo por disponer de un tipo de problema que se pudiera definir sobre la información de una sala SimSEE sin necesidad de especificar más parámetros. En la actualidad no hay ningún explorador implementado para este tipo de problema.
 - El tipo "PIGSimSEE" implica que se trata de un problema de Planificación de Inversiones de Generación y hay que editar la información adicional para este tipo de problemas que significa ingresar la lista de tecnologías posibles de introducir y los parámetros de costos de inversión y ventanas de fechas de decisión posibles.
 - El tipo "PAMSimSEE" es para los problemas de optimización de ventanas de mantenimiento e implica la edición de la información del conjunto de centrales para las que se quiere considerar el mantenimiento y los parámetros de dichos mantenimientos.

- Editar Información específica del tipo: Con este botón se seleccionan los actores que van a ser puestos en mantenimiento y las características de los mismos.

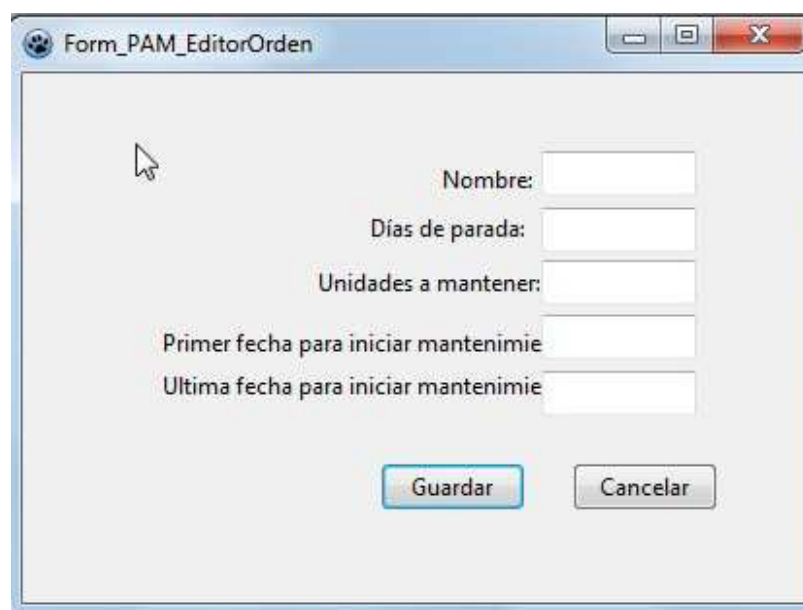
En la figura 4 se ve una lista de actores y los parámetros seleccionados para los mismos.



NID	Nombre.	Días de parada.	Unidades.	Primera fecha.	Última fecha.		
97	Bonete	15	1	2013-01-01	2013-12-31		
96	Palmar	30	1	2013-01-01	2013-12-31		
95	SG	15	1	2013-01-01	2013-12-31		
94	PTI_GO	15	1	2013-01-01	2013-12-31		
93	PTI_GO	15	1	2013-01-01	2013-12-31		
92	PTI_GO	15	1	2013-01-01	2013-12-31		
91	PTI_GO	10	1	2013-01-01	2013-12-31		
90	PTI_GO	10	1	2013-01-01	2013-12-31		
89	PTI_GO	10	1	2013-01-01	2013-12-31		
87	Motores	10	1	2013-01-01	2013-12-31		
86	CB 5to EOB	00	1	2013-01-01	2013-12-31		

Figura 4

Para agregar un nuevo actor se debe seleccionar Agregar Nueva y luego introducir los parámetros que aparecen a continuación en la Figura 5:



Form_PAM_EditorOrden

Nombre:

Días de parada:

Unidades a mantener:

Primer fecha para iniciar mantenimiento:

Última fecha para iniciar mantenimiento:

Figura 5

El nombre del actor debe ser el mismo que el que aparece en la sala utilizada. Se debe especificar la cantidad de días de parada del actor y la cantidad de unidades del actor que estarán en mantenimiento.

El formato de las fechas es año-mes-día (aaaa-mm-dd) y las mismas deben estar contenidas dentro del período de simulación de la sala. Se debe tener en cuenta que entre el final de la simulación y el final de la optimización haya un “tiempo de guardia” considerado para que se establezca el valor de la función de costo calculada, ya que la inicialización de los costos futuros de la sala debe ser cero.

Luego de “Guardar” aparecerá el actor, como se muestra en la figura 4, con un NID asociado que es un número de identificación asignado por defecto a cada actor. Los botones que aparecen a la derecha tienen las mismas funcionalidades que los de la figura 2. Se debe poner el botón “semáforo” en verde para que el mantenimiento que se quiera evaluar quede activado. Para que el mantenimiento de un actor no se evalúe, se debe dejar el semáforo en rojo.

- Sala: En este cuadro debe introducirse el camino completo al archivo de sala SimSEE que se utilizará como base en el problema. El botón "Buscar" permite explorar la computadora para ubicar el archivo de la sala. Una vez ubicado, el botón "Empaquetar" crea un archivo comprimido con el archivo de la sala y con todos los archivos que sean necesarios para su ejecución y los sube al SBD para que los exploradores puedan bajar el "paquete" y modificar la sala para hacer las simulaciones necesarias. Una vez que se haya empaquetado no es necesario repetir el procedimiento si se cambia cualquiera de los parámetros descriptos.
- Tipo de Codificación: OddFace puede aplicar el algoritmo genético con diferentes formas de codificación de los parámetros. En estos cuadros se puede especificar qué tipo de codificación se quiere usar, y para esto se debe ingresar un valor entre cero y uno en cada casilla de manera de elegir o combinar las opciones. La suma de todos los valores debe ser 1.

Actualmente se encuentran disponibles BINARY, GRAY, UNARY y fosil_agosto 2011.

BINARY: cada parámetro entero de optimización X / ($X \geq X_{min}$). ($X \leq X_{max}$) como la codificación binaria del número ($X - X_{min}$).

Si el parámetro es real, se especifica cómo Y / ($Y \geq Y_{min}$). ($Y \leq Y_{max}$) y la cantidad de bits a utilizar para la codificación del valor ($Y - Y_{min}$).

En el caso del tipo de problema PAM, todos los parámetros son enteros.

La operación de cruzamiento del algoritmo genético implica recorrer las cadenas binarias y tomar con probabilidad 50% el bit del progenitor A y con probabilidad 50% el bit del progenitor B. Sobre la cadena así obtenida, se aplica posteriormente la operación de mutación que implica alterar algunos bits con una probabilidad dada. Dos individuos que son "ceranos" desde el punto de vista del valor de los parámetros, pueden resultar lejanos en términos de las diferencias en las cadenas binarias que representan el ADN. Por ejemplo, la cadena binaria 0001 y 1110 pueden estar representando el valor 16 y 15 respectivamente para un parámetro que puede tomar valores entre 0 y 31. El cruzamiento de 0001 y 1110 puede dar lugar por ejemplo a 0000 o a 1111 (cero y 31) con lo cual, se puede apreciar que cruzar dos individuos cercanos desde el punto de vista de los parámetros lleva a individuos lejanos.

GRAY: dos números consecutivos tienen diferencia de solo un bit en la codificación.

Este código se utiliza para intentar reflejar la cercanía en las cadenas binarias, se utiliza el código GRAY. Cuando se utiliza la codificación GRAY, antes de realizar el cruzamiento y las mutaciones, el ADN se transforma de su representación BINARIA a código GRAY, se realiza entonces el cruzamiento y la mutación (generando así individuos que guardan relación con sus progenitores) y luego se transforma el resultado de código GRAY a BINARIO simple.

UNARY: sería el equivalente a suponer que el valor del parámetro está representado por la cantidad de 1 (unos) en un conjunto de bits. Por ejemplo, para representar números de 0 a 31 se podría suponer que se utilizan conjuntos de 310 bits (es sólo un ejemplo) así, la cantidad de 1 (unos) en los 310 bits, dividida 10 es el valor del parámetro. Con este tipo de codificación, al realizar la operación de cruzamiento se obtiene en valor esperado un promedio entre los dos conjuntos de 1 (unos) con dispersión alrededor de dicho valor. La implementación realizada en OddFace no es exactamente una representación por conjunto de Ceros y Unos, sino que se optó por operar directamente sobre los parámetros. Dados dos individuos con parámetros X_1 y X_2 , se genera por cruzamiento, el individuo $X = X_1 + (X_2 - X_1) \cdot (2 \cdot \text{random} - 0.5)$ (limitando X al rango $X_{\min}$, $X_{\max}$ que corresponda).

ro_fosil_agosto 2011: En esta versión de OddFace no está implementado. Se debe dejar en cero hasta que esté implementada este tipo de codificación.

iv. Consultas sobre el historial de evaluaciones.

El historial de evaluaciones se almacena en una tabla en la base de datos compartida por todos los nodos de cálculo.

Para consultar el historial se puede utilizar el botón  del formulario que despliega el listado de problemas, como se mostró en la Figura 2. Al presionar ese botón se despliega un formulario de consultas, como se muestra en la Figura 6, que permite la edición de los parámetros de una consulta SQL a realizar sobre la tabla del historial de evaluaciones del problema.

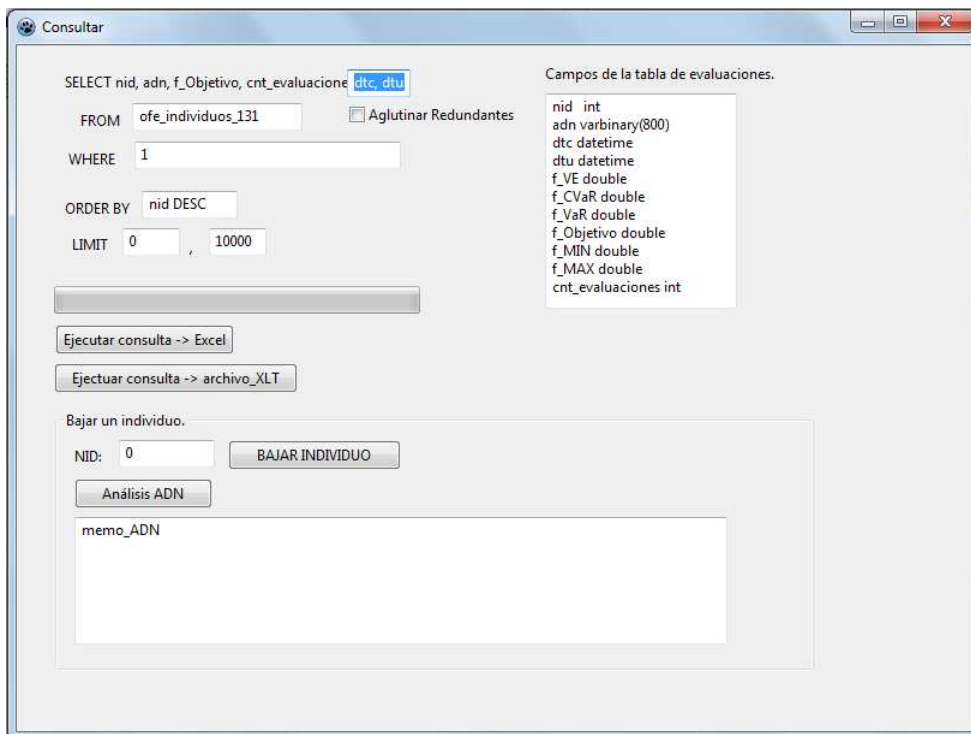


Figura 6

Este formulario selecciona por defecto los campos:

- "nid", número identificador único del punto evaluado.
- "adn", es el adn del individuo (número binario resultante de los cruzamientos y mutaciones) y se muestra en código hexadecimal.
- "f_Objetivo", valor de la función objetivo en el punto evaluado.
- "dtc", fecha y hora de la primer evaluación del punto.
- "dtu", fecha y hora de la última evaluación del punto.
- "cnt_evaluaciones", cantidad de evaluaciones realizadas del punto.

También se pueden seleccionar de la tabla que aparece a la derecha otros campos, se deben escribir en la casilla en donde aparecen escrito "dtc" y "dtu".

El casillero FROM tiene el nombre de la tabla del historial del problema y no puede ser editado.

El casillero WHERE tiene el "filtro" y por defecto tiene 1 lo que significa que todos los registros serán seleccionados. Por ejemplo si se pusiera "cnt_evaluaciones" > 4 en el casillero WHERE, sólo se seleccionarán aquellos puntos que cuentan al menos con 5 evaluaciones.

El casillero ORDER BY indica el orden en el que se ordenarán los resultados. En el ejemplo, están ordenados por orden decreciente del número identificador único (nid DESC), lo que significa que aparecerán por orden inverso de creación, o sea, al inicio los últimos creados.

Los casilleros a la derecha de LIMIT fijan el desplazamiento desde el primer registro y la cantidad de registros a bajar. En el ejemplo de la figura los valores son 0 (cero) desplazamiento y 10000 (diez mil) registros.

La opción Aglutinar redundantes permite agrupar individuos que aparecen como diferentes pero que en realidad decodifican la misma sala. Para esto se calculan las "f_Objetivo" ponderando cada uno por la cantidad de evaluaciones. Cuando se utilice esta opción se recomienda cambiar en LIMIT de 10000 a 10000000 ya que al aglutinar los resultados se resumen muchas filas de Excel.

Una vez establecidos los parámetros que permiten formar la consulta SQL, presionando el botón "Ejecutar consulta" se envía la consulta al servidor y se recibirán los registros que resulten seleccionados y son desplegados en un libro de Excel (Ejecutar consulta ->Excel) o en un archivo .xlt (Ejecutar consulta -> archivo_XLT).

En los registros desplegados en la planilla que resulta de la consulta elegida, se presenta la cantidad de días/etapas posteriores a la fecha de inicio de la primera etapa, que sumados a esa fecha se obtiene la fecha en la cual eligió iniciar el mantenimiento para cada actor.

El cuadro "Bajar individuo" le permite especificar el "nid" de un punto de evaluación (individuo) para su análisis. Presionando dicho botón, se baja el ADN del punto evaluado y se crea la sala SimSEE correspondiente, usando para ello el algoritmo de decodificación de ADN asociado al problema. Para ello, en la computadora en la que se está ejecutando "OddFace_prepare" debe estar instalada también una versión de SimSEE asociado al OddFace, para que se pueda ejecutar sobre el ADN bajado y crear la sala. Cuando se realiza esta consulta se debe verificar en la pantalla de consola de Oddface que se haya creado la sala (sale un mensaje como el de la figura 7) y luego presionar "Enter" en esta pantalla para volver a activar la ventana de consultas.


```

Cambiando el directorio a : \simsee\tmp_rundir\12
***** bajo archivos *****
***** fin copia de archivos *****
f.version: 65
uniformizar_promediando:
Sala creada, NID: 20078, archivo: \simsee\tmp_rundir\12\sala paso semanal 2012-2
015 DC_oddface_.ese
-

```

Figura 7

La sala se descarga en la carpeta donde se encuentra instalado el SimSEE, en la subcarpeta tmp_rundir\12 y como nombre, le pone por defecto el nombre que tiene la sala que se empaquetó, seguido de la palabra _oddface_.

El botón "Análisis ADN" permite ver la cadena binaria de un individuo.

Para esto se debe ingresar el NID del individuo que se quiera ver y luego presionar ese botón. Puede ser necesario hacer "click" en el cuadro de texto para que aparezcan los resultados.

En primer lugar se muestra la cadena tal cual está en la base de datos, luego se muestran los valores decodificados y después el ADN vuelto a codificar, pero "limpiando los bits redundantes".

Se obtiene un representante principal del conjunto de ADNs que decodificados tienen el mismo resultado.

En la figura 8 se muestra un caso en el cual nbits_Justo: 108, nbits_Resto:4. Eso es la cantidad de bits mínima necesaria y el resto de bits que le sobran a las cadenas para completar el múltiplo de 16.

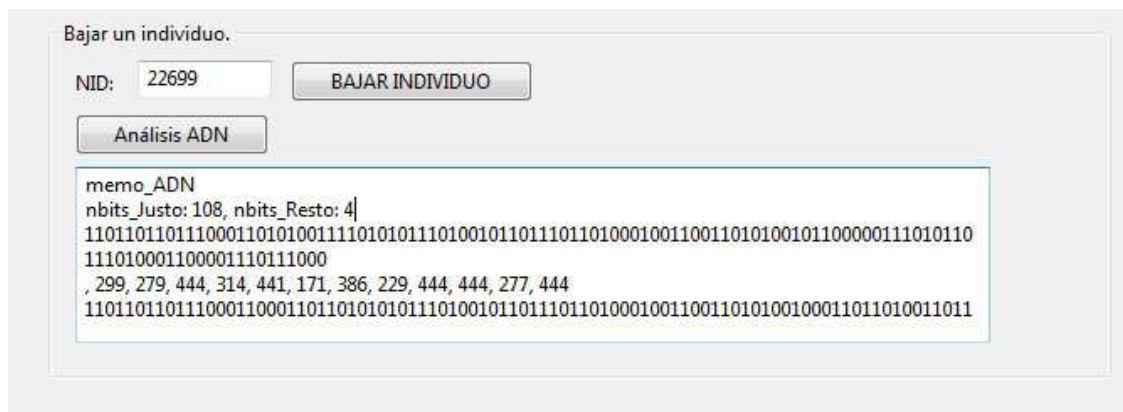


Figura 8

En este ejemplo se codifica en 108 bits, lo que da un espacio de búsqueda bastante grande (2^{108}) y sobran 4 bits, con lo que cada individuo tiene multiplicidad genética de 2^4 , es decir 16 (hay 16 variantes genéticas que dan el mismo comportamiento, o sea la misma sala). Además de esa multiplicidad, puede ser (dependiendo de los rangos de cada variable) que los máximos de los rangos tengan una multiplicidad mayor.

Aglutinar resultados encuentra los clones y resume los resultados.

Además también se pueden ver los siguientes resultados en la ventana de la pantalla de consulta de la figura 8:

memo_ADN

nbits_Justo: 108, nbits_Resto: 4

11011011011100011010100111101010111010010110111011010001001100110101001011000001110101101
11010001100001110111000

, 299, 279, 444, 314, 441, 171, 386, 229, 444, 444, 277, 444

11011011011100011000110110101011101001011011101101000100110011010100100011011010011011
011010001100011011010000

1101101101110001100011011010101011101001011011101101000100110011010100100011011010011011
011010001100011011010000

nid: 22699

cnt_evaluaciones: 3

f_VE: 1669.40833330102

f_VaR: 2030.39058815261

f_CVaR: 2030.39058815261

f_Objetivo: 1723.55567152876

f_Histo: [100| 1333.39729712299, 1356.65922658178, 1390.2637104659, 1404.49826297914,
1431.80404874466, 1437.62312717541, 1447.91226058092, 1457.34963916047, 1471.92162104701,
1474.98023323004, 1480.87443826666, 1484.24643715249, 1492.86950794142, 1500.18171494362,
1504.36829466843, 1509.83842110286, 1512.15684116083, 1513.93751311059, 1519.44275244992,
1523.45588617732, 1529.26238832754, 1534.65137392632, 1537.3297393572, 1539.27153374993,
1546.36963896319, 1550.18792580293, 1554.08754680869, 1563.24460089064, 1571.14261378092,
1577.57249674116, 1581.87728093536, 1585.62846368115, 1586.34764126481, 1593.89577688978,
1595.29458641297, 1600.26421083858, 1603.20092314468, 1612.01274474958, 1617.29980833009,
1621.82137785702, 1624.45125881176, 1629.44765421859, 1633.96668552046, 1639.02890185433,
1641.00789675266, 1644.27499815767, 1647.50979589226, 1658.79887189074, 1661.71523877671,
1667.29136797509, 1668.72845458103, 1675.000910312, 1677.40286030755, 1681.58351878432,
1686.7848208108, 1695.80987061204, 1699.26814934347, 1705.05702181344, 1708.95703807952,
1713.27782965892, 1716.60132932839, 1724.67996205101, 1730.02719930702, 1733.77146029194,
1741.20305254308, 1745.96595731175, 1750.32565024517, 1753.78747768884, 1758.06749813682,
1760.96427388653, 1763.993127831, 1766.49463249997, 1775.35206186535, 1779.8002887743,
1785.36731137679, 1790.08113073537, 1791.61275302002, 1797.3138892075, 1801.34779777944,
1803.16279004825, 1808.61581227366, 1815.00855546131, 1819.28027532366, 1823.5063011379,
1833.33026748623, 1836.17832712165, 1847.50487148093, 1850.71959948881, 1861.4889662272,
1870.59358497507, 1875.97412668879, 1878.66882383836, 1886.19991189041, 1889.89865971968,
1895.69670078905, 1908.97055734642, 1919.20032180107, 1928.88372270669, 1982.89666159574,
2030.39058815261]

f_Histo es el vector del promedio de la función de costo de las 100 crónicas simuladas. Estos valores son promedio de la función de costo de las evaluaciones del individuo.

Capítulo VII. OddFace-PAM1.

Milena Gurín y Daniel Cohn.

Diciembre 2012

Montevideo – Uruguay.

A. Introducción.

A continuación se presentan resultados obtenidos del estudio de un caso concreto de aplicación del módulo “OddFace_PAM”.

Los siguientes caso fueron hechos con una versión anterior de Oddface pero se presentan dos casos que se realizaron con la última versión.

Los principales cambios son:

- a) La semilla aleatoria se impone con un valor igual a 30 + cantidad de evaluaciones; de esta forma ocurrirá siempre que la primera evaluación tiene semilla 31, la segunda tiene 32 y así sucesivamente.
- b) Limpieza del resto de los bits en la codificación (por los cajones en múltiplos de 16).
- c) Ajuste de la redundancia binaria para representar las restricciones de caja de los parámetros.
- d) Se inicializa el valor de la probabilidad de excedencia (PE) para la estimación del VaR y del CVaR.

Breve descripción de la sala utilizada:

Horizonte de Optimización: 13/10/2012 al 31/12/2015.

Horizonte de Simulación: 13/10/2012 al 31/12/2013.

Paso de tiempo semanal (168 horas).

Nº Poste	1	2	3	4
Duración (horas)	5	30	91	42

Para representar los aportes hídricos se utilizó una variable aleatoria “Lluvias” compuesta por un sintetizador CEGH (CEGH20120712_BPS50) y para el viento se utilizó el sintetizador CEGH “vientos_uy”.

La demanda se modeló como una demanda detallada horaria y la representación de los escalones de falla fue la siguiente:

Profundidad (%)	2	5	7.5	85.5
Costo(U\$S/MWh)	291	400	4800	8000

Generadores:

Eólicos:

Se modeló el parque eólico existente y también las incorporaciones para los años futuros.

Térmicos:

Dentro de esta categoría se encuentra el parque generador térmico de UTE existente (Central Batlle, Punta del Tigre, CTR, Motores, Turbina de Maldonado y la generación arrendada de emergencia instalada en Punta del Tigre, MVA y MVB) y también el ciclo combinado a instalarse en el futuro. También se incluyó los generadores de biomasa existentes que están conectados a la red de distribución, UPM y las futuras incorporaciones al sistema de las licitaciones de UTE.

Hidráulicos:

Se incluyó las tres centrales del Río Negro (Bonete, Baygorria y Palmar) y la central de Salto Grande.

La central Bonete se modeló como una central con embalse, mientras las demás se modelaron como centrales de pasada.

Internacional y Otros:

Se representó la importación a través de la conversora de Rivera y la importación de origen Argentina en modalidad contingente.

También se representó la posibilidad de exportar.

i. Versiones Oddface:

La versión de Oddface utilizada fue: oddface_prepare_Aglutinante_v2, aunque se presentan dos casos con la versión OddFace_v11_GEA_binarios.

En esta versión no se respetaba la inicialización de la semilla madre.

Tampoco se inicializaba el valor de probabilidad de excedencia (PE) para la estimación del VaR y del CVaR. Por lo tanto $VaR = CVaR = f_MAX$ en los individuos que tienen una sola evaluación. Entonces, las corridas realizadas que tenga $ro_VaR > 0$ y/o $ro_CVaR > 0$ deben considerarse con una aversión al riesgo con el riesgo medido con $PE = 0\%$.

Además de estos cambios la nueva versión de Oddface (OddFace_v11_GEA_binarios) que se describe en este manual también realiza una limpieza del resto de los bits en la codificación y realiza un ajuste de la redundancia binaria para representar restricciones de caja de los parámetros.

B. Casos de estudio (con Aglutinar resultados y versión oddface_prepare_Aglutinante_v2):**i. Parámetros generales para todos los casos (no son modificados):**

Fecha de inicio primera etapa: 13/10/ 2012

Días/etapa: 1

Cantidad de etapas: 1200

N crónicas por vez: 100

N discretización del histograma: 100

Semilla madre: 31

ro_EG: 0

Tipo de problema: PAMSimSEE

Información específica del tipo:

Nombre actor	Días de parada	Unidades	Primera fecha	Ultima fecha
Bonete	15	1	01/01/2013	31/12/2013
Palmar	30	1	01/01/2013	31/12/2013
SG	15	1	01/01/2013	31/12/2013
PTI_GO	15	1	01/01/2013	31/12/2013
PTI_GO	15	1	01/01/2013	31/12/2013
PTI_GO	15	1	01/01/2013	31/12/2013
PTI_GO	10	1	01/01/2013	31/12/2013
PTI_GO	10	1	01/01/2013	31/12/2013
PTI_GO	10	1	01/01/2013	31/12/2013
Motores	10	1	01/01/2013	31/12/2013
CB-6ta-FOP	120	1	01/01/2013	31/12/2013
CB-5ta-FOP	90	1	01/01/2013	31/12/2013

Sala: “sala paso semanal 2012-2015 DC.ese”

ro_GRAY:0

ro_fosil_agosto2011:0

ii. CASO 1

NID_Problema:102

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.05

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14A_Grupo PAM VE=0.85 VaR=0.15

Tipo de codificación:

ro_BINARY:1

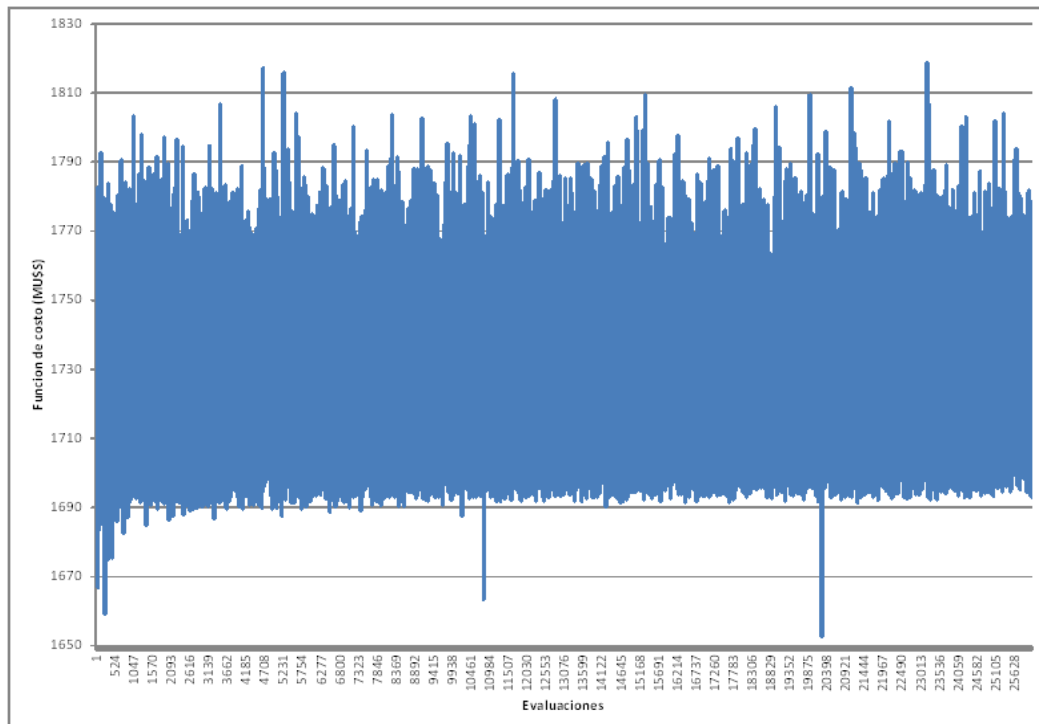
ro_UNARY:0

Resultados obtenidos:

Se realizaron 26119 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se observa que la función de costo varía entre 1820 y 1659 MU\$S aproximadamente, presentándose un mínimo de 1659 MU\$S en el individuo 40847, que fue evaluado una sola vez. Se presenta a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

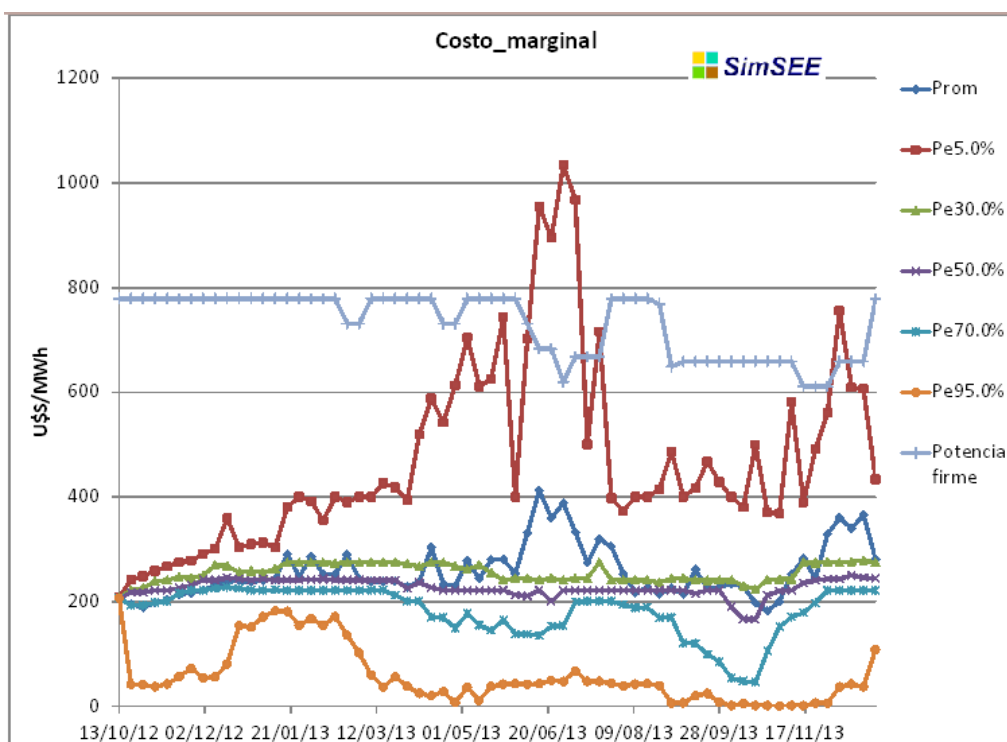
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	29/08/2013
CB-5ta-FOP	31/12/2013
Motores	21/08/2013
PTI_GO	31/12/2013
PTI_GO	25/02/2013
PTI_GO	22/04/2013
PTI_GO	08/06/2013
PTI_GO	19/11/2013
PTI_GO	19/06/2013
SG	31/12/2013
Palmar	01/07/2013
Bonete	31/12/2013

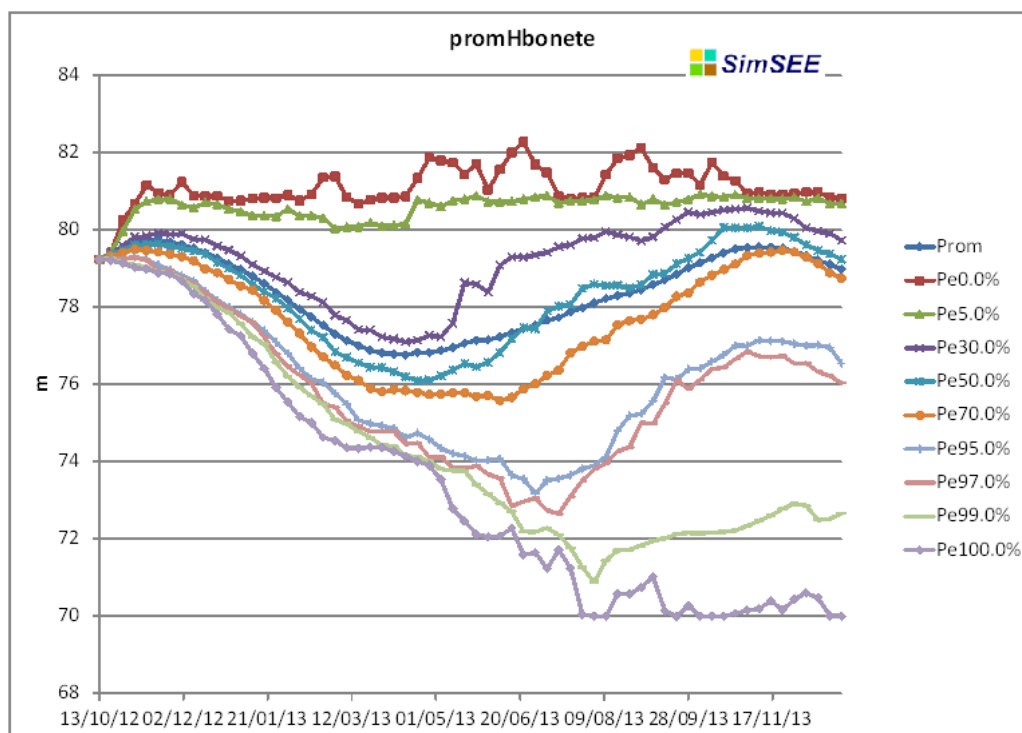
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la cota queda en 78.5 m al final de la simulación.

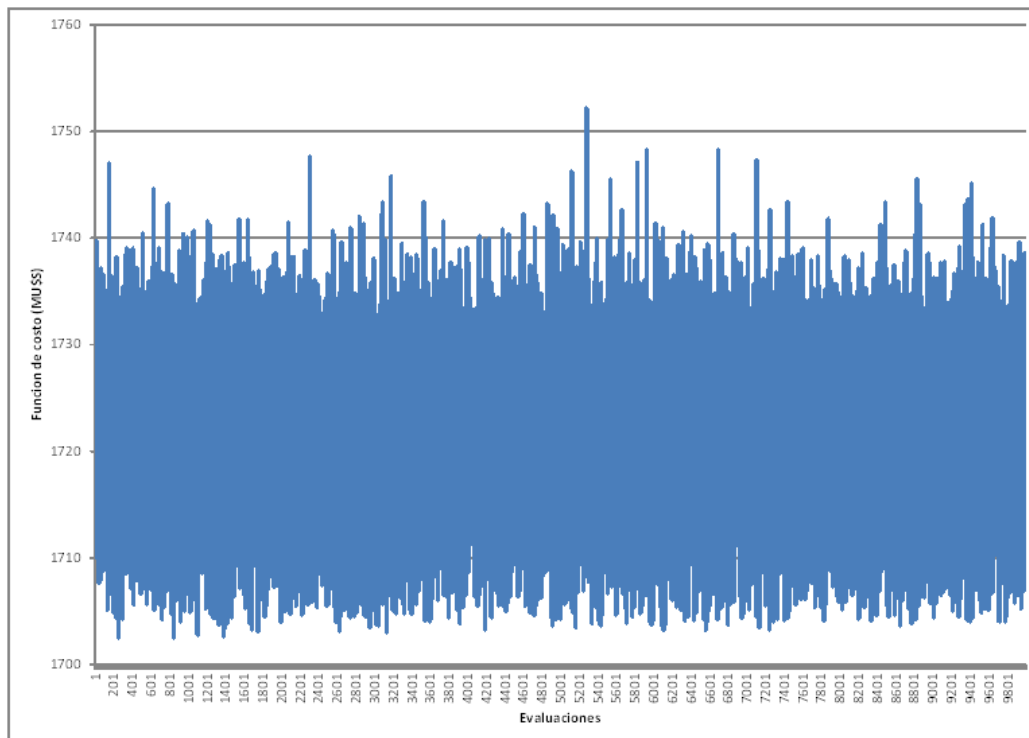
a) Ídem al anterior pero con version OddFace_v11_GEA_binarios

Resultados obtenidos:

Se realizaron 20349 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1750 y 1700 MU\$ presentándose un mínimo de 1702 en el individuo 20078, que fue evaluado una sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

Fechas de ubicación los mantenimientos:

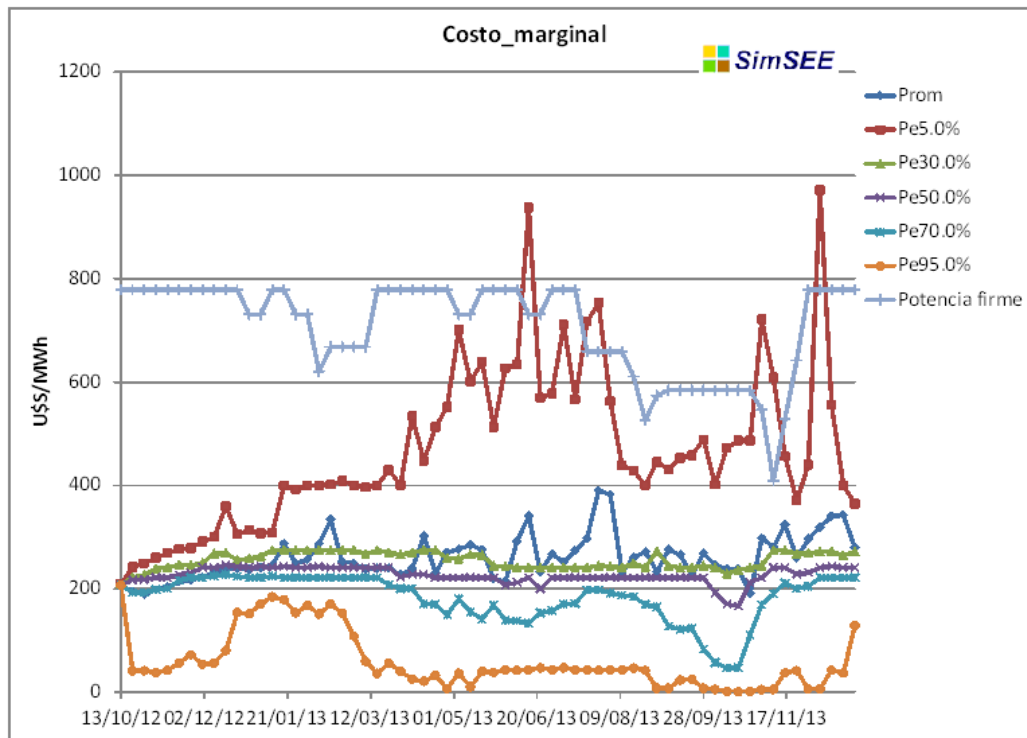
CB-6ta-FOP	23/07/2013
CB-5ta-FOP	25/08/2013
Motores	22/08/2013
PTI_GO	06/05/2013
PTI_GO	01/01/2013
PTI_GO	15/08/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	29/01/2013
SG	10/11/2013
Palmar	13/02/2013
Bonete	01/11/2013

En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.

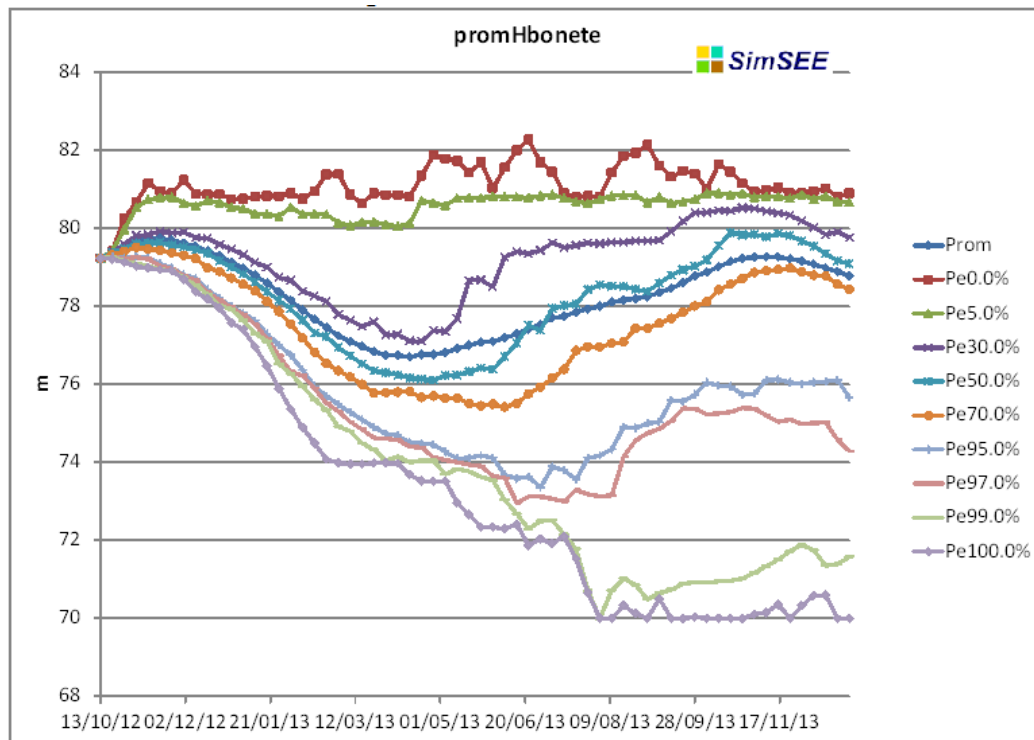


Se observa que la diferencia entre el mínimo y el máximo valor de la función de costo es mucho más pequeña que en el caso realizado con la versión de Oddface que no inicializaba la semilla madre ni la PE para la estimación del VaR y del CVaR.

En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

iii. CASO 2

NID_Problema:117

ro_VE:1

ro_VaR:0

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.05

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14AA_Grupo PAM VE=1 VaR=0

Tipo de codificación:

ro_BINARY:1

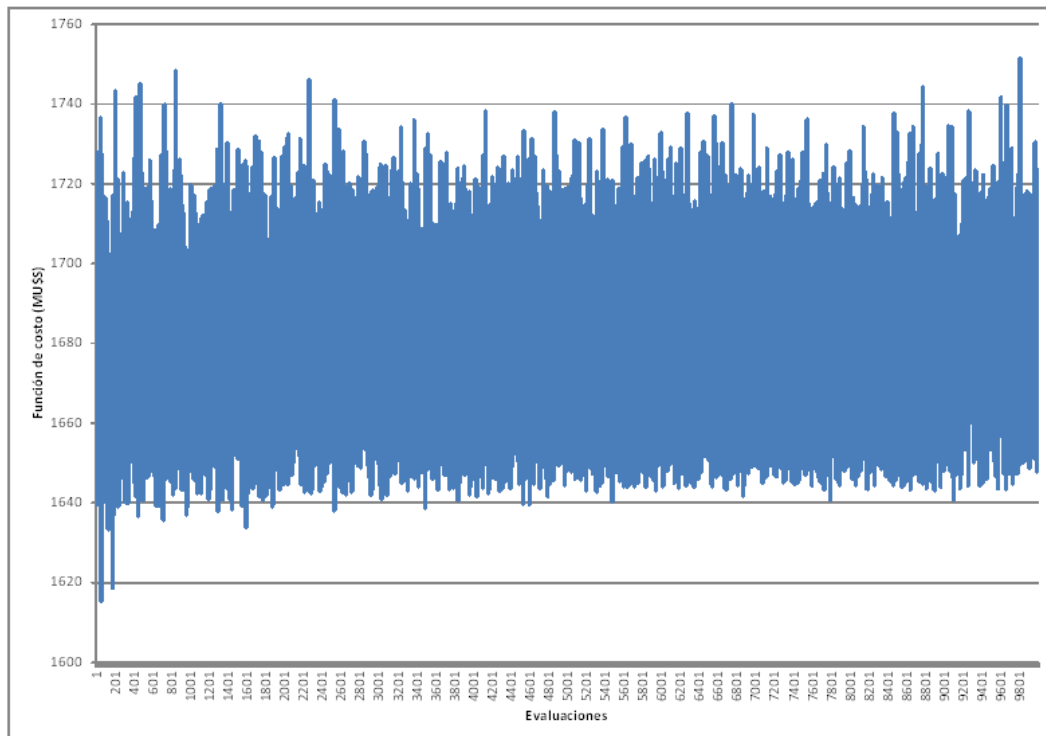
ro_UNARY:0

Resultados obtenidos:

Se realizaron 11266 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1620 y 1750 MU\$.

Se observa un mínimo de 1615 MU\$ en el individuo 21697, que fue evaluado 1 sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este

individuo.

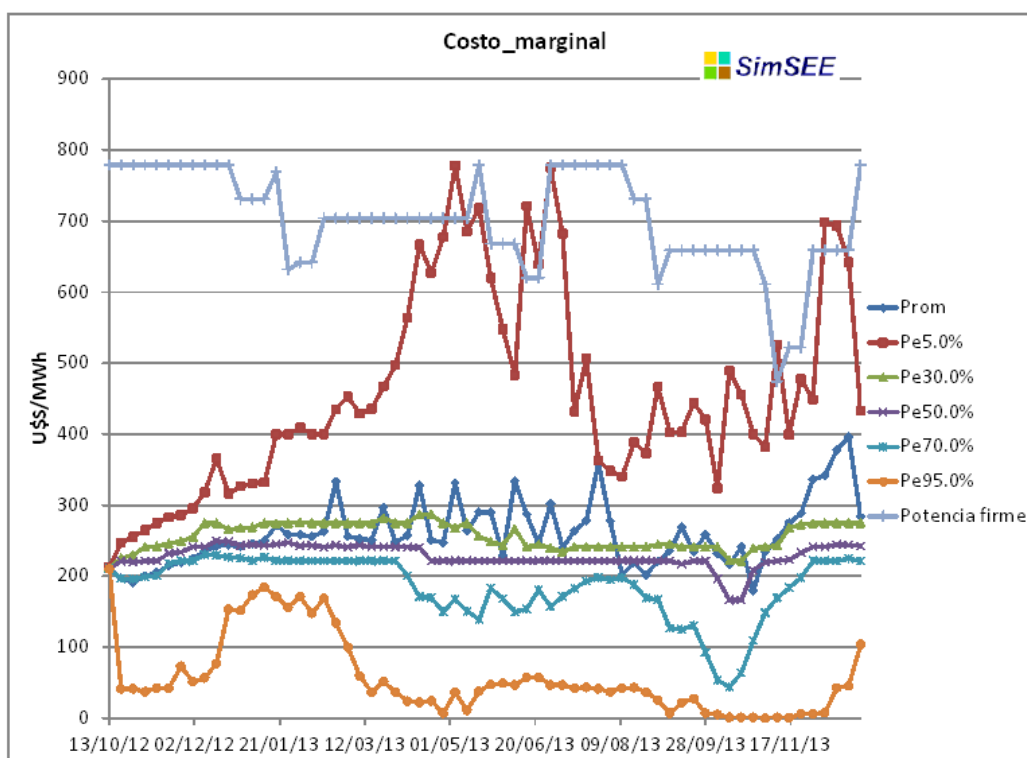
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	28/08/2013
CB-5ta-FOP	19/02/2013
Motores	19/01/2013
PTI_GO	31/12/2013
PTI_GO	05/11/2013
PTI_GO	31/12/2013
PTI_GO	19/08/2013
PTI_GO	31/12/2013
PTI_GO	03/01/2013
SG	25/01/2013
Palmar	24/05/2013
Bonete	31/12/2013

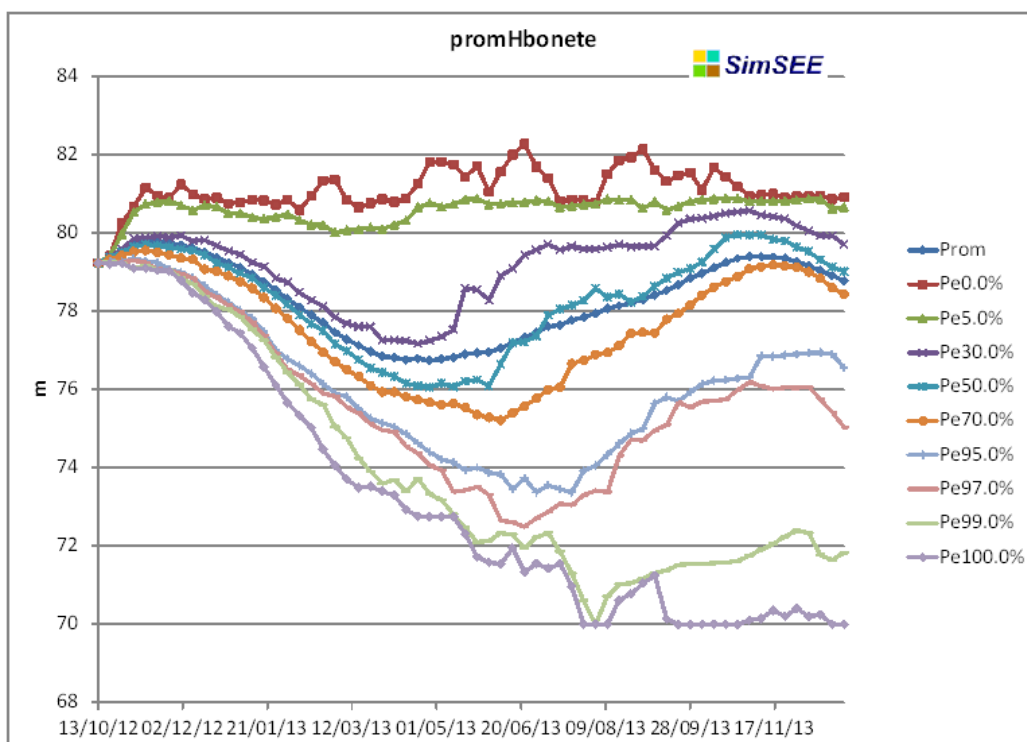
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

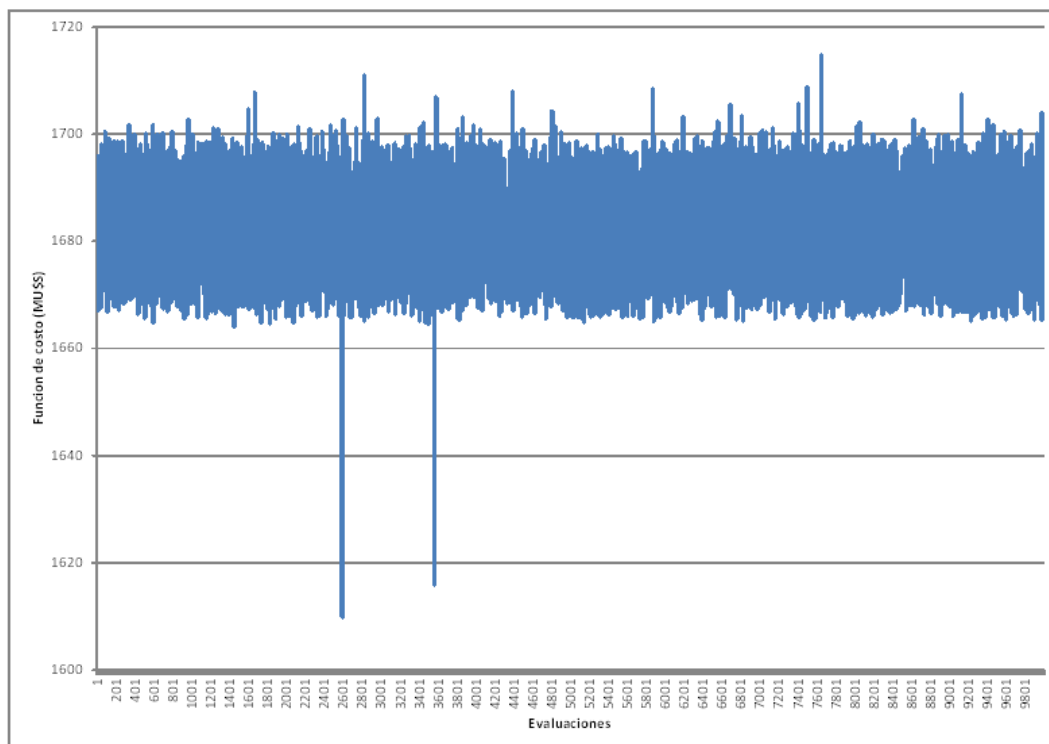
a) *Idem al 117 pero con version OddFace_v11_GEA_binarios*

Resultados obtenidos:

Se realizaron 41117 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1610 y 1720 MU\$. Se observa un mínimo de 1610 MU\$ en el individuo 38639, que fue evaluado 4 veces.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

Fechas de ubicación los mantenimientos:

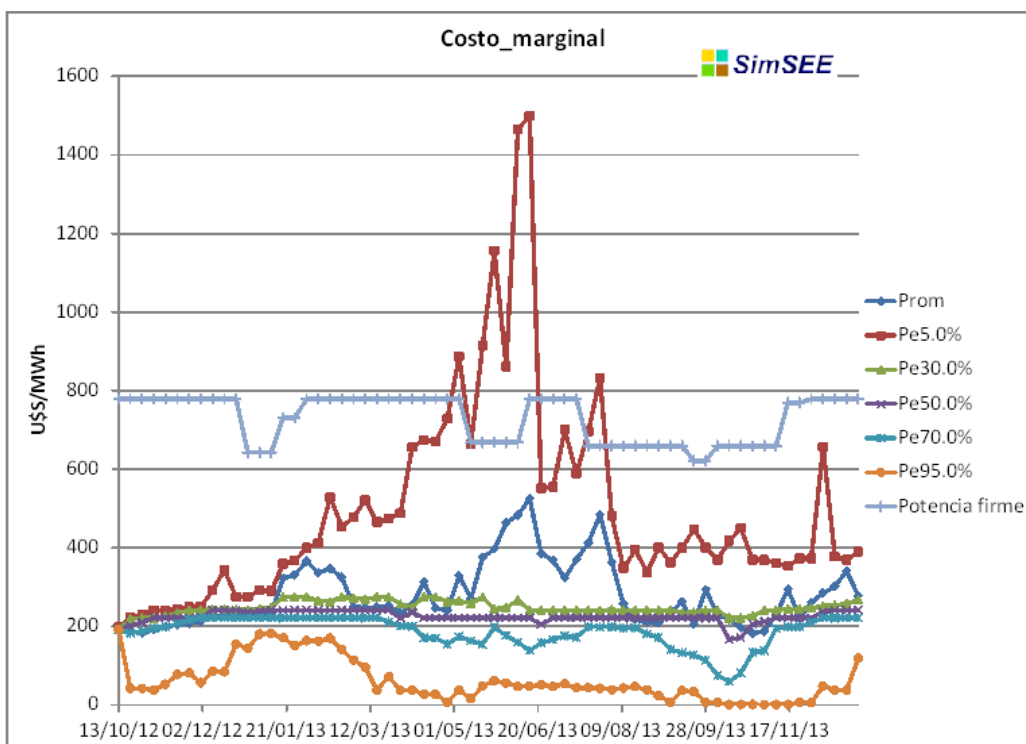
CB-6ta-FOP	22/07/2013
CB-5ta-FOP	31/12/2013
Motores	18/11/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	22/01/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
SG	01/01/2013
Palmar	12/05/2013
Bonete	23/09/2013

En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.

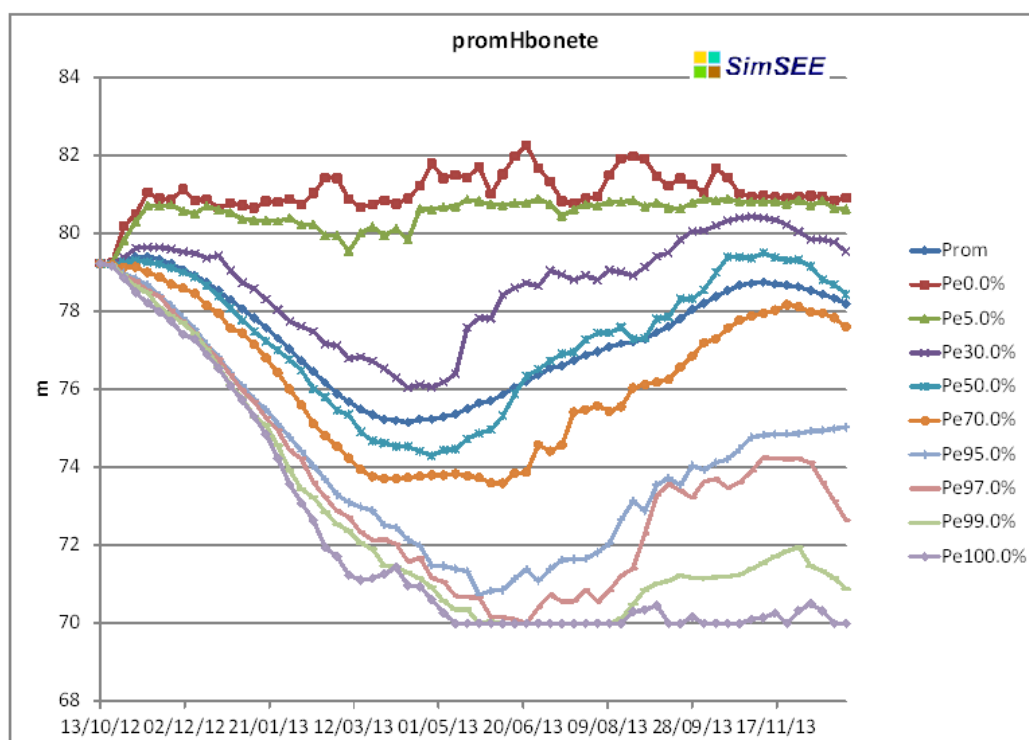


Se observa que la diferencia entre el mínimo y el máximo valor de la función de costo es mucho más pequeña que en el caso realizado con la versión de Oddface que no inicializaba la semilla madre ni la PE para la estimación del VaR y del CVaR.

En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78 m al final de la simulación.

iv. CASO 3

NID_Problema:118

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.1

GA_prob_mutacion:0.05

Descripción del problema: Prueba 14C_Grupo PAM premio_exito=0.10 mutacion=0.05

Tipo de codificación:

ro_BINARY:1

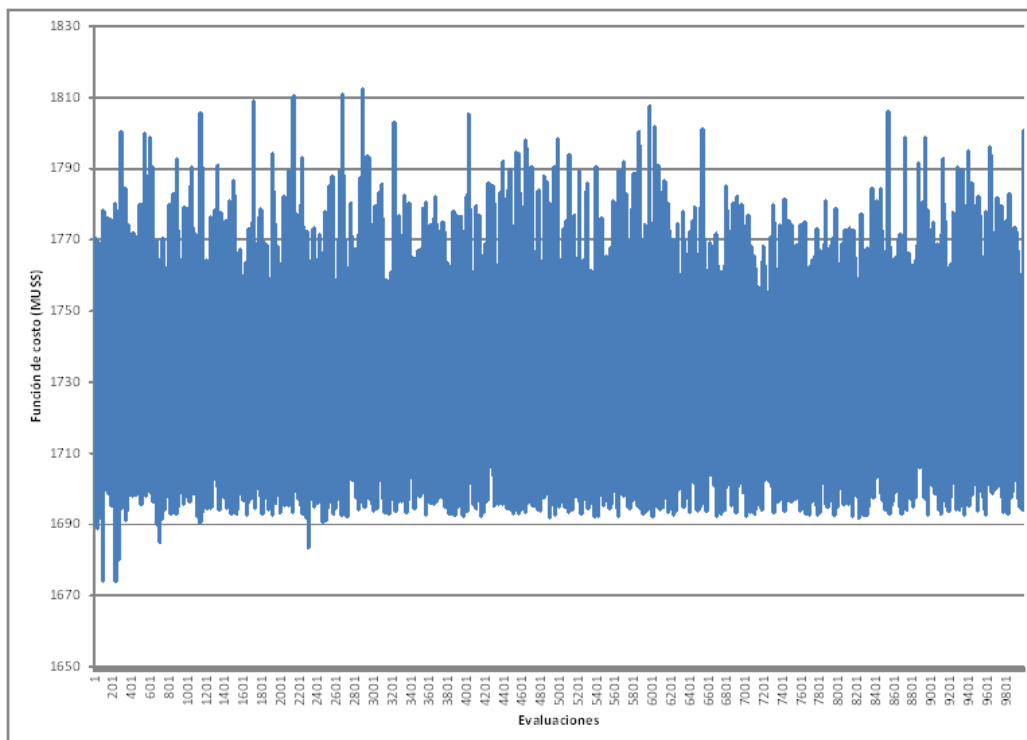
ro_UNARY:0

Resultados obtenidos:

Se realizaron 16472 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1810 y 1670 MU\$ presentándose un mínimo de 1674 MU\$ en el individuo 22690, que fue evaluado 1 sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este

individuo.

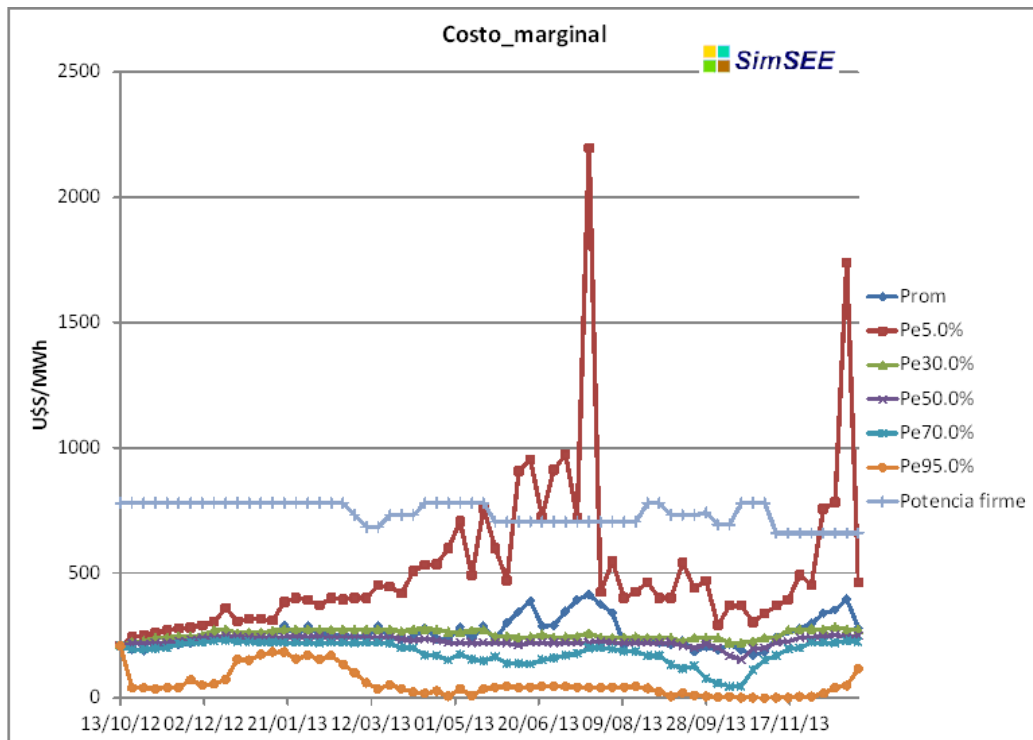
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	13/11/2013
CB-5ta-FOP	28/05/2013
Motores	31/12/2013
PTI_GO	31/12/2013
PTI_GO	27/03/2013
PTI_GO	07/10/2013
PTI_GO	04/03/2013
PTI_GO	10/09/2013
PTI_GO	07/03/2013
SG	31/12/2013
Palmar	31/12/2013
Bonete	30/09/2013

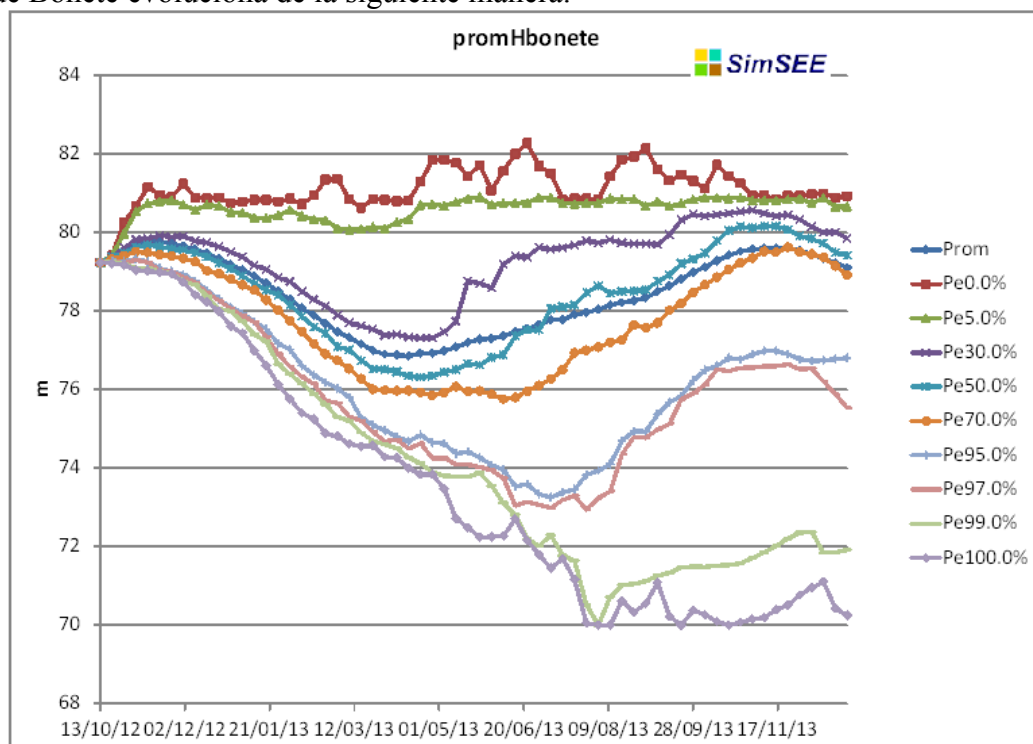
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

v. CASO 4

NID_Problema:120

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.05

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14DA_Grupo PAM BINARY=0 GRAY=0 UNARY=1

Tipo de codificación:

ro_BINARY:0

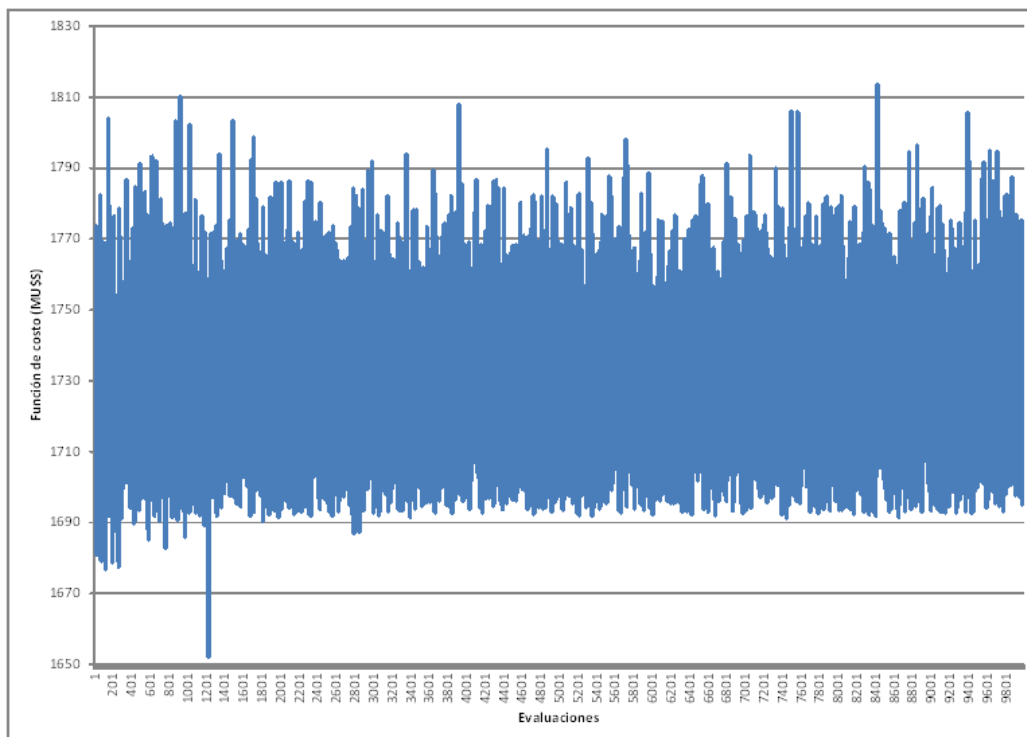
ro_UNARY:1

Resultados obtenidos:

Se realizaron 18690 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1810 y 1650 MU\$ presentándose un mínimo de 1652 MU\$ en el individuo 23166, que fue evaluado 1 sola vez.

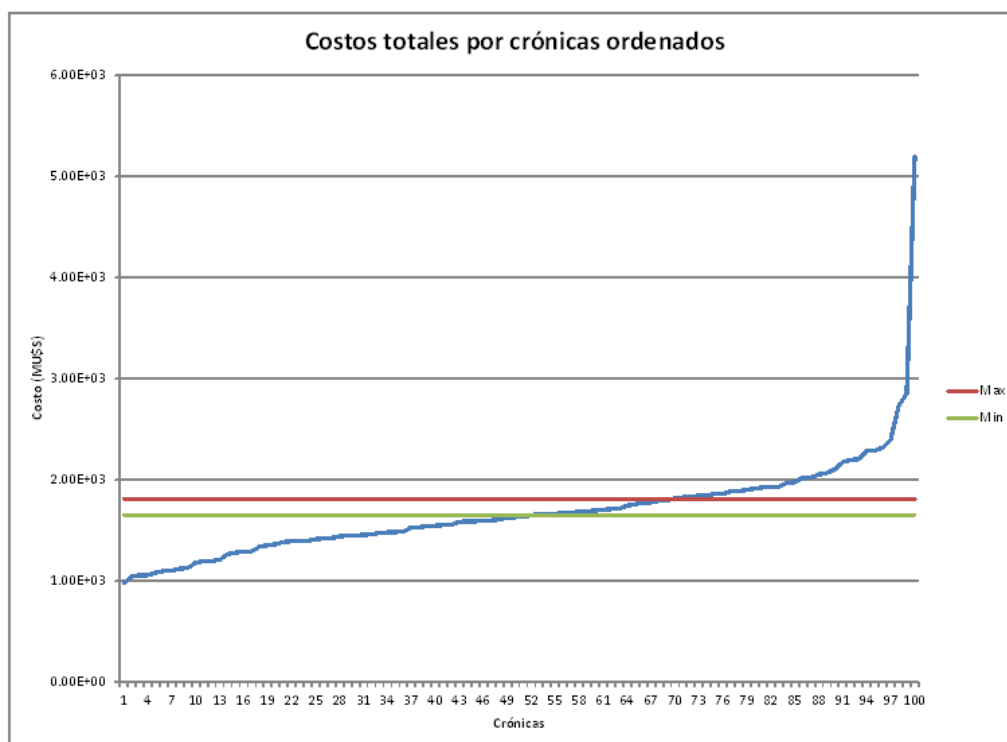
Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este

individuo.

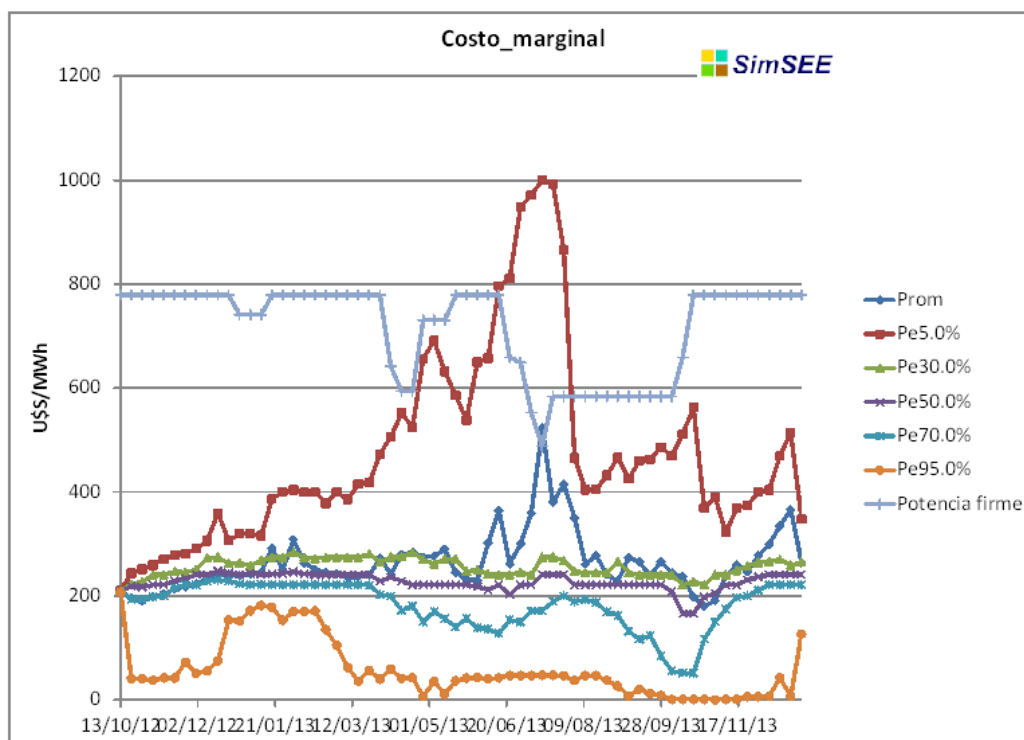
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	23/06/2013
CB-5ta-FOP	12/07/2013
Motores	26/06/2013
PTI_GO	08/07/2013
PTI_GO	12/04/2013
PTI_GO	08/07/2013
PTI_GO	31/12/2013
PTI_GO	25/04/2013
PTI_GO	31/12/2013
SG	06/04/2013
Palmar	31/12/2013
Bonete	30/01/2013

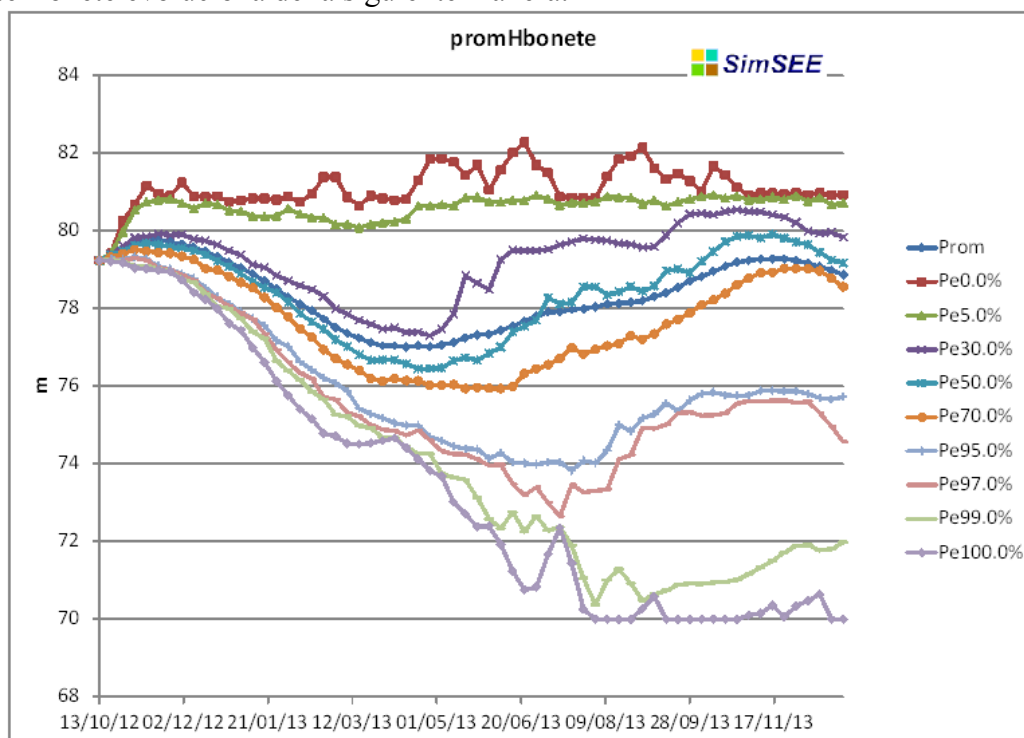
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

vi. CASO 5

NID_Problema:121

ro_VE:0.75

ro_VaR:0.25

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.05

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14AB_Grupo PAM VE=0.75 VaR=0.25

Tipo de codificación:

ro_BINARY:1

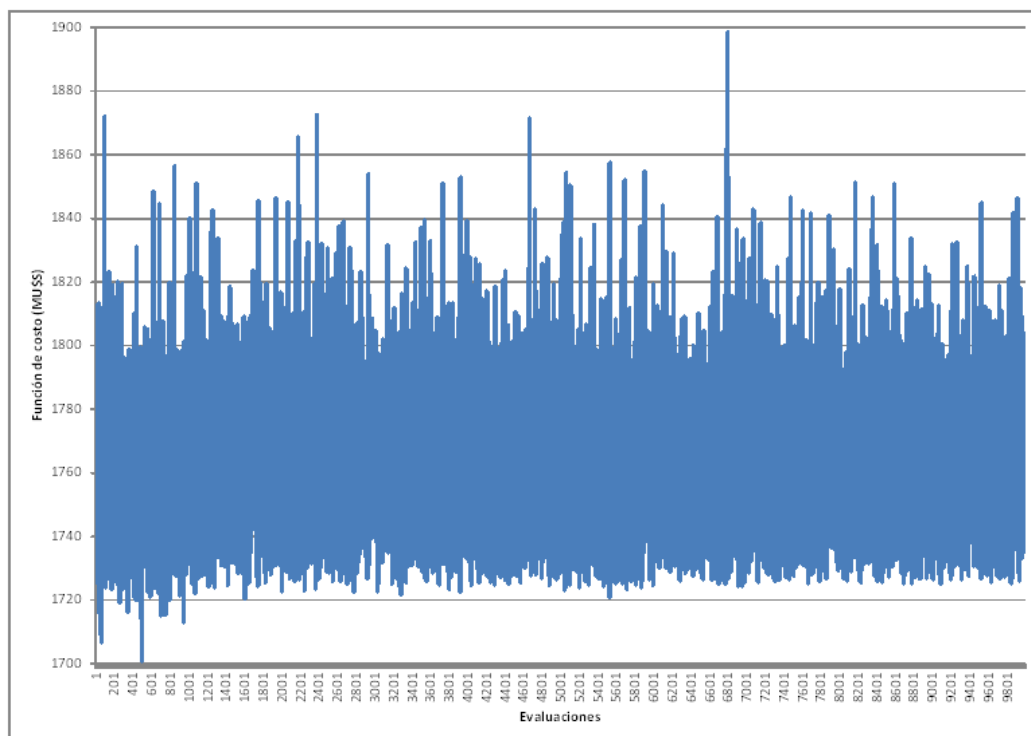
ro_UNARY:0

Resultados obtenidos:

Se realizaron 11484 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1700 y 1900 MU\$ presentándose un mínimo de 1701 MU\$ en el individuo 10991, que fue evaluado 1 sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

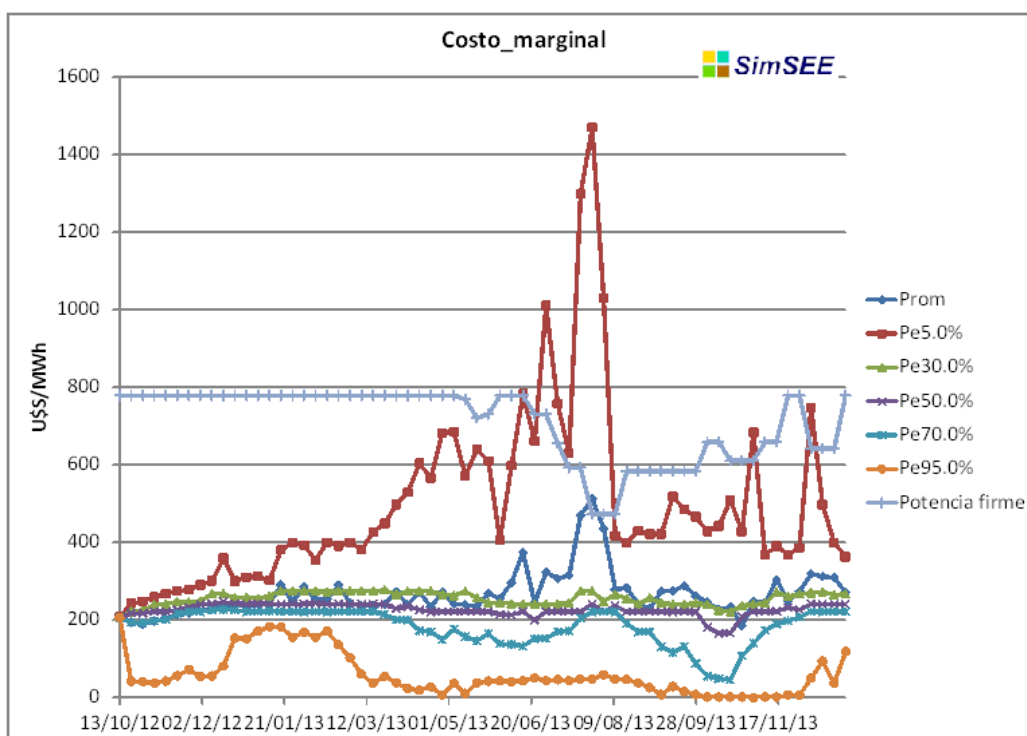
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	26/07/2013
CB-5ta-FOP	14/07/2013
Motores	14/05/2013
PTI GO	16/05/2013
PTI GO	31/12/2013
PTI GO	31/12/2013
PTI GO	20/10/2013
PTI GO	31/12/2013
PTI GO	23/06/2013
SG	06/12/2013
Palmar	15/07/2013
Bonete	31/12/2013

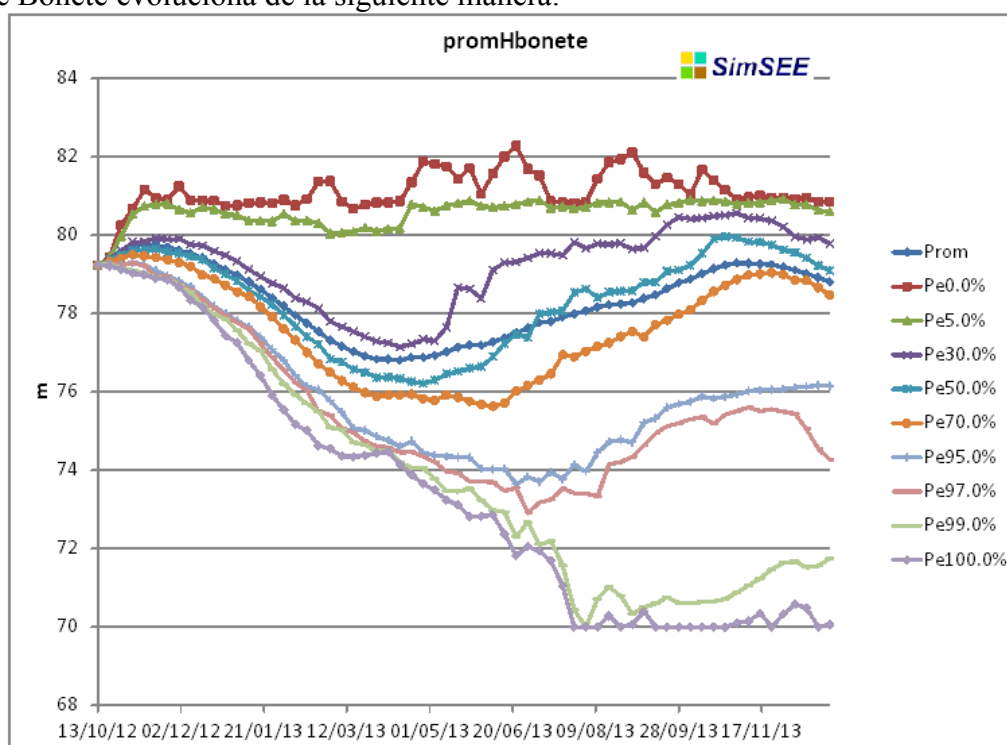
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

vii. CASO 6

NID_Problema:123

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.2

GA_prob_mutacion:0.05

Descripción del problema: Prueba 14CA_Grupo PAM premio_exito=0.20 mutacion=0.05

Tipo de codificación:

ro_BINARY:1

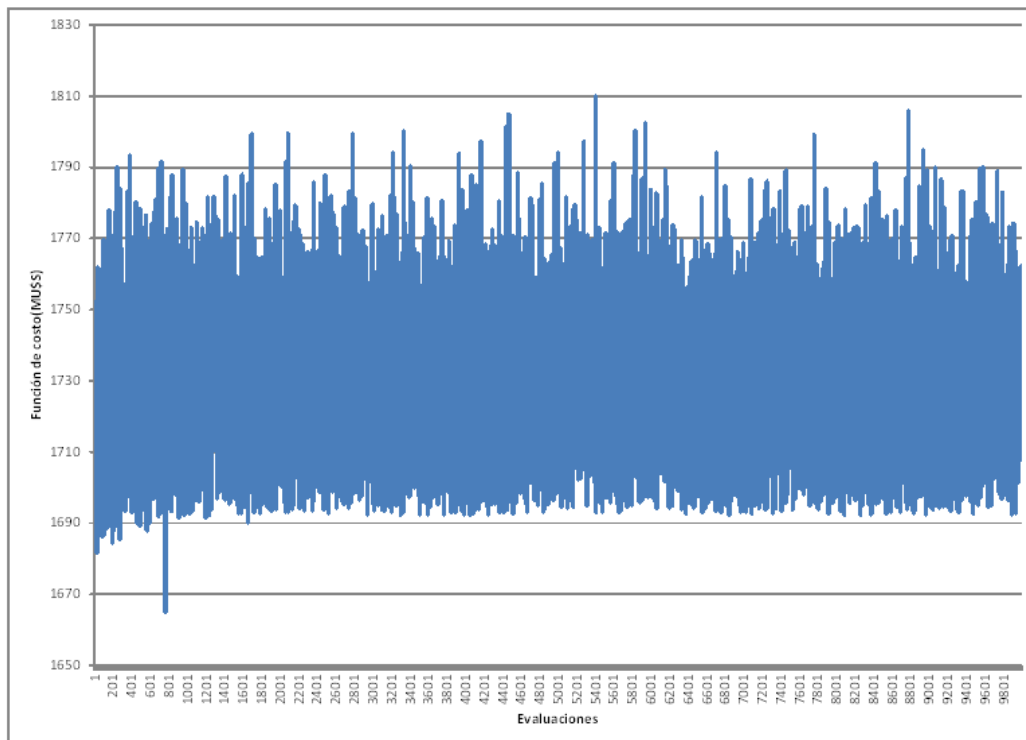
ro_UNARY:0

Resultados obtenidos:

Se realizaron 15675 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1660 y 1810 MU\$ presentándose un mínimo de 1665 MU\$ en el individuo 19333, que fue evaluado 8 veces.

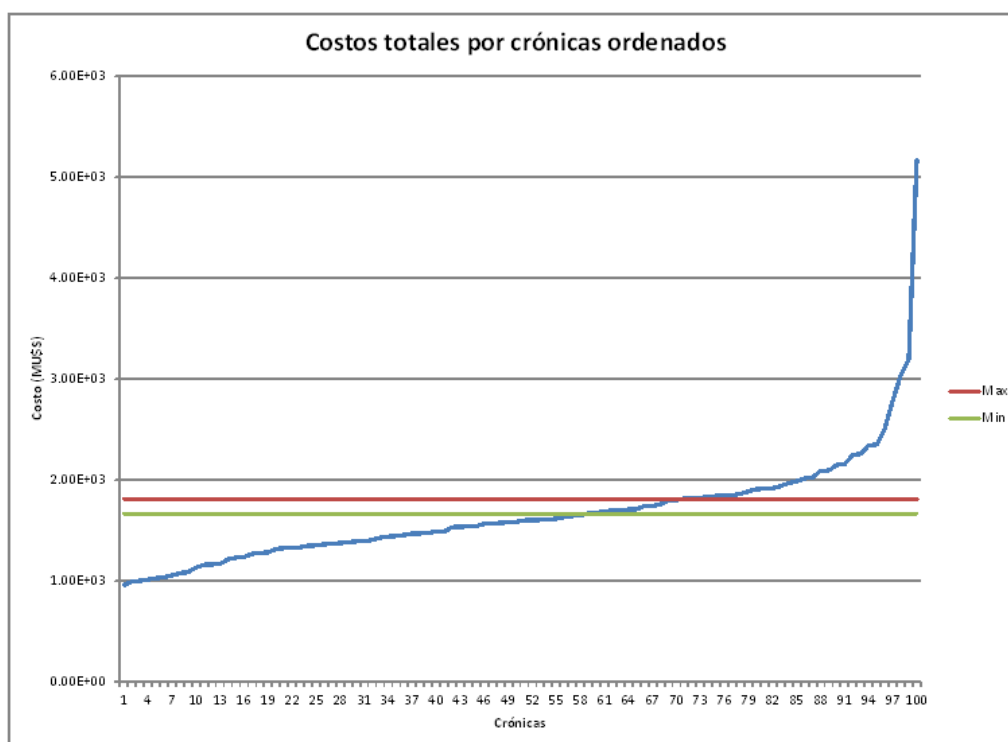
Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este

individuo.

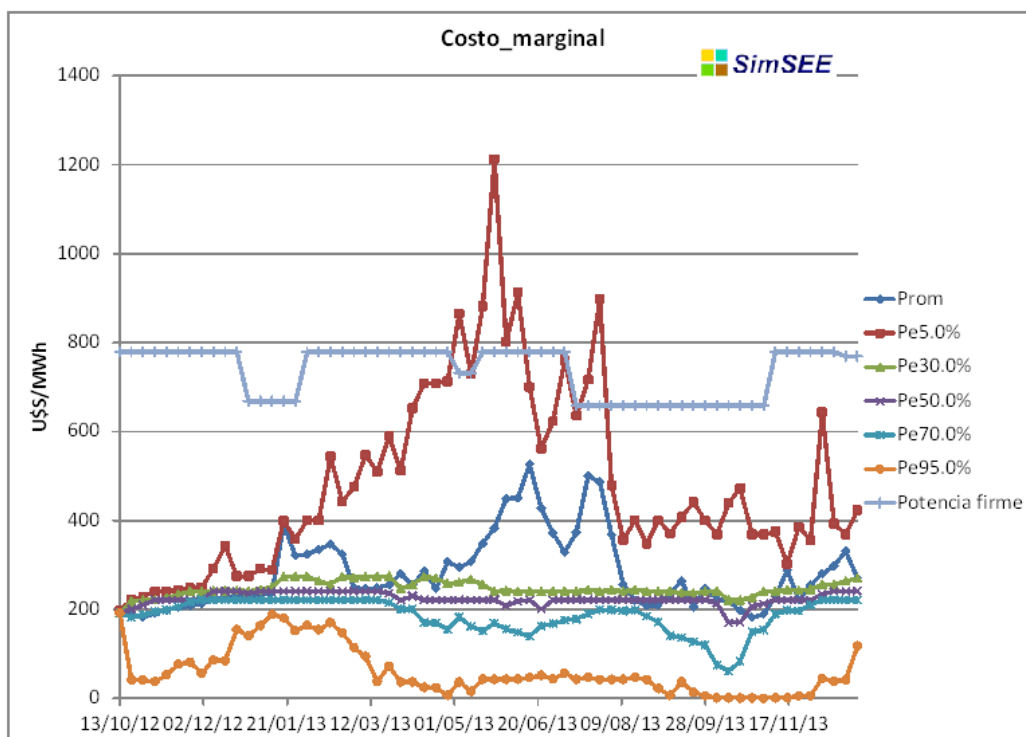
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	17/07/2013
CB-5ta-FOP	31/12/2013
Motores	24/12/2013
PTI GO	31/12/2013
PTI GO	07/05/2013
PTI GO	31/12/2013
PTI GO	31/12/2013
PTI GO	31/12/2013
PTI GO	31/12/2013
SG	31/12/2013
Palmar	01/01/2013
Bonete	31/12/2013

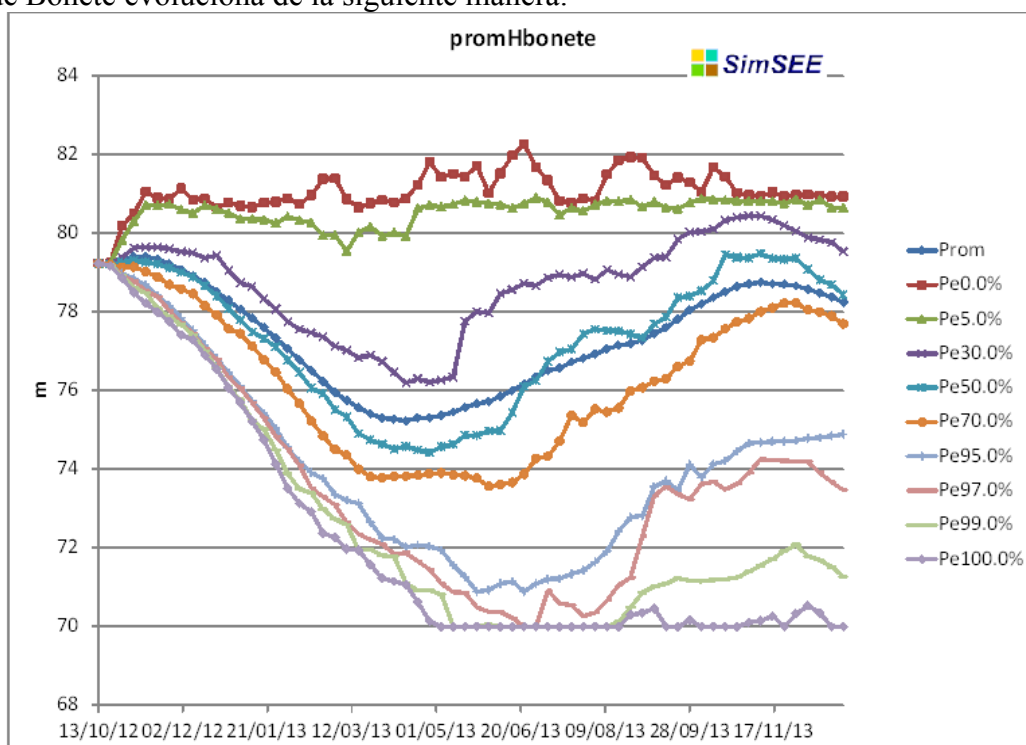
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78 m al final de la simulación.

viii. CASO 7

NID_Problema:124
ro_VE:0.85

ro_VaR:0.15
ro_CVaR:0

ro_GA:0.95
ro_MJ:0.05

GA_premio_exito:0.2

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14CB_Grupo PAM premio_exito=0.20 mutacion=0.01

Tipo de codificación:

ro_BINARY:1

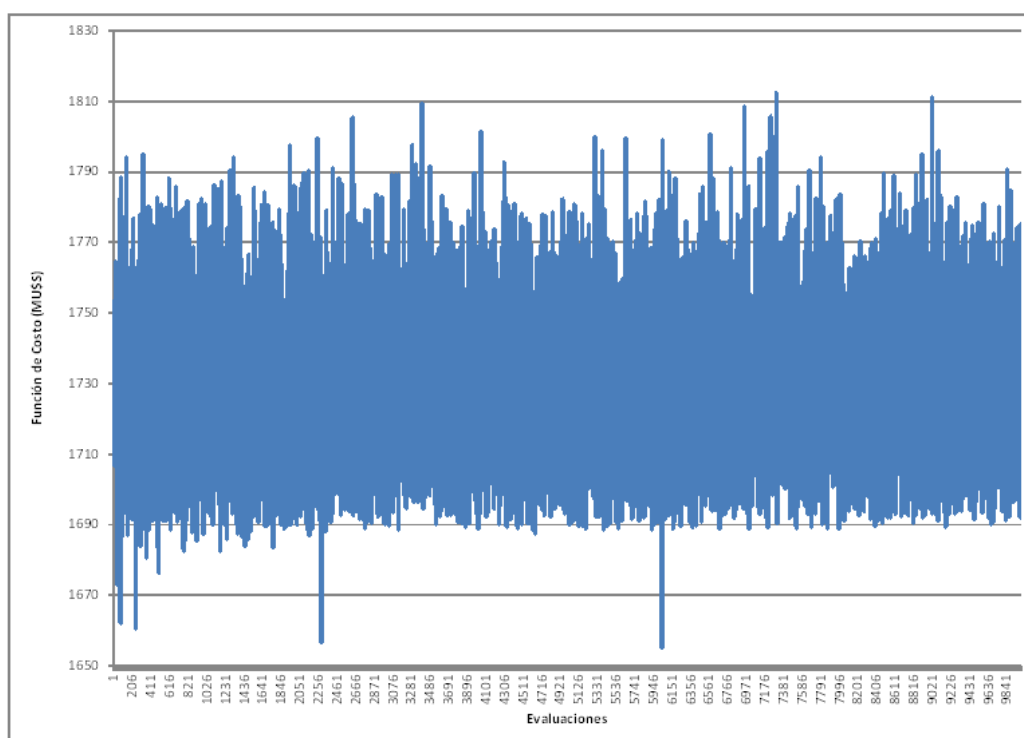
ro_UNARY:0

Resultados obtenidos:

Se realizaron 19978 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1650 y 1810 MU\$ presentándose un mínimo de 1655 MU\$ en el individuo 19482, que fue evaluado 64 veces.

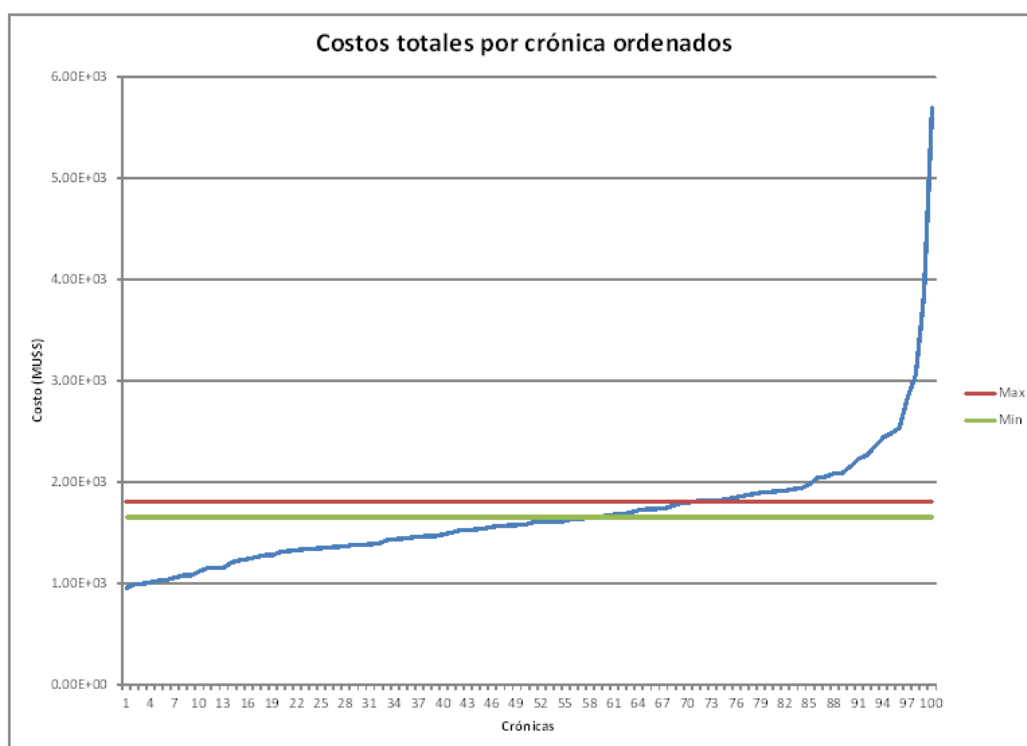
Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

Fechas de ubicación los mantenimientos:

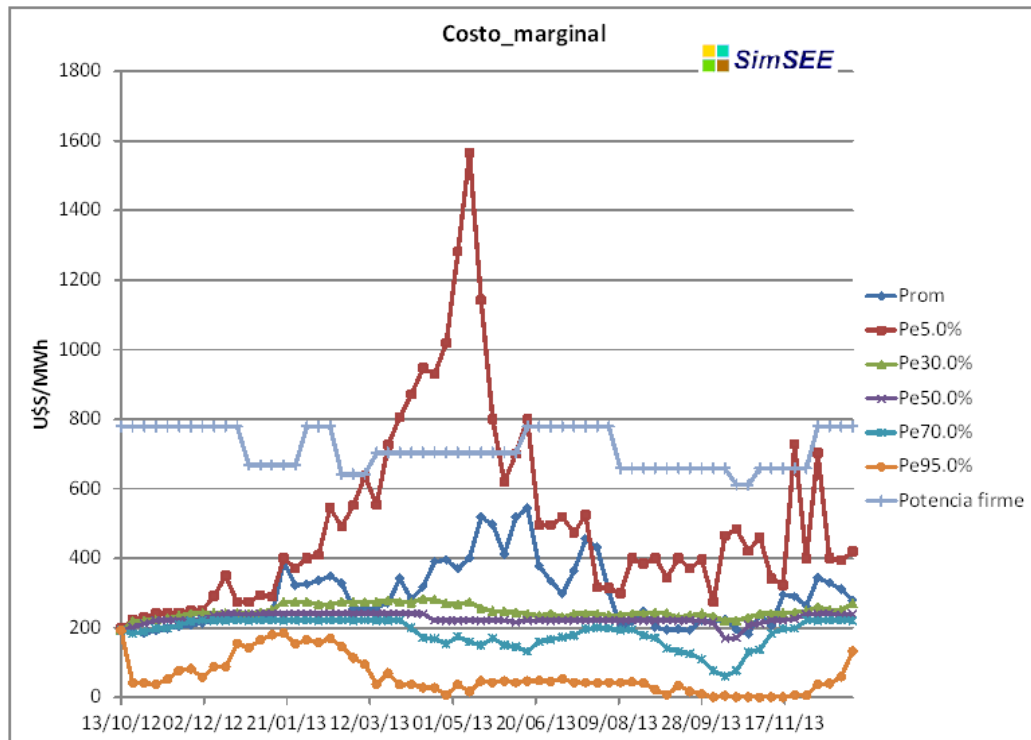
CB-6ta-FOP	11/08/2013
CB-5ta-FOP	17/03/2013

Motores	31/12/2013
PTI_GO	31/12/2013
PTI_GO	20/10/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
SG	19/12/2013
Palmar	01/01/2013
Bonete	31/12/2013

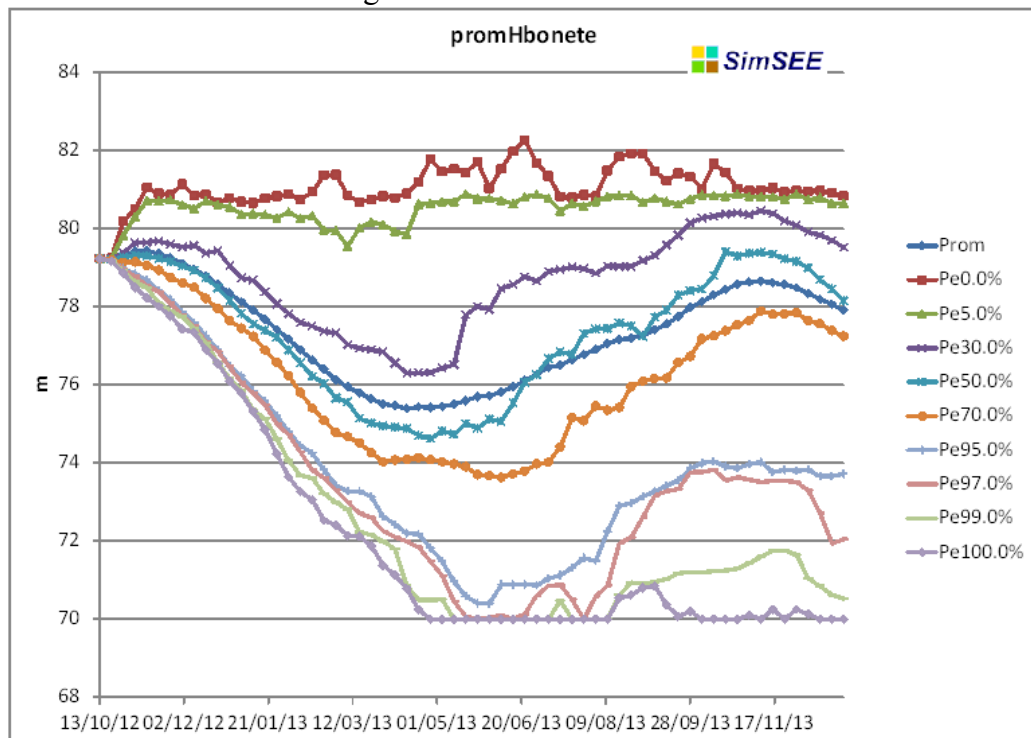
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se presenta el costo marginal del sistema y la potencia firme del conjunto de máquinas que participan en los mantenimientos:



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78 m al final de la simulación.

ix. CASO 8

NID_Problema:125

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:0.95

ro_MJ:0.05

GA_premio_exito:0.1

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14CC_Grupo PAM premio_exito=0.10 mutacion=0.01

Tipo de codificación:

ro_BINARY:1

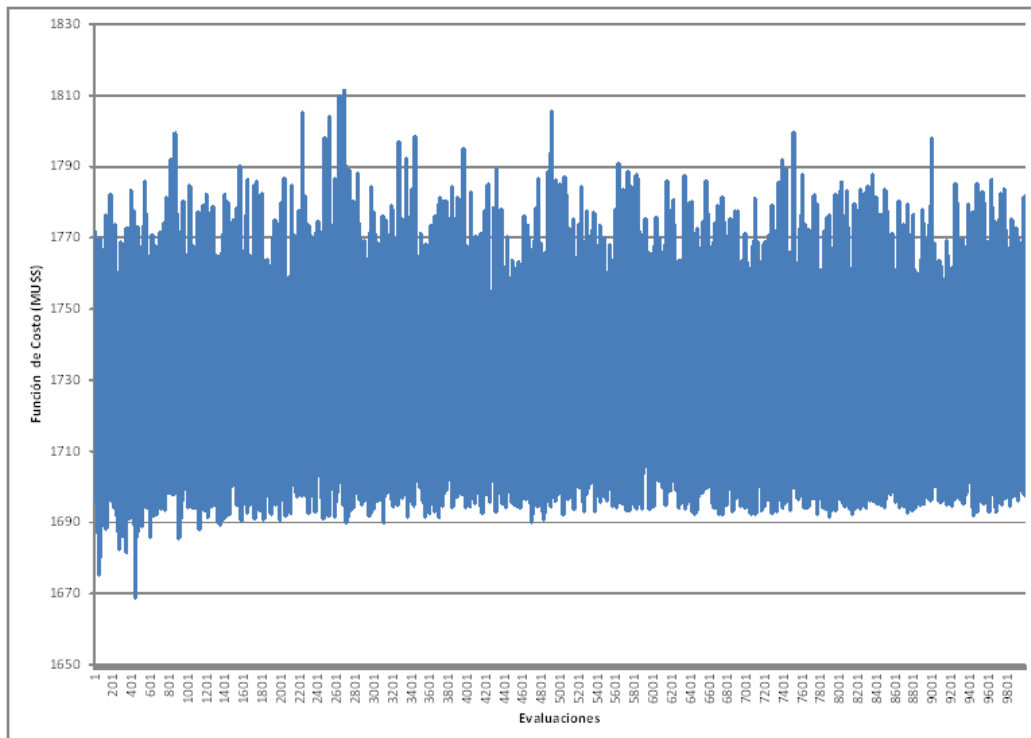
ro_UNARY:0

Resultados obtenidos:

Se realizaron 19185 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



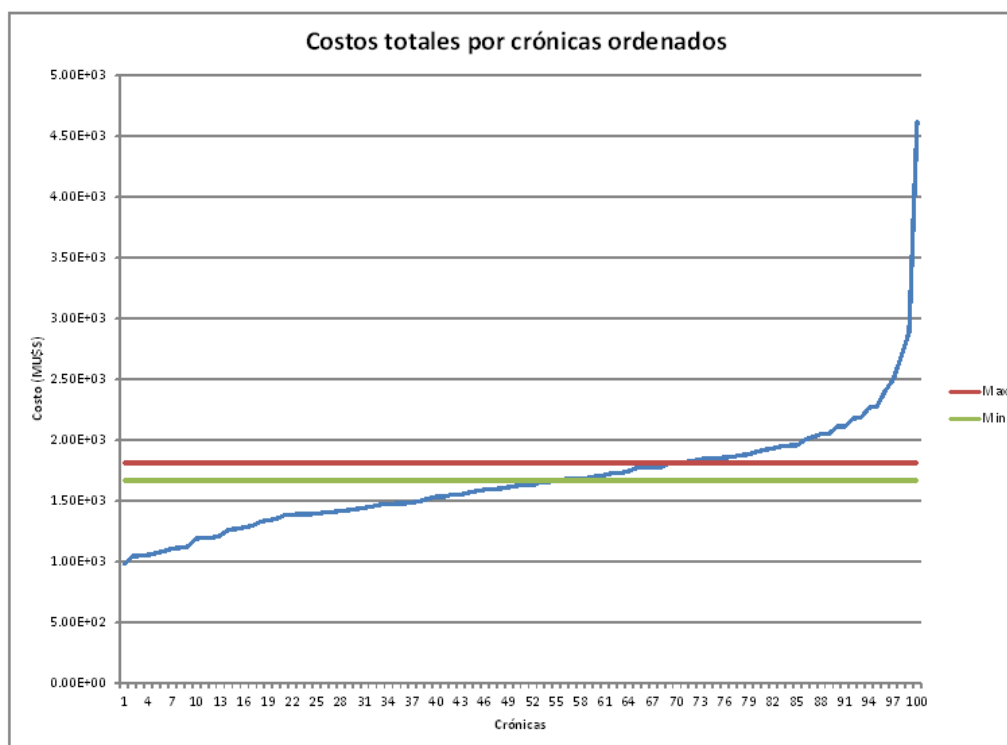
Se ve que la función de costo varía entre aproximadamente 1670 y 1810 MU\$ presentándose un mínimo de 1669 MU\$ en el individuo 23223, que fue evaluado 1 sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

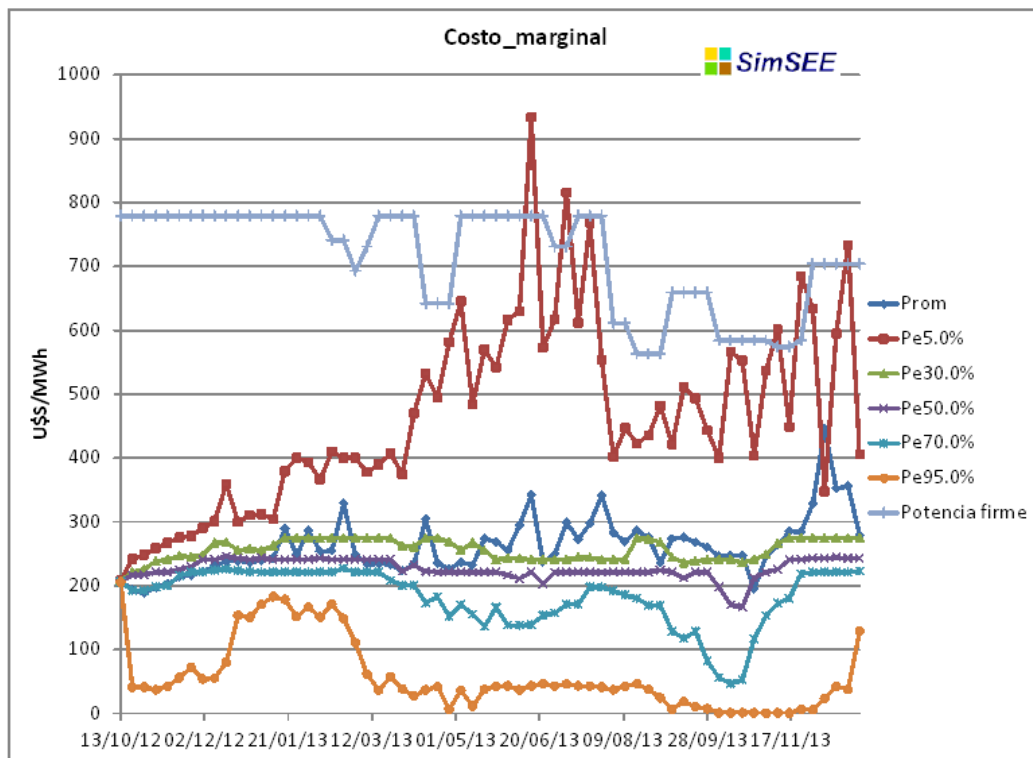
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	04/08/2013
CB-5ta-FOP	06/10/2013
Motores	07/11/2013
PTI_GO	02/07/2013
PTI_GO	03/08/2013
PTI_GO	05/03/2013
PTI_GO	17/08/2013
PTI_GO	16/08/2013
PTI_GO	31/12/2013
SG	12/04/2013
Palmar	30/12/2013
Bonete	19/02/2013

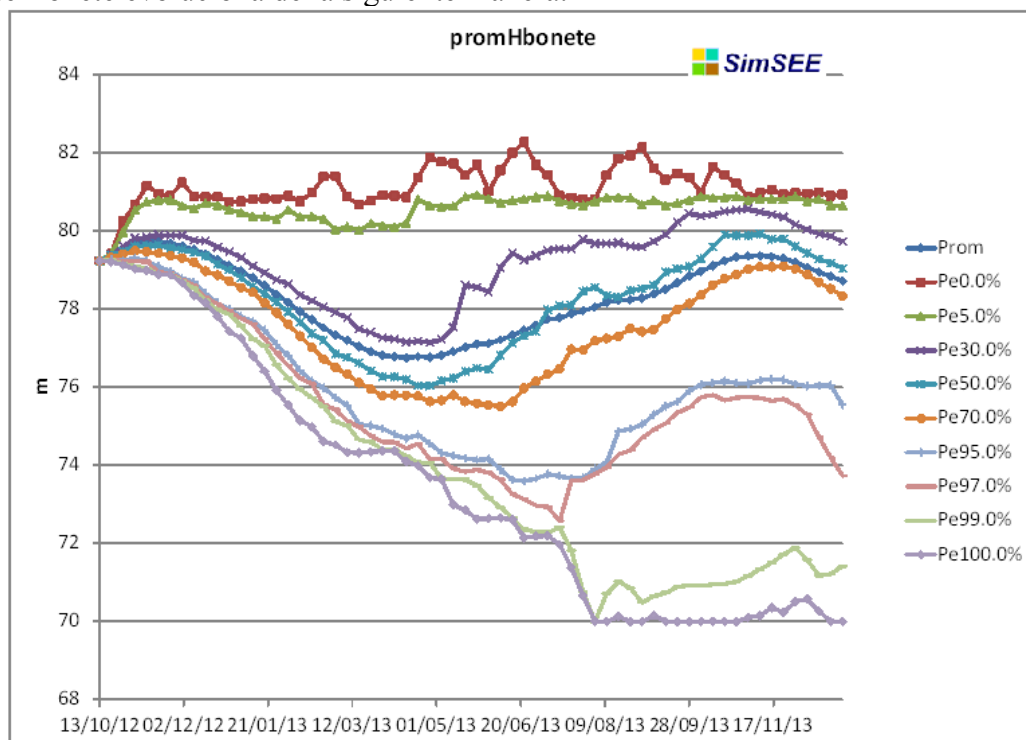
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

x. CASO 9

NID_Problema:131

ro_VE:0.85

ro_VaR:0.15

ro_CVaR:0

ro_GA:1

ro_MJ:0

GA_premio_exito:0.05

GA_prob_mutacion:0.01

Descripción del problema: Prueba 14AC_Grupo PAM ro_MJ=0

Tipo de codificación:

ro_BINARY:1

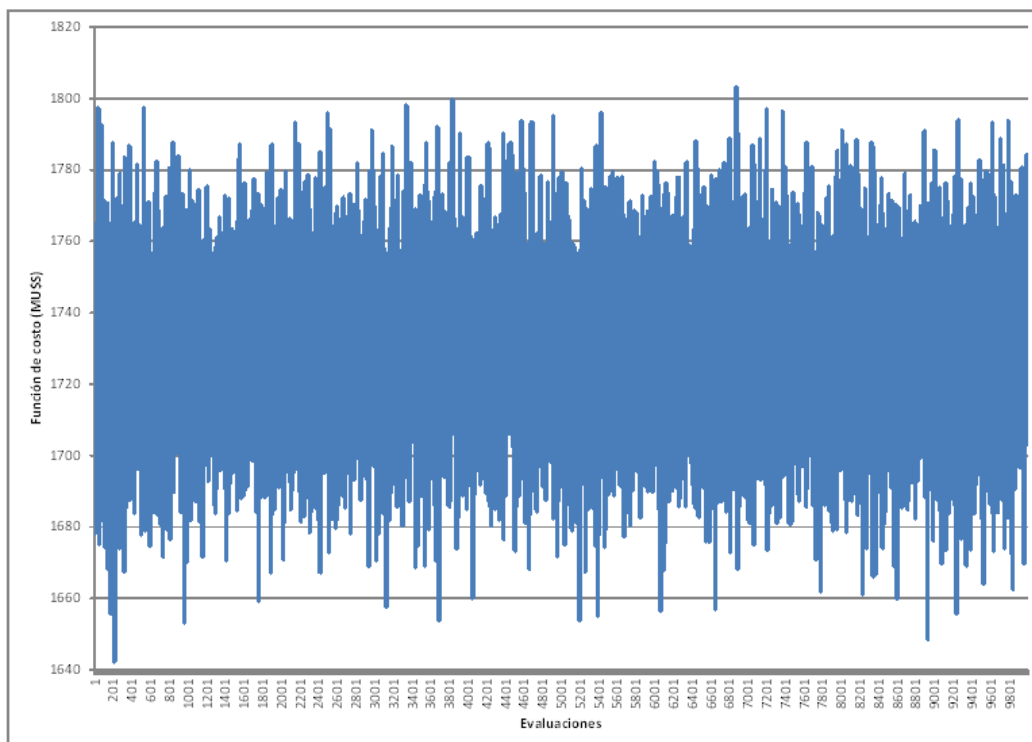
ro_UNARY:0

Resultados obtenidos:

Se realizaron 15237 evaluaciones.

Evolución de la función de costo en el tiempo:

En la siguiente gráfica se ve la evolución de la función de costo. La misma presenta en orden cronológico, comenzando desde la derecha, las evaluaciones realizadas y el costo resultante de las mismas.



Se ve que la función de costo varía entre aproximadamente 1640 y 1800 MU\$ presentándose un mínimo de 1642 MU\$ en el individuo 15028, que fue evaluado 1 sola vez.

Se presentan a continuación los resultados obtenidos en la simulación de la sala resultante para este individuo.

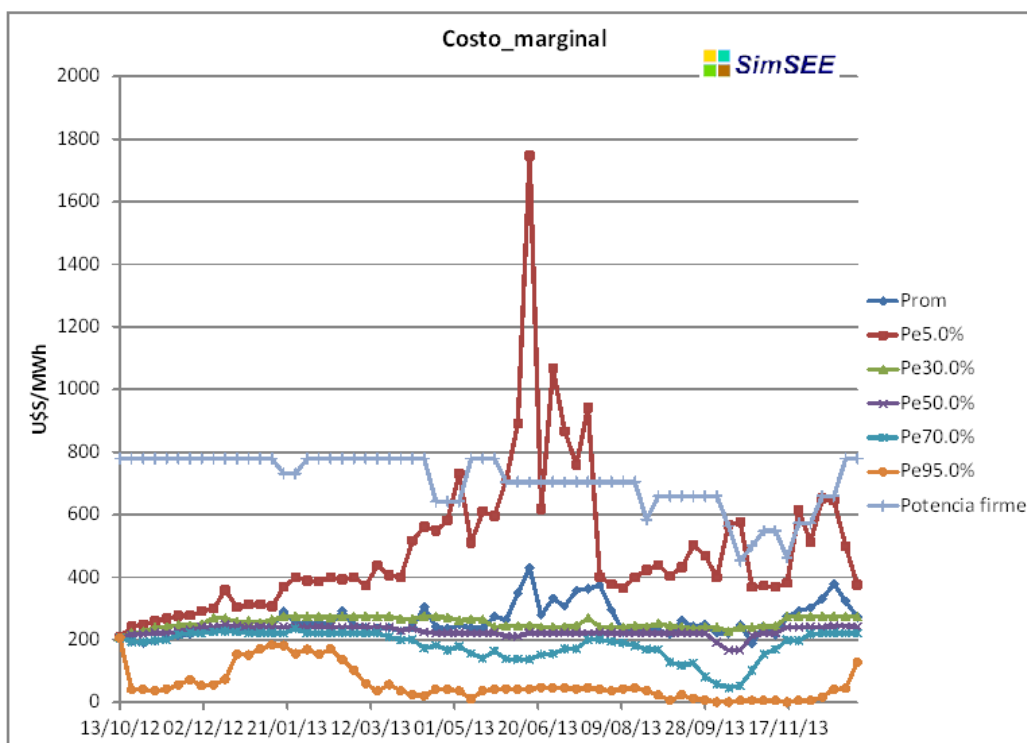
Fechas de ubicación los mantenimientos:

CB-6ta-FOP	23/08/2013
CB-5ta-FOP	03/06/2013
Motores	31/12/2013
PTI_GO	21/01/2013
PTI_GO	13/10/2013
PTI_GO	31/12/2013
PTI_GO	31/12/2013
PTI_GO	13/11/2013
PTI_GO	16/10/2013
SG	16/04/2013
Palmar	16/10/2013
Bonete	14/11/2013

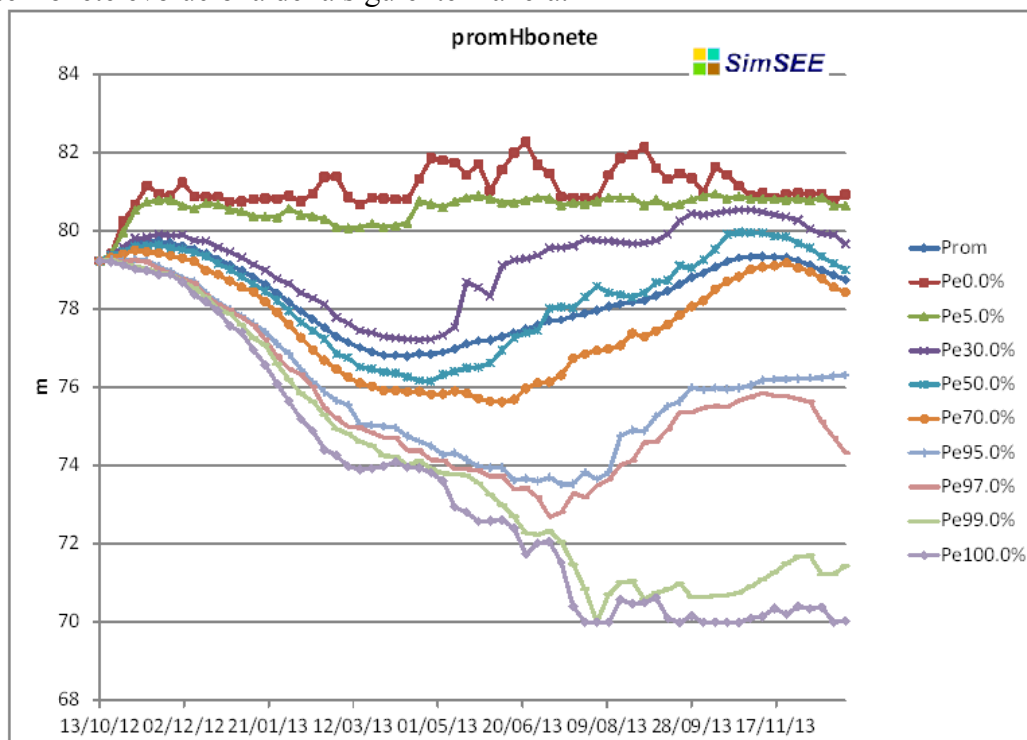
En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



En la siguiente gráfica se observa la función de costos por crónicas ordenados de menor a mayor y el máximo y mínimo obtenido en la función de costo del caso evaluado.



La cota de Bonete evoluciona de la siguiente manera:



Se observa que el valor esperado de la misma termina en 78.5 m al final de la simulación.

xi. Resumen de los casos:

	Caso 1	Caso 1 con nueva	Caso 2	Caso 2 con nueva	Caso 3	Caso 4	Caso 5	Caso 6	Caso 7	Caso 8	Caso 9
--	--------	------------------	--------	------------------	--------	--------	--------	--------	--------	--------	--------

		versión		versión							
ro_VE	0.85	0.85	1	1	0.85	0.85	0.75	0.85	0.85	0.85	0.85
ro_VaR	0.15	0.15	0	0	0.15	0.15	0.25	0.15	0.15	0.15	0.15
ro_CVaR	0	0	0	0	0	0	0	0	0	0	0
ro_GA	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	1
ro_MJ	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0
GA_premio_exit	0.05	0.05	0.05	0.05	0.1	0.05	0.05	0.2	0.2	0.1	0.05
GA_prob_mutacion	0.01	0.01	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.01
ro_BINARY	1	1	1	1	1	0	1	1	1	1	1
ro_UNARY	0	0	0	0	0	1	0	0	0	0	0
Total de Evaluaciones del problema	26119	20349	11266	41117	16472	18690	11484	15675	19978	19185	15237
Max Fcosto	1820	1750	1750	1720	1810	1810	1900	1810	1810	1810	1800
Min Fcosto	1659	1702	1615	1610	1674	1652	1701	1665	1655	1669	1642
Evaluaciones del individuo analizado	1	1	1	4	1	1	1	8	64	1	1

Fechas de ubicación de los mantenimientos:

Actor	Caso 1	Caso 1 con nueva versión	Caso 2	Caso 2 con nueva versión	Caso 3	Caso 4
CB-6ta-FOP	29/08/2013	23/07/2013	28/08/2013	22/07/2013	13/11/2013	23/06/2013
CB-5ta-FOP	31/12/2013	25/08/2013	19/02/2013	31/12/2013	28/05/2013	12/07/2013
Motores	21/08/2013	22/08/2013	19/01/2013	18/11/2013	31/12/2013	26/06/2013
PTI_GO	31/12/2013	06/05/2013	31/12/2013	31/12/2013	31/12/2013	08/07/2013
PTI_GO	25/02/2013	01/01/2013	05/11/2013	31/12/2013	27/03/2013	12/04/2013
PTI_GO	22/04/2013	15/08/2013	31/12/2013	22/01/2013	07/10/2013	08/07/2013
PTI_GO	08/06/2013	31/12/2013	19/08/2013	31/12/2013	04/03/2013	31/12/2013
PTI_GO	19/11/2013	31/12/2013	31/12/2013	31/12/2013	10/09/2013	25/04/2013
PTI_GO	19/06/2013	29/01/2013	03/01/2013	31/12/2013	07/03/2013	31/12/2013
SG	31/12/2013	10/11/2013	25/01/2013	01/01/2013	31/12/2013	06/04/2013
Palmar	01/07/2013	13/02/2013	24/05/2013	12/05/2013	31/12/2013	31/12/2013
Bonete	31/12/2013	01/11/2013	31/12/2013	23/09/2013	30/09/2013	30/01/2013

Actor	Caso 5	Caso 6	Caso 7	Caso 8	Caso 9
CB-6ta-FOP	26/07/2013	17/07/2013	11/08/2013	04/08/2013	23/08/2013
CB-5ta-FOP	14/07/2013	31/12/2013	17/03/2013	06/10/2013	03/06/2013
Motores	14/05/2013	24/12/2013	31/12/2013	07/11/2013	31/12/2013
PTI_GO	16/05/2013	31/12/2013	31/12/2013	02/07/2013	21/01/2013
PTI_GO	31/12/2013	07/05/2013	20/10/2013	03/08/2013	13/10/2013
PTI_GO	31/12/2013	31/12/2013	31/12/2013	05/03/2013	31/12/2013
PTI_GO	20/10/2013	31/12/2013	31/12/2013	17/08/2013	31/12/2013
PTI_GO	31/12/2013	31/12/2013	31/12/2013	16/08/2013	13/11/2013
PTI_GO	23/06/2013	31/12/2013	31/12/2013	31/12/2013	16/10/2013
SG	06/12/2013	31/12/2013	19/12/2013	12/04/2013	16/04/2013
Palmar	15/07/2013	01/01/2013	01/01/2013	30/12/2013	16/10/2013
Bonete	31/12/2013	31/12/2013	31/12/2013	19/02/2013	14/11/2013

C. CONCLUSIONES

En la gráficas Función de Costos vs. Evaluaciones de la mayoría de los casos estudiados, se percibe que el valor de la Función de Costos tiende a disminuir a medida que transcurre el tiempo.

Se puede ver que en todos los casos la variabilidad hidráulica es mucho mayor relativamente que la dispersión ocasionada por la movilidad de los mantenimientos.

De todas formas se puede observar que, en valores absolutos, la diferencia del costo total de operación en el período de simulación elegido puede llegar a valores del orden de 50 a 200 MU\$S.

Se observa que en la mayoría de los individuos hay una “tendencia” a ubicar el mantenimiento de una o más unidades en la última fecha “elegible”.

El programa sugiere, en algunos casos, fechas de mantenimiento que en general se consideran “restrictivas” en los planes de mantenimiento históricos utilizados para el sistema uruguayo, en los cuales se da preferencia a iniciar los grandes mantenimientos de unidades térmicas en otoño o primavera, que son las épocas del año en las que normalmente ocurren las mayores precipitaciones.

De los dos casos corridos con la última versión de OddFace, en uno de ellos se fijó un “ro_VaR”=0.15 y en el otro se fijó en 0 (cero). Se aprecia que la variación del costo entre el máximo y el mínimo es menor en el caso de “ro_VaR”=0.15

Surge la duda sobre la validez de un individuo “exitoso” (individuo con el valor mínimo de la función de costo), pero que fue evaluado sólo una vez, y se aparta mucho del rango de variación de la mayoría del conjunto de individuos.

Otra interrogante es respecto al “tiempo de prueba” de cada problema, es decir, cuánto tiempo se considera necesario para tener el semáforo en verde y llegar a un mínimo difícilmente superable (o incluso el absoluto).

D. ANEXO: “Bugs” detectados y mejoras propuestas por el equipo de trabajo que fueron implementadas durante la etapa de trabajo.

- ☐ En los primeros intentos de empaquetar una sala para correr el OddFace salía un mensaje que decía que no encontraba los costos futuros de la misma. A su vez, al usar el empaquetar dentro del SimSee aparecía un mensaje que indicaba que no encontraba el archivo CF.bin (era el mismo error que salía en el OddFace cuando se quería empaquetar):
- ☐ Cuando se creaba un nuevo problema se abría la pantalla del “Editor” y no permitía elegir “Tipo de Problema”; daba un error, se salía al listado de problemas y cuando volvíamos a entrar seleccionando el problema que habíamos definido, sí se podía elegir el “Tipo”.
- ☐ Se sugirió la inclusión de un botón “clonar”, tanto para problemas como para unidades a mantener, y en segunda instancia, que definiera como fecha de inicio del problema, la fecha de clonación, ya que en primera instancia, mantenía la fecha de creación del problema original.
- ☐ Mostrar la lista de campos de la tabla de evaluaciones seleccionables en la pantalla de consulta.
- ☐ Aparece un problema cuando terminan los mantenimientos de actores con más de una unidad y se quiere restablecer la cantidad inicial de unidades disponibles. También se descubre que las fechas en las cuales se iniciaban los mantenimientos no podían ser cualquier día del período del mantenimiento, si la duración de la etapa era mayor a 1, sino que tenían un delta tiempo igual a la duración de la etapa. Se corrige esto para poder ubicar el comienzo del mantenimiento en cualquier día del período elegido.
- ☐ Surgió una duda sobre el número evaluaciones para un individuo y las crónicas asociadas. Se plantea la incertidumbre en los casos en los que un individuo es re-evaluado respecto a si se usa el mismo conjunto de crónicas o es otro diferente.

Evolución de las versiones de Oddface durante el desarrollo del proyecto.

A continuación se presenta un resumen que muestra la evolución de OddFace durante la ejecución de esta etapa del proyecto, que incluye los comentarios asociados a cada versión expresados por el MSc. Ing. Ruben Chaer.

Primera versión: 10/10/2012:

SimSEE_OddFace_v362_RUIDA_C02_EnganchesProm_binarios

(http://iie.fing.edu.uy/simsee/instalador_simsee/SimSEE_OddFace_v362_RUIDA_C02_EnganchesProm_binarios.zip)

El binario que permite editar los problemas en la BD del IIE es: oddface_prepare.exe

Este binario se conecta usando http con el servidor web del IIE para acceder a la BD. Sin configurar la conexión vía PROXY.

Consulta y Versión 2: 15/10/2012

Se intenta empaquetar una sala para correr el OddFace, pero sale un mensaje que dice que no encuentra los costos futuros de la misma. Y al usar el empaquetar dentro del SimSee sale el siguiente mensaje (que es el mismo error que sale en el OddFace cuando se quiere empaquetar):

“Archivo no encontrado:

\simsee\rundir\pest_nov_abril_2011_dm_85_sesgo\CF.bin"

Al parecer la sala se enganchó por defecto con este CF.bin y no se puede borrar.

Había un BUG en el editor, el botón VACIAR limpiaba el cuadro de texto, pero no las variables de la sala con lo cual al guardar quedaba "almacenado el archivo". Normalmente esto no es problema, pues no intenta leer el archivo al ejecutar, pero el empaquetador no se fija si el radio-botton está que lo usa o no e intenta mandarlo.

Además de eso, fallaba también al intentar empaquetar el archivo de datos de la demanda detallada por un tema de que el archivo lo mantiene abierto el actor demanda y el ZIPPER de empaquetado fallaba al abrir.

Nueva Versión:

http://iie.fing.edu.uy/simsee/instalador_simsee/SimSEE_v364_ANARQUIA_binarios.zip

Versión 3: 15/10/2012

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare.zip

Versión 4: 18/10/2012

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_v2.zip

Tiene un botón para CLONAR problemas y las fichas de los problemas para facilitar crear modificaciones de uno dado.

Versión 5: 18/10/2012

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_v3.zip

- 1) mejora el CLONAR cambiando la fecha de creación del problema
- 2) En el formulario de consulta se agregó un cuadro de texto con los campos disponibles en la tabla de evaluaciones.

Versión 6: 19/10/2012

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_v4.zip

Se agregó en la CONSULTA que genera en el Excel la DECODIFICACION de los parámetros para que se pueda mirar "cómo es el individuo".

Los nombres de los parámetros son el nombre puesto durante la definición del problema específico (esto está codificado en la Clase pascal que define el tipo de problema) y al final se agregó '_nbts' que es la cantidad de bits con que se representa el rango resultante para el parámetro.

Versión 7: 20/10/2012

Se presenta un problema para la descarga del Excel.
Se corrige el problema:

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_v5.zip

Bajar la consulta ahora es más rápido y realimenta con un progress-bar para ver el progreso de la misma.

Había una torpeza en el anterior, que para cada renglón hacía de nuevo una consulta para bajar los individuos de a uno.

También se agregó un botón que en lugar de abrir el Excel y escribir directamente escribe un archivo de texto con extensión XLT con la información separada por TABs cosa que generalmente el Excel lee bien. Esto es mucho más rápido que escribir directamente a Excel.

Versión 8: 06/11/2012

Debido a estos cambios se tuvieron que parar todos los problemas y recompilar el cluster.

Nueva versión:

http://iie.fing.edu.uy/simsee/instalador_simsee/SimSEE_v366_Anarquia_OddFace_binarios.zip

Debido a este cambio se tuvieron que borrar los historiales de los problemas y comenzar de nuevo con las evaluaciones.

Versión 9: 12/11/2012

Se detectó que al probar simular una sala salía el mensaje:"error ... esta sala fue creada con una versión posterior al simulador ...". Esto fue debido a diferentes versiones entre los usuarios y el cluster.

Nuevas versiones:

Binarios SimSEE:

http://iie.fing.edu.uy/simsee/instalador_simsee/SimSEE_v367_Anarquia_efcm.zip

Binarios OddFace:

http://iie.fing.edu.uy/simsee/instalador_simsee/OddFace_v367_binarios.zip

Consulta: 13/11/2012

Surgió una duda sobre el número evaluaciones para un individuo y las crónicas asociadas.

Se cuestiona el hecho de que si un individuo es re-evaluado se usa el mismo conjunto de crónicas o es otro diferente.

Respuesta: 16/11/2012

Se descubre que luego de obtenido el ADN (sea por el método que sea) antes de hacer la corrida para evaluar la f_Objetivo se elige una semilla totalmente aleatoria. Por lo tanto se plantea el hecho de que es mucho mejor mantener un conjunto de sorteos coherentes.

Por ejemplo, que la primera evaluación de cualquier punto sea siempre con la semilla 31, la segunda siempre con 32 y así sucesivamente.

De esa forma, considerando puntos con igual cantidad de evaluaciones sabemos que sufrieron las mismas suertes y que las diferencias que vemos entre ellos no

están tan afectadas por la suerte que tuvieron (o por lo menos que para la misma suerte las diferencias son por su diferente posición en el espacio de búsqueda).

Redundancia en la codificación.

De una corrida sencilla, se descubre lo siguiente:

Esta tiene solo dos variables y cada una de las variables puede tomar solo 13 valores. Esto lleva a un espacio de búsqueda muy reducido 13 x13 valores. Sin embargo la tabla de Individuos evaluados tiene MUCHOS MAS. Analizando los resultados se ve que hay MUCHOS ADNs que al decodificarlos dan las mismas salas.

Lo que sucede es que las cadenas de ADN se implementaron como un tramo continuo de memoria en múltiplos de 16 bits. En el caso de este problema para codificar 13 valores es suficiente con 4 bits, por lo que en total el largo de la cadena de ADN es de 8 bits. Pero en la codificación se utiliza 16bits. Al operar en los cruzamientos y mutaciones opera sobre toda la cadena (los 16 bits) cuando decodifica el ADN a los parámetros reales, solo importan los primeros 8 bits.

Estos primeros 8 bits son capaces de representar 16 x 16 salas, pero los números a representar son 13 x 13.

El algoritmo implementado, primero decodifica el número entre 0 y 15 y lo mapea entre 0 y 12 asignando a 12 los valores 14 y 15.

De esta forma, el extremo superior de un parámetro está privilegiado en cuanto que hay más combinaciones que lo representan.

Entonces, para este ejemplo, los 8 primeros bits del ADN tomados en dos grupos de 4 bits y representan todos los puntos de exploración (13x13) con una redundancia sobre los valores del borde superior de cada una de las dos variables que tiene 3 representaciones en lugar de 1.

Pero los 8 bits siguientes pueden tomar cualquier número siendo que los 8 bits primeros son los que importan en la decodificación. Entonces cada sala (fijada por los primeros 8 bits) tiene $2^8 = 256$ representaciones posibles.

Esto NO es necesariamente un ERROR. Cuando se implementó se sabía que se estaba creando redundancia (por lo menos la del rango de cada parámetro) pero no se suponía que esta redundancia podría ser tan violenta como los 256 por cada una.

Llevado al extremo de una codificación que solo use un bit de los últimos 16 del ADN, sobran 15 bits al final que crean redundancia y eso significa un multiplicidad de $2^{15} = 32768$ valores.

El algoritmo tal como está tendría que ser capaz igual de encontrar los óptimos. Hay muchos clones sueltos nada más.

Aviso: 17/11/2012

Se avisa que existe un error en la codificación GRAY (una variable no inicializada), las corridas que usaban esta opción están mal.

Versión 10: 18/11/2012

Nueva versión del OddFace_prepare:

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_ConAglutinador.zip

Tiene la facilidad de AGLUTINAR los resultados agrupando los individuos que aparecen como distintos pero que en realidad decodifican en la misma sala. Lo que hace es calcular las f_Objetivo ponderando cada uno por la cantidad de evaluaciones.

Versión 11: 18/11/2012

Se detectó un problema al utilizar Aglutinar:

Nueva versión:

http://iie.fing.edu.uy/simsee/instalador_simsee/oddface_prepare_Aglutinante_v2.zip

También se agregó en la página que deja bajar un individuo un botón "Análisis ADN" que permite ver la cadena binaria de un individuo.

Primero muestra la cadena TAL CUAL está en la base de datos, luego muestra los valores decodificados y después el ADN vuelto a codificar, pero "limpiando los bits redundantes" Sería algo así como "El representante principal del conjunto de ADNs que dan decodificado lo mismo".

Consulta: 18/11/2012

No se encuentra dónde se fija la probabilidad de excedencia (que se suponía 5%) para cálculo de VaR y CVaR en OddFace.

Al parecer ese parámetro no se inicializa en ningún lado, con lo cual cualquier conclusión en corridas que tengan aversión al riesgo $ro_VaR > 0$ y/o $ro_CVaR > 0$ no vale.

El parámetro NO SE ESTA INICIALIZANDO y tal como está definida pa PE queda en CERO. Por esto $VaR = CVaR = f_MAX$ en los individuos que tienen una sola evaluación. Entonces, las corridas realizadas que tenga $ro_VaR > 0$ y/o $ro_CVaR > 0$ deben considerarse con una aversión al riesgo con el riesgo medido con $PE = 0\%$.

Versión 12: 22/11/2012

Los principales cambios son:

- a) el tema de la semilla aleatoria (imponiendo que la misma es $30 + cnt_evaluación$) de esa forma todas las evaluaciones 1 tienen 31, las 2 tienen 32 y así sucesivamente.
- b) limpieza del resto de los bits en la codificación (por los cajones en múltiplos de 16)
- c) ajuste de la redundancia binaria para representar las restricciones de caja de los parámetros.
- d) Corrección de errores como el valor de PE que no estaba inicializado.
- e) Habilitación del código GRAY.

Ultima versión SimSEE:

http://iie.fing.edu.uy/simsee/instalador_simsee/SimSEE_v368_ANARQUIA_EFC_binarios.zip

Ultima versión OddFace:

http://iie.fing.edu.uy/simsee/instalador_simsee/OddFace_v11_GEA_binarios.zip

La nueva versión de OddFace, en el botón "Analizar ADN" en la pantalla de

consulta imprime en el cuadro de texto `f_VE`, `f_VaR`, `f_CVaR`, `f_Objetivo` y el `f_histograma` (que es le vector con las 100 crónicas simuladas). Estos valores son el promedio de las evaluaciones del individuo.

E. Anexo: Propuestas de mejora al programa hacia el futuro

- ☐ Podría incluirse una lista desplegable con los actores que están en la sala de manera que sea más fácil elegirlos y no cometer errores en los nombres a la hora de agregarlos. Si esto no fuera fácil de implementar, se podría proponer que existiera una “advertencia” al ejecutar el problema, que indique que cierto actor no está definido en la sala empaquetada.
- ☐ Habilitar la posibilidad de incluir “mantenimientos partidos” o sea que un mismo mantenimiento se realice en dos períodos diferentes que pueda elegir el usuario.
- ☐ Posibilidad de clonar un caso y que se mantenga el historial de las evaluaciones de modo de evaluar el comportamiento de diferentes problemas que habiendo comenzado con los mismos parámetros, se ramifiquen en sendos problemas con variaciones de ciertos parámetros.
- ☐ Para el caso de centrales con más de una unidad, prever la posibilidad de que indiquemos la cantidad de unidades a mantener y que digamos cuántas puede sacar en forma simultánea.
- ☐ Alertar al usuario si ingresa valores fuera de un rango admisible en alguno de los campos habilitados en la pantalla de “Edición”.
- ☐ El botón “Guardar” en realidad es “Guardar y salir”; podría ser útil que existan ambas opciones.
- ☐ Los campos de la tablas de evaluaciones que uno desea seleccionar deben digitarse uno por uno, separados con una “coma”, por lo cual hay que tener cuidado de hacerlo correctamente o mejor “copiar y pegar” el nombre del campo de la lista que aparece en el recuadro “arriba a la derecha” en la ventana de “Consultas”. Se plantea si fuera posible que se pudiera seleccionar y deseleccionar de esa lista en la medida de las necesidades del usuario. Ocurre que si uno comete un error al escribir un nombre, no realiza la “bajada a Excel” y sin embargo, en la ventana de comando indica “fin copia de archivos”.

Capítulo VIII. OddFace+PAM (2)

Autores: Ing. María Cristina Álvarez y Felipe Palacio.

Este capítulo documenta el análisis de caso realizado por los autores en la aplicación de la herramienta OddFace para resolver un problema de PAM (Plan Anual de Mantenimientos) que consiste obtener el programa óptimo de ubicación de los mantenimientos de las unidades de generación. En este caso en particular se trabajó sobre el sistema de generación del Uruguay y se buscó optimizar la ubicación de las dos centrales Turbo Vapor de mayor porte.

Si bien el problema planteado puede parecer de muy pocas variables para el potencial de la herramienta OddFace, como se trata de validar la herramienta, se buscó tener un problema “maneable” y gracias a eso durante éste trabajo se lograron identificar varios bugs de la implementación de OddFace y corregirlos.

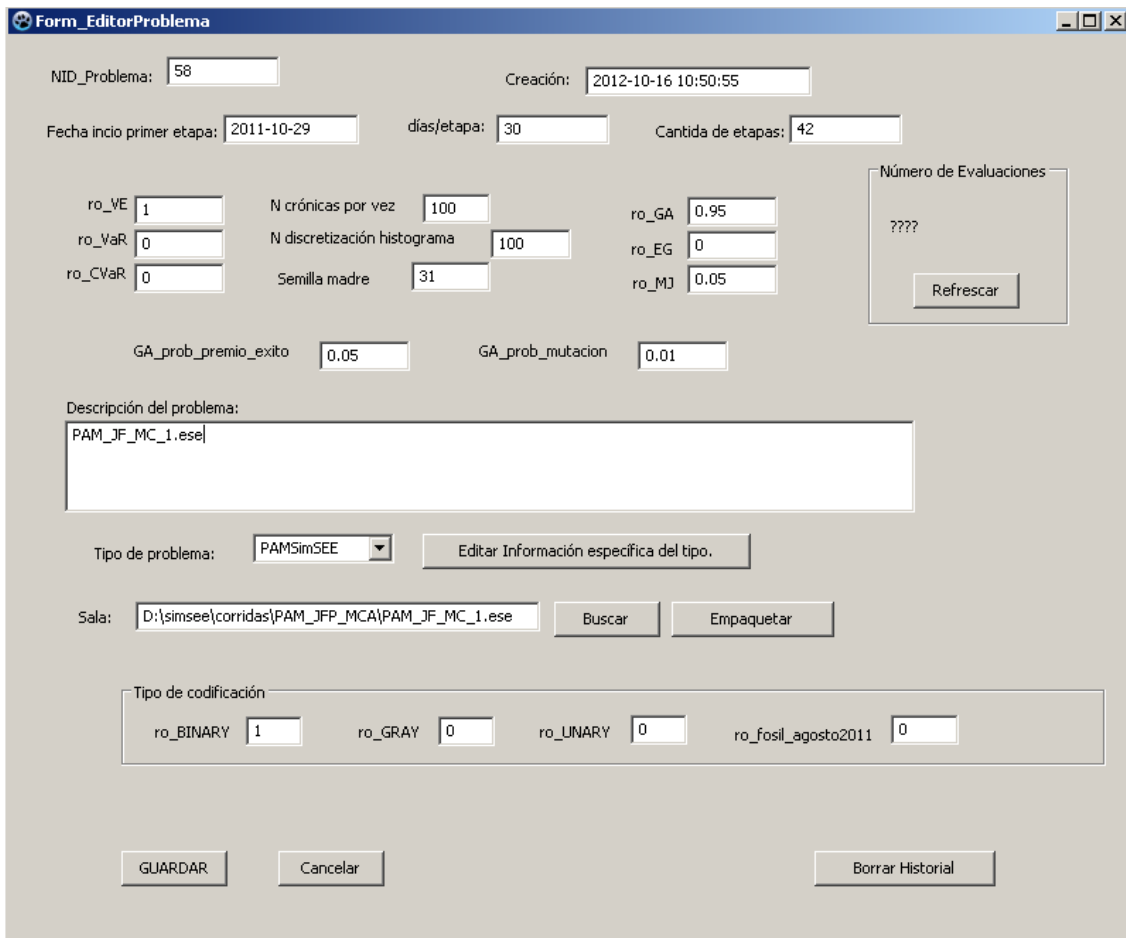
A. Definición del problema

Nos vamos a referir en todo momento al análisis del OddFace con un problema de PAM o sea de mantenimientos y los mantenimientos a ubicar dentro del período en consideración, son los de la 5ta y 6ta unidad de la Central Batlle.

Asimismo en todo momento la sala de SimSEE utilizada para este análisis tiene como fechas de optimización del 29/10/2011 al 29/10/2014 y de simulación del 29/10/20011 al 3/11/2012.

El objetivo principal del problema corrido con Oddface, es ubicar los mantenimientos de la 5ta y la 6ta, de forma tal de minimizar el costo de abastecimiento del sistema. Esta función de costo se llama “f_objetivo”.

Al editar el problema Oddface se deben definir una serie de parámetros que se detallan a continuación.



Form_EditorProblema

NID_Problema: 58 Creación: 2012-10-16 10:50:55

Fecha inicio primer etapa: 2011-10-29 días/etapa: 30 Cantidad de etapas: 42

ro_VE 1 N crónicas por vez 100 ro_GA 0.95
 ro_VaR 0 N discretización histograma 100 ro_EG 0
 ro_CVaR 0 Semilla madre 31 ro_MJ 0.05

GA_prob_premio_exito 0.05 GA_prob_mutacion 0.01

Número de Evaluaciones
 ????

Refrescar

Descripción del problema:
 PAM_JF_MC_1.esel

Tipo de problema: PAMSimSEE Editar Información específica del tipo.

Sala: D:\simsee\corridas\PAM_JFP_MCA\PAM_JF_MC_1.esel Buscar Empaquetar

Tipo de codificación
 ro_BINARY 1 ro_GRAY 0 ro_UNARY 0 ro_fosil_agosto2011 0

GUARDAR Cancelar Borrar Historial

a) Los campos NiD_Problema y Creación los crea el Oddface de forma automática y son el número que le asigna al problema en estudio y la fecha en que es creada.

b) El campo “fecha inicio primera etapa” es la fecha a partir del cual el Oddface considera como la primer fecha en la cual puede empezar a evaluar donde ubicar los mantenimientos en el período que se le define.

c) El campo “días/etapas” define los números de días entre etapas donde se podría empezar el mantenimiento

d) El campo “Cantidad de etapas” considera cuantas etapas hay para ubicar los mantenimientos, en nuestro caso 42 etapas, lo que si consideramos desde la fecha de inicio (29/10/2011) y con 42 etapas de 30 días, el período total del oddface iría hasta el 2015. Se puede ver el detalle de las etapas consideradas en este caso, mas adelante en el documento, cuando se detalla el caso de estudio analizado.

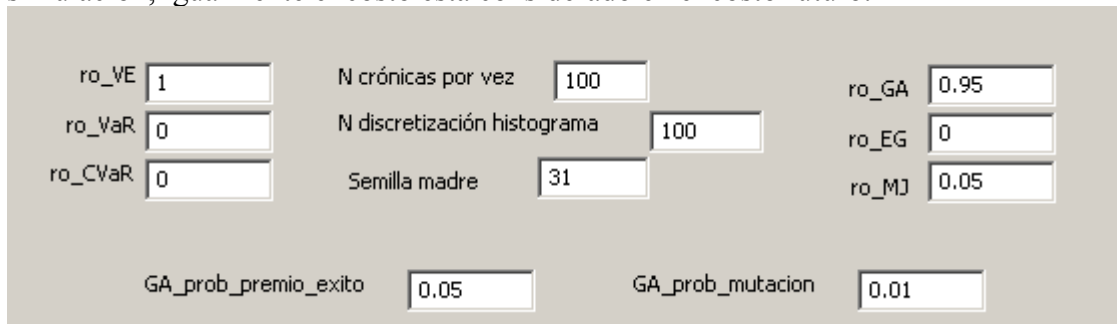
Pero si se clickea la ventana “Editar información específica del tipo” se observa:

NID	Nombre.	Días de parada.	Unidades.	Primera fecha.	Última fecha.		
11	CB-5ta-FOP	30	1	2011-10-29	2012-11-03		
10	CB-6ta-FOP	30	1	2011-10-29	2012-11-03		

Y se ve que la ultima fecha donde puede ubicar los mantenimientos es el 3/11/2012, con lo cual, mas allá de la cantidad grande de etapas consideradas el Oddface va a ubicar los mantenimientos en las primeras 13 etapas (ver detalle de etapas y fechas en el análisis del caso en estudio)

Pero si un mantenimiento empieza el 23/10/2012, va a finalizar el 22/11/2012, con lo cual excede la fecha final de la simulación y no estaría, por lo tanto, bien definido el problema Oddface analizado por nosotros, pero el costo que evalúa el SimSEE es el costo del paso mas el costo

futuro, con lo cual, si bien nos puede quedar algún mantenimiento fuera de la ventana de simulación, igualmente el costo está considerado en el costo futuro.



ro_VE	1	N crónicas por vez	100	ro_GA	0.95
ro_VaR	0	N discretización histograma	100	ro_EG	0
ro_CVaR	0	Semilla madre	31	ro_MJ	0.05
GA_prob_premio_exito		0.05		GA_prob_mutacion	
				0.01	

e) Ponderaciones Valor esperado, Valor en riesgo y valor en riesgo condicionado (ro_VE, ro_VaR y ro_CVaR)

Son los factores que se utilizarán para calcular la función objetivo del problema:

$$f_{\text{Objetivo}} = ro_VE * f_VE + ro_VaR * f_VaR + ro_CVaR * f_CVaR,$$

siendo f_VE , f_VaR y f_CVaR los valores estimados del Valor Esperado, del valor en riesgo y del condicional del valor en riesgo del costo por la simulación de la sala modificada.

d) A continuación se puede establecer el número de crónicas, en cuantas crónicas se quiere discretizar los histogramas y el posible cambio de la semilla madre. En esta primera versión del Oddface, se dejan los parámetros número de crónicas y número de crónicas para el histograma, iguales entre sí. La idea es que esto cambie para otras versiones.

e) La forma en que va a evaluar el próximo punto se fija en ro_GA, ro_EG y ro_MJ. Lo que significan es que se haga por el algoritmo genético (GA), o por estimación del gradiente (EG, no implementado en esta versión) o por una revaluación (MJ).

f) GA_premio_exito. Este parámetro regula el comportamiento de selección del algoritmo genético.

g) GA_prob_mutación. Cada individuo a evaluar se identifica con su ADN. Luego de que se selecciona el individuo se le puede aplicar con esta probabilidad, una mutación, donde se invierten bits de la cadena de ADN que identifica al individuo.

h) Número de evaluaciones

Al hacer click en “refrescar”, se da el número de evaluaciones que el Oddface ha hecho al problema en consideración, desde que se comenzó la corrida hasta ese momento

i) Descripción del problema: figura el nombre de la corrida SimSEE que el Oddface está analizando.

j) Tipo de problema: puede ser un problema de mantenimientos (PAM) o de planificación de Inversiones de Generación (PIG). En nuestro caso será PAM.

k) Editar información específica de tipo: muestra una ventana donde figura cual o cuales son las unidades a determinar el mantenimiento y en que período (ver d)).

l) Sala: con el botón “buscar” se busca la sala a aplicarle el Oddface y con el botón “empaquetar” se sube la corrida al cluster de facultad.

2) Consultas

Cuando se ingresa al Oddface, se puede ver las corridas que se han subido y si las mismas están corriendo (semáforo verde) o si están detenidas (semáforo en rojo y renglón en rojo).

OddFace - 1.1-GEA (c) IIE-FING-2012

Problemas Ingreso

Listados de Problemas Crear Nuevo

NID	dt_creacion	tipo	Descripcion		
141	2012-11-24 11:13:33	2	PAM_JF_MC_1_VE=0.85,CVaR=0.15.ese		
140	2012-11-24 10:32:09	2	PAM_JF_MC_1_10d_por_etapa.ese		
139	2012-11-24 10:29:14	2	PAM_JF_MC_1_15d_por_etapa.ese		
138	2012-11-23 15:39:53	2	Idem 102 con nueva version oddface		
137	2012-11-22 20:57:37	2	Idem 117 con nueva version oddface		
136	2012-11-22 10:54:47	1	Plan generación 2013-2029_99 (GEA_CLON del 129) -- Va de nuevo con corrección Factor UG/UI _		
135	2012-11-17 20:27:18	2	Prueba 14DC_Grupo PAM BINARY=0.5 UNARY=0.5		
134	2012-11-13 22:45:37	2	Prueba 14E_Grupo PAM 5ta=30 6ta=30		
132	2012-11-12 21:14:20	2	PAM_JF_MC_1_VE=0.5,VaR=0.5.ese		
131	2012-11-12 20:46:16	2	Prueba 14AC_Grupo PAM ro_MJ=0		
130	2012-11-12 20:24:45	2	PAM_JF_MC_1_VE=0.85,VaR=0.15.ese		
129	2012-11-10 21:45:43	1	Plan generación 2013-2029_99 (CLON del 82) -- Va de nuevo con corrección Factor UG/UI		
125	2012-11-08 21:17:30	2	Prueba 14CC_Grupo PAM premio_exito=0.10 mutacion=0.01		
124	2012-11-08 21:15:23	2	Prueba 14CB_Grupo PAM premio_exito=0.20 mutacion=0.01		
123	2012-11-08 21:13:40	2	Prueba 14CA_Grupo PAM premio_exito=0.20 mutacion=0.05		
122	2012-11-08 20:57:06	2	Prueba 14DB_Grupo PAM BINARY=0.33 GRAY=0.33 UNARY=0.33		
121	2012-11-08 20:51:43	2	Prueba 14AB_Grupo PAM VE=0.75 VaR=0.25		

Refrescar

Si se clickea en “editar” se observa todo lo descrito en el punto anterior
 Pero si se opta por realizar una consulta, se observa:

Consultar

SELECT nid, adn, f_Objetivo, cnt_evaluacione dtc, dtu

FROM ofe_individuos_58 Aglutinar Redundantes

WHERE 1

ORDER BY nid DESC

LIMIT 0, 10000

Ejecutar consulta -> Excel

Ejecutar consulta -> archivo_XLT

Bajar un individuo.

NID: 0 BAJAR INDIVIDUO

Análisis ADN

memo_ADN

Campos de la tabla de evaluaciones:

nid int
 adn varbinary(800)
 dtc datetime
 dtu datetime
 f_VE double
 f_CVaR double
 f_VaR double
 f_Objetivo double
 f_MIN double
 f_MAX double
 cnt_evaluaciones int

Donde se puede elegir cuantos individuos bajar, como se quiere ordenar la consulta y si se quiere bajar la consulta a un archivo Excel o simplemente a un archivo xlt.

Si se escribe “ DESC” en “ORDER BY” se ordena la consulta en forma decreciente. Puede ser útil no solo bajarla por NID (identificador del problema) sino también por f_Objetivo para ver las de menor costo directamente.

Al efectuar la consulta, se observan varias columnas:

2012-11-14 15:50:58, sql: SELECT nid, adn, f_Objetivo, dtc, dtu, cnt_evaluaciones FROM ofe_individuos_58 WHERE 1 ORDER BY nid DESC LIMIT 0, 10000

nid	adn	f_Objetivo	dtc	dtu	cnt_evaluaci	e_CB-6ta-FO	e_CB-5ta-FOP_11_4
70947	1811	1688.9864	14/11/2012 15:50	14/11/2012 15:50	1	8	1
70946	6F1B	1670.13063	14/11/2012 15:50	14/11/2012 15:50	1	12	6
70945	FFA3	1706.3567	14/11/2012 15:48	14/11/2012 15:48	1	12	12
70944	52C4	1723.86703	14/11/2012 15:48	14/11/2012 15:48	1	2	5
70943	69D1	1663.64975	14/11/2012 15:47	14/11/2012 15:47	1	9	6
70942	466	1723.92773	14/11/2012 15:47	14/11/2012 15:47	1	4	0
70941	676D	1734.35927	14/11/2012 15:46	14/11/2012 15:46	1	7	6
70940	66BC	1710.08916	14/11/2012 15:45	14/11/2012 15:45	1	6	6

La primer fila muestra cuando y como se hizo la consulta.

La primer columna especifica el individuo evaluado, identificándose con su NID y su ADN. Posteriormente se observa el valor de f_Objetivo y los dtc y dtu que son la fecha y la hora, respectivamente, de la primera y la última evaluación del punto.

Posteriormente el “Cnt_evaluaciones” muestra el numero de evaluaciones que tuvo el punto.

Las dos últimas columnas muestran en que etapa comienzan los mantenimientos buscados para la unidad o las unidades generadoras objeto del estudio, en el caso que estamos analizando, serían la 6ta y la 5ta unidad de Central Batlle.

B. Caso de estudio analizado

Como ya se mencionó nos vamos a referir en todo momento al análisis del OddFace con un problema de PAM o sea de mantenimientos y los mantenimientos a ubicar dentro del período en consideración, son los de la 5ta y 6ta unidad de la Central Batlle.

Asimismo en todo momento la sala de SimSEE utilizada para este análisis tiene como fechas de optimización del 29/10/2011 al 29/10/2014 y de simulación del 29/10/20011 al 3/11/2012 y la sala SimSEE corresponde a la corrida realizada por ADME (Administración del Mercado Eléctrico) para la Programación Estacional de noviembre 2011 a abril 2012, vale decir que posee el parque generador Uruguayo existente o entrante en un futuro cercano, tal cual lo consideró, en su momento, el grupo de técnicos que realizaron esta corrida. Los escalones de falla considerados en la corrida no son los reglamentarios sino que fueron calculados para que el SimSEE no despachara falla por despacho económico. Es así que la falla 1 se considera en esta corrida en 302 USD/MWh, falla 2 se considera en 400 USD/MWh, falla 3 en 1200 USD/MWh y falla 4 en 2000 USD/MWh.

Al elegir como fecha de inicio de la primera etapa del Oddface el 29/10/2011 y considerar 42 etapas de 30 días, las fechas de inicio de cada etapa son las que se muestran a continuación, donde las etapas que están en amarillo son las que entran dentro del período de simulación.

Si el mantenimiento llegara arrancar en la última etapa o sea la que comienza el 23/10/2012, terminaría fuera de la ventana de simulación, el 22/11/2012, con lo cual se ve que el problema Oddface no fue correctamente realizado para que el periodo donde colocar los mantenimientos entrara por completo dentro del período de simulación. Igualmente se considera el costo de este último mantenimiento, dado que el costo total es la suma del costo de cada paso más el costo futuro.

A continuación se muestran las fechas de las etapas, en amarillo se muestran las que están contempladas dentro del período de simulación de la sala SimSEE y el resto son las etapas comprendidas dentro de la optimización de la sala.

0	29/10/2011
1	28/11/2011
2	28/12/2011
3	27/01/2012
4	26/02/2012
5	27/03/2012
6	26/04/2012

7	26/05/2012
8	25/06/2012
9	25/07/2012
10	24/08/2012
11	23/09/2012
12	23/10/2012
13	22/11/2012
14	22/12/2012
15	21/01/2013
16	20/02/2013
17	22/03/2013
18	21/04/2013
19	21/05/2013
20	20/06/2013
21	20/07/2013
22	19/08/2013
23	18/09/2013
24	18/10/2013
25	17/11/2013
26	17/12/2013
27	16/01/2014
28	15/02/2014
29	17/03/2014
30	16/04/2014
31	16/05/2014
32	15/06/2014
33	15/07/2014
34	14/08/2014
35	13/09/2014
36	13/10/2014
37	12/11/2014
38	12/12/2014
39	11/01/2015
40	10/02/2015
41	12/03/2015

Al analizar el número total de evaluaciones que realizó el Oddface en el período considerado, se vio que hubieron 20.892 evaluaciones y que el resultado no tendía a un valor de f_{Objetivo} sino que iba cambiando, así como iban cambiando las etapas donde colocaba los dos mantenimientos objetivos de la sala SimSEE. Si se analiza el problema que consideramos en el oddface, se observa que es una situación donde se tienen dos variables a colocar cada una en 13 etapas, lo cual da un problema de 13×13 posibles soluciones totales, vale decir en total 169 soluciones posibles.

Esto hizo que se revisara como había sido implementado la codificación de los problemas, llamado ADN, ya que se podía observar que el Oddface volvía a evaluar soluciones ya encontradas pero que las consideraba distintos entre si, por su codificación. El ADN, o sea cada codificación de cada posible solución, se realiza con 16 bits cuando este problema solo necesita 8 bits para su codificación. Asimismo se observa que la etapa 12, o sea los extremos, tiene más probabilidad de

ser elegida por la redundancia de la codificación ya que cuando se decodifica un ADN hay bits redundantes que son asignados a la etapa 12.

Para poder eliminar las redundancias y solo considerar las 169 posibles soluciones, se realizó una nueva versión del Oddface, que permite bajar la tabla de soluciones eliminando la redundancia y esta tabla, es la que se analiza para examinar y documentar los resultados de Oddface en el presente documento.

La mejor solución encontrada es la de NID 52.398 con una f_{objetivo} de 1.709,1 MUSD y ubicando el mantenimiento de la 6ta en la etapa 3 (27/1/2012 a 26/2/2012) y el mantenimiento de la 5ta en la etapa 1 (28/11/2012 a 28/12/2012). A continuación se detalla y analiza dicha solución.

i. ANALISIS DE LA MEJOR SOLUCION

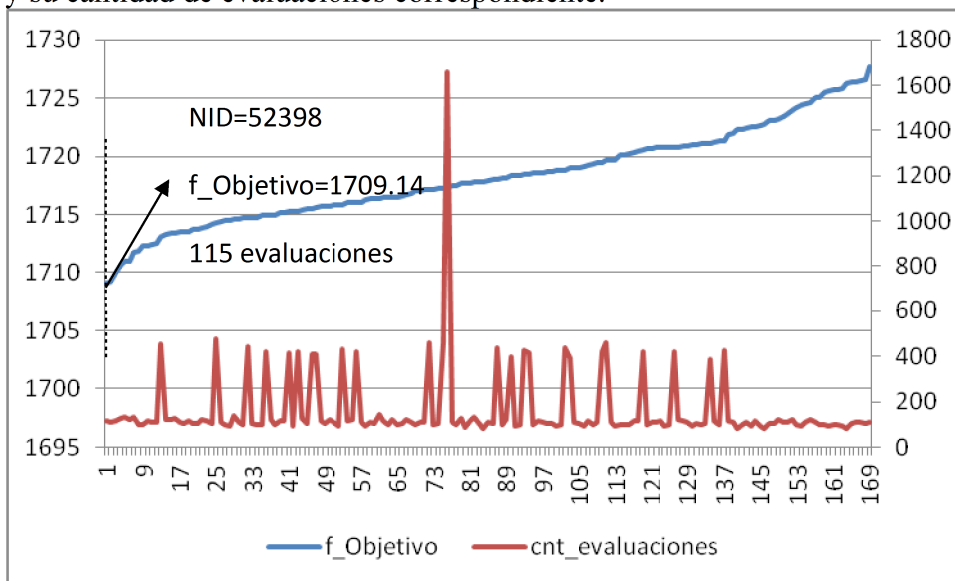
CASO BASE –prob. 58 (Versión original)

Como ya se mencionó, se tienen dos variables a colocar, cada una en 13 etapas, lo cual da un problema de 13×13 posibles soluciones totales, vale decir en total 169 soluciones posibles.

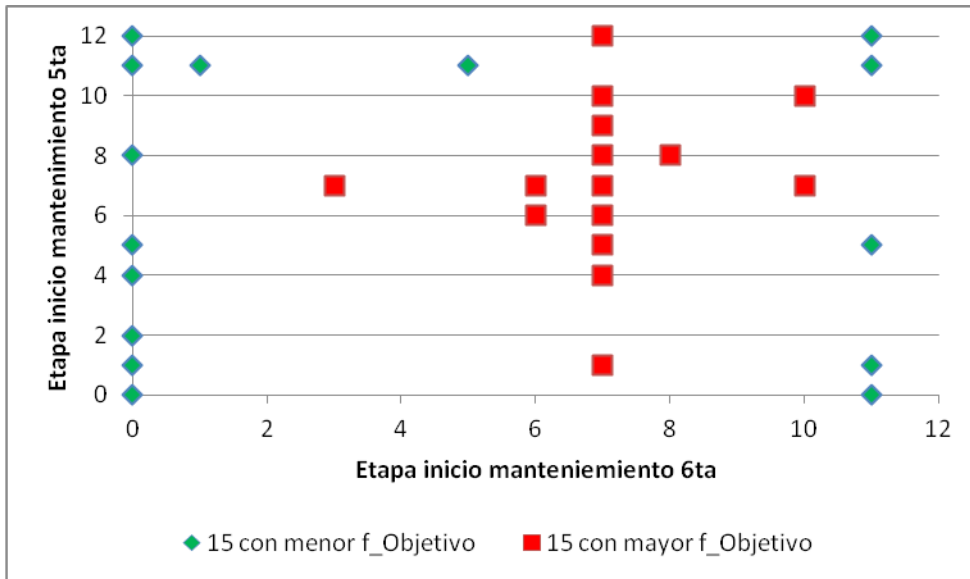
Se optimizó y buscó el individuo con menor f_{objetivo} pero que a su vez tuviera un número considerable de evaluaciones para asegurarnos que dicho valor bajo no fuera producto de un azar favorable hacia el individuo.

En nuestro caso, se consideró que el individuo 52398 cumplía con dicho requisito, ya que cuenta con 115 evaluaciones y presenta el menor valor de f_{objetivo}

La siguiente gráfica muestra los valores antes mencionados de los 169 individuos posibles (13×13) y su cantidad de evaluaciones correspondiente.



La siguiente gráfica muestra la dispersión de los mantenimientos para los 50 mejores y peores individuos

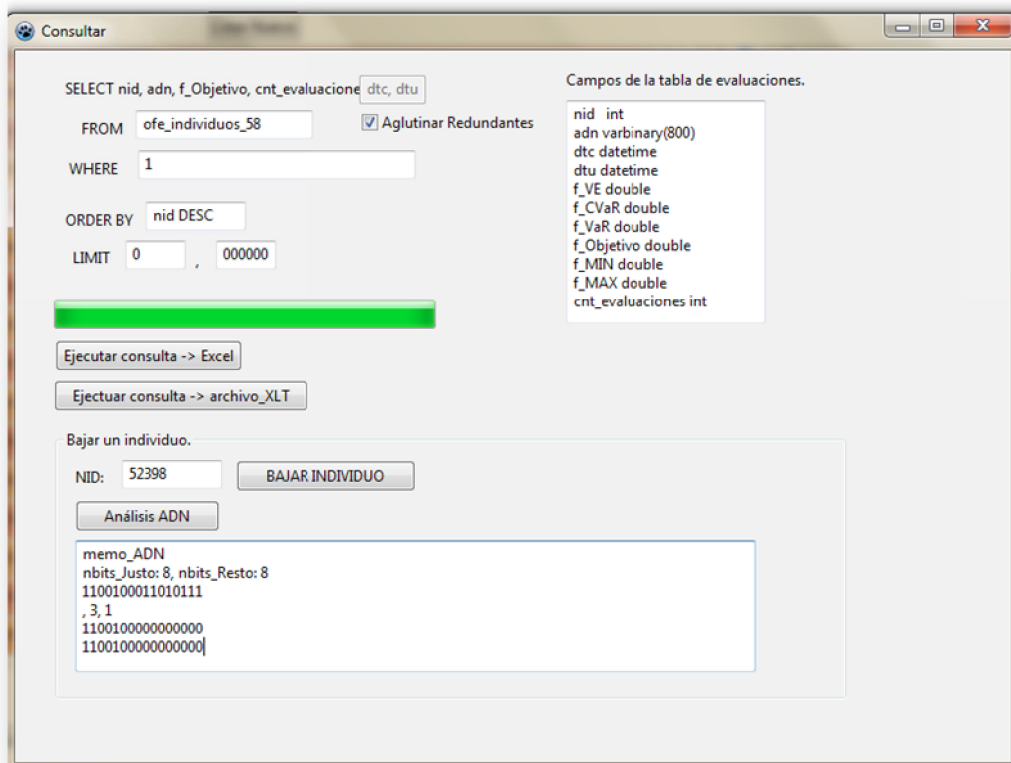


Para obtener los datos se tuvo que realizar una consulta del problema en cuestión, marcando la opción “aglutinar redundantes”. Esta opción se implementó en una versión posterior a la original, en la que cada individuo se identificaba a partir de 16 bits fijos. Como en nuestro caso de estudio hay 169 posibles combinaciones (13 posibles por cada uno de los 2 generadores), tan solo se necesitan 2 cadenas de 4 bits (16 posibilidades por cadena). De modo que quedan 8 bits sobrando que la herramienta modificaba y consideraba como distintos cuando en realidad representaban la misma elección de mantenimientos.

Por tanto, con el agregado de la opción “aglutinar redundantes”, se calculan las f_{Objetivo} del individuo real promediando de forma ponderada entre todos los redundantes según la cantidad de evaluaciones para un mismo ADN.

Se aconseja cambiar el límite de individuos (“LIMIT”) de 10000 (valor genérico) a 10000000 o un valor grande, de modo que se consideren todas las evaluaciones de cada individuo.

Consulta:



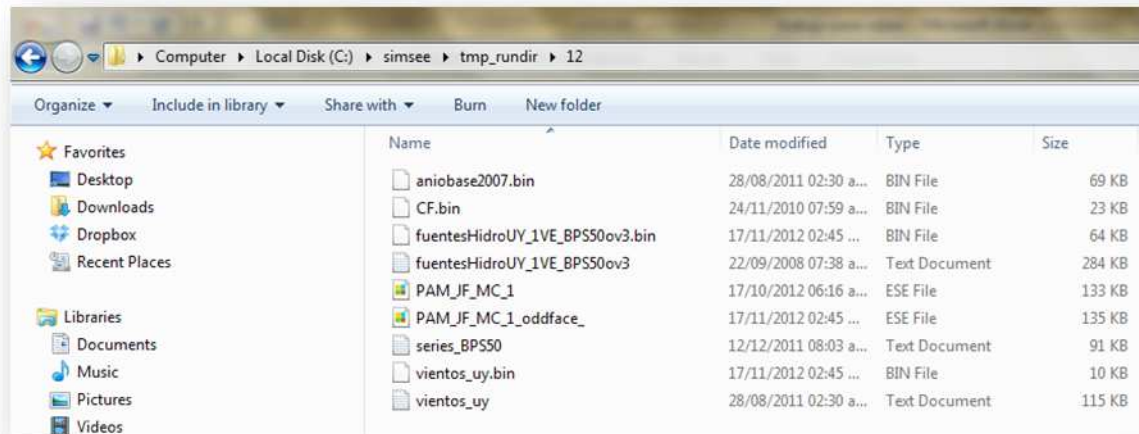
1		2012-11-18 15:27:28, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu FROM ofe
2		nid adn f_Objetivo cnt_evaluaciones CB-6ta-FOP CB-5ta-FOP_11_4
3	Elegido	52398 1300 1709.1431 115 3 1
4		75084 B500 1709.2317 111 5 11
5		70117 1800 1710.0168 115 11 1
6		57776 3000 1710.6036 125 0 3

A partir de la consulta realizada, se observa que para el individuo elegido, los mantenimientos se efectúan en la etapa 3 y 1 para la 6ta y 5ta unidad, que fueron nuestros parámetros de optimización. Como en nuestra sala oddface (58) las etapas son de 30 días y la misma arranca el 29/10/2011, debemos sumar a la fecha inicial el producto de la etapa elegida con los 30 días/etapa:

$$\text{Mant } 5^{\text{ta}} = 29/10/2011 + 1 \times 30 = 28/11/2011$$

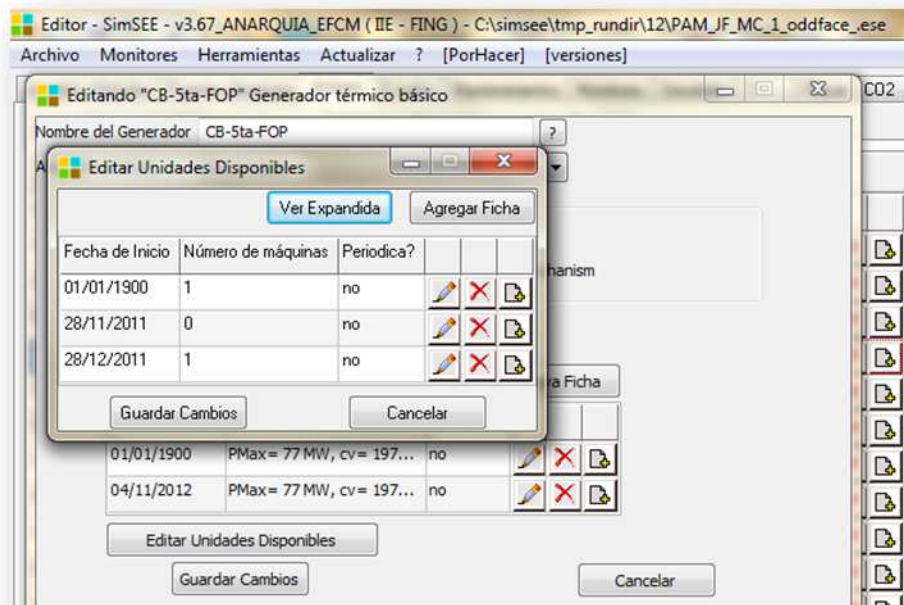
$$\text{Mant } 6^{\text{ta}} = 29/10/2011 + 3 \times 30 = 27/01/2012$$

Luego, se procedió a bajar el individuo para constatar que el mismo tuviera los mismos mantenimientos. La sala correspondiente se guarda automáticamente en C:\simsee\tmp_rundir\12 y lleva el nombre de la sala original seguido de “_oddface_”. También aparecen los archivos necesarios para la corrida (demanda, CF, series vientos, series aportes, CEGH aportes, etc) junto con la corrida original.

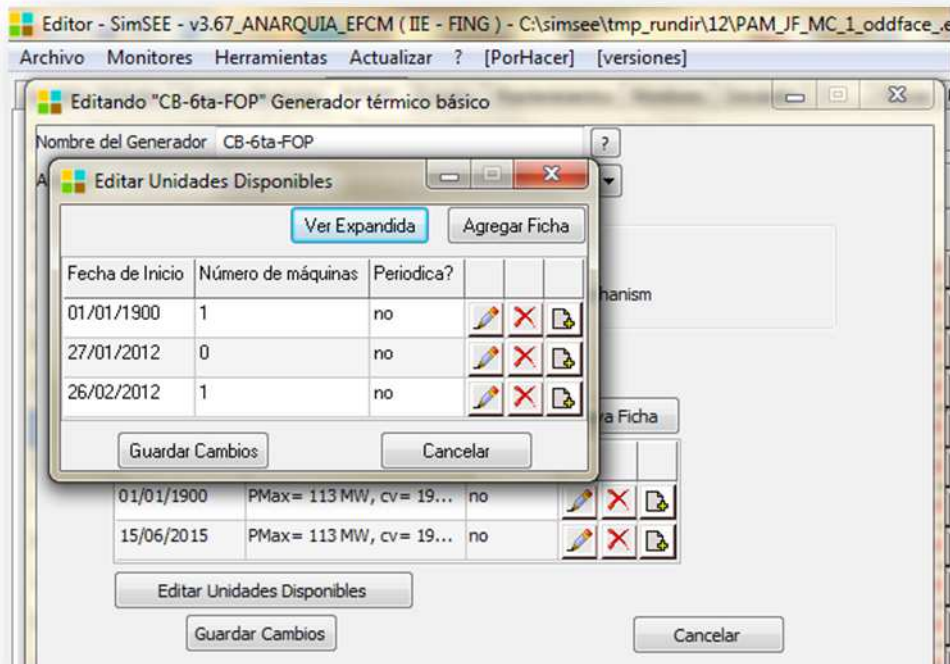


Abriendo el individuo elegido (52398), constatamos que los mantenimientos coinciden con lo señalado por la consulta:

5ta



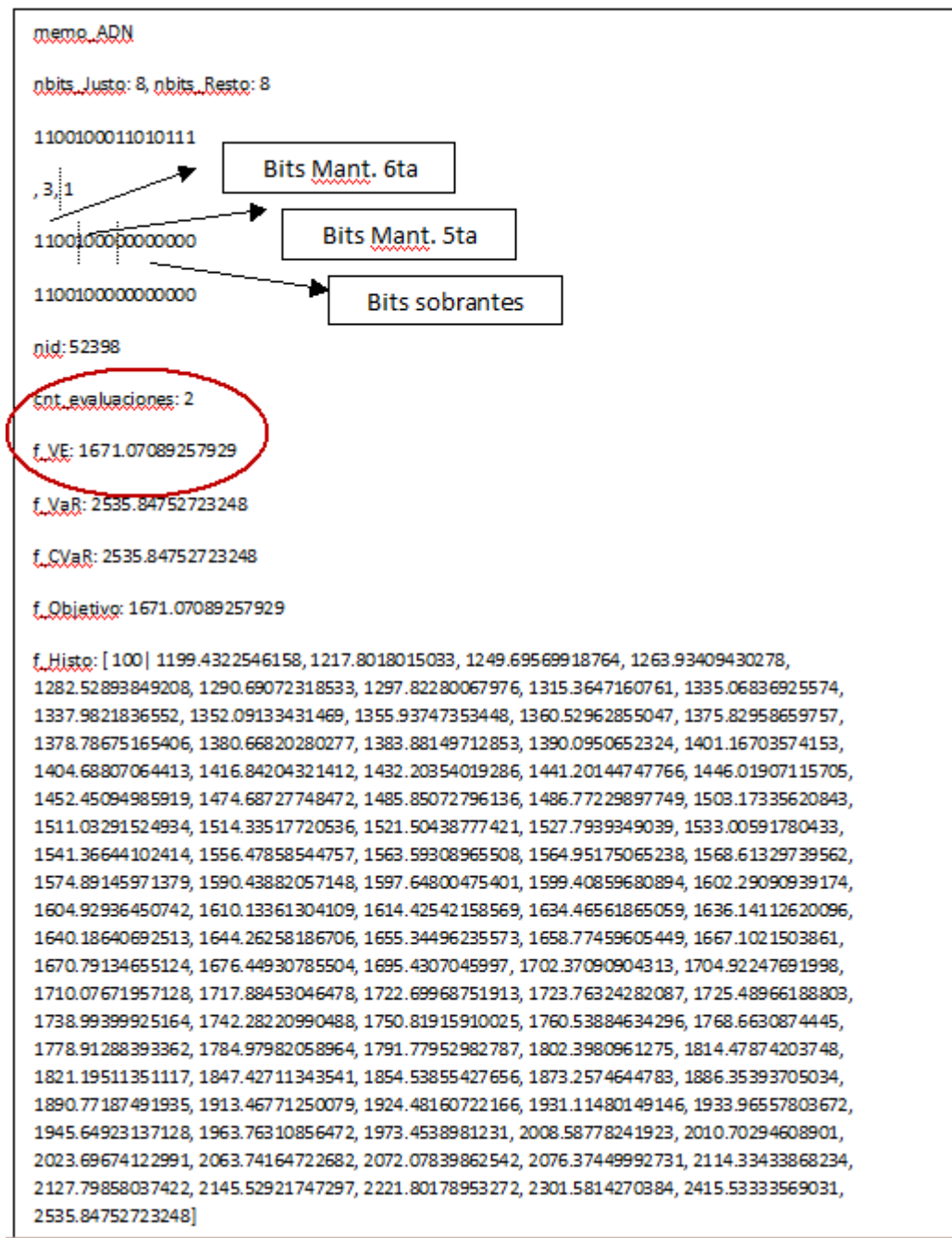
6ta



En el cuadro de abajo se observa el análisis de ADN del individuo 52398. Aparece en el cuadro luego de escribir el número de individuo y hacer click en la opción “Análisis ADN”.

En el mismo se muestra f_{VE} , $f_{Objetivo}$, f_{VaR} , f_{CVaR} , nid , $cnt_evaluaciones$ y f_Histo que es el histograma ordenado de valores medios promediados en las distintas evaluaciones.

Análisis ADN ind. 52398



Se muestra la decodificación binaria para la 6ta y 5ta unidad así como los bits sobrantes o redundantes. Como se puede apreciar, los valores de cantidad de evaluaciones y f_VE o $f_Objetivo$ (son iguales en este problema) no coinciden con los resultados arrojados por la consulta. Estos valores así como el historial corresponden al individuo 52398 pero sin aglutinar resultados redundantes. Para comprobarlo, se realiza una nueva consulta pero sin aglutinar redundantes, filtrando solo los individuos con 6ta y 5ta en las etapas 3 y 1.

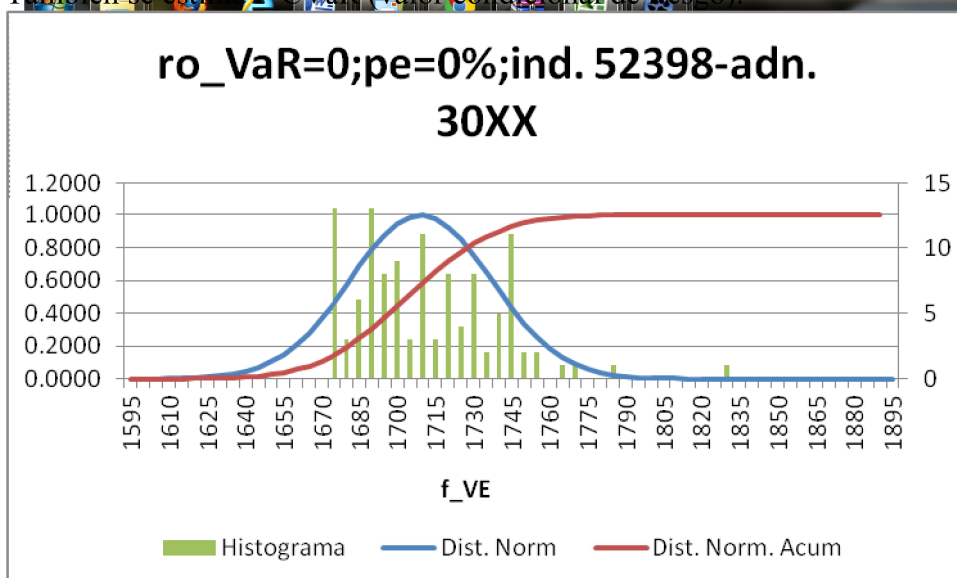
1	2012-11-29 21:25:53, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu, f_VE, f_VaR, f_CVaR, f_MIN, f_MAX FROM ofe_individuos_58 WHERE 1 ORDER BY nid DESC LIMIT 0, 100000000																	
2	nid	adn	f_Objetivo	cnt_evaluas	dtc	dtu	f_VE	f_VaR	f_CVaR	f_MIN	f_MAX	e_CB-6ta-FO	e_CB-5ta-FOP	11_4				
72	52790	13CC	1707.42625	1	12:36:57	12:36:57	1707.42625	2667.70374	2367.70374	1124.18602	2667.70374	3	1	1707.42625				
73	52764	13EC	1681.85668	3	12:24:50	17:09:57	1681.85668	2697.41518	2397.41518	1104.46887	2804.59938	3	5	1045.57005				
74	52657	134E	1689.6678	3	11:39:08	10:42:06	1689.6678	2703.49902	2703.49902	1076.2273	2894.88152	3	1	5069.0034				
75	52398	13EB	1671.07089	2	8:58:17	20:51:24	1671.07089	2535.84753	2335.84753	1158.35028	2652.35971	3	3	3342.14179				
76	51972	139A	1717.44112	1	4:54:16	4:54:16	1717.44112	2665.07688	2665.07688	1146.90093	2665.07688	3	1	1717.44112				
77	51174	132A	1748.49552	1	21:49:11	17:48:49	1748.49552	2621.63471	2321.63471	1179.24977	2621.63471	3	1	1748.49552				
78	51088	1313	1728.22209	1	21:18:58	21:18:58	1728.22209	2695.99003	2395.99003	1165.87316	2695.99003	3	1	1728.22209				
79	50611	1370	1725.01269	2	18:07:37	14:37:09	1725.01269	2574.22497	2374.22497	1197.40572	2589.34061	3	1	3450.02538				
80	50578	13FC	1750.91315	1	17:54:39	17:54:39	1750.91315	3142.8306	3142.8306	1203.90551	3142.8306	3	1	1750.91315				
81	50180	130F	1686.60213	2	15:02:37	19:11:38	1686.60213	2567.63617	2367.63617	1099.30125	2652.35971	3	1	3373.20427				
82	49936	1369	1747.51579	1	13:06:18	13:06:18	1747.51579	2668.23497	2368.23497	1149.17672	2668.23497	3	1	1747.51579				
83	49883	135B	1686.12922	1	12:43:01	12:43:01	1686.12922	2492.30967	2492.30967	1220.54371	2492.30967	3	1	1686.12922				
84	49662	1379	1682.40455	2	10:56:10	21:38:55	1682.40455	2592.35943	2392.35943	1132.35929	2663.53297	3	1	3364.80911				
85	Evaluaciones ADN 13XX=			115														
ee													f_VE ADN 13XX=				1709.14313	

Se muestra subrayado el individuo 52398 y recuadrados los valores de cantidad de evaluaciones y f VE calculados a mano, coherentes con los obtenidos de la consulta.

En posteriores versiones se corrige este defecto, mostrándose en el análisis los resultados del ADN en cuestión y no el de un individuo en particular.

A partir de éstos valores, se construye el histograma de valores medios y su distribución normal estimada.

También se está f CVaR (valor condicional de riesgo)



Eval. Totales Problema	27321
Eval individuo	115
f_Objetivo	1709.14
f_VE	1709.14313
f_CVaR (est.)	1880
(f_CVaR- f_VE)/f_VE	10 %
Desvío Standard	27.7604434
%desv.est/f_VE	1.62423163

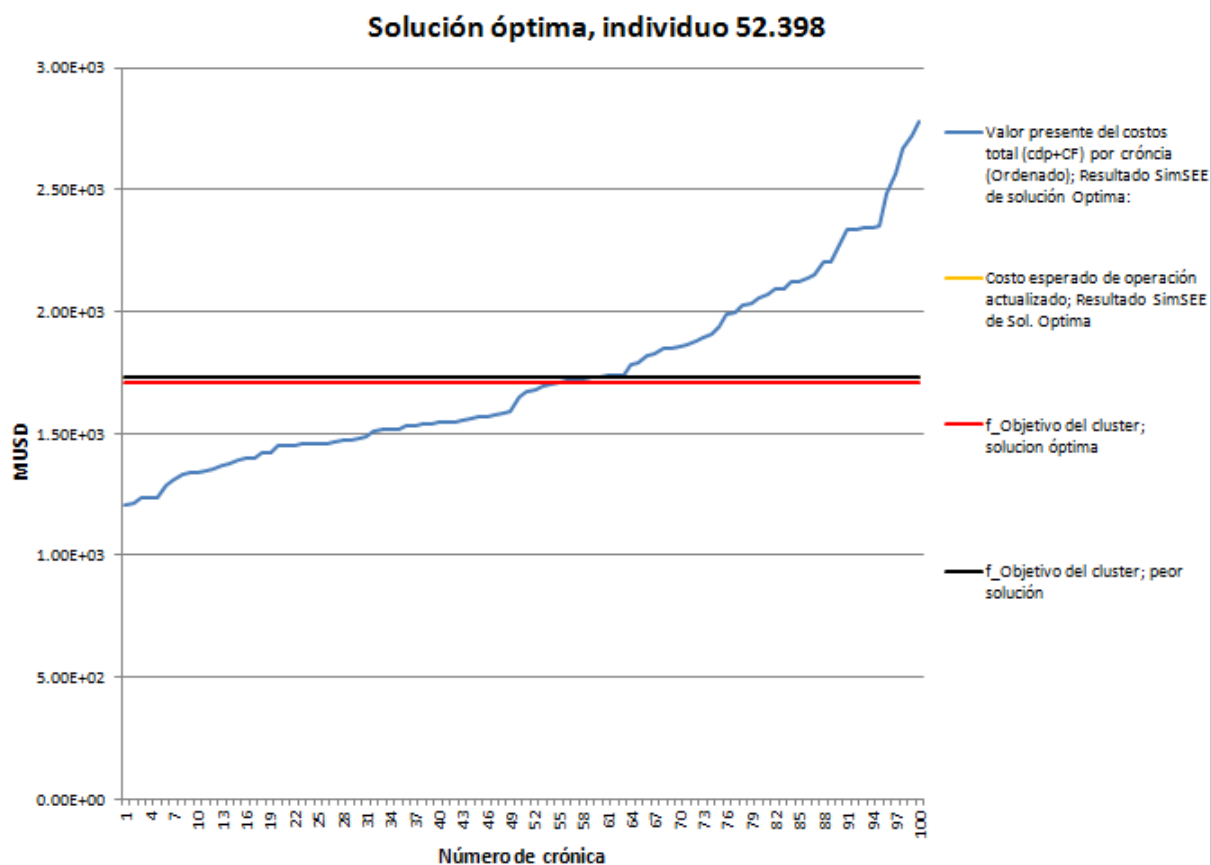
ii. ANALISIS DEL CONJUNTO DE SOLUCIONES FACTIBLES

Como ya se mencionó, en el problema principal analizado, se tienen dos variables que pueden tomar 13 valores cada una, vale decir 13×13 o sea 169 soluciones factibles.

a) Análisis de la solución óptima

La solución óptima (NID 52.398) se baja a un pc común y se optimiza y simula con 100 crónicas en el SimSEE. Esto se realiza para mostrar el nivel de variabilidad de la solución óptima.

Se muestra a continuación un gráfico del resultado económico de la solución óptima encontrada, analizando la tabla simcosto 31×100 obtenida después de optimizar y simular en el SimSEE, comparándolo con el f_{objetivo} obtenido corriendo el Oddface en el cluster de la facultad de ingeniería.

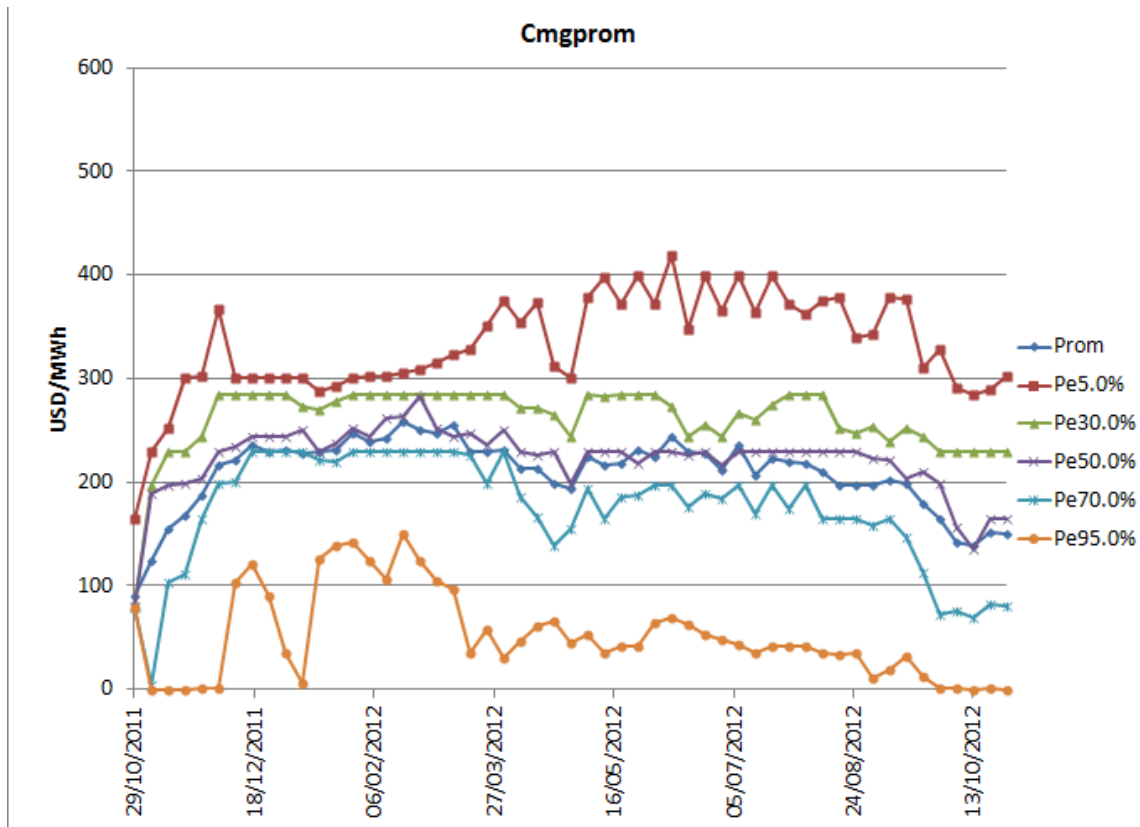


La curva en azul muestra la variabilidad de la solución óptima encontrada y fue hallada realizando la corrida en el SimSEE con 100 crónicas de optimización y simulación, la línea roja muestra la solución del cluster f_{objetivo} óptima y la línea negra muestra la solución peor obtenida en el cluster.

Se observa que la diferencia entre el costo esperado de operación actualizado (1.732,6 MUSD), obtenido de la planilla resultante de optimizar y simular en el SimSEE y la f_{objetivo} obtenida (1.709,1 MUSD) de correr el Oddface en el cluster de facultad, es de 1,38 %, o sea no se aprecian diferencias significativas entre la realización de la corrida con el SimSEE y lo obtenido con el Oddface en el cluster de la facultad de ingeniería.

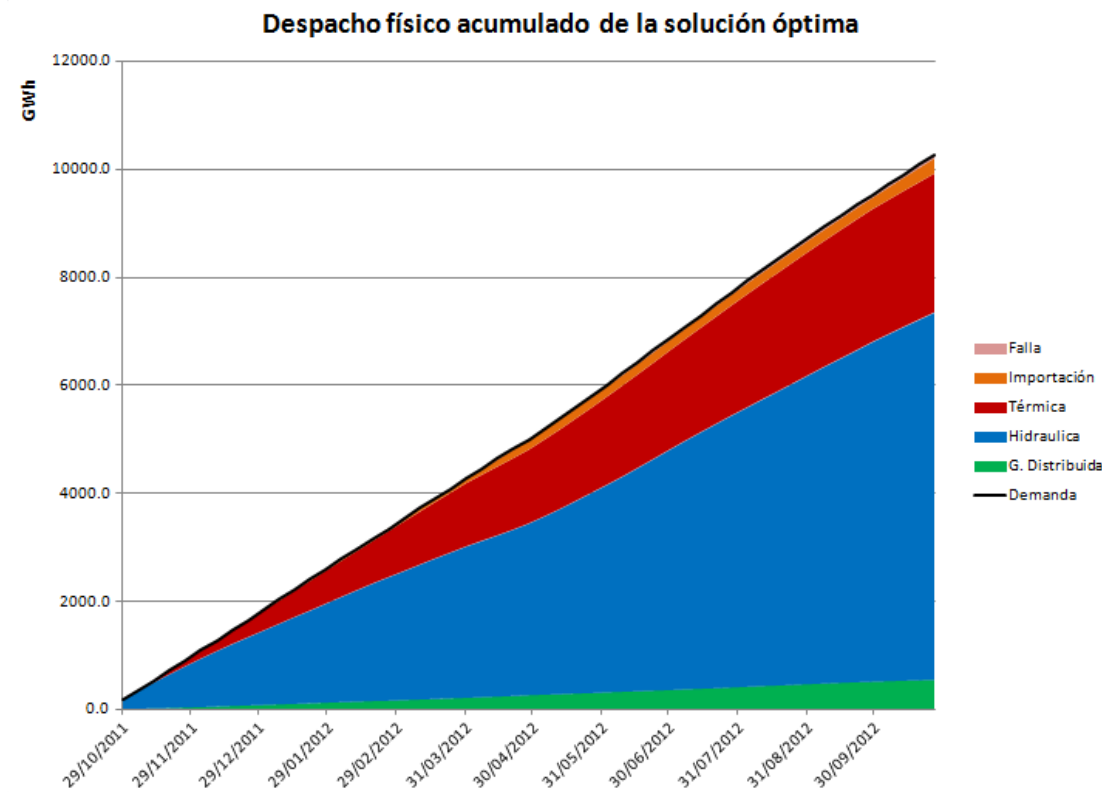
La diferencia entre la f_{objetivo} de la solución óptima y la peor solución, ambas obtenidas del cluster de facultad, es tan solo de un 1,09%. Por lo que podemos ver que las diferencias entre las soluciones del problema objetivo son muy pequeñas.

Asimismo se muestra el costo marginal de operación promedio obtenido optimizando y simulando la solución óptima en el SimSEE:



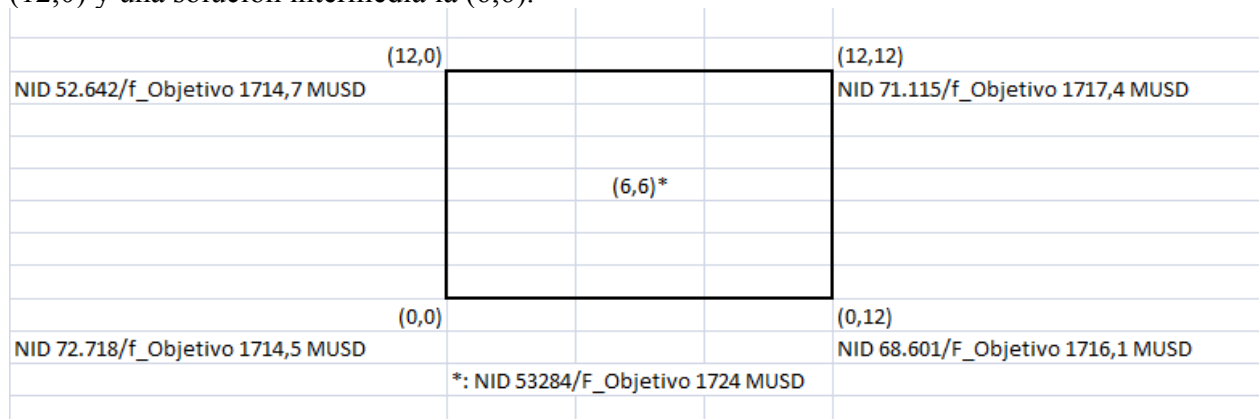
Siendo el promedio total en todo el periodo de simulación de 208,7 USD/Mwh.

A continuación se muestra el despacho físico de la energía acumulada en el período por tipo de fuente, según surge de la optimización y simulación con el SimSEE:



b) Análisis del resto de las soluciones factibles

Para analizar las variaciones en este conjunto de 169 soluciones posibles y que la función de costo se aproxima a la hallada con la herramienta Oddface, se optimiza y simula en el SimSEE, cada solución en los extremos de las soluciones factibles, esto es las soluciones (0,0); (0,12); (12,12) y (12,0) y una solución intermedia la (6,6).



Nota: el dibujo anterior muestra el “cuadrado” formado por las 169 posibles soluciones (13*13) y los individuos que bajamos y optimizamos y simulamos con 100 crónicas sintéticas en el SimSEE.

Resumen de las principales características de estos individuos:

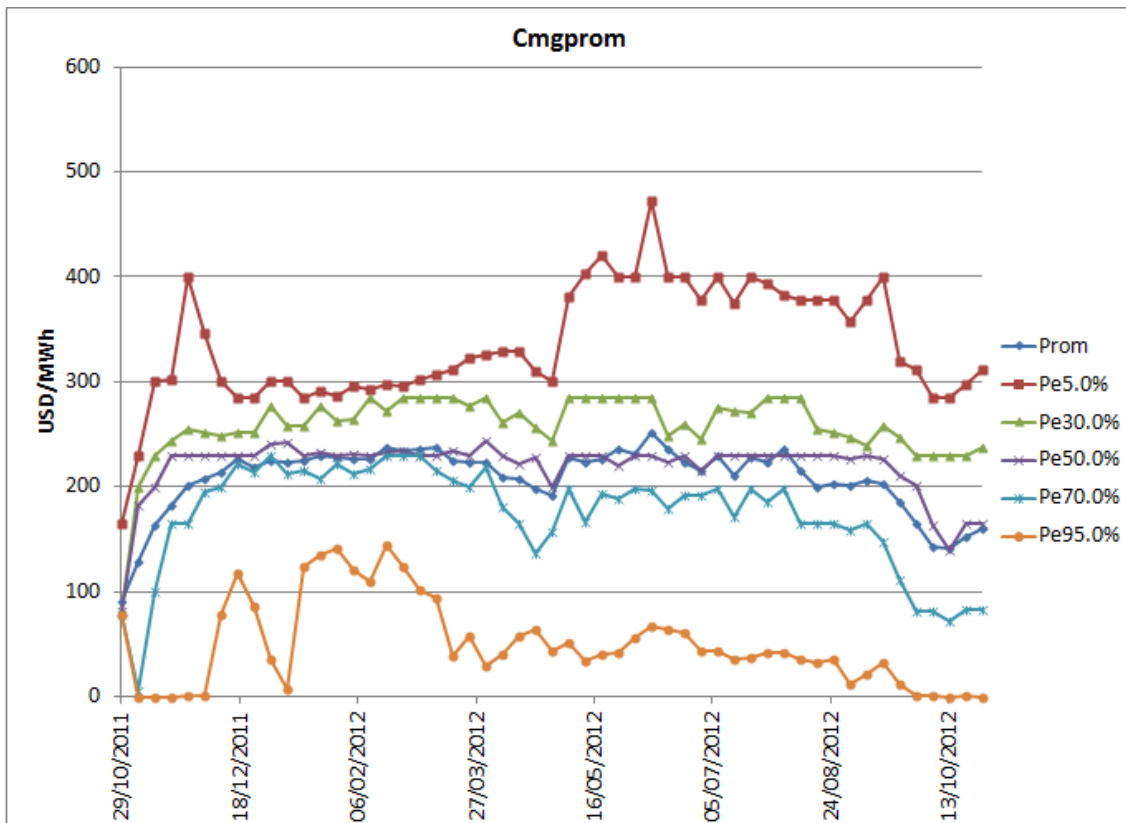
Solucion	NID	F_objetivo (MUSD)	Desv standard de valor presente de costos total
(0,0)	72718	1,714.5	364.1
(0,12)	68601	1,716.1	364.6
(12,12)	71115	1,717.4	364.7
(12,0)	52642	1,714.7	364.3
(6,6)	53284	1,724.0	364.6

Nota: La desviación estándar se realiza sobre los archivos resultantes de realizar la optimización y simulación con el SimSEE.

A continuación procederemos a analizar las corridas de las esquinas de este cuadrado y compararemos el “costo esperado de operación actualizado” de estos individuos con el del individuo elegido como mejor solución con NID 52398, cuyo valor de costo de f_objetivo es de 1709,1 MUSD. Para ello, identificaremos a estos individuos, bajaremos las salas SimSEE correspondientes y realizaremos la optimización y simulación de las mismas.

B1). Es así que el vértice superior izquierdo, o sea el que considera los mantenimientos de la 6ta en la etapa 12 y la 5ta en la etapa 0, corresponde al individuo con NID 52.642. Este individuo tiene una f_objetivo de 1714,7 MUSD.

Al optimizar y simular este individuo, el resultado da un costo marginal promedio que se muestra a continuación:

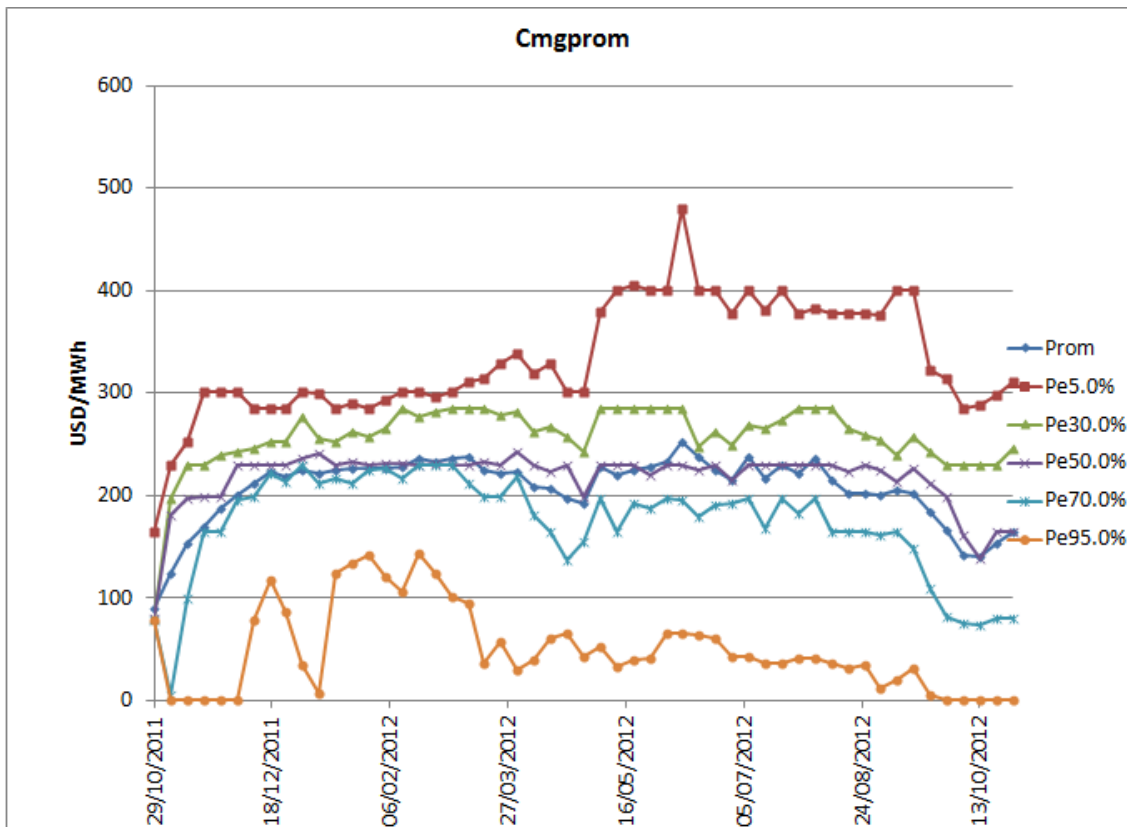


Donde el promedio del cmg promedio en todo el período de simulación considerado es de 207,7 USD/MWh.

El resultado Simcosto 31×100 del resultado de correr la sala con el NID 52.642 en el SimSEE, nos muestra que el costo esperado de operación actualizado es de 1730,6 MUSD. Con lo cual vemos que tenemos un 1,26% mas en el costo que el resultado del costo del f_{objetivo} del Oddface.

B2) A continuación se realiza el mismo análisis para el vértice superior derecho del cuadrado, vale decir el individuo cuyo mantenimiento de la 6ta y de la 5ta sea en la etapa 12, que corresponde al individuo 71.115 con una f_{objetivo} de 1717,4 MUSD.

Se muestra a continuación el gráfico de costo marginal promedio del sistema en el periodo de estudio:

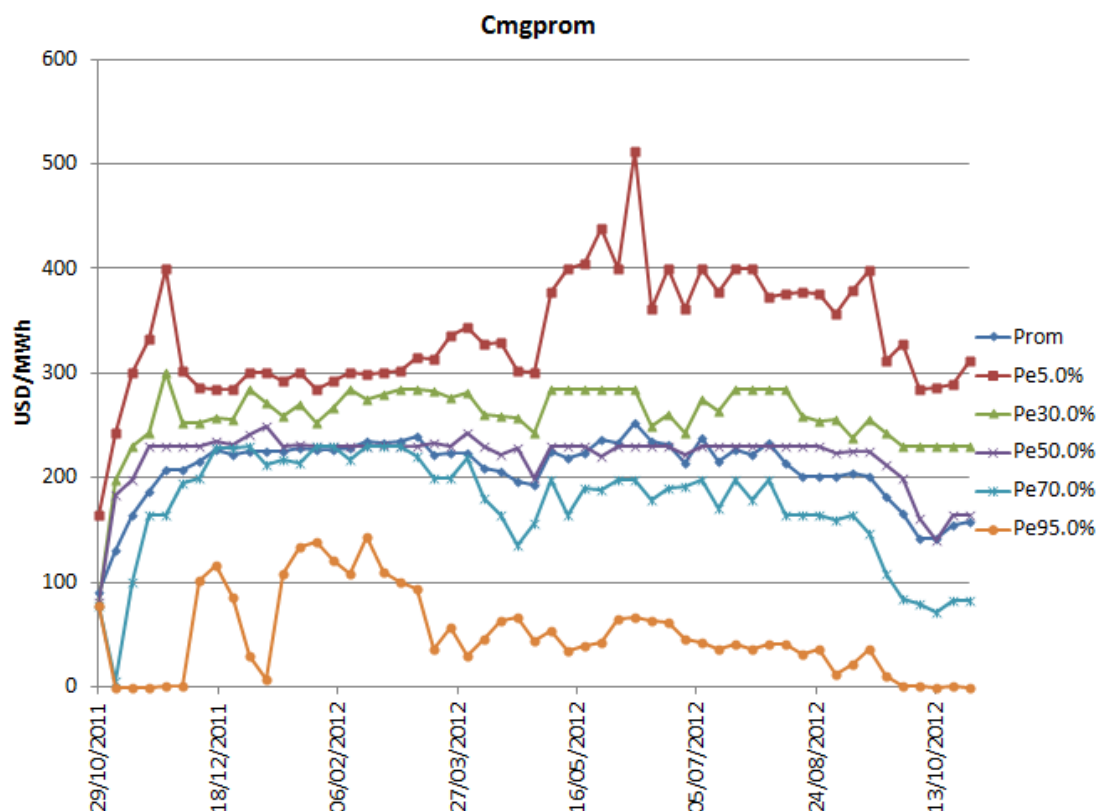


Donde el promedio del costo marginal promedio en todo el período de simulación considerado es de 206,8 USD/MWh.

El costo esperado de operación actualizado es de 1731.2 MUSD, valor obtenido luego de optimizar y simular con el SimSEE, siendo un 1,29 % mas alto que el valor de 1709,1 MUSD de la solución obtenida con el Oddface.

B3) A continuación se realiza el mismo análisis para el vértice inferior derecho del cuadrado, vale decir el individuo cuyo mantenimiento de la 6ta es en la etapa 0 y de la 5ta sea en la etapa 12, que corresponde al individuo 68.601 con una f_{objetivo} de 1716,1 MUSD.

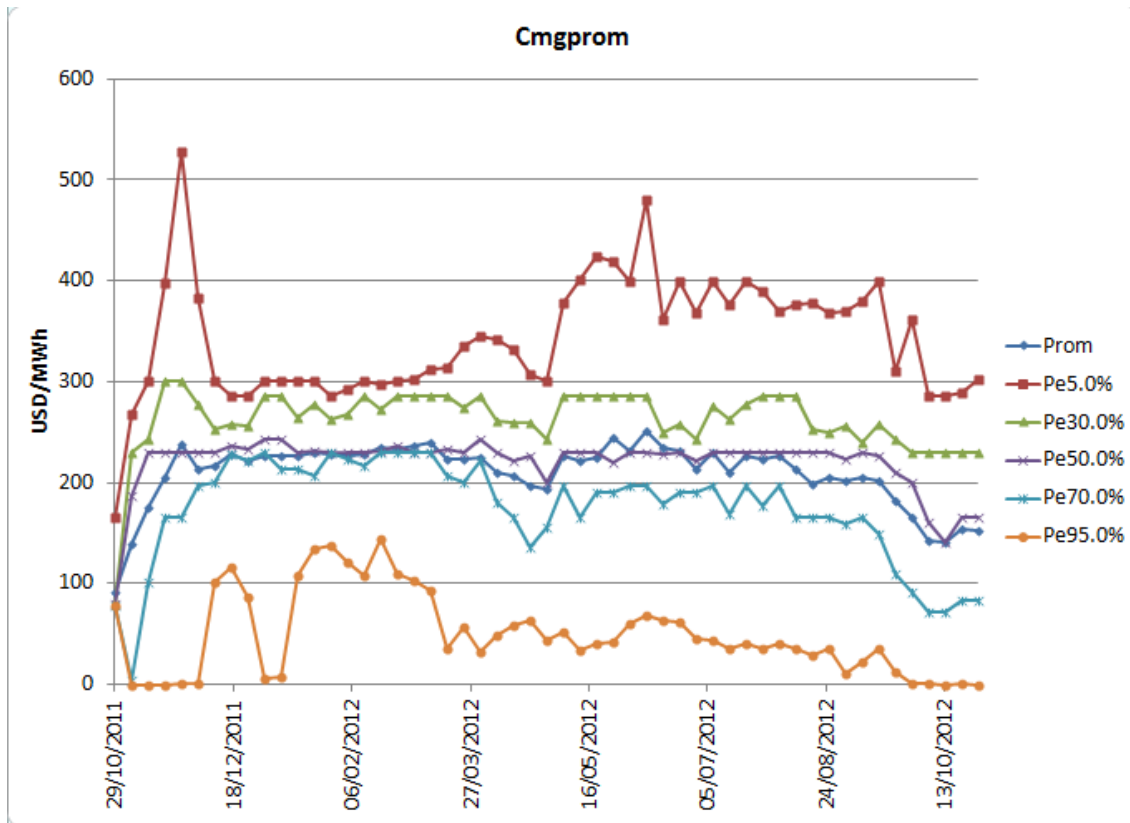
A continuación se muestran los datos resultantes al ser optimizada y simulada la sala en el SimSEE:



Dando el costo marginal promedio 208 USD/MWh y el valor del costo esperado de operación actualizado es de 1730,4 MUSD, siendo un 1,25 % superior al obtenido del individuo elegido.

B4) A continuación se realiza el mismo análisis para el vértice inferior izquierdo del cuadrado, vale decir el individuo cuyo mantenimiento de la 6ta y de la 5ta sea en la etapa 0, que corresponde al individuo 72.718 con una f_{objetivo} de 1714,5 MUSD.

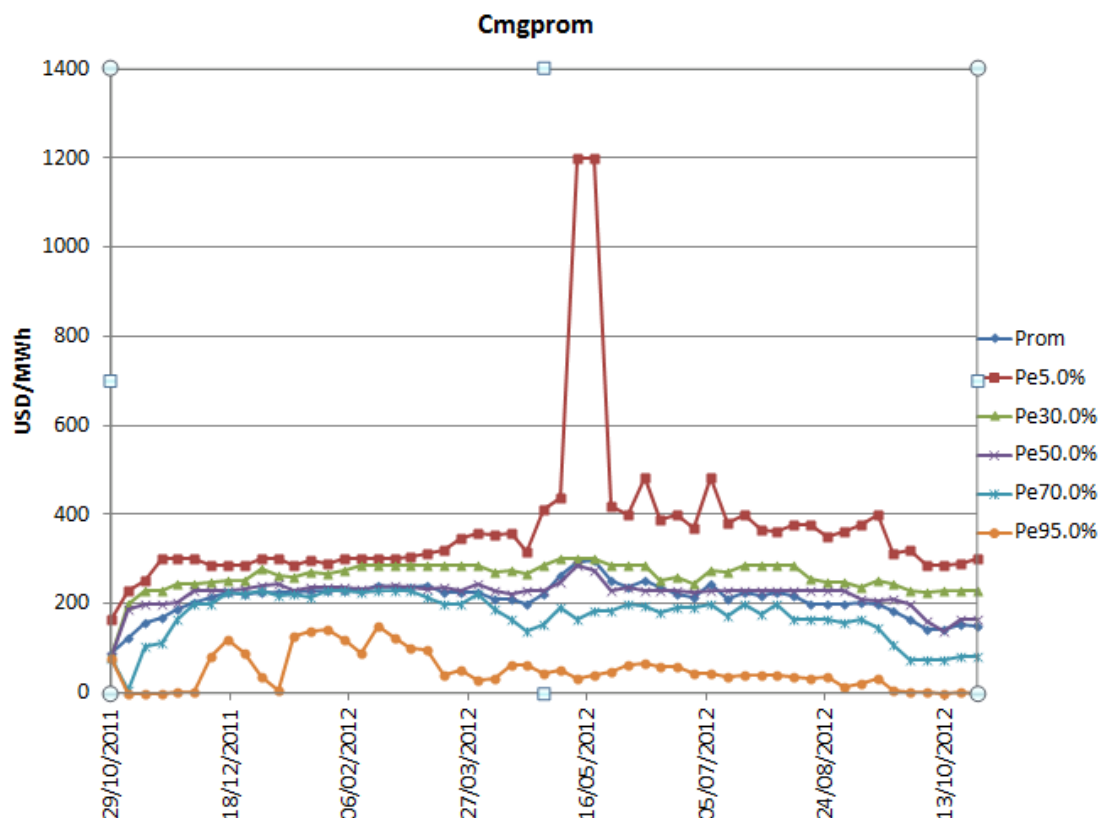
A continuación se muestran los datos resultantes al ser optimizada y simulada la sala en el SimSEE:



Siendo el promedio, en todo el período de simulación considerado, del costo marginal promedio de 209,4 USD/MWh. el valor del costo esperado de operación actualizado es de 1730,6 MUSD, siendo un 1,26 % superior al obtenido del individuo elegido.

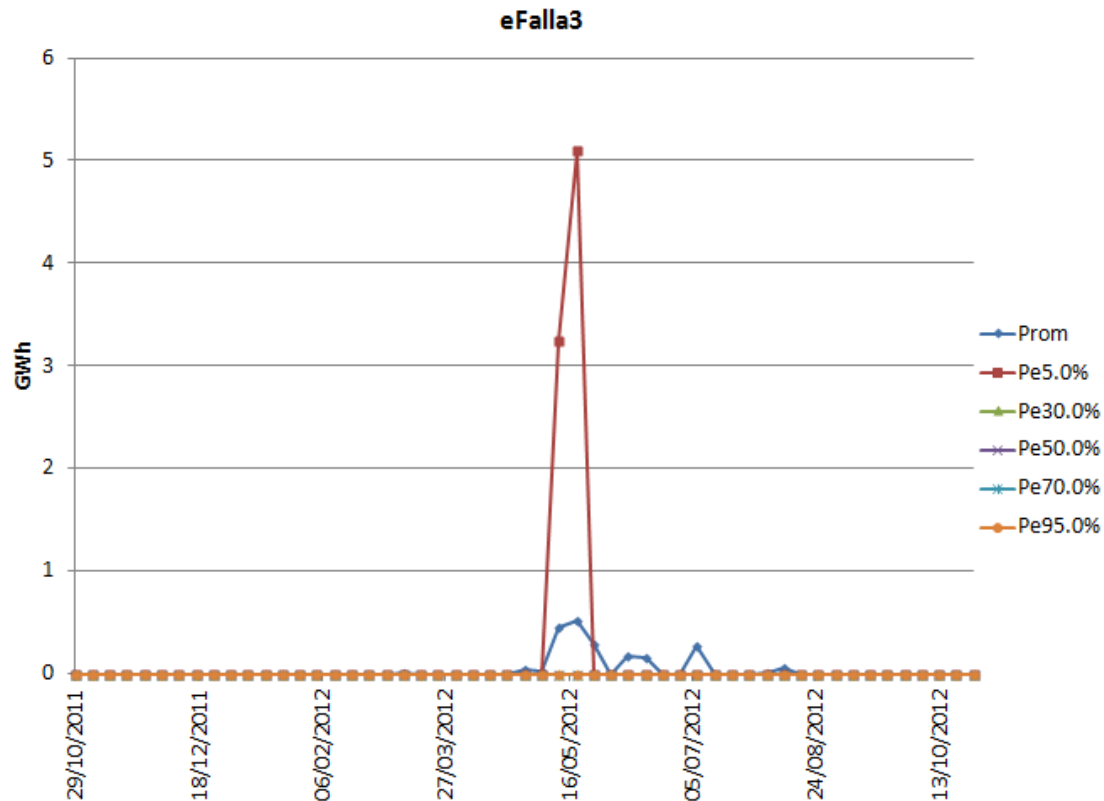
B5) A continuación se realiza el mismo análisis para una solución del medio del cuadrado de soluciones factibles, vale decir el individuo cuyo mantenimiento de la 6ta y de la 5ta sea en la etapa 6, que corresponde al individuo 53.284 con una f_{objetivo} de 1724 MUSD.

A continuación se muestran los datos resultantes al ser optimizada y simulada la sala en el SimSEE:



Siendo el promedio, en todo el período de simulación considerado, del costo marginal promedio de 211,4 USD/MWh. el valor del costo esperado de operación actualizado es de 1734,9 MUSD, siendo un 1,5 % superior al obtenido del individuo elegido.

Se observa que para las 100 crónicas sorteadas en el SimSEE, se da que con probabilidad de excedencia del 5 %, en los resultados de la corrida, el despacho de falla 3 para mediados de mayo del 2012.



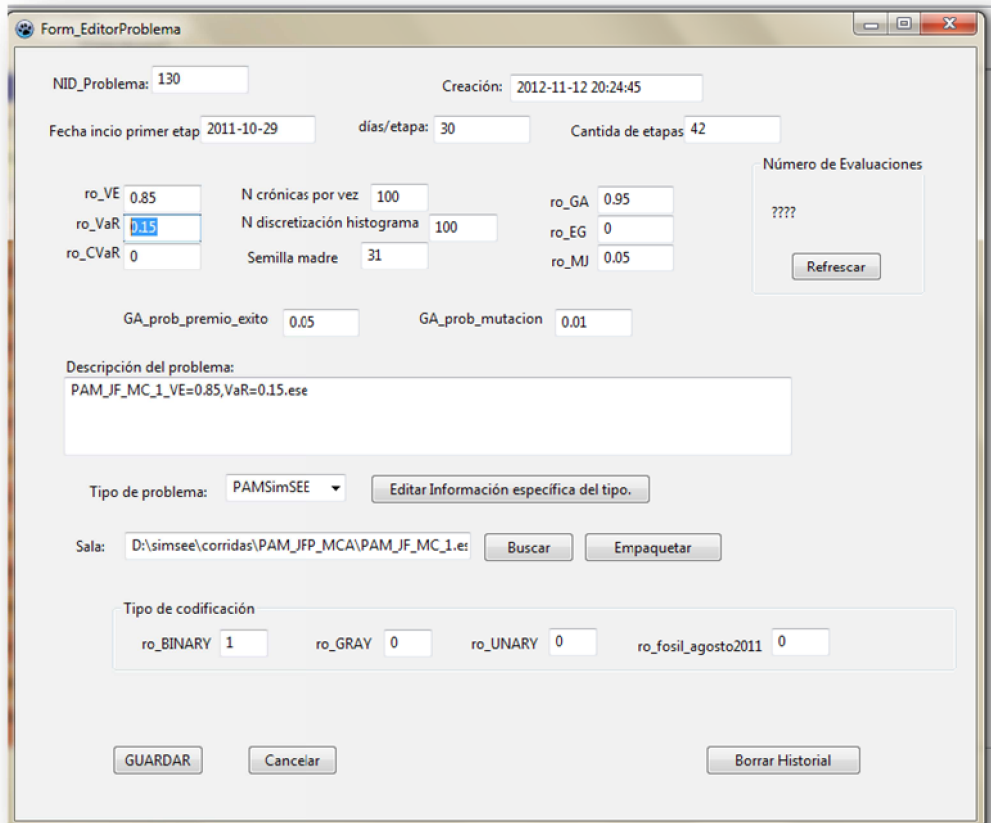
c) CONCLUSIONES

Como conclusión general de este análisis se observa que las diferencias entre las soluciones de los vértices de los cuadrados no difieren en forma notoria, respecto a la solución óptima encontrada. Asimismo se observa una diferencia leve entre una solución elegida al azar en el medio del cuadrado de soluciones factibles y la solución óptima.

iii. ANALISIS DE SOLUCIONES VARIANDO VALOR ESPERADO, AVERSION AL RIESGO Y DIAS POR ETAPAS

a) CASO BASE con $ro_VE=0.85$ y $ro_VaR=0.15$ –prob. 130 (Versión original)

Se decidió optimizar los mantenimientos cambiando el valor de ro_VE y ro_CVar con los restantes parámetros iguales. Para ello se clonó el problema 58 creando el 130 como se muestra a continuación:



The screenshot shows the 'Form_EditorProblema' window with the following configuration:

- NID_Problema:** 130
- Creación:** 2012-11-12 20:24:45
- Fecha inicio primer etap:** 2011-10-29
- días/etapa:** 30
- Cantida de etapas:** 42
- ro_VE:** 0.85
- ro_VaR:** 0.15
- ro_CVaR:** 0
- N crónicas por vez:** 100
- N discretización histograma:** 100
- Semilla madre:** 31
- ro_GA:** 0.95
- ro_EG:** 0
- ro_MJ:** 0.05
- Número de Evaluaciones:** ????
- GA_prob_premio_exito:** 0.05
- GA_prob_mutacion:** 0.01
- Descripción del problema:** PAM_JF_MC_1_VE=0.85,VaR=0.15.ese
- Tipo de problema:** PAMSimSEE
- Sala:** D:\simsee\corridas\PAM_JFP_MCA\PAM_JF_MC_1.ese
- Tipo de codificación:**
 - ro_BINARY:** 1
 - ro_GRAY:** 0
 - ro_UNARY:** 0
 - ro_fosil_agosto2011:** 0

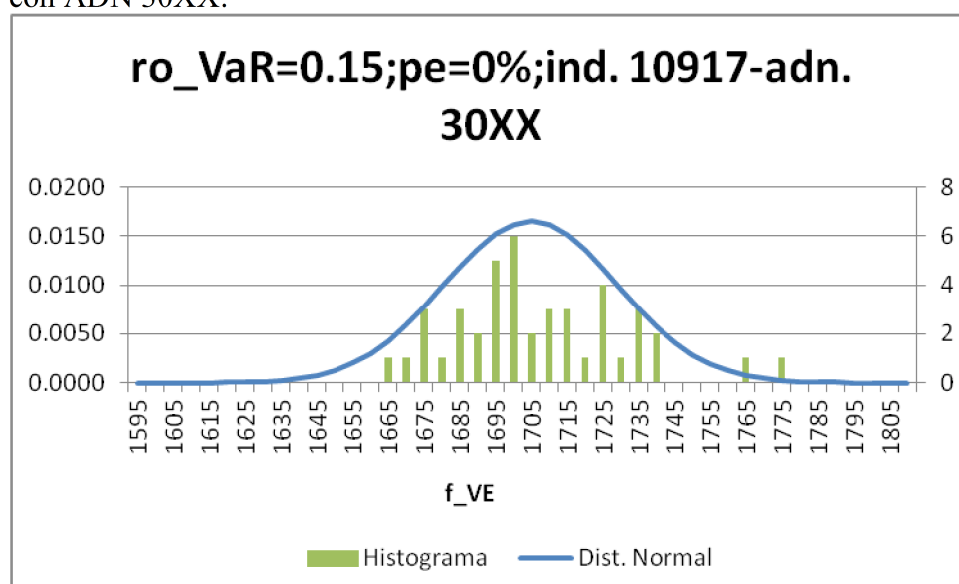
Buttons visible: GUARDAR, Cancelar, Borrar Historial, Refrescar, Editar Información específica del tipo, Buscar, Empaquetar.

Con este cambio se busca encontrar individuos más adversos al riesgo sacrificando seguramente el valor esperado f_VE .

Consulta

1	2012-11-30 00:02:16, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu FROM ofe_individuos_130 WHERE					
2	nid	adn	f_Objetivo	cnt_evaluaci	e_CB-6ta-FO	e_CB-5ta-FOP_245_4
3	10917	0300	1851.25131	43	3	0
4	11281	3100	1853.76243	49	1	3
5	11372	AB00	1856.04659	32	11	10
6	11230	7500	1857.0011	55	5	7
7	10848	4100	1857.60276	57	1	4
8	10631	A400	1857.60489	59	4	10
9	11227	2500	1857.82403	45	5	2
10	11416	0500	1858.20295	62	5	0
11	11095	1200	1858.31664	48	2	1
12	10975	1800	1859.02477	55	8	1
13	11273	3C00	1859.39854	227	12	3
14	11165	4900	1860.16913	41	9	4
15	10980	3200	1860.22761	37	2	3
16	11146	C200	1860.40222	208	2	12
17	10899	3000	1860.5865	50	0	3
18	10978	5700	1860.84902	44	7	5
19	11262	0200	1861.39912	52	2	0

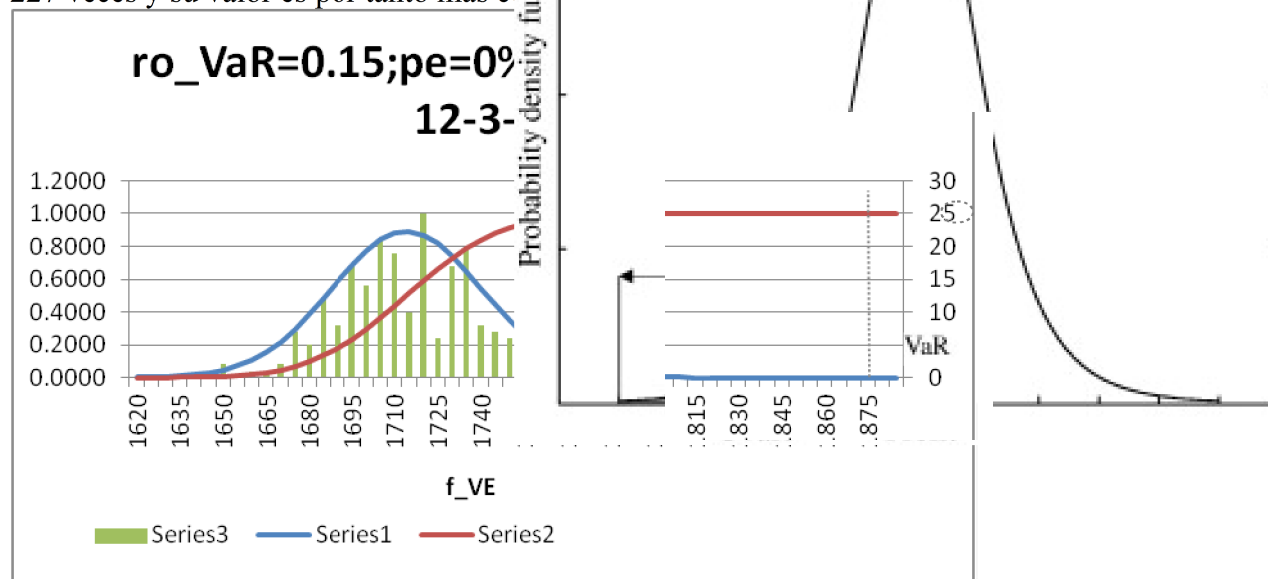
De igual manera que para el caso anterior se procedió a realizar el histograma del individuo 10917 con ADN 30XX:



Eval.	Totales	13356
Problema		
Eval individuo		43
f_Objetivo		1851.25
f_VE		1704.64321
Desvío Standard		24.2155383
%desv.est/f_VE		1.42056344

Como se observa en el cuadro, este problema tiene mayor f_Objetivo que el base, probablemente debido que tiene menor cantidad de evaluaciones totales (13356 contra 27321 del caso base). A su vez tiene 43 evaluaciones contra 115 del individuo elegido en el base, con lo que está menos probado.

El individuo 11273 con ADN 12-3-X-12727 veces y su valor es por tanto más co



Eval.	Totales	13356
Problema		
Eval individuo	227	
f Objetivo	1859.4	
f VE	1714.03693	
f VaR (est.)	1875	
f CVaR (est.)	1875	
(f CVaR- f VE)/f VE	9.4 %	
Desvío Standard	26.2959924	
%desv.est/Prom	1.53415553	

Con este nuevo individuo se confirma la intuición de la baja de la dispersión con el “costo” de subir el valor esperado. Como decíamos antes, el problema tuvo menos tiempo de prueba pero se confirma la tendencia a buscar individuos con menor valor condicionado en riesgo en porcentaje (9.4 % contra 10 % del base) y menor dispersión (1.53 % contra 1.62 % del caso base). El valor esperado sube en aprox. 5 MUSD (0.3 %).

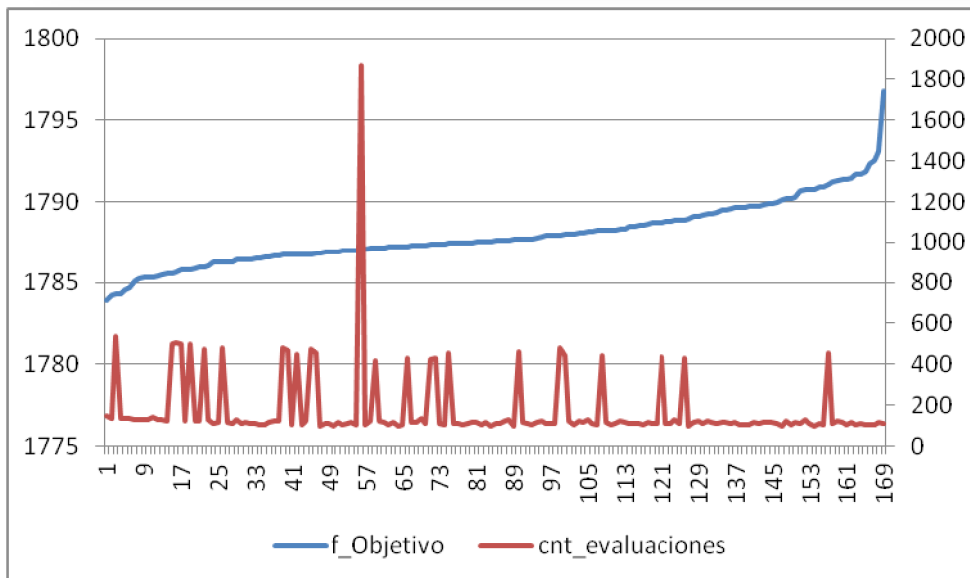
b) CASO BASE-prob. 146 (Versión nueva)

Los principales cambios de la nueva versión fueron:

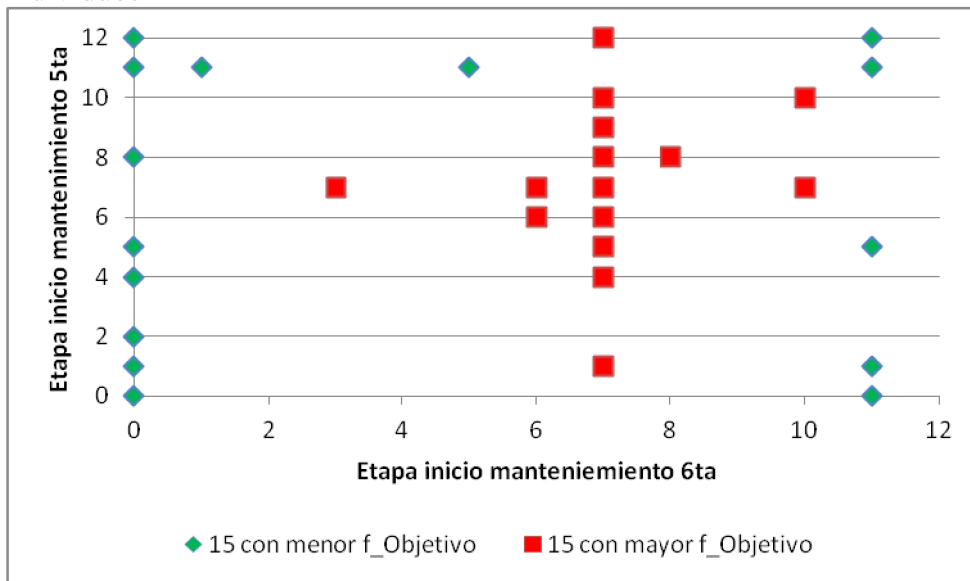
- Variación de la semilla aleatoria para cada evaluación del individuo.
- Limpieza de bits sobrantes en la codificación.
- Ajuste de la redundancia binaria para representar las restricciones de caja de los parámetros.
- Corrección de errores como el valor de PE que no estaba inicializado (se inicializó en 5%, en versión anterior estaba en 0%)

Respecto de la versión anterior se observa un aumento de 84 MUSD para el problema con casi la misma cantidad de evaluaciones totales e individuales para el ind 130 (elegido). En esta última, cada evaluación del individuo se realiza con una nueva semilla. El nuevo cambio de considerar como semilla $31 + \text{cnt_evaluaciones}$ mejora la estadística de los individuos.

2	nid	adn	f_Objetivo	cnt_evaluaci	e_CB-6ta-FO	e_CB-5ta-FOP_336_4
3	130	B000	1783.76586	148	0	11
4	186	5000	1784.14896	133	0	5
5	256	C000	1784.29828	540	0	12
6	29	0B00	1784.31709	132	11	0
7	87	0000	1784.49865	135	0	0
8	139	5800	1784.72375	132	11	5
9	164	1000	1785.1149	128	0	1
10	43	4000	1785.22037	130	0	4



La siguiente gráfica muestra la dispersión de los mantenimientos para los 50 mejores y peores individuos



Al igual que en el problema base con la versión anterior se observa una tendencia hacia la elección de puntos en la frontera del espacio de optimización.

Con la nueva versión, no fue necesario estimar los parámetros, puesto que los mismos aparecen correctamente cuando analizamos el ADN de cada individuo.

Eval. Problema	Totales 29371
Eval individuo	148
f_Objetivo	1783.76
f_VE	1783.76
f_VaR	2433.26
f_CVaR	2681.93
$(f_CVaR - f_VE)/f_VE$	50.3 %

c) CASO BASE con $ro_VE=0.85$ y $ro_CVaR=0.15$ –prob.141 (Nueva versión)

Consulta

1	2012-11-29 23:24:57, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu FROM ofe_indiv					
2	nid	adn	f_Objetivo	cnt_evaluaci	e_CB-6ta-FO	e_CB-5ta-FOP_317_4
3	269	5000	1918.63665	125	0	5
4	213	B100	1918.67845	132	1	11
5	188	B000	1918.68978	122	0	11
6	142	1000	1918.89099	115	0	1
7	134	C000	1918.92398	490	0	12

En la consulta se observa que el individuo correspondiente al adn 50XX presenta el menor f_Objetivo con 125 evaluaciones (el problema tiene 27922 evaluaciones en total). De manera que

se demora prácticamente el mismo tiempo de cálculo y el individuo esta casi igualmente probado (caso base 29371 y elegido 148 evaluaciones).

<u>Eval.</u>	Totales	27922
<u>Problema</u>		
<u>Eval individuo</u>		125
<u>f_Ojetivo</u>		1918.6
<u>f_VE</u>		1784.91
<u>f_VaR</u>		2451.64
<u>f_CVaR</u>		2676.42
<u>(f_CVaR- f_VE)/f_VE</u>		50 %

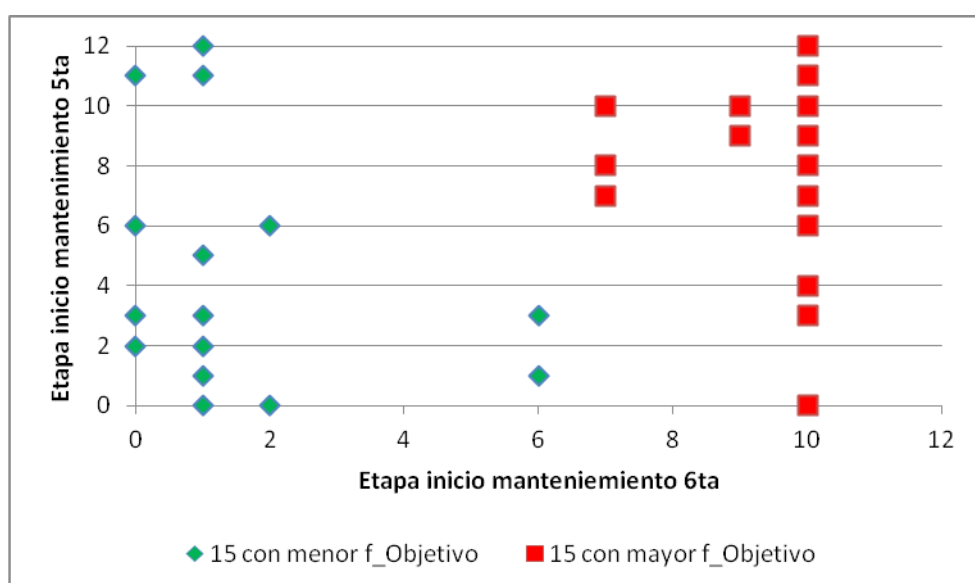
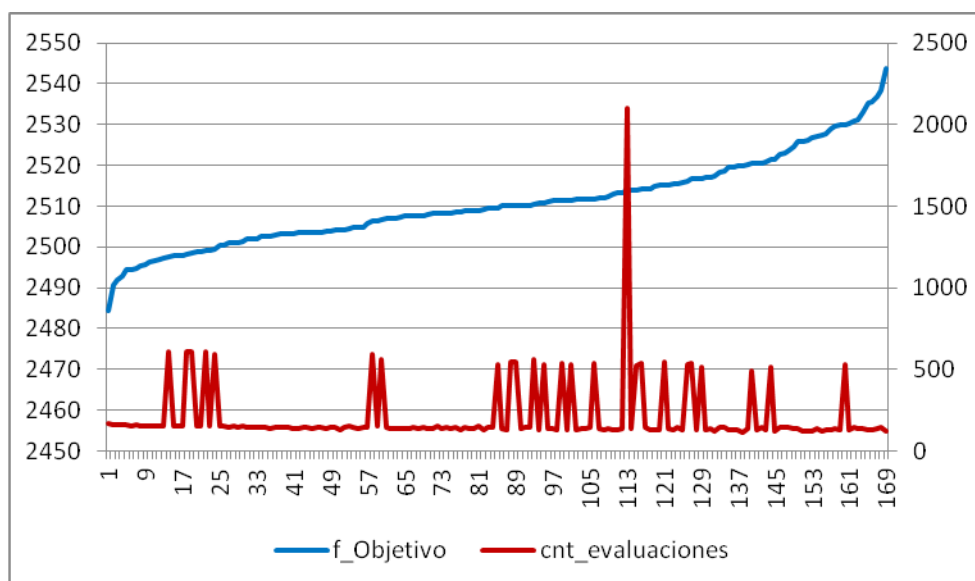
Se comprueba, al igual que para la versión original, que se obtiene un individuo menos expuesto al riesgo que para el caso anterior. A su vez, se observa que el valor esperado es mayor (1784.91 contra 1783.76). Se observa que el riesgo aumenta significativamente del entorno del 10 % al 50 % con la nueva versión.

d) **CASO BASE con $ro_VE=0.2$ y $ro_CVaR=0.8$ – prob. 156 (Versión nueva)**

Consulta

75	278	3800	2508.27603	137	8	3
76	258	0300	2508.31176	142	3	0
77	110	4400	2508.3245	136	4	4
78	319	5400	2508.56604	143	4	5
79	320	8300	2508.66103	125	3	8
80	152	2300	2508.76256	145	3	2
81	99	7000	2508.77941	140	0	7
82	323	7500	2508.81993	135	5	7
83	222	3800	2508.96756	154	11	3

Se elige el individuo 320 con ADN 3-8-X-X que tiene prácticamente la misma cantidad de evaluaciones que el caso anterior.



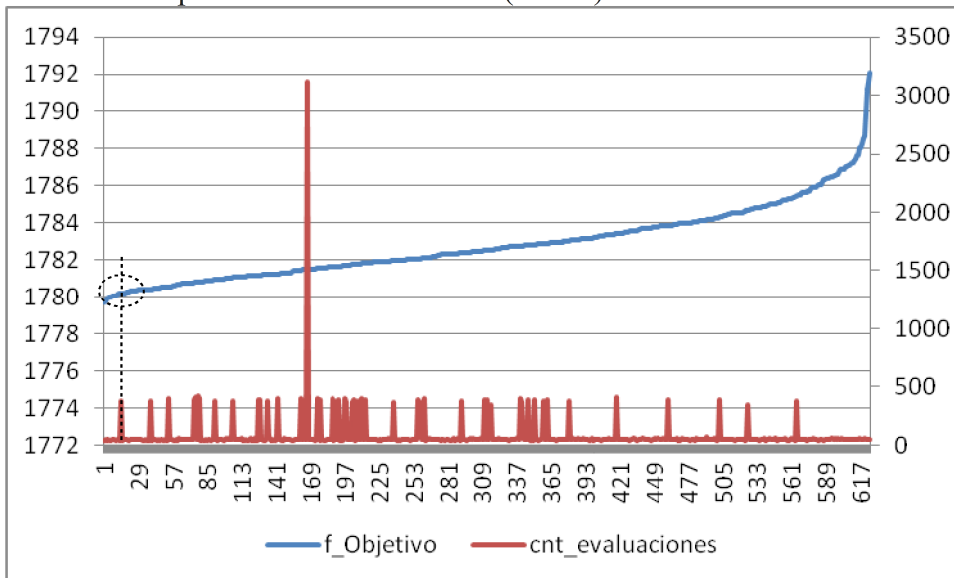
En este caso se deja de observar la tendencia hacia la elección de los bordes del espacio de optimización.

Eval.	Totales	35413
Problema		
Eval individuo		126
f_{Objetivo}		2508.81
f_{VE}		1788.8
f_{VaR}		2455.94
f_{CVaR}		2688.82
$(f_{\text{CVaR}} - f_{\text{VE}})/f_{\text{VE}}$		50.31 %

Como se preveía, hubo un aumento en el valor esperado del abastecimiento, aunque llamativamente aumenta el riesgo respecto del caso anterior en el que se ponderaba menos el riesgo (50.3 % contra 50 %).

e) **CASO BASE con 15 días/etapa-prob. 139 (Versión nueva)**

Se decidió bajar de 30 a 15 días por etapa, aumentando la cantidad de etapas (84 etapas) de modo que el horizonte de optimización Oddface fuera el mismo que en los casos anteriores. Habiendo hecho la consulta, se obtuvo la siguiente gráfica que muestra f_{Objetivo} y cantidad de evaluaciones para cada uno de los 676 (26x26) individuos.

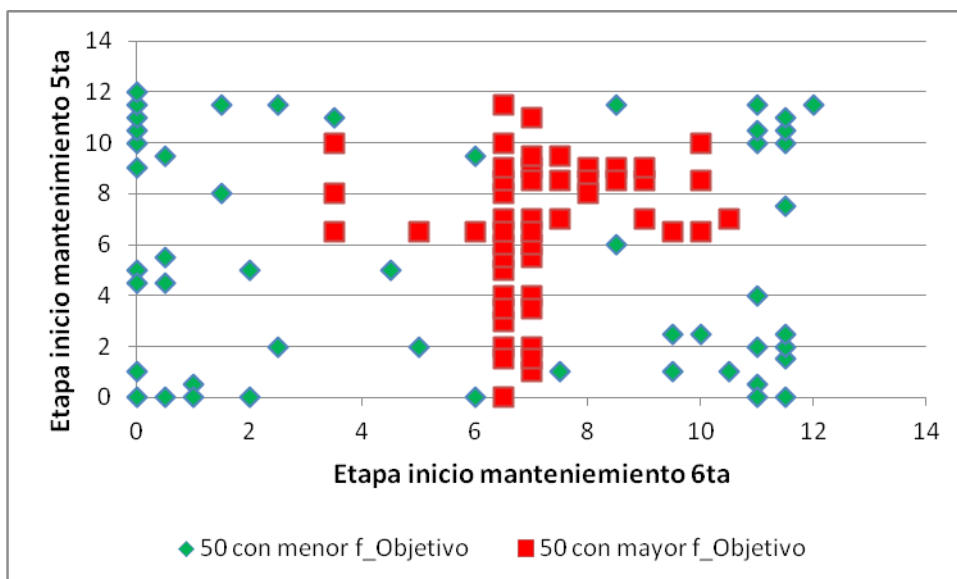


Para poder comparar resultados con lo obtenido en el caso base, se decidió elegir el individuo 952 con 378 evaluaciones y apenas un 0.025 % superior en f_{Objetivo} que el menor (1780.17 contra 1779.72).

1	2012-12-05 23:11:49, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu FROM ofe_individuos_139 WHERE 1 ORDEF									
2	nid	adn	f_Objetivo	cnt_evaluaciones	e_CB-6ta-FO	e_CB-5ta-FOP	313_5			
3	143	F701	1779.72487	42	23	15				
4	661	E302	1779.76529	45	3	23				
5	563	8002	1779.92498	42	0	20				
6	481	3600	1779.92974	47	22	1				
7	37	A002	1779.94301	44	0	21				
8	374	8500	1779.98766	40	5	4				
9	316	4901	1779.99586	40	9	10				
10	922	4001	1780.01954	47	0	10				
11	621	1601	1780.05684	46	22	8				
12	823	2001	1780.06684	46	0	9				
13	92	2200	1780.07149	42	2	1				
14	631	7700	1780.09977	47	23	3				
15	533	4000	1780.15082	46	0	2				
16	204	5500	1780.17349	41	21	2				
17	952	F802	1780.17535	378	24	23		Diferencia=	0.02531169	
18	890	6C02	1780.18278	39	12	19				
19	298	2101	1780.20796	44	1	9				
20	208	9602	1780.21442	42	22	20				

Comparando resultados notamos que el f_{Objetivo} empeora respecto del base aunque cabe decir que este problema tiene 48941 evaluaciones, cuando para lograr la misma proporción (evaluaciones/ind. Posibles) se hubieran necesitado $4 \times e_{\text{caso_base}}$ (108000 aprox.). Esto hubiera implicado tiempos de ejecución elevados.

La siguiente gráfica muestra la dispersión de los mantenimientos para los 50 mejores y peores individuos. Las etapas de mantenimientos se dividen entre 2 para comparar con el caso base.



Se observa también una tendencia hacia la elección de puntos en el borde.

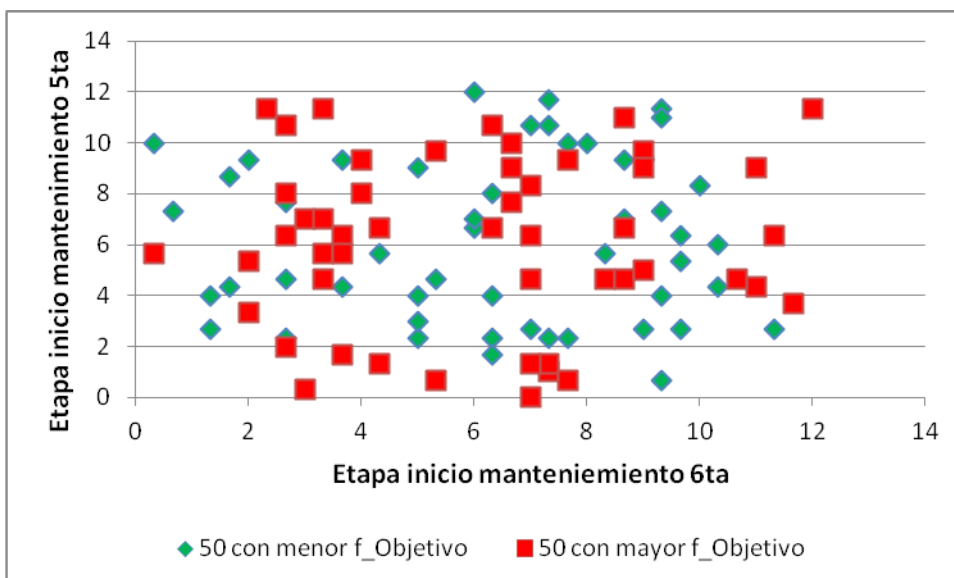
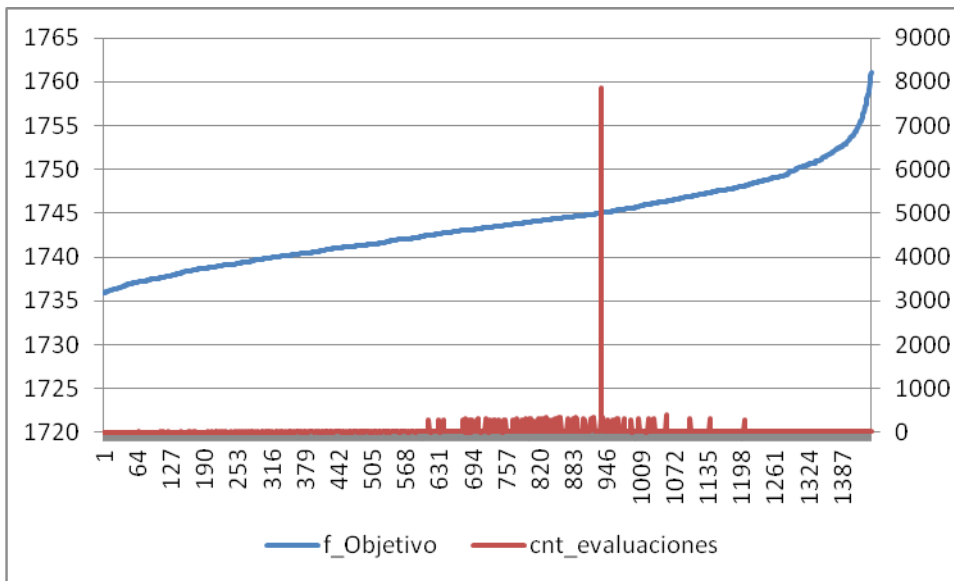
f) CASO BASE con 10 días/etapa-prob. 140 (Versión nueva)

En esta ocasión, se disminuyó a 10 días/etapa con las mismas consideraciones respecto del horizonte de optimización que con 15 días/etapa.

Al igual que para el caso anterior, para poder comparar con el caso base, se hubieran necesitado 9*caso base evaluaciones (243000). Debido a la demora que ello conllevaría, se logró llegar a las 46918 evaluaciones.

1	2012-12-05 23:32:36, sql: SELECT nid, adn, f_Objetivo, cnt_evaluaciones, dtc, dtu FROM of					
2	nid	adn	f_Objetivo	cnt_evaluaci	e_CB-6ta-FO	e_CB-5ta-FOP_315_6
935	546	8002	1745.00003	14	0	10
936	1394	4E01	1745.01152	13	14	5
937	3507	6009	1745.05071	351	32	37
938	4118	6509	1745.07292	7865	37	37
939	3936	4509	1745.07809	335	5	37
940	4046	2505	1745.07858	286	37	20
941	3896	4109	1745.0876	351	1	37
942	1429	C107	1745.10823	14	1	31
943	1495	C105	1745.1144	13	1	23
944	842	8607	1745.11609	13	6	20

De todas maneras, si elegimos el individuo 3896 con 351 evaluaciones (casi igualmente probado que el ind. elegido en 15d/etapa) notamos una disminución de f_Objetivo en un 2 % (35 MUSD).



En este caso no parece existir una predilección hacia la elección de individuos.

g) ANÁLISIS Y CONCLUSIONES

AVERSIÓN AL RIESGO

Para los individuos seleccionados anteriormente, se decidió optimizarlos y simularlos en el Simsee convencional (100 crónicas de Optimización Y 100 crónicas de Simulación) para verificar que se mantuviera la misma la tendencia que se observa entre problemas, cuando se le cambia algún parámetro.

En la siguiente tabla se muestran los valores relevantes de los resultados obtenidos en los diferentes problemas:

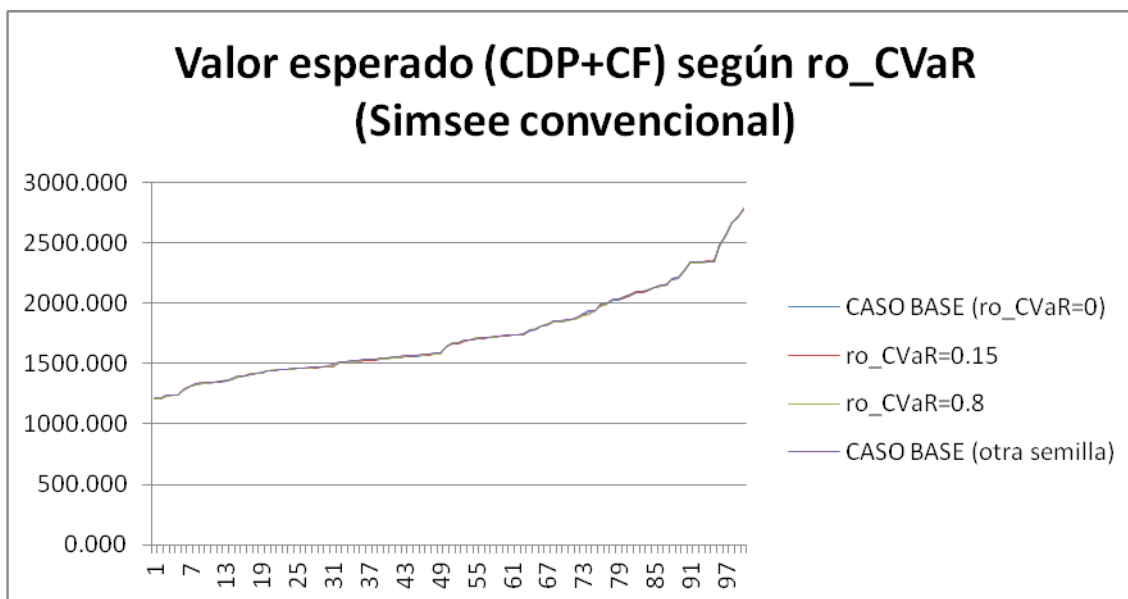
Simsee Convencional						
PROBLEMA	f_VE	f_CVaR	(f_CVaR-f_VE)/f_VE	f_Objetivo	cant. Evaluaciones	Eval. Totales
ro_CVaR=0 (146)	1729.616	2637.191	52.47%	1729.616	1	1
ro_CVaR=0.15 (141)	1730.541	2639.252	52.51%	1866.84791	1	1
ro_CVaR=0.8 (156)	1731.287	2635.490	52.23%	2454.64924	1	1
Oddface						
PROBLEMA	f_VE	f_CVaR	(f_CVaR-f_VE)/f_VE	f_Objetivo	cant. Evaluaciones	Eval. Totales
ro_CVaR=0 (146)	1783.77	2681.93	50.35%	1783.77	148	29371
ro_CVaR=0.15 (141)	1782.73284	2676.42	50.13%	1918.6	125	27922
ro_CVaR=0.8 (156)	1788.80497	2688.82	50.31%	2508.81589	126	35413

Como fuera señalado, se buscó que los individuos tuvieran la misma cantidad de evaluaciones para asegurarnos que fueran estadísticamente igualmente fiables. También se procuró que los problemas tuvieran similar cantidad de evaluaciones totales de modo de comparar problemas con igual tiempo de ejecución.

Se nota un descenso en el riesgo $((f_CVaR - f_VE)/f_VE)$ del problema 156 contra el 146 y un aumento del valor esperado entre ambos problemas. Esto es coherente con el compromiso existente entre riesgo y valor esperado. No obstante, también notamos que esto mismo no se verifica del problema 141 al 156. Esto podría ser consecuencia del aumento sustantivo de $f_Objetivo$, indicando que a medida que aumentamos ro_CVaR también la búsqueda se torna más ineficiente para similares tiempos de ejecución.

Si comparamos con las corridas hechas en Simsee, notamos que los valores de riesgo se encuentran en el entorno de lo obtenido por Oddface obteniéndose también una reducción del riesgo y como contrapartida un aumento del valor esperado. Se nota una diferencia importante en el valor esperado de suministro dado por Oddface y Simsee. Como los individuos hallados por Oddface tienen en el entorno de 130 evaluaciones, son más fiables que los obtenidos en una corrida convencional y la herramienta permite conocer con mayor exactitud el costo esperado.

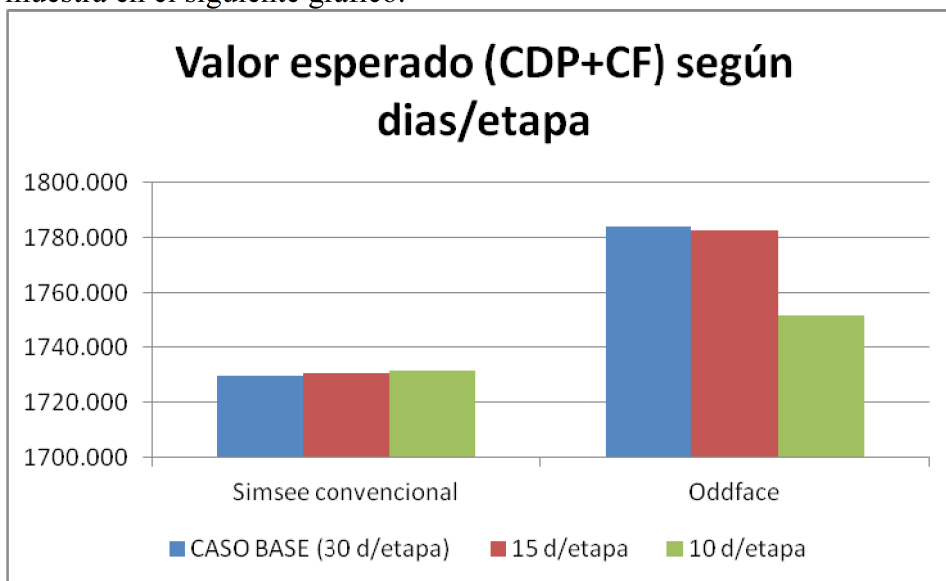
A continuación se muestra en una misma gráfica los valores ordenados del valor esperado de cada crónica obtenidos en Simsee. También se incluyen los resultados del caso base ($ro_CVaR=0$) con otra semilla (32).



Prácticamente no se nota diferencia alguna entre los distintos casos. Esto indica que en este problema en particular, en el que se analiza el mantenimiento de tan solo 2 unidades, no influiría la elección de los mismos, estando los valores en el orden del error estadístico.

C7.2) DÍAS POR ETAPA

Variando este parámetro y dándole mayor libertad de elección al software, aparece una reducción significativa de aprox. 35 MUSD en el costo esperado para el caso de 10 días/etapa, como se muestra en el siguiente gráfico.



Por tanto, se resalta la importancia de disminuir la cantidad de días por etapa considerando que se logran reducciones en el valor esperado, habiendo aumentado en tan solo un 60 % el tiempo de ejecución (46918 contra 29371 evaluaciones).

C. CONCLUSIONES GENERALES

Se creó la herramienta Oddface para realizar un Plan de Mantenimiento de unidades optimizado. En nuestro caso lo aplicamos a un caso simple y no detectamos diferencias notorias entre las soluciones.

A medida que fuimos familiarizándonos y utilizando la herramienta, fuimos encontrando errores, que iban siendo corregidos.

En general se observa que no hay grandes diferencias de costos entre las soluciones, por lo que se puede concluir que termina siendo más grande la variabilidad hidráulica del Sistema Eléctrico Uruguayo que la asociada a los mantenimientos de las unidades térmicas. Pero esta herramienta nos permite obtener el plan de mantenimientos optimizado para el mencionado sistema, algo con lo que hasta ahora no se contaba.

Capítulo IX. OddFace+PIG

Eliana Cornalino y Ruben Chaer.

Diciembre 2012

Montevideo – Uruguay.

Si al editar el problema, se seleccionó que el tipo de problema es PIGSimSEE entonces, se trata de un problema de optimización de Plan de Inversiones en Generación. El problema consiste en determinar el cronograma óptimo de decisiones de inversión.

A. Sobre la planificación de inversiones en generación.

La planificación en inversiones de generación de energía eléctrica es un problema complejo de toma de decisiones que afectan fuertemente el futuro. Lo complejo del problema, radica en la conjunción de más de una característica siendo las principales: El desconocimiento del futuro, los grandes montos de inversión involucrados, los plazos de construcción y la irreversibilidad de algunas decisiones. Como las decisiones tomadas afectan el futuro, se está ante un problema de optimización de un sistema dinámico. Se trata entonces, en todo momento, teniendo en cuenta el estado actual del sistema tomar “la mejor decisión” en el presente, teniendo en cuenta los efectos sobre el futuro.

La función de costo a minimizar es directamente el Costo Futuro (CF) de abastecimiento de la demanda eléctrica, incluyendo tanto los costos operativos directos como los costos de inversiones. Para el cálculo de este CF, normalmente se utilizan flujos de fondos en moneda constante y una tasa de descuento del orden del 10%.

La plataforma SimSEE es útil para calcular el Costo Futuro (CF) de operación del sistema conocido cuál será el crecimiento de la demanda del sector, el conjunto de inversiones que se realizarán y por lo tanto las fechas de las nuevas incorporaciones.

El manejo de las incertidumbres sobre el futuro, puede realizarse en forma cómoda dentro de SimSEE modelando el proceso estocástico que genera la incertidumbre, por ejemplo, el crecimiento de la demanda o las posibles evoluciones del precio de petróleo; o pueden tratarse como “casos” o escenarios diferentes y que sean simulados por separado. Por ejemplo, el precio del petróleo, en lugar de modelarlo dentro del SimSEE con un proceso estocástico, puede optarse por ejecutar dos simulaciones una con precio “alto” y otra con precio “bajo” y luego sacar conclusiones en base a ambos escenarios. Al momento de decidir cómo tratar una incertidumbre, es importante analizar si la misma “tiene estado relevante”, entendiendo por tal aquella información que puede hacer cambiar la Política de Operación (PO) del sistema. Si el proceso tiene estado relevante, es importante representar adecuadamente las probabilidades de transición entre los diferentes estados para que esa información pueda ser tenida en cuenta en la formación de la Política de Operación Óptima (POO). El carácter de relevante o no de la información del estado del proceso estocástico asociado a la incertidumbre tiene también que ver con los plazos de la POO que estamos buscando. A modo de ejemplo, cuando la POO se refiere al despacho de los recursos del sistema para los siguientes seis meses es relevante en la valorización del agua embalsada en los lagos del sistemas

el conocimiento del precio del petróleo dado que sus variaciones tiene constantes de tiempo del orden del período analizado. Si la POO se refiere a las inversiones a realizar en los siguientes 30 años, el conocimiento del precio de petróleo hoy no es una información demasiado útil, pues la dinámica de dicho proceso hace que al cabo de un año, pueda existir un valor esperado y una distribución del precio, pero muy poco condicionada con el valor de hoy.

No cabe duda que lo mejor sería poder representar todos los procesos estocásticos con sus estados y que de la optimización realizada en SimSEE surjan cuales contienen información relevante o no. Pero en la práctica, este enfoque adolece de la “Maldición de la dimensionalidad de Bellman”. Para minimizar el impacto del crecimiento de la dimensionalidad, una estrategia clásica es crear árboles de escenarios. Es decir simplificar el problema, imponiendo que solo en determinados instantes de tiempo pueden ocurrir las transiciones y limitar en cada bifurcación las transiciones posibles.

B. Planteo del Problema:

Para especificar este tipo de problemas, lo primero que se debe hacer es crear una sala SimSEE con todas las posibles centrales (o tecnologías) de generación existentes y que se quieran considerar como alternativas para la expansión (sala base). A las alternativas a considerar se las debe incluir en la sala con cero (0) unidades de generación en las fichas de unidades configurando los pagos por energía, costo variables y pagos por potencias en forma correcta para reflejar tanto los costos fijos como variables.

En el formulario de edición del problema, luego de seleccionar el tipo PIGSimSEE, presionando el botón "Editar información específica del tipo" se accede al formulario que permite editar el listado de tecnologías a considerar.

Para cada tecnología se debe incluir la siguiente información:

- Nombre: nombre de la máquina dado en la sala base
- Meses de construcción: plazo desde la toma de decisión hasta la puesta en operación
- Años de vida útil: plazo desde la puesta en operación hasta la salida de servicio
- MUSD/Unidad: costo de decisión asignado al momento de toma de decisión, por cada Unidad de Inversión
- Fecha de Inicio: comienzo de la ventana de tiempo para la toma de decisiones. Si se pone cero (0) valdrá la fecha de inicio de la simulación ingresada en la sala base
- Fecha de Fin: final de la ventana de tiempo para la toma de decisiones. Si se pone cero (0) valdrá la fecha de fin de la simulación ingresada en la sala base
- Máximo de Unidades por vez: Número máximo de Unidades de Inversión que pueden decidirse en una etapa
- Máximo de Unidades activas: Número máximo de Unidades de Inversión que pueden estar operativas en cada etapa.

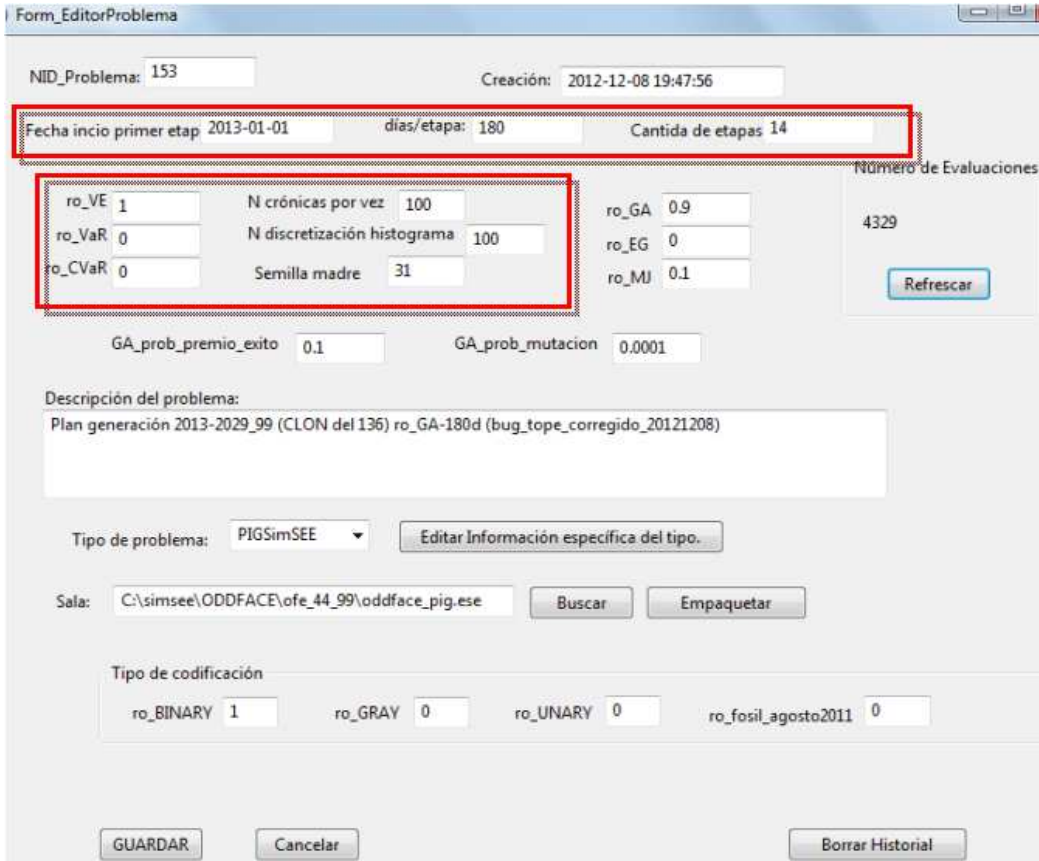
- Factor UG/UI: Factor por el que se multiplican las Unidades de inversión para generar las fichas de Unidades de Generación. Las Unidades de Generación son las definidas en la sala con su potencia correspondiente. Por ejemplo, en el caso de los parques eólicos puede pensarse que las Unidades de Inversión son parques de 50 MW, pero las Unidades de Generación en la sala base son aerogeneradores de 2MW y entonces el Factor UG/UI sería 25.

C. Resolución del Problema:

Al poner en marcha la resolución del problema el programa traducirá las cadenas de ADN que irá generando en vectores de entrada de unidades de generación para cada una de las tecnologías definidas con los que se determinarán las salas a ser evaluadas.

En concreto estos vectores indicarán para cada etapa de la evaluación y para cada tecnología, el número de Unidades de Inversión (n) que se decide instalar en dicha etapa (e), lo cual es equivalente a ingresar en la sala ($n \times$ Factor UG/UI) Unidades de Generación en la fecha ($e + \text{Meses de construcción}$) y darles de baja en la fecha ($e + \text{Meses de construcción} + \text{Años de vida útil}$)

Manteniendo fijos los parámetros que inciden directamente sobre el cálculo de la función objetivo (ro_VE , ro_VaR , N crónicas, etc.), mediante el seteo del resto de los parámetros del problema se deberá dirigir la búsqueda del óptimo tomando distintas estrategias para recorrer el dominio dependiendo de la etapa de la evaluación en la que nos encontremos.



Form_EditorProblema

NID_Problema: 153 Creación: 2012-12-08 19:47:56

Fecha inicio primer etap 2013-01-01 días/etapa: 180 Cantida de etapas 14

ro_VE 1 N crónicas por vez 100 ro_GA 0.9
 ro_VaR 0 N discretización histograma 100 ro_EG 0
 ro_CVaR 0 Semilla madre 31 ro_MJ 0.1

GA_prob_premio_exit 0.1 GA_prob_mutacion 0.0001

Descripción del problema:
 Plan generación 2013-2029_99 (CLON del 136) ro_GA-180d (bug_tope_corregido_20121208)

Tipo de problema: PIGSimSEE Editar Información específica del tipo...

Sala: C:\simsee\ODDFACE\ofe_44_99\oddface_pig.es Buscar Empaquetar

Tipo de codificación
 ro_BINARY 1 ro_GRAY 0 ro_UNARY 0 ro_fosil_agosto2011 0

GUARDAR Cancelar Borrar Historial

Numero de Evaluaciones: 4329 Refrescar

Fig. 11: Parámetros que inciden sobre el cálculo de la Función Objetivo

Al comenzar de cero la evaluación de un problema dado, se deberá tener en cuenta que el punto de partida será seleccionado al azar por lo que no es bueno restringir demasiado la búsqueda en torno a las evaluaciones más exitosas si no que debe favorecerse en esta etapa la exploración al azar de distintas zonas del dominio. Debe ser preponderante entonces la selección por algoritmo genético (ro_GA) frente a la reevaluación de puntos (ro_MJ).

Un alto valor de probabilidad de mutación (GA_prob_mutacion) favorecerá también la dispersión de la exploración. Se debe tener en cuenta que esta probabilidad es la que tiene cada bit individual de la cadena de sufrir una mutación por lo que para cadenas largas la probabilidad total de sufrir al menos un cambio en un bit puede ser muy grande.

D. Consulta y análisis de la solución:

Las consultas más relevantes a realizar durante la evaluación de este problema son básicamente:

- para ver las mejores soluciones encontradas al problema: realizo la consulta por función objetivo ascendente (f_Objetivo ASC)
- para observar si la evolución del problema es la esperada: realizo la consulta por última fecha de evaluación (dtu_DESC)

- incluyendo además la cantidad de evaluaciones (cnt_evaluaciones) totales reales realizadas sobre cada individuo se tiene información acerca de la representatividad del valor de f_objetivo correspondiente
- la consulta incluye siempre por defecto el número de identificación del punto evaluado (nid) y el código del ADN. Con el nid o con la fecha de creación del individuo (dtc) puede evaluarse cronológicamente que tan rápido se alcanzó por primera vez el mínimo de la función o cualquier otro valor buscado.

Para el análisis de las posibles soluciones encontradas se incluye en los datos de la consulta el vector de Unidades de Inversión por etapa y por tecnología a partir del cual se puede analizar la expansión propuesta en cada solución sin necesidad de descargar la sala correspondiente en cada caso.

A partir del número identificador (nid) se puede descargar la sala que el programa ha utilizado para la evaluación, consistente en la sala base modificada con los ingresos de máquinas correspondientes al individuo analizado.

Si se realizan corridas manuales utilizando la sala debe tenerse en cuenta que las inversiones no son consideradas en el cálculo del valor actualizado del costo total (archivo simcosto) por lo que las mismas deben ser sumadas aparte para la comparación de este valor con la función objetivo calculada por el programa. Al bajar la sala, en la pantalla del Oddface, se especifica el valor actualizado de las inversiones que deberá sumarse al de costos totales calculados con SimSEE.

La tasa de descuento utilizada por la sala y por el OddfacePIG para la actualización de todos los valores es la ingresada en la sala base, en la pestaña "Simulación".

E. Caso de estudio

i. Planteo:

El caso de estudio propuesto para la aplicación de este módulo consiste en la determinación de la expansión eléctrica óptima a llevar adelante en el período 2013-2020 considerando sólo dos tecnologías:

- Energía eólica: modelada como actor eólico con fuente de vientos "vientos_uy" y unidades de generación de 2 MW de potencia, con un pago por energía de 70 USD/MWh.
- Ciclo Combinado: modelado como generador térmico básico, con unidades de generación de 180 MW, con un costo variable de 64 USD/MWh y un pago por potencia de 17,56 USD/MWh conectados a una fuente de índices de precios de GNL.

Los parámetros específicos del problema son los detallados en la figura 2.

NID	Nombre.	Meses constr.	Años vida.	MUSD/UI.	Fecha ini.	Fecha fin.	Máx. UI/vez.	Máx. UIs.	Factor UG/UI		
47	eolico	18	20	0	2013-01-01	2030-01-01	10	100	25		
46	CC180_GN	18	40	180	2013-01-01	2030-01-01	4	30	1		

Fig. 12: Parámetros específicos del problema PIG.

ii. Resolución:

Se plantean en paralelo dos problemas para la resolución del caso en cuestión dando en uno de los casos mayor probabilidad de mutación sobre los individuos resultantes del algoritmo genético.

Problema	157	158	2° etapa 158
ro_GA	0.9	0.9	0.5
ro_MJ	0.1	0.1	0.5
GA_prob_premio_éxito	0.1	0.1	0.5
GA_prob_mutación	0.1	0.0001	0.0001
Codificación	Bin	Bin	Bin
Días por etapa	180días	180días	180días

Tabla 1 Parámetros de la búsqueda

El resultado obtenido con el Problema n°157 en un tiempo similar de cálculo (del orden de las 100.000 evaluaciones) es aproximadamente un 5% mejor que el del n°158.

Aplicando la segunda etapa para el problema n°158 tal como se muestra en la Tabla, se logra reducir rápidamente (en aproximadamente 1000 evaluaciones) la función objetivo hasta un nivel similar al obtenido con el otro problema.

Para observar la evolución de ambos problemas se ordenaron los resultados según la fecha de la última evaluación realizada. Debe observarse que los individuos evaluados aparecen en los resultados una sola vez, por lo cual los mejores individuos pueden haber aparecido en cualquier etapa de la evaluación pero en esta presentación de los resultados es más probable que aparezcan sobre el final ya que es probable que hayan sido reevaluados varias veces luego de su creación.

Al filtrar la evolución mediante promedios de 1000 evaluaciones se puede observar una convergencia mayor en la evolución del n°158 que es el que tiene menor probabilidad de mutación.

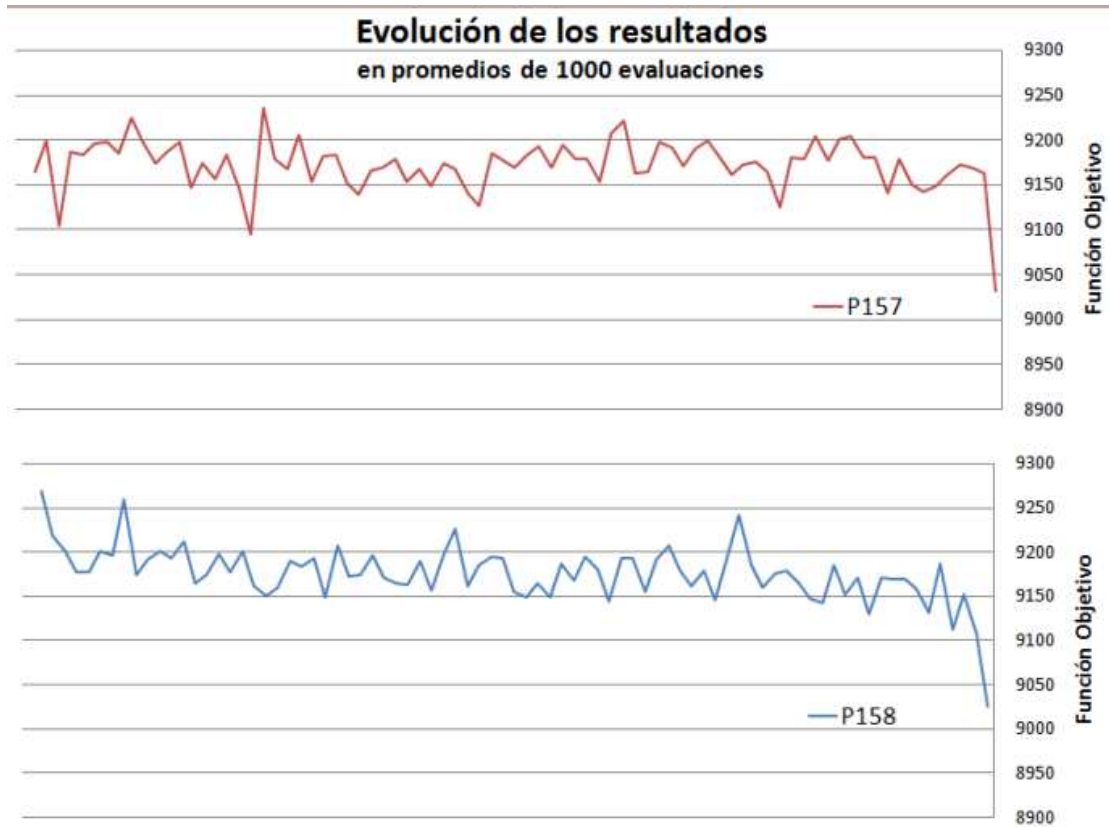


Fig. 13: Comparación de la evolución de los resultados promediados.

Dado el objetivo de encontrar el mínimo de la función se analizó también la evolución de los mínimos de las evaluaciones filtrados cada 1000 evaluaciones. Se observó un comportamiento globalmente más monótono del problema n°158, con menos probabilidad de mutación, como se ve en la Figura 4.

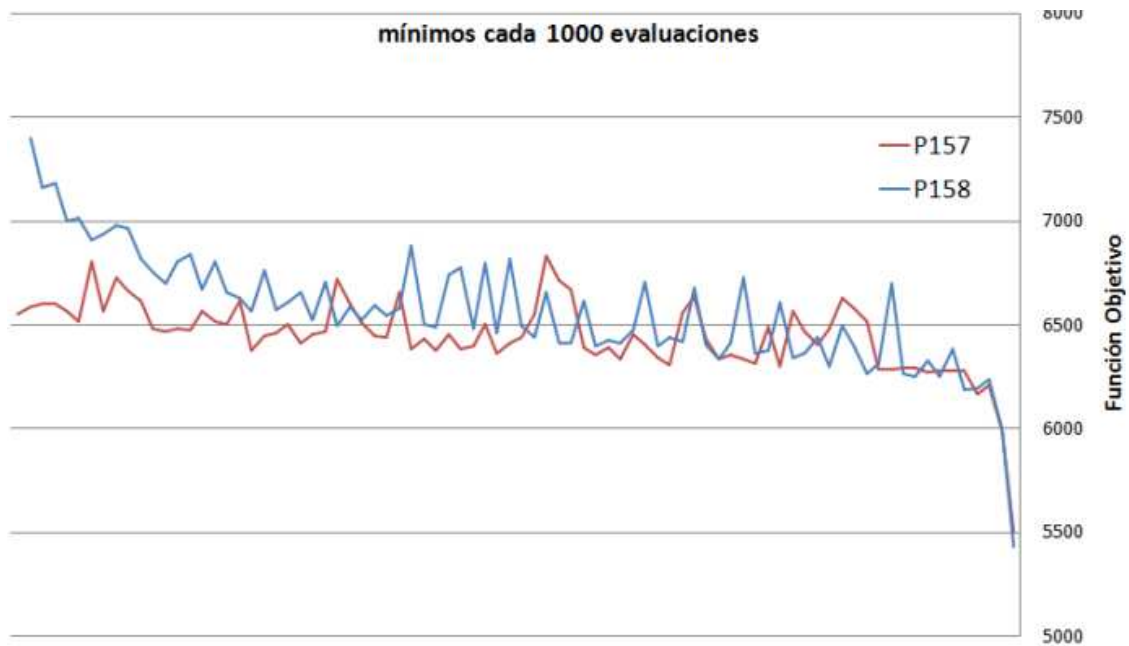


Fig. 14: Comparación de la evolución de los resultados mínimos.

iii. Análisis de resultados:

A continuación se muestran los resultados obtenidos para el caso concreto aquí presentado.

En la Figura 5 se presenta gráficamente el plan de expansión de mínimo costo (CC_min_1 y eol_min_1) y el de segundo menor costo (CC_min_2 y eol_min_2) con costos totales de 4989.7 MUSD y 5147.1 MUSD respectivamente.

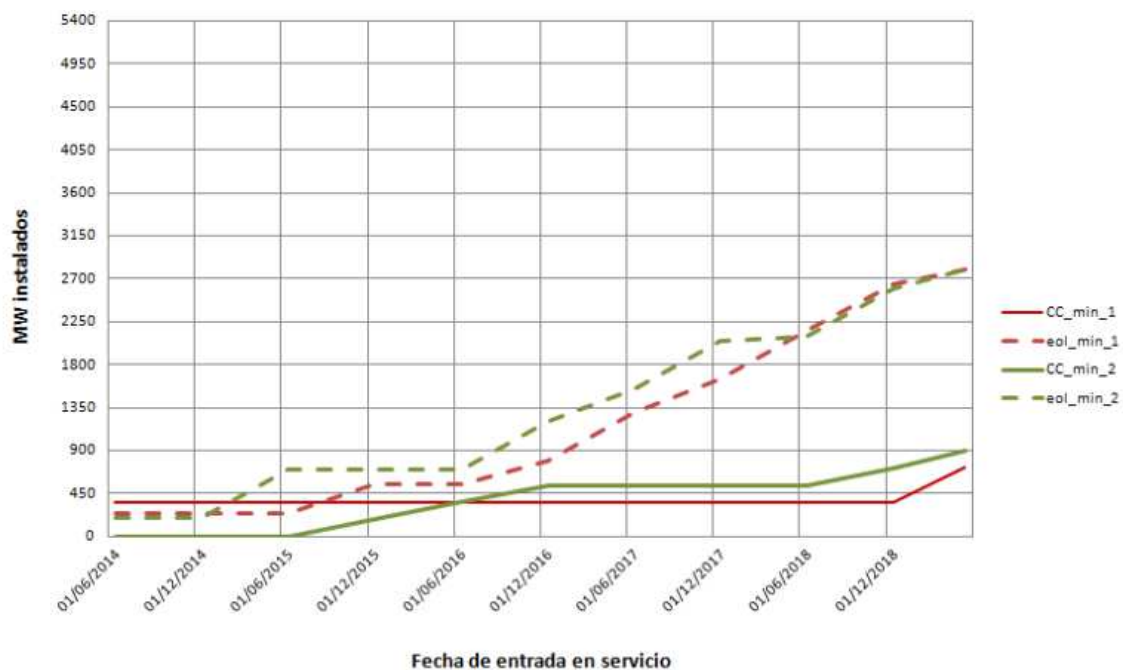


Fig. 15: Plan de expansión óptimo según los dos mejores resultados obtenidos.

El valor de costo mínimo obtenido es un promedio basado en un total de 40 evaluaciones de 100 crónicas cada una, mientras que el segundo mejor valor obtenido cuenta con 125 evaluaciones.

Como forma de evaluar la dispersión que puede haber detrás de estos resultados se realizaron 1000 simulaciones manualmente en SimSEE, obteniéndose el resultado mostrado en la Figura 6.

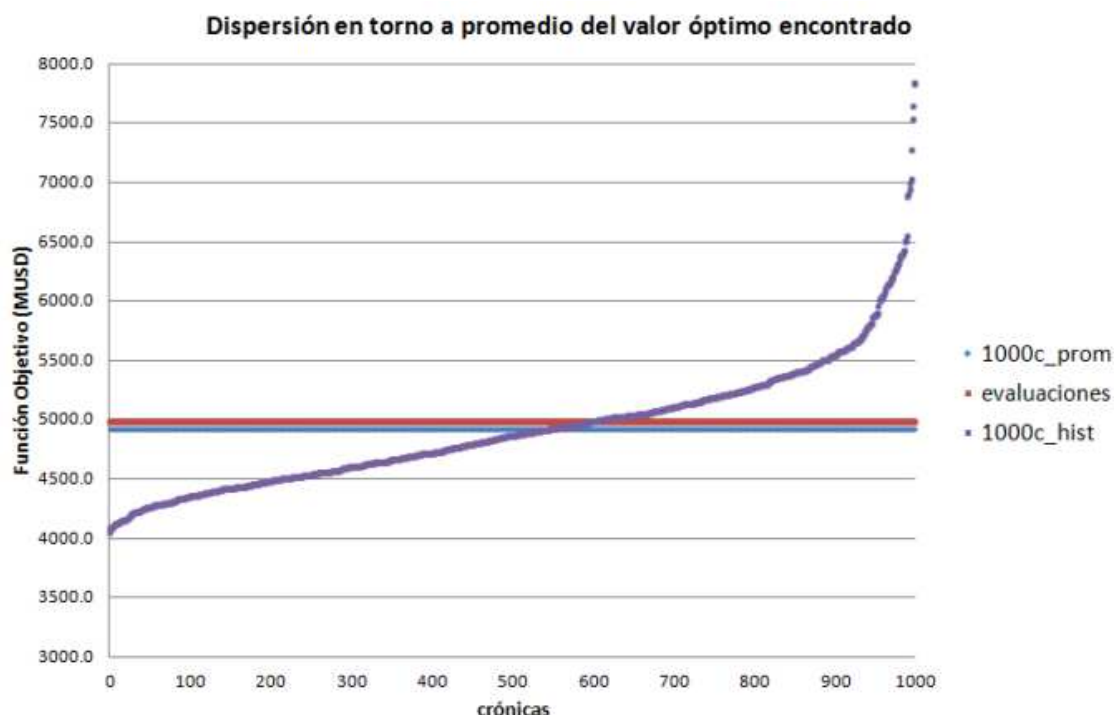


Fig. 16: Histograma de resultados de la simulación manual.

El costo promedio de las mil crónicas fue de 4358.8 MUSD, a los que se debe adicionar el costo de las inversiones en centrales de Ciclo Combinado, que asciende a 565.9 MUSD lo cual da un total de 4924.8 MUSD, un 1,3% menor que el valor obtenido con las 40 evaluaciones OddFacePIG. La dispersión de los resultados en el histograma va entre -29% y + 48% del promedio, con una probabilidad del 91% de que los resultados se encuentran entre +/- 15% del promedio de las 1000 crónicas.

F. Errores encontrados

Los principales errores que se lograron detectar durante las pruebas realizadas fueron los siguientes:

- Los ADN creados para evaluar no eran filtrados por “Máximo de unidades activas”. Al no filtrar los ADN a evaluar por se gasta tiempo de cálculo en evaluar individuos no factibles en toda su extensión.
- La ventana de etapas consideradas para configurar la sala estaba tomada sobre la fecha de toma de decisión y no sobre la fecha de entrada en operación de las máquinas lo cual generaba un desfase de

fechas que como resultado truncaba antes de la fecha de fin de simulación el ingreso de unidades.

- El ingreso de unidades al configurar la sala se hacía con fechas fijas por etapas con una discretización de las mismas cada 3 meses. Esto hacía perder precisión en las fechas de ingreso, quedando por ejemplo ingresadas en la misma fecha dos centrales cuya decisión de ingreso se tomaba en la misma fecha pero tenían diferentes plazos de construcción, con una diferencia menor a 3 meses.

i. Sugerencias.

Se listan algunas sugerencias que se detectaron como posibles mejoras para la solución de problemas PIG:

- 1) Sería conveniente poder sugerir manualmente un individuo para su evaluación. Esto sería muy útil poder ahorrar tiempo de cálculo partiendo de un punto cercano al óptimo y no evaluar al azar cualquier punto del dominio, por ejemplo para el caso en que se tienen ya evaluados problemas similares.
- 2) Poder comenzar con una discretización gruesa que se pueda ir refinando en la medida en que se avanza en la resolución del problema. Por ejemplo, en caso que se quieran realizar evaluaciones de muy largo plazo comenzar por etapas del orden de años para luego ir aumentando la precisión de las incorporaciones de potencia.
- 3) Se observó que se mejoraría la eficiencia del proceso de optimización si la base del problema fuera la determinación de la fecha de entrada de las unidades y no la cantidad de unidades a ingresar por etapa. Esto permitiría mejorar el resultado del proceso de cruzamiento entre individuos. La implementación actual partiendo de un individuo que ingresa n unidades en la etapa e y otro que ingresa n unidades en la etapa $e+1$ podría generar un individuo que ingresa n unidades en e y n unidades en $e+1$, lo cual seguramente se aleje del resultado buscado. Con la implementación sugerida se este tipo de problemas se evitaría optimizando el proceso de búsqueda.

G. Conclusiones

La herramienta evaluada mostró tener gran potencialidad para la resolución de los problemas de Planificación de Inversiones en generación.

Los resultados obtenidos en menos de 12 horas de tiempo de cálculo, del orden de 15.000 evaluaciones, presentaban ya una gran concordancia con lo razonablemente esperado y resultados muy próximos a los finalmente obtenidos con del orden de 120.000 evaluaciones.

En la medida que se implementen las mejoras sugeridas se contará con una muy potente herramienta para la resolución de estos problemas en tiempos de cálculo realmente bajos.

H. Referencias

[1] [Planificación de las inversiones de generación eléctrica con control de la volatilidad de los costos anuales de abastecimiento](#)

Marisa León, [Nicolás Castromán](#), Daniel Larrosa, [Gonzalo Casaravilla](#), [Ruben Chaer](#)
Encuentro Latinoamericano de Economía de la Energía, 3ero. ELAEE 2011. Buenos Aires,
Argentina - Apr. 2011

Capítulo X. SimSEE+CO2

Manual de cálculo del factor de emisión del sistema eléctrico uruguayo en el simulador de sistemas de energía eléctrica (SimSEE)

Fernanda Maciel., Ruben Chaer y Mario Vignolo.

Diciembre 2012

Montevideo - Uruguay.

Proyecto de mejoras a la plataforma SimSEE

ANII-FSE-18-2009.

Capítulo XI. Introducción

El presente manual, sus definiciones y metodologías de cálculo se basan en el documento “Tool to calculate the emission factor for an electricity system” de UNFCCC (United Nations Framework Convention on Climate Change). Asimismo, se utiliza la nomenclatura de dicho documento.

El objetivo es presentar una metodología para estimar, a partir de las simulaciones del SimSEE, el factor de emisiones del sistema eléctrico para el cálculo de la **línea de base de emisiones** para un proyecto que sustituye electricidad de la red. Es decir, para un proyecto que suministra electricidad a la red o una actividad/proyecto que resulta en un ahorro de electricidad que hubiera sido proporcionada por la red (ej. proyectos de generación de energía renovable o de eficiencia energética). Este tipo de proyectos se denominan **CDM**, por las siglas en inglés de mecanismos de desarrollo limpio.

El presente documento se encuentra dividido 5 capítulos que se enumeran a continuación con una breve descripción de su contenido:

- Introducción.
- Definiciones.
- Factor de emisión de una unidad/planta generadora: contiene la metodología de cálculo del factor de emisión de dióxido de carbono (CO_2) de cada unidad o planta generadora del sistema eléctrico, que deben ser calculados por el usuario e ingresados al SimSEE como datos. Asimismo, se presenta un ejemplo de cálculo y las fuentes para la obtención de los datos necesarios en el caso del sistema eléctrico uruguayo.
- Factor de emisión del sistema eléctrico: contiene una guía de cómo especificar en el Editor del SimSEE que se desea calcular las emisiones de CO_2 del sistema eléctrico, la descripción del documento generado por el SimSEE en este caso, y la metodología de cálculo que utiliza.
- Conclusión y recomendaciones: las conclusiones describen las ventajas y desventajas de utilizar la herramienta SimSEE para el cálculo de emisiones de CO_2 del sistema eléctrico y su aplicabilidad. Finalmente, se proponen algunas recomendaciones para su uso.

Capítulo XII. Definiciones

Factor de emisión de “margen combinado” (combined margin, CM) del sistema eléctrico ($\text{EF}_{\text{grid,CM},y}$): toneladas de CO_2 emitidas por cada MWh de energía generado en el año y . Se calcula como un promedio ponderado de dos factores de emisión pertenecientes al sistema eléctrico: el “margen de operación” (operating margin, OM) y el “margen de construcción” (build margin, BM). El factor de emisión OM ($\text{EF}_{\text{grid,OM},y}$) se refiere al grupo de plantas existentes cuya generación actual de electricidad se verá afectada por el proyecto u actividad propuesto. El factor de emisión BM ($\text{EF}_{\text{grid,BM},y}$) se refiere al grupo futuro (prospectivo) de plantas cuya construcción y futura operación se verán afectadas por el proyecto/actividad propuesto.

Factor de emisión de una unidad/planta generadora ($\text{EF}_{\text{EL},m,y}$ o $\text{EF}_{\text{EL},k,y}$): factor de emisión de CO_2 (en tCO_2/MWh) de una unidad de potencia m (o k) en un año y . Las unidades generadoras se denominan k o m según sean un recurso low-cost/must-run o no, respectivamente.

Recursos low-cost/must-run (LCMR): se definen como las plantas con costo marginal de generación bajo o plantas que son despachadas independientemente de la carga diaria o estacional

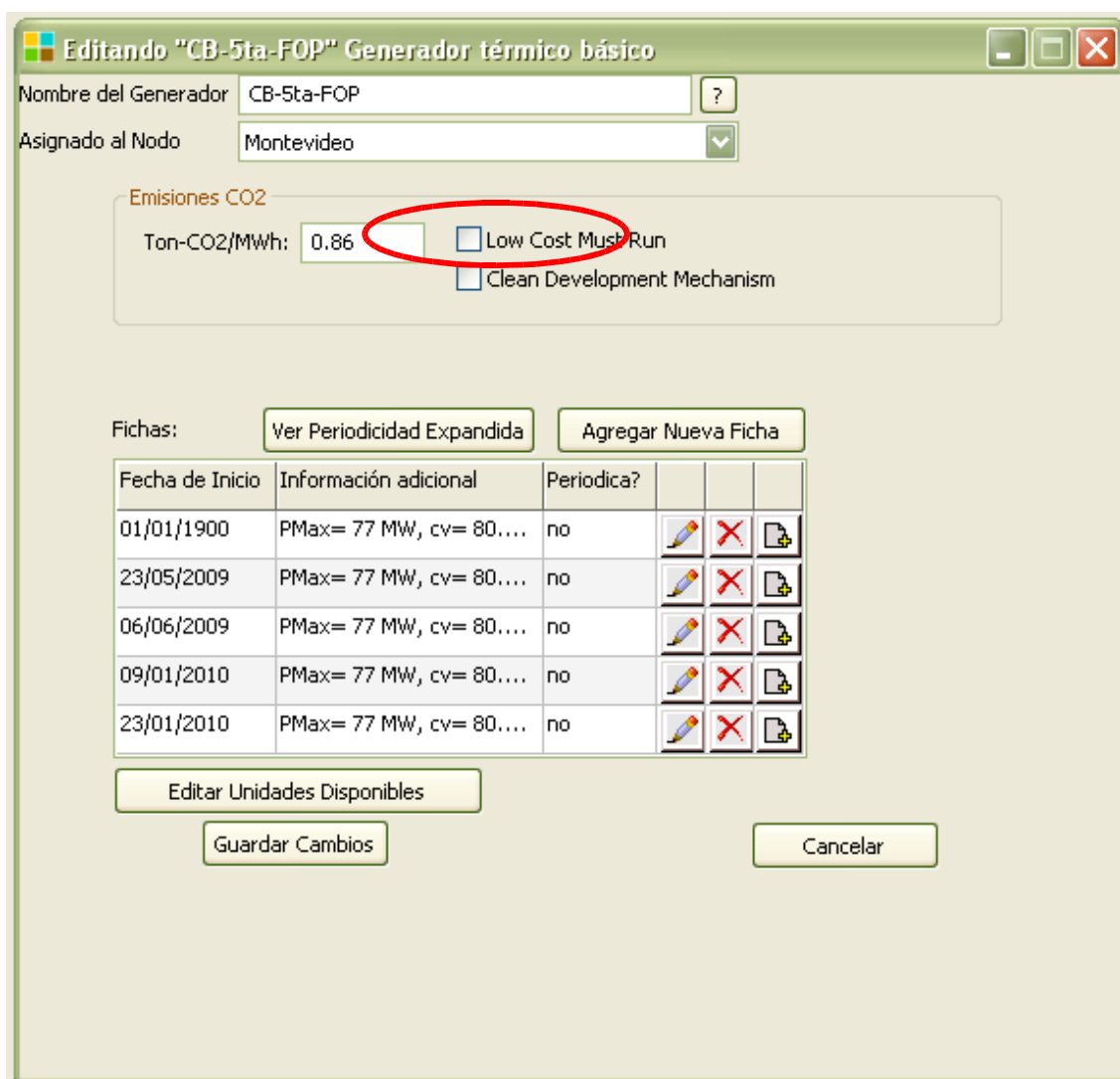
de la red. Típicamente está incluidas: generación hidroeléctrica, geotermal, viento, biomasa de bajo costo, nuclear y solar. De lo contrario se denominan high-cost/may-run (HCMR).

Capítulo XIII. Factor de emisión de una unidad/planta generadora

Cada unidad o planta generadora de energía puede clasificarse, según lo mencionado anteriormente, como LCMR o HCMR. El Editor del SimSEE permite al usuario determinar para cada Actor/Generador si es LCMR.

En la Figura 1 se presenta un ejemplo para un generador térmico. En caso que no se marque el casillero “Low Cost Must Run” indicado en la figura, el generador es considerado como HCMR.

Asimismo, a la izquierda de dicho casillero se debe ingresar el factor de emisión de la unidad. Su cálculo se presenta a continuación.



Editando "CB-5ta-FOP" Generador térmico básico

Nombre del Generador: CB-5ta-FOP

Asignado al Nodo: Montevideo

Emisiones CO2

Ton-CO2/MWh: 0.86 ☒ Low Cost Must Run ☐ Clean Development Mechanism

Fichas: Ver Periodicidad Expandida Agregar Nueva Ficha

Fecha de Inicio	Información adicional	Periodica?			
01/01/1900	PMax= 77 MW, cv= 80....	no			
23/05/2009	PMax= 77 MW, cv= 80....	no			
06/06/2009	PMax= 77 MW, cv= 80....	no			
09/01/2010	PMax= 77 MW, cv= 80....	no			
23/01/2010	PMax= 77 MW, cv= 80....	no			

Editar Unidades Disponibles

Guardar Cambios

Cancelar

Figura 1

Para una unidad m de la que se conoce datos de generación de energía y el tipo de combustible que utiliza, el factor de emisión se calcula en base al factor de emisión del CO₂ del tipo de combustible que utiliza y de la eficiencia de la unidad, según:

$$EF_{EL,m,y} = \frac{EF_{CO_2,m,i,y} \times 3.6}{\eta_{m,y}} \quad (1)$$

Donde:

$EF_{EL,m,y}$ = factor de emisión de CO₂ de la unidad m en el año y (tCO₂/MWh).

$EF_{CO_2,m,i,y}$ = factor de emisión de CO₂ promedio para el tipo de combustible i utilizado en la unidad m en el año y (tCO₂/GJ).

$\eta_{m,y}$ = eficiencia promedio de conversión de energía neta de la unidad generadora m en el año y (tasa).

Cuando distintos tipos de combustible son utilizados en una unidad, se debe considerar el tipo de combustible con el menor factor de emisión de CO₂.

Si para una unidad solamente se conocen datos de generación de energía eléctrica, se puede asumir de manera conservadora un factor de emisión de 0 tCO₂/MWh.

En el caso de las unidades LCMR, $EF_{EL,k,y}$ deben ser calculados utilizando los mismos procedimientos que $EF_{EL,m,y}$.

A. Ejemplo de cálculo

Se presenta el ejemplo de la central térmica de Punta del Tigre, pero la metodología de cálculo es igual para otras unidades.

En primer lugar se debe conocer el tipo de combustible que utiliza, en este caso gasoil, y el consumo de combustible de la central por unidad de energía generada: $FC/EG = 224$ gr/kWh para la central de Punta del Tigre.

En segundo lugar se debe conocer el poder calorífico neto (contenido de energía) del combustible: $NVC = 43$ MJ/kg (para el gasoil). Con FC/EG y NVC se puede calcular la eficiencia de conversión de energía de la central:

$$\eta = \frac{3600}{\frac{FC}{EG} \times NVC} \quad (2)$$

En este caso se obtiene $\eta=0.374$.

En tercer lugar, es necesario conocer el factor de emisión del combustible ($EF_{CO_2,i}$), para el gasoil es 0.0741 kgCO₂/MJ.

Finalmente se calcula el factor de emisión de la central a partir de la ecuación (1), obteniéndose un valor de 0.713 tCO₂/MWh.

B. Obtención de datos para los cálculos

La información de tipo de combustible y FC/EG para las centrales de UTE se pueden encontrar en:

- http://www.ute.com.uy/pags/Institucional/memoria_anual.html

La información del poder calorífico neto (NVC) de diversos combustibles y su factor de emisión ($EF_{CO_2,i}$) se pueden encontrar en:

- http://www.ipcc-nggip.iges.or.jp/EFDB/find_ef.php.

A modo de ejemplo, NVC es igual a 40.4 MJ/kg para fueloil y 48 MJ/kg para el gas natural; $EF_{CO_2,i}$ es igual a 0.0774 kgCO₂/MJ para el fueloil y 0.0561 kgCO₂/MJ para el gas natural.

Capítulo XIV. Factor de emisión del sistema eléctrico

En primer lugar es importante mencionar que para el cálculo de la línea de base de emisiones para un proyecto CDM que sustituye energía de la red, el proyecto en cuestión no debe estar incorporado en el SimSEE.

El Editor del SimSEE, en la pestaña “CO2”, permite al usuario marcar la opción “Calcular Emisiones CO2” (indicado en rojo Figura 2).

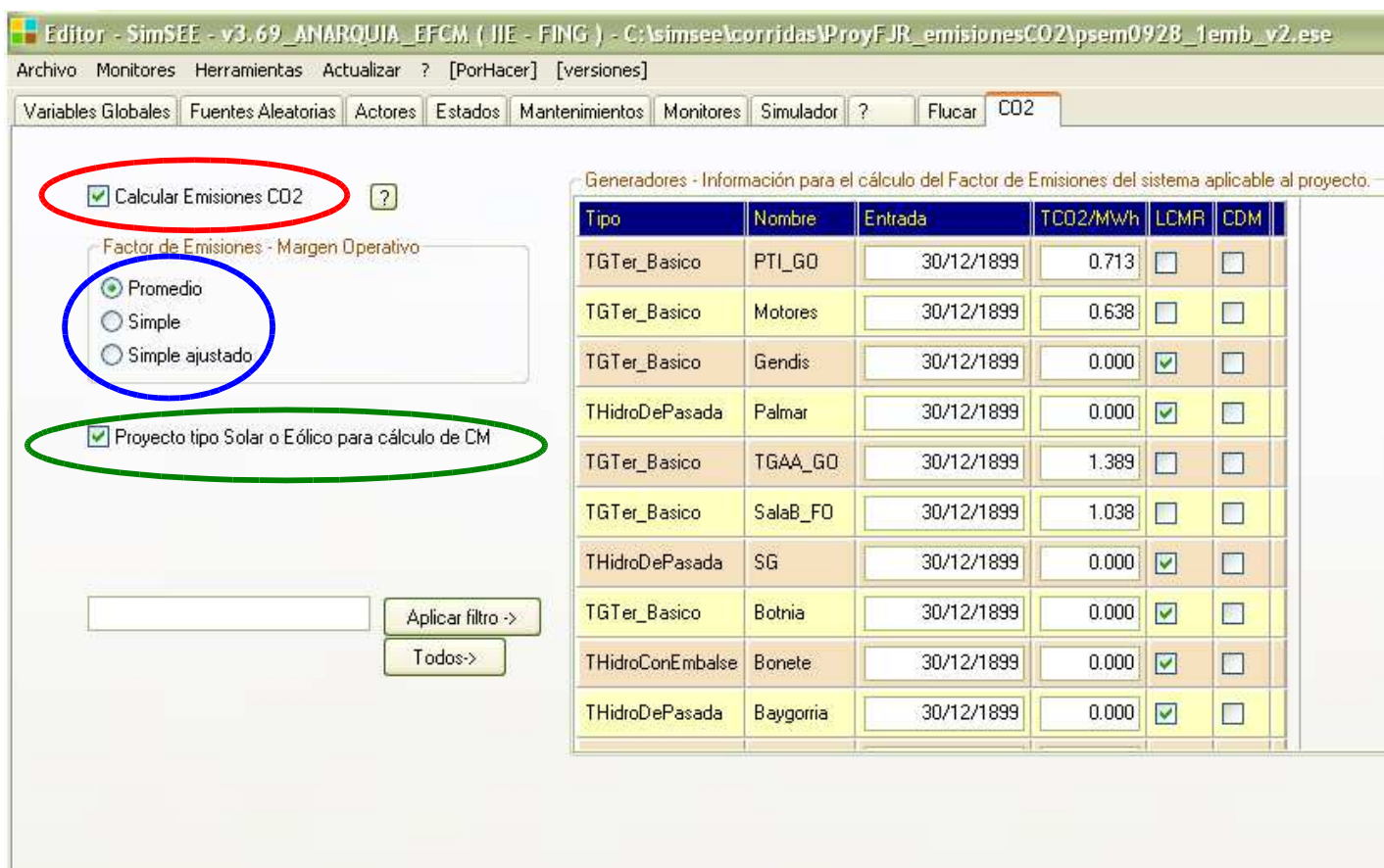
En esta misma pestaña el usuario debe elegir el factor de emisión OM (margen operativo) que desea utilizar para el cálculo del factor de emisión CM (margen combinado), indicado en azul en la Figura 2. La diferencia de cálculo entre los tres factores OM se detalla en la sub-sección Factor de emisión OM.

El usuario también puede seleccionar si el proyecto para el cual desea calcular la línea de base de emisiones es de tipo solar o eólico (indicado en verde en la Figura 2). Esta información se utiliza para el cálculo del factor de emisión CM (ver sub-sección Factor de emisión CM).

Asimismo, al seleccionar el botón *Todos->* en la pestaña “CO2”, se muestra la tabla de generadores del sistema eléctrico a utilizar para el cálculo de las emisiones de CO₂ (Figura 2). Allí el usuario debe ingresar las fechas de entrada de cada generador (inicio del suministro de energía a la red) y marcar los proyectos aprobados como CDM (Clean Development Mechanism), datos que son necesarios para el cálculo del factor de emisión BM (margen de construcción, ver sub-sección Factor de emisión BM). La clasificación CDM se puede indicar también al editar cada generador (Figura 1).

Finalmente, al realizar las simulaciones se genera el archivo “simres_CO2_SemAleatoria_Simulaciones” (*SemAleatoria* y *Simulaciones* corresponden a los valores definidos por el usuario en la pestaña “Simulador” para la semilla aleatoria de la simulación y el número de crónicas de la simulación, respectivamente).

Dicho archivo se genera en el directorio C:\simsee\rundir\NombreSala, donde *NombreSala* corresponde al nombre que el usuario asignó a la sala (archivo con extensión .ese) que utilizó para realizar la corrida en el SimSEE.



Tipo	Nombre	Entrada	TCO2/MWh	LCMR	CDM
TGTer_Basico	PTI_GO	30/12/1899	0.713	☐	☐
TGTer_Basico	Motores	30/12/1899	0.638	☐	☐
TGTer_Basico	Gendis	30/12/1899	0.000	☒	☐
THidroDePasada	Palmar	30/12/1899	0.000	☒	☐
TGTer_Basico	TGAA_GO	30/12/1899	1.389	☐	☐
TGTer_Basico	SalaB_FO	30/12/1899	1.038	☐	☐
THidroDePasada	SG	30/12/1899	0.000	☒	☐
TGTer_Basico	Botnia	30/12/1899	0.000	☒	☐
THidroConEmbalse	Bonete	30/12/1899	0.000	☒	☐
THidroDePasada	Baygorria	30/12/1899	0.000	☒	☐

Figura 2

El archivo “simres_CO2_SemAleatoria_Simulaciones” es del tipo: Microsoft Excel Template y contiene, para cada año y crónica simulada, la siguiente información:

- Energía anual generada por cada unidad/planta LCMR en MWh. Para diferenciarlas fácilmente de las HCMR, la unidad de generación aparece con un símbolo “*”, es decir: MWh*.
- Energía anual generada por cada unidad/planta HCMR en MWh.
- Emisión anual de CO₂ de cada unidad/planta LCMR en toneladas de CO₂ (así como la energía, la unidad aparece como ton-CO₂*).
- Emisión anual de CO₂ de cada unidad/planta HCMR en toneladas de CO₂.
- El factor de emisión OM ($EF_{grid,OM,y}$) en tCO₂/MWh, calculado de tres maneras distintas, que se presentan en la sub-sección Factor de emisión OM.
- El factor de emisión BM ($EF_{grid,BM,y}$) en tCO₂/MWh, su metodología de cálculo se presenta en la sub-sección Factor de emisión BM.
- El factor de emisión CM ($EF_{grid,CM,y}$) en tCO₂/MWh, a partir del factor OM seleccionado, del factor BM y del tipo de proyecto CDM propuesto (ver sub-sección Factor de emisión CM).
- Lambda: factor que expresa el porcentaje de horas del año (horas/horas) en las que marginan las unidades/plantas LCMR de acuerdo a la estimación de tomar la curva horaria de generación de energía ordenada de forma monótona decreciente (monótona de generación) y completar en la base con la energía del grupo LCMR. Este factor es necesario para calcular uno de los tipos de OM (ver sub-sección Factor de emisión OM).

- 100 valores que representan la monótona de generación utilizada para calcular el factor lambda.

La información de mayor utilidad para el usuario, de acuerdo con el objetivo del presente manual, es contar con los factores de emisión CM ($EF_{grid,CM,y}$) para cada año y crónica; siendo el resto de la información complementaria, y con la cual es posible reproducir los cálculos según se presenta en la metodología a continuación.

El tipo de factor de emisión OM utilizado para calcular CM depende del criterio del usuario, aunque existe una limitación para el uso del factor “OM simple”, según se presenta en la subsección Factor de emisión OM.

Para un año dado, las distintas crónicas muestran el rango de variación (la distribución) de los valores de la información presentada, y en particular del factor de emisión CM. La elección de un factor CM anual queda a criterio del usuario, siendo conveniente elegir alguna medida de la tendencia central de la muestra, como por ejemplo la mediana o la moda.

A. Metodología de cálculo

i. Factor de emisión OM

Se incorporaron al SimSEE tres procedimientos de cálculo del factor de emisión OM ($EF_{grid,OM,y}$):

- OM simple
- OM simple ajustado
- OM promedio

a) OM simple

El factor de emisión OM simple ($EF_{grid,OMsimple,y}$) puede ser utilizado solamente si los recursos LCMR constituyen menos del 50% de la generación total del sistema: 1) promediando los cinco años más recientes, o 2) basado en promedios de largo plazo para la producción de energía hidroeléctrica. Se debe tener presente que esto en la actualidad no se cumple para el sistema eléctrico uruguayo.

$EF_{grid,OMsimple,y}$ se calcula como el promedio ponderado -según la generación de energía- de las emisiones de CO₂ de todas las plantas generadoras del sistema, sin incluir las que son LCMR, por unidad de energía eléctrica neta generada (tCO₂/MWh). Entonces:

$$EF_{grid,OMsimple,y} = \frac{\sum_m (EG_{m,y} \times EF_{EL,m,y})}{\sum_m EG_{m,y}} \quad (3)$$

Donde:

$EG_{m,y}$ = generación neta de electricidad entregada a la red por la unidad m e el año y (MWh).

$EF_{EL,m,y}$ = factor de emisión de CO₂ de la unidad m en el año y (tCO₂/MWh).

m = todas las unidades generadoras en el año y exceptuando las que son LCMR.

El factor de emisión de CO₂ de cada unidad m debe ser ingresado por el usuario en el Editor del SimSEE según se indicó en la sección Factor de emisión de una unidad/planta generadora.

El SimSEE acumula anualmente, para cada crónica, la energía neta generada por cada unidad/planta ($EG_{m,y}$). A partir de ésta y del $EF_{EL,m,y}$ calcula el $EF_{grid,OMsimple,y}$ para cada año y de las simulaciones, considerando solamente las unidades/plantas generadoras LCMR.

b) OM simple ajustado

El factor de emisión OM simple ajustado ($EF_{grid,OM-adj,y}$) es una variante del OM simple, que considera por separado las unidades/plantas generadoras (incluyendo las importaciones) LCMR (k) y HCMR (m). Al igual que el OM simple, se calcula en base a la energía neta generada por cada unidad y su factor de emisión, según la siguiente ecuación:

$$EF_{grid,OM-adj,y} = (1 - \lambda_y) \times \frac{\sum_m (EG_{m,y} \times EF_{EL,m,y})}{\sum_m EG_{m,y}} + \lambda_y \times \frac{\sum_k (EG_{k,y} \times EF_{EL,k,y})}{\sum_k EG_{k,y}} \quad (4)$$

Donde:

- λ_y = (lambda) factor que expresa el porcentaje de tiempo en que las unidades LCMR marginan en el año y (h/h).
- $EG_{m,y}$ = generación neta de electricidad entregada a la red por la unidad m en el año y (MWh).
- $EG_{k,y}$ = generación neta de electricidad entregada a la red por la unidad k en el año y (MWh).
- $EF_{EL,m,y}$ = factor de emisión de CO_2 de la unidad m en el año y (tCO_2/MWh).
- $EF_{EL,k,y}$ = factor de emisión de CO_2 de la unidad k en el año y (tCO_2/MWh).
- m = todas las unidades generadoras en el año y exceptuando las que son LCMR.
- k = todas las unidades generadoras en el año y que son LCMR.

Los factores de emisión de CO_2 de cada unidad (m o k) deben ser ingresados por el usuario en el Editor SimSEE según se indicó en la sección Factor de emisión de una unidad/planta generadora.

El SimSEE acumula anualmente, para cada crónica, la energía neta generada por cada unidad/planta ($EG_{m,y}$, $EG_{k,y}$). A partir de éstas, de λ_y y de los factores $EF_{EL,m,y}$ y $EF_{EL,k,y}$ calcula el $EF_{grid,OM-adj,y}$ para cada año y de las simulaciones, considerando todas las unidades/plantas generadoras: LCMR y HCMR.

A continuación se presenta: **1)** la definición del parámetro λ_y según el documento de referencia (“Tool to calculate the emission factor for an electricity system”), y **2)** cómo calcula el SimSEE dicho parámetro.

1) El parámetro λ_y se define como:

$$\lambda_y = \frac{\text{Número de horas que marginan las fuentes LCMR en el año } y}{8760 \text{ horas por año}} \quad (5)$$

Según el documento de referencia lambda (λ_y) debe ser calculada siguiendo los pasos a continuación (ver Figura 3):

- i. Graficar la curva de duración de carga. Ordenar de forma monótona decreciente la potencia entregada por el sistema en cada hora del año y (típicamente en MW). Graficar MW versus las 8760 horas en el año, en orden decreciente.
- ii. Calcular la generación total anual (en MWh) de las plantas/unidades LCMR ($\sum_k EG_{k,y}$).

- iii. Llenar la curva de duración de carga. Graficar una línea horizontal a través de la curva de duración de carga de tal forma que el área debajo de la curva (MW por hora) iguale la generación total de las unidades/plantas LCMR ($\sum_k EG_{k,y}$).
- iv. Determinar el “Número de horas que marginan las fuentes LCMR en el año y ”. Primero se debe localizar la intersección de la línea horizontal graficada en el paso (iii) y la curva de duración de carga graficada en el paso (i). El número de horas a la derecha de la intersección es el número de horas que marginan las fuentes LCMR. Si las líneas no se intersectan, se debe concluir que las fuentes LCMR no marginan y λ_y es igual a cero.

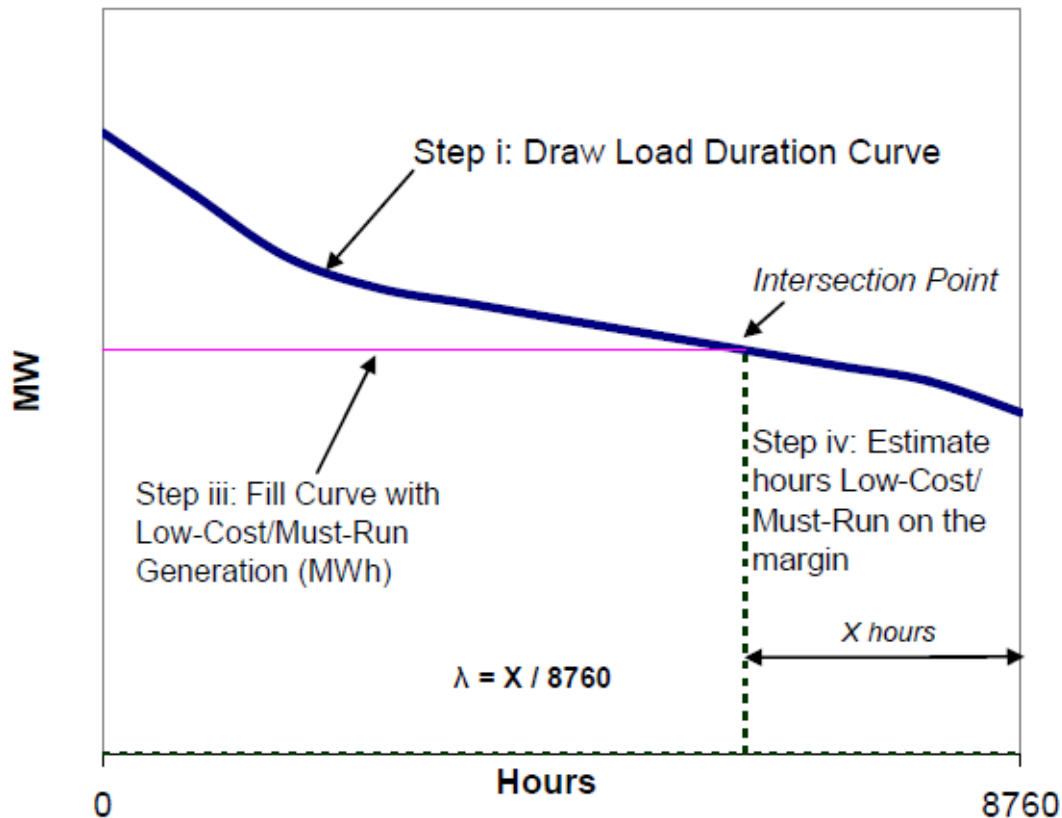


Figura 3 – Ilustración del cálculo de lamda (fuente: Tool to calculate the emission factor for an electricity system).

2) La duración del paso de tiempo en el SimSEE es definida por el usuario dependiendo de la longitud del período de simulación y optimización. Para simulaciones de algunos años, en general se utiliza paso de tiempo semanal. Por lo tanto, no se posee la información horaria de carga/potencia.

Para estimarla, se utiliza la división del paso de tiempo en postes que permite el SimSEE.

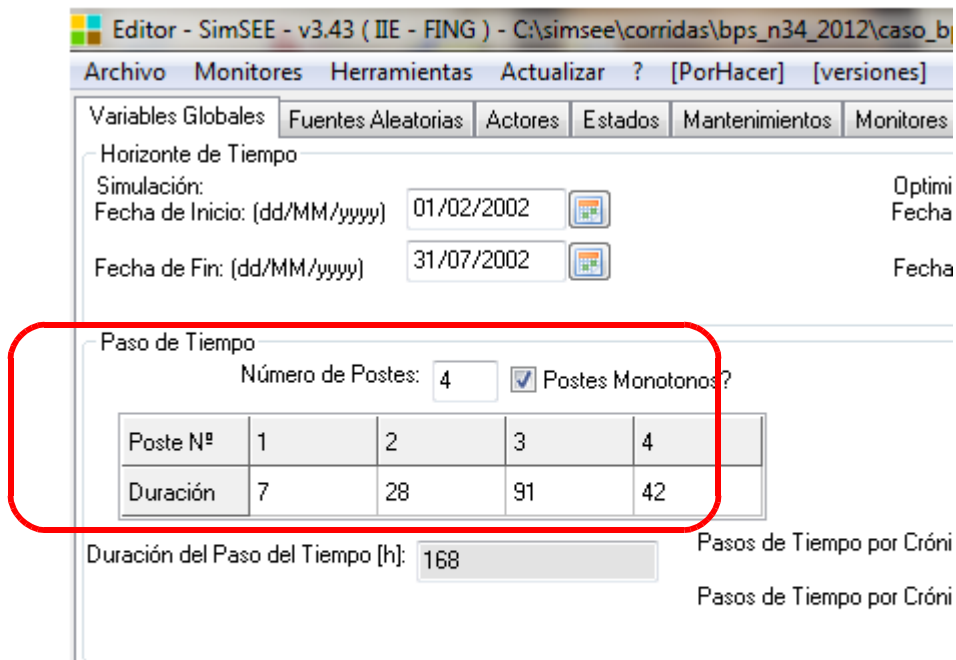
El número de postes y su duración es definido por el usuario en la pestaña Variables Globales del Editor SimSEE (Figura 4). A mayor número de postes más se parecerá la curva de duración de carga estimada a la horaria (en el extremo de que se eligieran tantos postes de duración de una hora como horas hay en el paso de tiempo, las curvas serían iguales), pero mayor será también el tiempo de cálculo de las simulaciones.

Se debe verificar que esté seleccionado la opción de Postes Monótonos, ya que esto implica un ordenamiento de las curvas de demandas de forma que la mayor potencia se agrupe en el poste 1, luego el poste 2 y así sucesivamente hasta agrupar las horas de menor potencia en el último poste. Entonces, para cada paso de tiempo la potencia queda ordenada en tantos intervalos como postes

se elijan. Para cada poste se tiene la potencia media en la cantidad de horas definidas para dicho poste.

Para cada año y crónica se ordena la potencia por poste de manera monótona decreciente y se grafica potencia versus horas del año, asignándole a la cantidad de horas de cada poste la potencia media del mismo.

De esta forma la curva horaria queda escalonada, por ejemplo: si el paso de tiempo se divide en 4 postes de 42 horas de duración cada uno (paso de tiempo semanal de 168 horas, que equivale a una semana), en la gráfica habrá 42 horas consecutivas con cada valor de potencia.



Poste Nº	1	2	3	4
Duración	7	28	91	42

Figura 4

El total de horas en un año en el SimSEE puede no ser exactamente 8760 debido a que, por ejemplo, considera que el año tiene 52 semanas. De todas formas para cada año tiene un contador de la cantidad de horas, y calcula lambda como:

$$\lambda_y = \frac{\text{Número de horas que marginan las fuentes LCMR en el año } y}{\text{Número total de horas en el año}} \quad (6)$$

En el documento “simres_CO2_SemAleatoria_Simulaciones” aparece, como se mencionó anteriormente, cien valores de potencia que representan la curva de duración de carga para cada año de la simulación y cada crónica (ver ejemplo en la Figura 5).

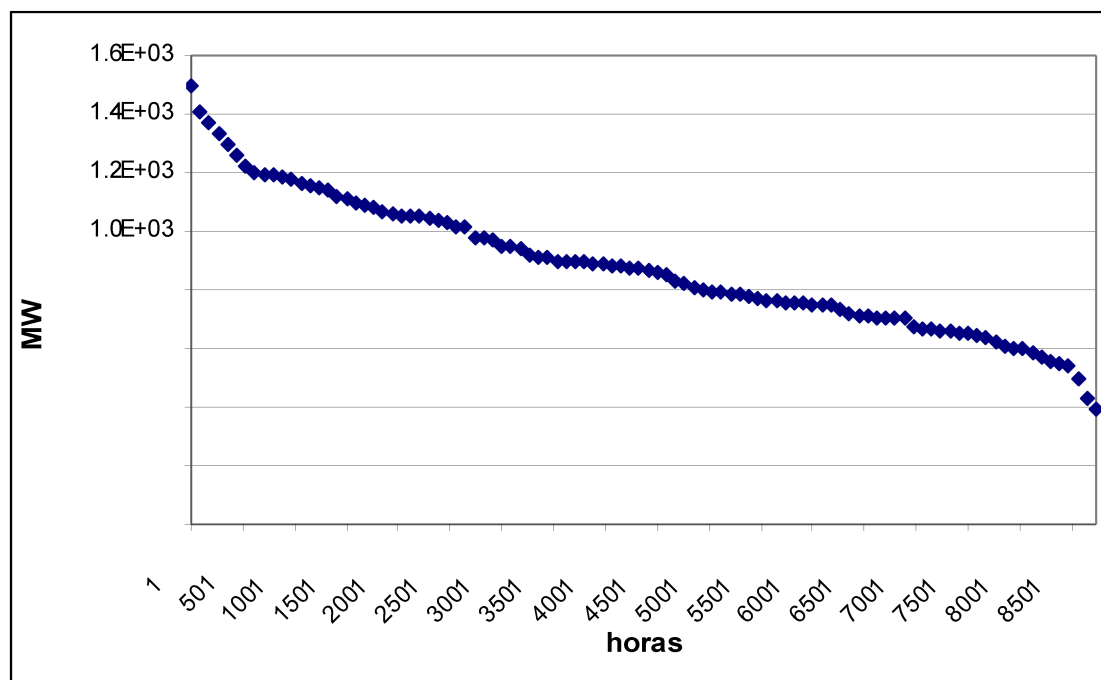


Figura 5 – Ejemplo de curva de duración de carga para un año y una crónica dados (representada mediante 100 valores).

c) OM promedio

El factor de emisión OM promedio ($EF_{grid,OM-ave,y}$) se calcula como la tasa de emisión promedio de todas las unidades/plantas generadoras del sistema eléctrico. Se utiliza la misma metodología descrita arriba para el OM simple, pero incluyendo también las fuentes LCMR en la ecuación (3).

ii. Factor de emisión BM

El grupo de unidades/plantas generadoras que se utiliza, cada año y , para el cálculo del factor de emisión BM ($EF_{gris,BM,y}$) se determina siguiendo el procedimiento a continuación:

- Identificar las cinco unidades más recientemente incorporadas al sistema eléctrico ($SET_{5-units}$), excluyendo aquellas que estén registradas como proyectos CDM, y determinar su generación anual de electricidad ($AEG_{SET-5-units}$, en MWh).
- Identificar las unidades más recientemente incorporadas al sistema eléctrico que generaron en el año y al menos el 20% del total de la generación anual de electricidad ($SET_{\geq 20\%}$), excluyendo aquellas que estén registradas como proyectos CDM, y determinar su generación anual de electricidad ($AEG_{SET \geq 20\%}$).
- De (a) y (b) seleccionar el set de unidades cuya generación anual de electricidad sea mayor (SET_{sample}). Identificar las fechas en que las unidades del SET_{sample} fueron incorporadas al sistema eléctrico. Si ninguna de estas unidades comenzó a suministrar electricidad a la red más de 10 años atrás, entonces utilizar SET_{sample} para calcular el factor de emisión BM. En este caso ignorar los pasos (d) y (e).
- De lo contrario:** excluir de SET_{sample} las unidades que comenzaron a suministrar energía a la red más de 10 años atrás. Incluir las unidades registradas como proyectos CDM, comenzando con las incorporadas más recientemente al sistema, hasta que la generación de energía del grupo supere el 20% del total de la generación anual de electricidad o hasta incorporar todos los proyectos CDM. Determinar para el set resultante ($SET_{sample-CDM}$) su

generación anual de energía ($AEG_{SET-sample-CDM}$). Si $AEG_{SET-sample-CDM}$ es mayor o igual al 20% del total de la generación anual de electricidad, utilizar $SET_{sample-CDM}$ para calcular el factor de emisión BM.

- (e) **De lo contrario:** incluir en el grupo las unidades que comenzaron a suministrar electricidad a la red más de 10 años atrás hasta que la generación anual del nuevo set ($SET_{sample-CDM->10yrs}$) sea mayor o igual al 20% del total de la generación anual de electricidad. $SET_{sample-CDM->10yrs}$ se debe utilizar para el cálculo del factor de emisión BM.

Con el set de unidades/plantas generadoras seleccionadas, el factor de emisión BM se calcula, para cada año y y para cada crónica de simulación, como el promedio ponderado (según la generación de energía) de los factores de emisión (tCO_2/MWh) de cada unidad/planta n del set. Entonces:

$$EF_{grid,BM,y} = \frac{\sum_n (EG_{n,y} \times EF_{EL,n,y})}{\sum_n EG_{n,y}} \quad (7)$$

Donde:

- $EG_{n,y}$ = generación neta de electricidad entregada a la red por la unidad n en el año y (MWh).
 $EF_{EL,n,y}$ = factor de emisión de CO_2 de la unidad n en el año y (tCO_2/MWh).
 n = todas las unidades generadoras incluidas en el grupo seleccionado para el cálculo de BM en el año y , siguiendo el procedimiento (a)-(e) explicado arriba.

El factor de emisión de CO_2 de cada unidad n debe haber sido ingresado por el usuario en el Editor del SimSEE según se indicó en la sección Factor de emisión de una unidad/planta generadora.

Utilizando los datos proporcionados por el usuario acerca del año de ingreso al sistema eléctrico de cada unidad/planta generadora y si se tratan de proyectos registrados como CDM, el SimSEE sigue los pasos (a)-(e) indicados arriba y determina el grupo de unidades/plantas a considerar para el cálculo del factor de emisión BM.

Como se mencionó anteriormente para el cálculo del factor OM, el SimSEE acumula anualmente, para cada crónica, la energía neta generada por cada unidad/planta ($EG_{n,y}$). A partir de ésta y del $EF_{EL,n,y}$ calcula el $EF_{grid,OB,y}$ para cada año y de las simulaciones, considerando solamente las unidades/plantas incluidas en el grupo seleccionado para el cálculo de BM de dicho año.

iii. Factor de emisión CM

Finalmente, a partir de los factores de emisión OM y BM, se calcula el factor de emisión de margen combinado ($EF_{grid,CM,y}$) para cada año y y para cada crónica de simulación.

El factor de emisión de margen combinado se calcula como un promedio ponderado de los factores OM y BM, según la siguiente ecuación:

$$EF_{grid,CM,y} = EF_{grid,OM,y} \times w_{OM} + EF_{grid,BM,y} \times w_{BM} \quad (8)$$

Donde w_{OM} y w_{BM} son los pesos de los factores de emisión OM y BM, respectivamente. w_{OM} y w_{BM} dependen del tipo de proyecto CDM propuesto por el usuario:

- Para proyectos de energía eólica o solar $w_{OM}=0.75$ y $w_{BM}=0.25$
- Para otros proyectos $w_{OM}=0.5$ y $w_{BM}=0.5$

Estos pesos son los propuestos por el documento “Tool to calculate the emission factor for an electricity system” para el primer período de crédito del proyecto CDM.

Como se mencionó al inicio de esta sección (Factor de emisión del sistema eléctrico), el usuario puede seleccionar en la pestaña “CO2” del Editor SimSEE si su proyecto corresponde a generación de energía solar o eólica, con lo que el SimSEE utiliza $w_{OM}=0.75$ y $w_{BM}=0.25$. Si esta opción no se marca, utiliza por defecto $w_{OM}=w_{BM}=0.5$.

Capítulo XV. Conclusión y recomendaciones

El presente documento no sustituye al procedimiento descrito en el documento “Tool to calculate the emission factor for an electricity system” (documento base), sino que trata de adaptar la metodología allí descrita para su aplicación en el SimSEE. Esta adaptación y el presente manual están pensados para su utilización en el cálculo de emisiones del sistema eléctrico uruguayo. No se incluyen en los cálculos plantas/unidades generadoras “off-grid”.

El documento base propone la utilización de datos observados para realizar los cálculos. El SimSEE permite calcular el factor de emisión en años futuros, a partir de simulaciones de la operación del sistema eléctrico con una política de operación óptima.

Las mayor ventaja es que permite calcular el factor de emisión en años donde el proyecto CDM considerado estará en funcionamiento, teniendo en cuenta aspectos del futuro como el aumento de la demanda de energía, o la incorporación de nuevas unidades/plantas generadoras que están proyectadas para un futuro cercano.

Una posible desventaja es que, debido a incertidumbres acerca del futuro, como los aportes a las represas, la rotura de máquinas, variación del precio del petróleo, entre otras, existen diversas posibilidades en cuanto a la operación del sistema eléctrico, las cuales se representan mediante el número de crónicas de la simulación. Para cada crónica el factor de emisión anual de CO₂ calculado es distinto. Por lo tanto, para cada año del período de simulación, no se tiene un solo valor del factor de emisión del sistema, sino su rango de variación, la distribución de los valores.

Finalmente, las recomendaciones de uso del SimSEE para el cálculo de la línea de base de emisiones para un proyecto que sustituye electricidad de la red se pueden resumir como las siguientes:

- El período de simulación elegido debería extenderse en el tiempo en la medida que se posea cierto conocimiento sobre cómo evolucionará el parque generador.
- Elegir el número de crónicas de la simulación suficientemente grande (ejemplo: 100 crónicas) para tener una muestra amplia de la distribución de los valores anuales del factor de emisión de CO₂.
- La elección de un factor CM anual que represente la emisión del sistema para dicho año queda a criterio del usuario, siendo conveniente elegir alguna medida de la tendencia central de la muestra (crónicas), como por ejemplo la mediana o la moda.
- Es conveniente definir algún procedimiento de monitoreo del factor de emisión elegido para cada año.