

Proyecto: ANII_FSE_1_2017_1_144926
SimSEE+Variabilidad+Red+DemandaConRespuesta.

Memoria Final.

"TRACTORCITO"

Planificación de inversiones con energías variables, restricciones de red y gestión de demanda.



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



<https://simsee.org/investigacion/tractorcito.html>

Instituto de Ingeniería Eléctrica.

Facultad de Ingeniería.

Universidad de la República Oriental del Uruguay.

Setiembre 2020.

Investigadores:

Pablo Soubes, Ximena Caporale, Felipe Palacio, Damián Vallejo y Sergio Martinez.

Investigador y Co-repsonable científico:

Ignacio Ramírez

Investigador y Responsable científico:

Ruben Chaer

Índice

1. Resumen.....	5
2. Algoritmo de aproximación eficiente del Costo Futuro.....	6
2.1. Introducción.....	6
2.2. Revisión de técnicas de aproximación de funciones.....	12
2.2.1 Transformaciones y agregaciones.....	13
Espacio de Características. (features).....	14
Agregaciones espaciales.....	14
Agregaciones temporales.....	14
2.2.2 Estructuras de modelado.....	15
Mínimos Cuadrados. (LSM).....	16
Redes Neuronales (NN).....	17
Radial Basis Functions (RBF).....	19
Fuzzy Logic Models. (FLM).....	19
Support Vector Machines (SVM).....	19
Mixture Density Networks (MDN).....	20
Espacios Grillados.....	21
Kernels. Aproximación por cercanía.....	21
Redes Neuronales Recurrentes. (RNN).....	22
2.3. Regularización.....	22
2.4. Alternativas para representar la información útil para el despacho óptimo de un paso de tiempo.....	23
2.4.1 Aproximación a la función de Valor.....	23
2.4.2 Aproximación a las diferencias espaciales desde uno de los estados visitados.....	24
2.4.3 Aproximación a las diferencias espaciales desde cualquiera de los estados visitados.....	24
2.4.4 Aproximación a la función de Valor con nivel libre.....	25
3. Incorporación a la plataforma SimSEE.....	28
3.1. Bucle de mejora de la política de operación (PO).....	29
3.2. Función de Valor y Función de Valor-Acción.....	30
3.3. El Tratorcito.....	31
3.4. Bucle del Tratorcito.....	32
3.5. El Hiperplanador.....	32
4. Inicialización y Convergencia, Redes Neuronales.....	32
4.1. Valores iniciales, regularización y especialización de las neuronas.....	32
4.2. Exploración vs. Dinámica.....	39
4.2.1 Exploración del espacio de estado.....	43
4.2.2 Distribución inicial de estados.....	43
4.2.3 CONVERGENCIA DEL ALGORITMO.....	48
4.2.4 IMPLEMENTACIÓN.....	49
4.2.5 Condiciones de convergencia:.....	50
4.3. Reducción de la Varianza. Coherencia de la Suerte.....	50
4.4. Inicialización de las Neuronas para permitir su especialización en la explicación de zonas del espacio de estado.....	57
4.4.1 trbr = 0. "Aleatorio radio 1.3 anidado.".....	58
4.4.2 trbr = 1. "Lineal entre 8 y 10.".....	58

5. Enganche de las Políticas de Operación.....	60
6. Parsimonia Temporal.....	63
7. Comparación con Política de Operación Clásica.....	67
7.1. Consideraciones del efecto de la cantidad de discretizaciones del volumen embalsado y de la imposición de valor del agua positivo sobre la Política de Operación del lago de Rincón de Bonete obtenida con el algoritmo clásico.....	67
7.1.1 Resumen ejecutivo.....	68
7.1.2 Introducción.....	68
7.1.3 Efecto de la cantidad de discretizaciones.....	68
7.1.4 Efecto de imponer $cva > 0$	70
7.1.5 Estadística de la cota del lago.....	70
7.1.6 Comparación de los Costos Futuros.....	71
7.1.7 Intermediario de Costo Futuro.....	72
8. Resultados sobre el Sistema dinámico simple.....	74
8.1. Configuración del sistema de prueba.....	74
8.2. Resultados.....	75
8.3. Resultados en algoritmo clásico.....	76
9. Convergencia de la aproximación lineal de orden 1 para el caso de la Sala de Largo Plazo (Planificación de Inversiones).....	77
9.1. Tipo de aproximador.....	79
9.2. Parsimonia temporal.....	81
9.3. Propagación de la información y Pasos Hacia Delante.....	82
10. Sistema Hidro-térmico con Aportes con Estado y costo térmico exponencial.....	86
10.1. Sala Caso 1. Resolución clásica.....	88
11. Caso de prueba sobre la operación de Corto Plazo del Sistema.....	92
12. Revisión bibliográfica.....	98
12.1. Abstract Dynamic Programming (Bertsekas).....	98
12.2. Learning and Soft Computing Support Vector Machines, Neural Networks, and Fuzzy Logic Models, Vojislav Kecman, 2001 MIT.....	98
12.3. DYNAMIC PROGRAMMING.....	98
12.4. Iterative Gaussianization: From ICA to Random Rotations.....	98
12.5. RADIAL GAUSSIANIZATION WITH CLUSTER-SPECIFIC BIAS COMPENSATION... ..	99
12.6. A DAM : A M ETHOD FOR S TOCHASTIC O PTIMIZATION.....	99
12.7. Analysis of stochastic dual dynamic programming method.....	99
12.8. Reinforcement Learning: An Introduction.....	100
12.9. Variance Reduction in Population-Based Optimization: Application to Unit Commitment... ..	100
12.10. SOME GUIDELINES AND GUARANTEES FOR COMMON RANDOM NUMBERS*.....	101
12.11. Dynamic Programming and Optimal Control - Vol I.....	101
12.12. Dynamic Programming and Optimal Control 3rd Edition, Volume II Approximate Dynamic Programming.....	101
12.13. Improving the performance of Stochastic Dual Dynamic Programming.....	101
12.14. Optimal Long-Term Distributed Generation Planning and Reconfiguration of Distribution Systems: An Accelerating Benders' Decomposition Approach.....	102
12.15. On variance reduction of mean-CVaR Monte Carlo estimators.....	102
12.16. Multi-stage stochastic optimization applied to energy planning.....	103
12.17. SDDP vs. ADP: The Effect of Dimensionality in Multistage Stochastic Optimization for Grid Level Energy Storage.....	103

12.18. Neuro-Dynamic Programming.....	103
12.19. On the solution variability reduction of Stochastic Dual Dynamic.....	104
12.20. Approximate Dynamic Programming.....	104
12.21. Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling	105
12.22. Stochastic dual dynamic programming applied to nonconvex hydrothermal models.....	105
12.23. Programación Dinámica Estocástica Revisión Bibliográfica.....	106
13. Conclusiones y trabajos futuros.....	106
14. Anexo: Cronograma del proyecto.....	107
15. Anexo: Introducción al lenguaje de optimización.....	108
16. Anexo: Caso base para prueba de administrador de estados multiplano.....	115
16.1. Introducción.....	115
16.2. Pruebas Realizadas.....	116
17. Anexo: Informe con estudio de caso de Planificación de Inversiones.....	122

1. Resumen

El objetivo principal del proyecto era la búsqueda de técnicas de aproximación a la Política Óptima de Operación de un Sistema Dinámico Estocástico. En particular aplicada al sistema de generación de energía eléctrica. La resolución de este tipo de problemas sufre de lo que se conoce como La Maldición de la Dimensionalidad de Bellman que establece que la complejidad de resolución del problema de encontrar la Política Óptima para la operación de un sistema dinámico crece exponencialmente con el aumento de las dimensiones del Espacio de Estado del sistema. El proyecto no es más que un intento de luchar contra dicha maldición.

El conjunto de herramientas desarrolladas dieron lugar a una aplicación que llamamos: **Tractorcito** que dada una Sala SimSEE y una estructura de aprendizaje realiza simulaciones para ir aprendiendo la manera de comportarse (Política de Operación) que resulte en un menor costo de producción. A la aplicación la llamamos Tractorcito, pues dado que, sobre todo al inicio, las simulaciones conducen al Sistema "por cualquier lugar" y el Simulador, debe funcionar igual para poder calcular el costo de operación, el simulador en este modo recuerda al funcionamiento de un Tractor capaz de atravesar por cualquier camino, bañado, cerro de piedra, etc.

El desarrollo del Tractorcito, implicó la adaptación de la herramienta SimSEE para poder actuar en conjunto con la capacidad de aprendizaje por refuerzo del Tractorcito.

El Tractorcito está dotado de un conjunto de Redes Neuronales cuyos parámetros van cambiando durante la experiencia del propio Tractorcito. La capacidad de aprendizaje está dada por la complejidad asignada a dichas Redes Neuronales. Para un sistema dado y complejidad de representación, el Tractorcito que logra menor costo es el más exitoso. El problema de aprendizaje está planteado en el sentido de que sabiendo que la complejidad de representación es limitada, dicha capacidad sea optimizada (durante el aprendizaje continuo) a aquellos aspectos que resultan más relevantes en el objetivo de reducción el valor esperado del costo de la operación futura.

El resultado clásico (recursión de Bellman) es que la información que determina la Política de Operación se encuentra en las derivadas espaciales de la función que determina el costo esperado de la operación futura a partir de un estado dado.

En la primer etapa del proyecto se tomaron decisiones respecto de las técnicas a implementar para la resolución del problema principal (Programación Dinámica Estocástica Aproximada).

Cumpliendo con lo resuelto se implementó un módulo de Aproximación General a cualquier tipo de función $CF(X): \mathcal{R}^n \rightarrow \mathcal{R}$.

En la aplicación del algoritmo se distingue la necesidad de combinar las iteraciones de mejora de la Política de Operación en base a simulaciones hacia adelante (forward) en las que el estado del sistema evoluciona siguiendo la ecuación de evolución de estado con simulaciones en las que el estado evoluciona de forma "guiada" a los efectos de obtener una exploración más robusta del espacio de Estados.

Para la implementación de esta nueva aproximación del valor esperado del costo de la operación futura (Costo Futuro), se tuvieron que implementar varios cambios en la plataforma SimSEE, las cuales se detallan en la sección 3

El objetivo principal de este trabajo es la identificación, desarrollo e implementación de técnicas de representación aproximada de la información contenida en CF , este aspecto de la implementación se documenta en la sección 2, en donde se hace un listado de las técnicas estudiadas de las cuales se implementaron dos de ellas; Mínimos Cuadrados y Redes Neuronales.

La información útil de la Política de Operación se encuentra en las derivadas del Costo Futuro respecto a los estados por lo que es importante poder representar la variación δCF asociada a un movimiento del estado, esto lo desarrollaremos en detalle en la sección 2.4.

En la sección 4 se analizan aspectos claves en la resolución de aproximar la función de costo futuro utilizando RN, como por ejemplo cuales son las mejores configuraciones para las redes neuronales en el caso de querer representar el CF, inicialización de las neuronas para evitar mínimos locales, modos de evolución para lograr abarcar todo el espacio de estados en el entrenamiento de la red neuronal y la especialización de cada neurona para la explicación del CF.

A medida que fue transcurriendo el proyecto fueron surgiendo simplificaciones muy importantes, las cuales implican ahorros en tiempo y poder de computo importantes, entre ellos se encuentra la capacidad de poder realizar enganches de Políticas de Operación, esto facilita en el sentido de poder dividir los horizontes de simulación en etapas, para esto es necesario realizar correctamente los enganches, que si bien esto ya se hacia en la política de operación convencional a partir de la grilla de valores de CF, se vio la necesidad de lograr los enganches de las derivadas del CF, este aspecto se detalla en la sección 5. La otra simplificación es la posibilidad de incorporar una parsimonia en los datos, ya sea continuidad o estacionalidad anual, que claramente existe en los sistemas físicos, como el sistema eléctrico, en la sección 6 se muestra la implementación.

En las secciones 7, 8, 9 y 10 se analizan distintos resultados, con enfoque en la salas de largo plazo del sistema uruguayo, comparando los resultados con el método convencional. Finalmente en la sección 12 se realiza un revisión de la bibliografía con la cual se baso este proyecto.

En el algoritmo clásico (recursión de Bellman) la disponibilidad de las unidades de generación queda mal representada al perderse la inercia de los procesos de rotura/reparación (por realizar sorteos independientes en cada paso de la etapa de optimización). En el algoritmo propuesto (aproximación de CF por simulaciones hacia adelante) la inercia de los procesos rotura/reparación está correctamente representada aunque no se defina en forma explícita el estado de las unidades. Las simulaciones hacia adelante no tienen las simplificaciones que son necesarias hacer en la recursión "hacia atrás" en el tiempo.

2. Algoritmo de aproximación eficiente del Costo Futuro.

2.1.Introducción.

En la actualidad, buena parte de los algoritmos de resolución de problemas de optimización de sistemas dinámicos se basan en utilizar lo que se conoce como Función de Valor del Estado, función de Costo Futuro (CF) o función de Valor de Bellman [1]. Esta función representa el costo de la

operación futura del sistema a partir de un estado y tiempo dados si se utiliza una Política de Operación dada.

El objetivo principal de este trabajo es la identificación, desarrollo e implementación de técnicas de representación aproximada de la información contenida en CF que resulta útil a los propósitos de su uso en una Política de Operación de un Sistema Dinámico. Por facilidad del lenguaje hablaremos de técnicas de representación aproximada de CF (función de Costo Futuro), pero no se debe perder de vista que lo que nos interesa es representar la información contenida en dicha función que resulta útil a la Política de Operación más que la función en si mismo.

El sistema dinámico sobre el que vamos a aplicar las técnicas desarrolladas son sistemas de generación eléctrica. Estos sistemas se caracterizan por tener un espacio de estado de múltiples dimensiones, por presentar pseudo-estacionalidades (anuales y diarias), por ser variantes en el tiempo (cambia al estructura al aumentar la demanda y cambiar las tecnologías utilizables de generación) y por tener componentes estocásticas relevantes (variabilidad de la generación hidráulica, eólica y solar y precios de los combustibles). Estas características, hacen que en la práctica sea muy difícil el problema de obtener una política óptima de operación aplicando el algoritmo recursivo planteado por Bellman[1]. En la misma referencia, Bellman establece que dicho algoritmo recursivo tiene complejidad explosiva con el aumento de las dimensiones del estado, de las acciones de control y de las componentes aleatorias. Este aumento explosivo de la complejidad es lo que se ha dado en llamar la "Maldición de la Dimensionalidad de Bellman" y que se deba recurrir a técnicas aproximadas [2] y [3].

La implementación de las herramientas de simulación de la operación están basadas en disponer de la información del sistema en un Horizonte de Tiempo y discretizar dicho horizonte en Pasos de Tiempo (o Etapas). En este trabajo identificaremos el Paso de Tiempo con la letra k .

Si $X \in \Omega_X(k)$ es el vector de Estado¹ del sistema y $k \in E$ representa el tiempo (o paso de tiempo) la función CF será un mapeo como se expresa en la ec.1, en donde $\Omega_X(k)$ es el conjunto que puede variar el estado (o dominio del estado). Como se puede apreciar el dominio del estado depende del tiempo k contemplando así que la estructura del sistema puede cambiar.

$$CF(X, k) : \Omega_X(k) \times E \rightarrow \mathfrak{R} \quad \text{ec.(1) Función de Costo Futuro.}$$

El modelo del sistema se puede expresar a partir de su **Ecuación de Evolución de Estado**, también denominada **Restricción Dinámica** del sistema que tendrá la forma de la ec.2. Esta ecuación nos permite conocer el estado al que evolucionará el sistema al final del paso de tiempo k conociendo el estado, las entradas no controlables y el valor de las entradas de control al inicio del paso.

$$X_{k+1} = f(X_k, r_k, u_k, k) \quad \text{ec.(2) Modelo del sistema. Ecuación de evolución del estado.}$$

¹Estado del sistema con el significado utilizado en teoría de control, es decir un vector que contiene la información relevante del pasado del sistema para poder conocer su evolución futura a partir del conocimiento de sus entradas futuras.

Para completar el modelo del sistema, es necesario una ecuación que nos permita calcular el **Costo de Etapa** en función de la información del paso. El costo de etapa tendría entonces la expresión de la ec.3.

$$ce(X, r, u, k) : \Omega_X(k) \times \Omega_r(k) \times \Omega_u(k) \times E \rightarrow \mathfrak{R} \quad \text{ec.(3) Costo de Etapa.}$$

Si $X \in \Omega_X(k)$ es el vector de Estado del sistema, $k \in E$ representa el tiempo (etapa de cálculo en un horizonte de tiempo discretizado), $r \in \Omega_r(k)$ es el vector de entradas no controlables y $u \in \Omega_u(k)$ es el vector de control, una **Política de Operación** (PO) es un mapeo del espacio definido por el producto cartesiano de los dominios de X , r y k en el dominio del vector de control u lo que se puede expresar con la notación de la ec.4.

$$u = PO(X, r, k) : \Omega_X(k) \times \Omega_r(k) \times E \rightarrow \Omega_u(k) \quad \text{ec.(4) Política de Operación.}$$

Dada una PO , y un ensamble de realizaciones de las entradas no controlables es posible calcular el costo futuro de la operación con dicha política a partir de un estado y tiempo dado, usando el modelo (ec.2 y ec.3) como se expresa en la ec.5.

$$CF_{PO}(X_k, k) = \left\langle \sum_{j=k}^{j=+\infty} q^{(j-k)} ce(X_j, r_{hj}, u_j) \right\rangle_h \quad \text{ec.(5).}$$

En la ec.5 q es un factor positivo menor que 1 (uno) y representa una tasa de actualización del dinero. Es así que la ec.5 representa el valor presente esperado de la operación futura con la política $u_j = PO(X_j, r_{hj}, j)$ en el ensamble de las realizaciones de las entradas no controlables. Cada realización de dichas entradas está etiquetada por el subíndice h .

En este documento se entiende por Operación Óptima de un sistema dinámico, a aquella secuencia de los controles del sistema que llevan al mínimo valor esperado del costo de la operación futura. Una política $\check{P}O$ será óptima a partir de una ubicación (X_k, k) del sistema si aplicando la ecuación 5 en el conjunto de políticas posibles CPP se cumple la ec.6.

$$CF_{\check{P}O}(X, k) \leq CF_{PO}(X, k) \forall PO \in CPP \quad \text{ec.(6) Política Óptima}$$

En la referencia [1] Bellman desarrolla lo que se ha convertido en el "planteo clásico" de la resolución del problema de obtención de una política de operación óptima de un sistema dinámico planteando lo que se conoce como la recursión de Bellman. Esta recursión, nos permite plantear que si se conociera la función $CF_{\check{P}O}$ correspondiente a la Política Óptima al final de un paso de tiempo, es posible calcular su valor al inicio del paso de tiempo para un estado dado, minimizando la suma del costo de etapa más el valor de $CF_{\check{P}O}$ en el estado de llegada, en el conjunto de las realizaciones posibles de las variables no-controlables en el paso de tiempo, como se muestra en la ec.7. Dónde el operador $\langle \cdot \rangle_{r_k}$ significa el valor esperado en el ensamble de las realizaciones de las entradas no controlables r_k en el paso k .

$$CF_{\check{P}O}(X, k) = \min \left\langle ce(X, r, u, k) + CF_{\check{P}O}(f(X, r, u, k), k+1) \right\rangle_{r_k} \quad \text{ec.(7) Recursión de Bellman}$$

Para pasos de tiempo suficientemente pequeños, en relación a las variaciones de las variables no-controlables, el valor de r_k se puede suponer conocido durante el paso de tiempo y la ec.7 admite ser reformulada intercambiando la esperanza $\langle \cdot \rangle_{r_k}$ con el problema de búsqueda del mínimo como se muestra en la ec.8.

$$CF_{\check{P}O}(X, k) = \left\langle \min_u \{ ce(X, r, u, k) + CF_{\check{P}O}(f(X, r, u, k), k+1) \} \right\rangle_{r_k} \quad \text{ec.(8) Recursión de Bellman con entradas del paso conocidas.}$$

Observar que si es posible suponer conocidas las entradas r_k al inicio de la resolución del paso, el problema de minimización que se resuelve para cada valor de r_k en la ec.8 es la Política de Operación Óptima. No es objetivo de este trabajo discutir si las formulaciones de las ec.8 y 7 son o no equivalentes, lo relevante aquí es que ambas utilizan la función $CF_{\check{P}O}$ para resolver el problema de la Operación Óptima en un paso de tiempo y que en ambas implementaciones la suma de un valor constante para cada paso de tiempo a $CF_{\check{P}O}$ no cambia la operación en lo que refiere a la determinación del control óptimo (es decir en la determinación de $\check{P}O$). Entonces, a los efectos del uso de $CF_{\check{P}O}$ para la determinación de $\check{P}O$ usando la Recursión de Bellman, dada una $CF_{\check{P}O}(X, k)$ (correspondiente a una política óptima) entonces las función $CF(X, k)$ obtenidas como se muestra en la ec.9 corresponden también a $\check{P}O$.

$$CF(X, k) = CF_{\check{P}O}(X, k) + \gamma(k) \quad \text{ec.(9) Políticas equivalentes.}$$

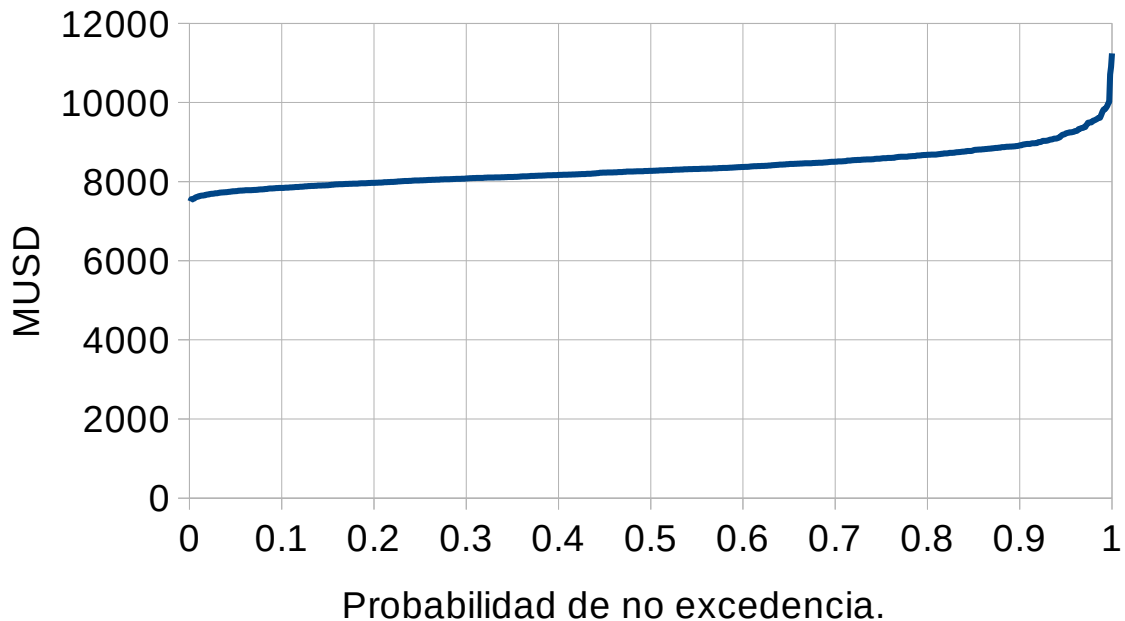


Figura 1: Costo Futuro de Uruguay 30 años actualizado al 5%.

Esto puede resultar algo confuso en el sentido de que $CF(X, k) = CF_{\tilde{p}O}(X, k) + \gamma(k)$ obviamente no representa El Costo de la Operación Futura si $CF_{\tilde{p}O}(X, k)$ si lo representaba, pero se quiere dejar claro que nuestro objetivo no es representar fielmente el costo de la operación futura por medio de una función $CF(X, k)$ sino representar la información relevante para la resolución de la Política Óptima de operación. Otro aspecto importante a considerar es que con $CF(X, k)$ estamos intentando representar el Valor Esperado del Costo de la operación Futura, pero la información que se recibe para realizar la aproximación proviene de simulaciones y lo que se tiene de dichas simulaciones es el costo futuro de la operación de cada realización simulada. La Fig.1 muestra la distribución del Costo Futuro del sistema uruguayo partiendo de un estado conocido para los siguientes 30 años obtenido en base a la simulación de 1000 realizaciones de los procesos estocásticos y operando con la política óptima obtenida con los métodos actualmente en uso en SimSEE. El valor esperado es 8348 MUSD, pero como se puede apreciar, aunque se esté observando el valor actualizado de 30 años, el costo futuro tiene dispersión. De la gráfica, el valor en riesgo 5% de ser excedido es: 9220 MUSD.

Es importante entonces tener en cuenta que las estimaciones de CF se están realizando en base a simulaciones y que la dispersión de CF llevaría a que si se comparan valores aislados de las simulaciones, para el mismo punto (X, k) hay diferencias del orden de 1000 MUSD. Esta dispersión (salvo que se tomen precauciones) ingresaría como un ruido en el término $CF_{\tilde{p}O}$ de la ec.8. Si la magnitud de ese ruido es comparable al costo de etapa entonces se tendrá un problema en la resolución de la ec.8. La Fig.2 muestra el histograma de permanencia del Costo de Etapa (paso semanal) de la misma simulación. Como se puede apreciar, el costo de etapa es sustancialmente inferior a la dispersión del Costo Futuro por lo que el tratamiento que se haga al ruido de estimación del valor esperado de CF puede ser determinante en la resolución de la ec.8.

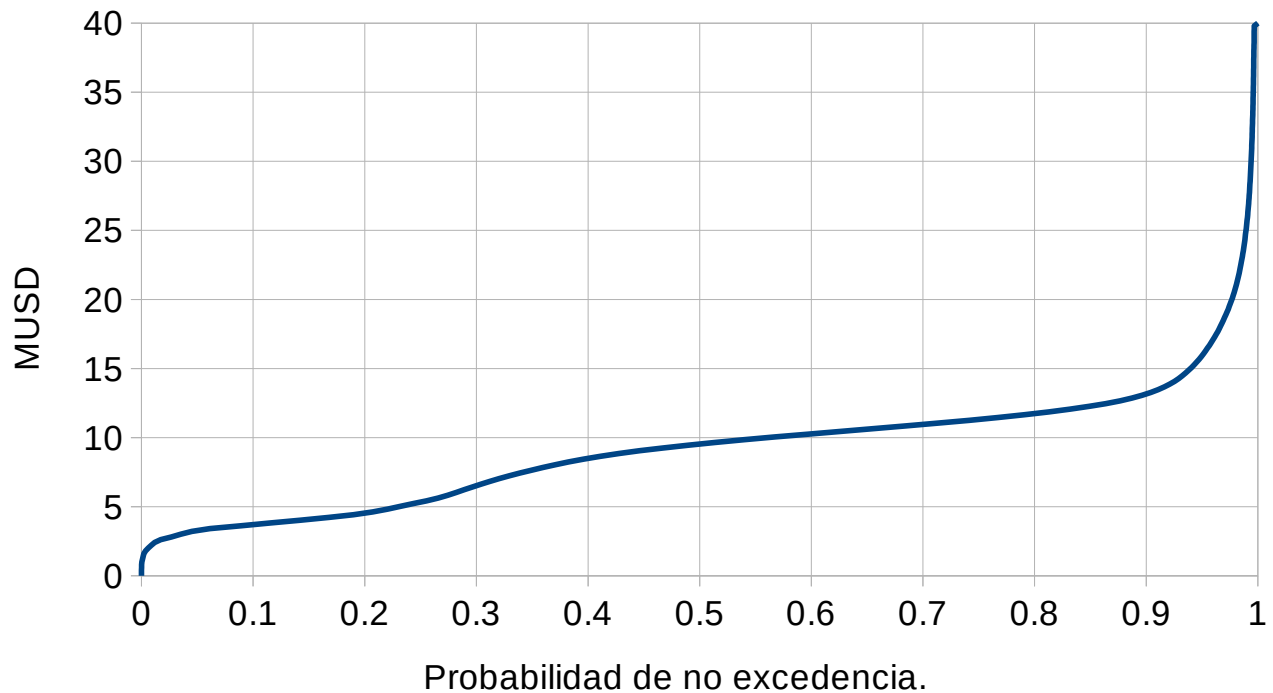


Figura 2: Distribución del Costo de Etapa (semanal). Simulación 1000 crónicas Uruguay 30 años.

Otra característica del sistema, es que las distribuciones asociadas al costo no son simétricas. Generalmente, el costo de abastecer la Demanda en una hora crece más que linealmente con el nivel de Demanda a cubrir. Como consecuencia de ello, puede observarse que la distribución de la Fig.2 es claramente no-simétrica. Esta misma asimetría se traduce (amortiguada por el efecto de la actualización y acumulación en 30 años) en la Fig.1.

Las referencias [1] y [2] recopilan el conjunto de técnicas actualmente en uso para la lucha contra la maldición de Bellman. La mayoría de estas técnicas hacen uso de la resolución de la ec.8 para obtener en forma iterativa la funciones CF que aproximan a la óptima CF_{FO} . De la lectura de ambos libros surge claramente que no existe un único algoritmo para lograr el objetivo de obtener la Política Óptima, que todas las soluciones planteadas son aproximadas y que en los sistemas reales no tienen garantizada la convergencia. Lo que también queda claro es que para la mayoría de las soluciones, es necesaria una representación la información de CF que es relevante en la solución de la ec.8 y es este precisamente el objetivo del presente trabajo.

La referencia [3], dedicada al Aprendizaje por Refuerzo, hace un repaso de las técnicas en uso intentando dar una nomenclatura ordenada que resulte útil en las diferentes áreas de aplicación especialmente en la de Aprendizaje por Refuerzo como su nombre lo sugiere. En esta referencia, al igual que en [1] y [2] (y en la literatura clásica de control de sistemas dinámicos) se plantea, además de la formulación de las ec.7 y 8 para la resolución de La Política Óptima, otra formulación en que en lugar de considerar la función de Valor del Estado $CF(X,k)$ se considera la función de Valor del (Estado, Acción) $Q(X,u,k)$ con el mismo sentido de representar el costo esperado de la operación

futura a partir del estado X para la acción u . Esta formulación resulta necesaria cuando se desconoce la ecuación de evolución de estado, lo que no es nuestro caso. Nosotros conocemos el modelo del sistema y no necesitamos aprenderlo mediante ensayos. Nuestro sistema objetivo es el sistema de generación de energía eléctrica de un país y obviamente no permite realizar ensayos como para aprender por ensayo y error el modelo del sistema. El objetivo de este trabajo no son los algoritmos de resolución del problema de operación óptima, este trabajo se limita a la representación de la información útil en la función de costo futuro para ese propósito. No obstante lo anterior, se consideró necesario aclarar porqué nos dedicaremos solo a representar la función de valor del estado y no incluimos en el trabajo la función de valor del estado-acción.

2.2. Revisión de técnicas de aproximación de funciones.

Nuestro objetivo implementar una Plataforma de Aproximación Genérica (PAG) con la cual lograr una aproximación flexible de CF y que nos permita ir mejorando dicha aproximación en base a iteraciones en las que se recibe nueva información e incorporar el conocimiento experto (humano) de forma de colaborar en la lucha contra la Maldición de Bellman.

Sería imposible rastrear en la historia el primer uso de un Modelo, como conjunto de ecuaciones que permiten emular el comportamiento de un sistema, así como sería imposible enumerar todas las técnicas que se han desarrollado para la aproximación de Funciones. De ese vasto conjunto, solo enumeraremos aquellas, de las que tuvimos oportunidad de conocer, que consideramos valen la pena sean incluidas en la PAG.

Simulación y Trayectorias. Nuestro propósito es ir mejorando la representación de la Política Óptima de operación en base a ir mejorando la representación de $CF(X, k)$ que la contiene. El método consiste en, a partir de un propuesta inicial $CF_0(X, k)$, ir obteniendo sucesiva aproximaciones $CF_1(X, k), CF_2(X, k) \dots CF_i(X, k)$. La aproximación $CF_{i+1}(X, k)$, se realiza incorporando la información recibida al realizar simulaciones de la operación del sistema con la política de operación contenida en $CF_i(X, k)$.

Para realizar las simulaciones, dada $CF_i(X, k)$, se utiliza la Política de Operación implícita en la resolución de la ec.8. Se parte de un conjunto de estados iniciales y de realizaciones de los procesos estocásticos y al realizar la simulación, se obtienen trayectorias del estado del sistema con información del costo de etapa y del costo futuro en el estado al final de la etapa (sumandos del lado derecho de la ec.8). Entonces, la información resultante de la simulación, está formada por Trayectorias: $T_1, T_2 \dots T_j \dots T_{n_x \times n_r}$ donde n_x y n_r son la cantidad de estados iniciales y la cantidad de realizaciones de los proceso estocásticos simulados respectivamente. Cada trayectoria, está formada por una serie de “puntos de información” como se muestra en la ec.10.

$$T_j^i = \{(X_k^{ij}, ce_k^{ij}, cf_k^{ij})\} \quad \text{ec.(10) Trayectoria}$$

Donde X_k^{ij} , ce_k^{ij} y cf_k^{ij} corresponden, para la trayectoria j de la iteración de cálculo i , al estado del sistema al inicio de la etapa k , el costo incurrido en la etapa k y el costo

futuro del estado de llegada al final de la etapa k con la estimación $CF_i(X, k)$ utilizada para realizar la simulación.

Las ternas de datos $P_k^{ij} = (X_k^{ij}, ce_k^{ij}, cf_k^{ij})$ es lo que llamaremos un punto de dato (o de información). Todo punto de dato necesariamente pertenece a una trayectoria j de una iteración de cálculo i . Cuando por el contexto sea claro la iteración o la trayectoria a la que se refiere el punto se omitirán el/los superíndices para agilizar la notación.

La mejora de la aproximación en base a iteraciones, puede verse como un aprendizaje en el que un Actor (o Agente) va aprendiendo (mejorando la aproximación) mediante sucesivos ensayos de los que recopila información. Con este enfoque, el problema de obtención de una política óptima de operación de un sistema dinámico ha sido incorporado en disciplinas más amplias (y sin duda más de moda) que la de control de sistemas como ser: Aprendizaje a partir de datos, Aprendizaje por Refuerzo, Aprendizaje Automático, Aprendizaje de Máquinas, Inteligencia Artificial, etc. Estas disciplinas tienen aplicaciones en campos muy diversos como el tratamiento de imágenes, reconocimiento de patrones o análisis lógico de expresiones y representaciones con fines de inferencias estadística y sin duda las herramientas desarrolladas en esos campos resultan valiosas para nuestro propósito.

En particular la referencia [5] realiza una revisión de las técnicas de aproximación de funciones con el enfoque de analizar su aporte en el **aprendizaje a partir de datos** y en la flexibilidad para **incorporación del conocimiento experto**. De la lectura de esta referencia, se rescata la importancia de las Redes Neuronales y de las Máquinas de Soporte Vectorial en lo que hacen a la implementación de estructuras que puedan aprender a partir de datos y de los Modelos de Lógica Difusa como modelos flexibles para incorporación del conocimiento humano. La referencia [6] desarrolla en profundidad la noción de Kernels como noción de similitud entre puntos de información. Está claro que el primer aporte experto a nuestro problema, es que existen zonas del espacio de estados en las que la Política de Operación debe ser continua. A modo de ejemplo, no puede ser que por variar 20 cm el nivel de un lago, las decisiones de operación deban cambiar drásticamente para las cotas que se encuentren en el rango de operación normal. Y que llegando al fondo del lago o a la cota en la que sea necesario abrir el vertedero por seguridad de la presa esa variación pueda resultar en cambios significativos. Por tanto es de esperar que existan zonas en que, con cierta medida de similitud entre estados, la Política de Operación sea igual o con variación suave entre Estados Similares. Entre las referencias [4] y [5] se encuentran las bases de las técnicas utilizadas en este trabajo, son técnicas conocidas y bien establecidas. El esfuerzo del presente trabajo está en cómo aplicar esas técnicas incorporando el conocimiento experto del sistema tanto en el andamio esperado de dicha función respecto de las variables de estado como en el tiempo.

2.2.1 Transformaciones y agregaciones.

Las prácticas comunes utilizadas en la lucha contra la Maldición de Bellman, incluyen transformación del espacio de estado $Z = R(X, k)$ en un espacio que permita una explicación más sencilla de la función $CF(Z, k)$. Solo a modo de ejemplo, si se conoce una transformación

$Z=R(X, k)$ que permite expresar el costo futuro como $CF(Z, k)=a_k^T Z+y(k)$, para la calibración del modelo $CF(X, k)$ para su uso en la ec.8 será suficiente con estimar los vectores a_k en base a la información recolectada usando los métodos conocidos de regresión lineal.

Espacio de Características. (features)

Se podría decir que la transformación $Z=R(X, k)$ transforma la posición del sistema de espacio de estado a un espacio de características (cada componente del vector Z) que logran explicar el costo futuro con una relación más sencilla que si se utilizará directamente el vector de estado X . Sin lugar a dudas, seleccionar adecuadamente el espacio de características puede ser clave para la obtención de un modelo aproximado y es una parte del modelado que sin lugar a dudas permite la incorporación del conocimiento experto.

Agregaciones espaciales.

Si la transformación $Z=R(X, k)$ es tal que la dimensión de Z es inferior a la de X se está ante lo que se conoce como una técnica de Agregación. A modo de ejemplo, en un sistema con dos centrales hidroeléctricas, el vector de estado puede considerarse formado por los volúmenes embalsados en los respectivos embalses teniendo así un espacio de estado de dimensión 2 (dos). Si las restricciones de potencia del sistema no juegan un papel importante, seguramente lo que importa a los efectos de la operación es cuánta energía hay almacenada en el conjunto de ambos embalses y por tanto se podrá resumir dicha energía en una ecuación del tipo: $Z=ce_1(x_1)x_1+ce_2(x_2)x_2$ siendo x_1 y x_2 los volúmenes embalsados y $ce_1(x_1)$ y $ce_2(x_2)$ los respectivos coeficientes energéticos de las centrales calculados para las alturas correspondientes al nivel embalsado. En este caso la Agregación resume de un espacio de 2 dimensiones a un espacio de 1 dimensión lo que permite un mejor uso de la información recolectada para aproximar $CF(Z, k)$ y por tanto es de esperar que la misma se más sencilla de aproximar. Se observa que en la aplicación de la aproximación en las ec.7 y 8 no es necesario invertir la transformación $Z=R(X, k)$ y lo único relevante es si la aproximación lograda $CF(Z, k)$ contiene la información relevante.

Agregaciones temporales.

Una técnica que es usada en las resoluciones actuales del problema de obtención de una política óptima es la utilización de diferentes pasos de tiempo. En los sistemas de energía eléctrica, se suelen usar pasos de tiempo semanales o mensuales para horizontes temporales de más de 3 años, pasos de tiempo diarios para horizontes de meses o años y pasos de tiempo horario para horizontes de días o semanas. Los Organismos Encargados del Despacho (OED) utilizan esta técnica generalmente. Se utiliza lo que se conoce como “encadenamiento de las políticas de operación”. Primero se resuelve el problema de Largo Plazo, luego el de Mediano Plazo tomando como inicialización el resultado del Largo Plazo y luego el de Corto Plazo inicializando el problema con el resultado del Mediano Plazo. El uso de diferentes pasos de tiempo, generalmente implica variaciones en la agregación espacial también. A modo de ejemplo, en una simulación de paso horario, la producción actual del conjunto de parques

eólicos es información útil para la siguiente hora y por tanto forma parte de la información de estado del sistema, pero para una simulación de paso mensual, difícilmente la generación actual del conjunto de parques eólicos resulte relevante.

2.2.2 Estructuras de modelado.

De la literatura consultada, por su posible aporte al modelado de $CF(Z, k)$ se destacan las siguientes técnicas:

- Mínimos Cuadrados (LSM)
- Neural Networks (NN).
- Radial Basis Functions (RBF).
- Fuzzy Logic Models. (FLM).
- Support Vector Machines (SVM).
- Mixture Density Networks (MDN).
- Espacios Grillados.
- Kernels. Aproximaciones por cercanía.
- Redes Neuronales Recurrentes.

A continuación se hace un muy breve repaso de las características de las mismas y cómo pueden proyectar en nuestro objetivo. En todos los casos, aunque no se mencione explícitamente, la estructura puede estar aplicada sobre las entradas X o sobre un espacio de características $Z = R(X)$

En cuanto a la representación de la información, podemos identificar dos dimensiones que resultarán de interés. La primera es la capacidad de representar (o de habilitar) el aprendizaje y la segunda la posibilidad de incorporar conocimiento de los expertos (humanos).

Las NN, SVM y MDN son estructuras que proyectan preferentemente en la dimensión de la representación del "aprendizaje" (por ajuste de sus parámetros) y los modelos del tipo FLM sobre la dimensión de representar el conocimiento humano en los algoritmos. Si bien estos enfoques de modelado tienen preferencias sobre las dimensiones de interés (aprendizaje y conocimiento previo), estas no son excluyentes y la frontera entre las técnicas se solapan.

La referencia [4] está dedicada al "aprendizaje a partir de datos" y a la "asimilación del conocimiento humano" en el contexto de una nueva disciplina llamada "soft computing" que en contraposición de "hard computing". La idea detrás de "soft computing" es tener algoritmos más flexibles haciendo hincapié más en el comportamiento del sistema aunque sea a costa de sacrificar la precisión del modelo.

Nuestro foco está en la aproximación a partir de datos de la función CF, tratando de captar aquello que importa e ignorando lo que no importa para lo cual será importante detectar aquellos aspectos relevantes de la operación del sistema así como tener la posibilidad de incorporar el conocimiento de los expertos en dicha aproximación.

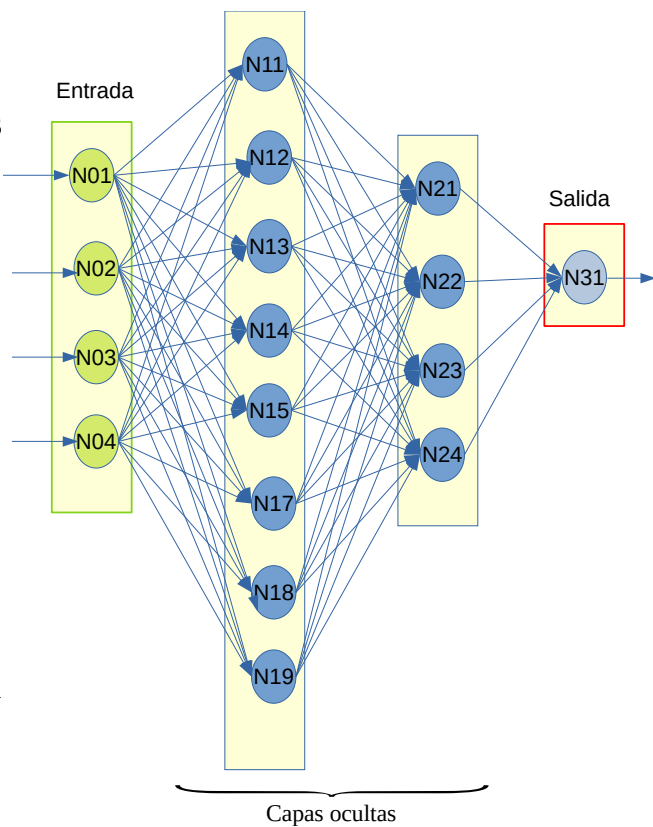
En todos los casos, se trata de dado el conjunto de datos $\{X_j, CF_j\}$ buscar explicar la dependencia de $CF = M(X, \rho)$, donde M representa el tipo de modelo y ρ es el vector de parámetros.

Mínimos Cuadrados. (LSM).

Dado el conjunto de datos $\{X_j, CF_j\}$ el modelo **LSM** busca explicar la dependencia de $M(X, \rho, \rho_0) = \rho^T X + \rho_0$.

La solución a este problema es clásica y abunda la literatura respecto del mismo. Aquí solo recordaremos que siendo el error cuadrático el valor a minimizar en la calibración de los parámetros, se tiene que $\rho_0 = \langle CF_j \rangle$ y $\rho = \langle X_j X_j^T \rangle^{-1} \langle X_j CF_j^T \rangle$. En nuestro caso, que la información puede ser

escasa (pocos puntos muchas dimensiones) la matriz $\langle X_j X_j^T \rangle$ puede resultar no invertible o mal condicionada (puede resultar semi-definida positiva, o con algún cero o muy cercano a cero). Es práctica común utilizar algún tipo de regularización como son la de Ridge o Lasso (ver sec.2.3). En particular la de Ridge es de implementación inmediata pues consiste en simplemente adicionar el coeficiente de regularización multiplicado por la identidad en la diagonal de $\langle X_j X_j^T \rangle$. Para la regularización de Lasso si bien es un poco más complejo de implementarlo, está resuelto ver por ej. la referencia [4]. La LSM es ampliamente usada en todas las ramas de la ciencia y de la ingeniería y no podía faltar en un trabajo sobre aproximación de funciones, pero obviamente, para nuestro caso en que la función es no-lineal resulta limitada y por esa razón no daremos más profundidad sobre este tipo de modelo, sino que veremos que el mismo queda comprendido dentro de lo que son las Redes



Neuronales. Un aproximador LSM no es otra cosa que una NN de una sola Neurona con función de activación la identidad.

Figura 3: Tipo de Red Neuronal simple.

Redes Neuronales (NN).

En este trabajo, nos limitaremos al uso de **Redes Neuronales** simples, formadas por un conjunto de capas cada una con un número dado de Neuronas como se muestra en la Fig.3. Las capas están encadenadas de forma tal que las salidas de una capa alimentan la siguiente. La primer capa (izquierda) es la que llamaremos Capa de Entrada y es la que recibe la información Z (vector de características) y la última capa Capa de Salida es la que expone en la salida la estimación de $CF(Z, k)$. Este tipo de arquitectura es lo que se conoce como Perceptrón Multi Capa, o red Feed Forward. Las Capa de entrada se suele utilizar simplemente para acondicionar las entradas quitando el valor esperado y aplicando un factor de escala. Las Neuronas de las capas internas (ocultas) pueden ser de varios tipos. Si bien podríamos llamar Neurona a cualquier mapeo de un conjunto de entradas en una salida, el tipo clásico de Neurona está formado en una etapa de entrada que realiza una combinación lineal de las entradas seguida de una función de activación como se muestra en la Fig.4. Los parámetros ρ_i son los parámetros del modelo y son el objeto de la calibración. Las funciones de activación más usadas son:

- Lineal, $s = u$
- Tangente Hiperbólica, $s = (e^u - e^{-u}) / (e^u + e^{-u})$
- Sigmoide, $s = 1 / (1 + e^{-u})$
- Logica, $s = u > 0$
- ReLu, $s = (u > 0) u$

Las siguientes dos no son de uso frecuente, y no usan la combinación lineal de las entradas, pero resultan necesarias para nuestro propósito:

- Max, $s = \max(\rho_i z_i)$, la salida es el máximo de las entradas escaladas por los ρ_i .
- PonderSoftmax, $s = \frac{\sum_i e^{\rho_i z_i}}{\sum_j e^{\rho_j z_j}}$, la salida es una combinación lineal de las entradas, pero los pesos varían con las entradas, dando mayor peso a las entradas con mayor $\rho_i z_i$.

Se puede demostrar que la estructura de la Fig.3, con una sola capa interna, es capaz de aproximar con la precisión deseada cualquier función. En la práctica se suelen utilizar estructura de 1 y dos capas internas y rara vez más capas.

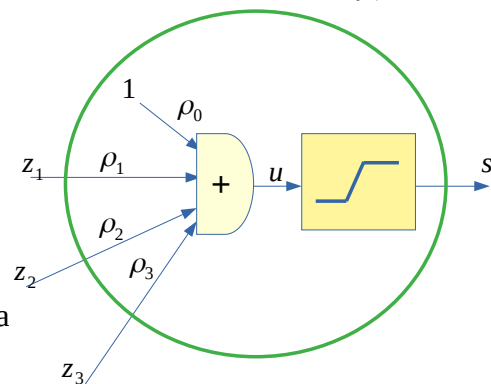


Figura 4: Esquema de Neurona Clásica.

La calibración de los parámetros ρ_i se realiza intentando minimizar el error cuadrático de la salida. El método conocido como Back Propagation, permite calcular el gradiente del error respecto de

los parámetros y es utilizado para implementar algoritmos de minimización. El problema es no lineal y presenta generalmente mínimos locales. El éxito en realizar la calibración depende por tanto de la inicialización de los algoritmos y de utilizar técnicas como las de la aplicación de un "momento" en la aplicación de los movimientos. De los tipos de neuronas antes enumerados, Lineal, Tangente Hiperbólica, Sigmoide y PonderSoftmax son diferenciables y por tanto permiten una implementación sencilla del método de Back Propagation. Las funciones Lógica, ReLu y Max presentan discontinuidades en las derivadas que dificultan su implementación.

Dado un conjunto de muestras $\{(Z_j, CF_j)\}$ y el conjunto de los parámetros de la red ρ , para cada muestra (Z_j, CF_j) sea $Y_i = M(X_i, \rho)$ la salida de la NN. El error sobre esa muestra es $e_i = (Y_i - CF_i)$ y el error cuadrático medio sobre el conjunto de muestras se calcula como se muestra en la ec.11 y es el objetivo de calibración de los parámetros ρ .

$$J = \frac{1}{N} \sum_{i=1}^N e_i^2 \quad \text{ec.(11) Error cuadrático medio de la NN}$$

La calibración se realiza buscando el vector ρ que minimiza J . El gradiente de J respecto de los ρ_i se calcula para cada muestra utilizando el método Back Propagation. El gradiente sobre el conjunto de muestras (usando la ec.11) es la suma de los gradientes calculados para cada muestra. Hay algoritmos que utilizan el gradiente sobre el conjunto de muestras para aplicar un "paso" en la búsqueda de un mejor ρ y otros utilizan el gradiente de cada muestra para dar pasos (stochastic gradient). En cualquiera de los dos casos, se puede utilizar el gradiente calculado para dar el paso o aplicar un "momento" que implica considerar una combinación entre el vector utilizado para dar el paso anterior y el nuevo gradiente. La ec.12 muestra la secuencia de obtención del vector v_{j+1} para ser utilizado en el cálculo del nuevo ρ . El coeficiente $\alpha_j \in [0, 1]$ es el factor de "momento". En el caso de $\alpha_j = 0$ (sin momento) se utilizará directamente el gradiente.

$$v_{j+1} = \alpha_j v_j + (1 - \alpha_j) \frac{\partial}{\partial \rho} J \quad \text{ec.(12) Gradiente con momento}$$

Para actualizar ρ se realiza en base a la ec.13 siendo λ_j el paso de actualización.

$$\rho_{j+1} = \rho_j - \lambda_j v_{j+1} \quad \text{ec.(13) Actualización de } \rho$$

Las variaciones principales entre los algoritmos de calibración están en cómo modificar el momento y el paso en durante la evolución del algoritmo y en como imponer un juego inicial de parámetros.

Para que estos algoritmos que se basan en el gradiente sean aplicables, es necesario poder calcular dicho gradiente para lo cual las funciones de activación de las neuronas tienen que ser diferenciables.

La aproximación de mínimos cuadrados **LSM** se puede representar como una NN de una sola neurona con función de activación lineal y por tanto puede ser modelada con la estructura de la Fig.3.

La calibración de la neurona en este caso se puede realizar por los algoritmos clásicos de resolución de aproximaciones de mínimos cuadrados.

Radial Basis Functions (RBF).

Las **RBF**, pueden considerarse dentro de la estructura de la Fig. 3, con una sola capa oculta y sustituyendo las Neuronas clásicas (combinación lineal de las entradas y función de activación) de la primer capa, por funciones que tengan su máximo en un "centro" asociado a cada neurona y decaigan su valor de salida en la medida en que el vector de entradas se aleja del centro. Un ejemplo de este tipo de funciones puede ser el que se muestra la ec.14 donde z_0 sería "el centro" asociado a la Neurona y la matriz A una matriz definida positiva. La neurona de salida sería simplemente la suma ponderadas de las salidas de la capa RBF.

$$s = e^{-(z-z_0)^T A (z-z_0)} \quad \text{ec.(14) Ejemplo de RBF.}$$

Los parámetros a calibrar en este caso son los centros de las neuronas y las matrices A y los parámetros de ponderación de la neurona de salida.

En teoría las RBF, con una suficiente cantidad de Neuronas pueden aproximar cualquier función (al igual que las NN), pero pensando en nuestro objetivo de que con pocos datos se quieren estructuras que capten el modelo las RBF presentan desventajas frente a las NN comunes pues nuestra función

$CF(Z)$ suele presentar monotonía en las variables y por tanto no presenta centros o estructuras radiales.

Fuzzy Logic Models. (FLM).

Los FLM facilitan la incorporación del conocimiento humano dentro de la estructura del modelo. Al igual que las RBFs puede considerarse dentro de la estructura de la Fig.3 con una capa interna compuesta por Neuronas que calculan el Grado de Pertenencia de la entrada a la red y una neurona de salida que realiza la combinación lineal de las salidas. En definitiva, el conocimiento humano es incorporado al seleccionar y parametrizar las funciones que determinan el Grado de Pertenencia. No visualizamos que este tipo de estructura pueda utilizarse eficientemente en nuestro caso, dado que disponemos de la posibilidad de simular y evaluar el costo de operación dado un modelo. El conocimiento humano incorporable en nuestro caso son cosas como "el sistema se expande naturalmente por el crecimiento vegetativo de la Demanda", "tanto la Demanda como la hidraulicidad imponen una estacionalidad anual importante", "La Demanda, la generación solar y eólica imponen un ciclo diario".

Una estructura FLM podría tener aplicación si se quisiera por ejemplo separar el tratamiento de un estado especial del sistema. Por ejemplo EMERGENCIA por sequía extrema.

Support Vector Machines (SVM).

Las **SVM** son estructuras más especializadas en el problema de clasificación en el que dada una muestra el modelo tiene que clasificarla y decidir si pertenece o no a un conjunto. En su aplicación al modelado de una función continua multi-variable (lo que se conoce como regresión-SVM) la estructura

SVM es la misma que la de regresión lineal sobre un conjunto de propiedades cambiando el objetivo de minimizar el error cuadrático medio por el de minimizar el valor absoluto del error que supera un umbral dado como se muestra en la ec.15.

$$e_i = \max(|CF_i - Y_i| - \varepsilon, 0) \quad \text{ec.(15) Error a minimizar en la estructura SVM}$$

Este cambio en la función de error, puede tener cierta ventaja en nuestro problema dado el nivel de "ruido" que se tiene asociado a que $CF(Z, k)$ obtenida por simulaciones como se mostró en el ej. de la Fig.1. Es así que la tolerancia ε en la ec.15 puede jugar a favor de evitar que los parámetros de la red se especialicen en ajustar al error producido por ese ruido (lo que se conoce como overfitting).

El cambio en la función de error ec.e15 respecto del error cuadrático dificulta la implementación directa por la dificultad del cálculo del gradiente. Para el caso de regresión lineal, hay soluciones en base a resolver el dual de un problema con restricciones agregada para representar la tolerancia en el error. No se profundiza en esta solución pues a los efectos de nuestro problema no es suficiente una regresión lineal.

Mixture Density Networks (MDN).

Los valores de $CF(X)$ provenientes de simulaciones presentan una distribución. La Fig.1 es un ejemplo de dicha distribución para un estado de partida dado X correspondiente a 1000 posibles operaciones futuras del sistema uruguayo durante 30 años. Como ya se mencionó, la dispersión de valores es importante cuando es comparada con los costos de etapa (ver Fig.2) y por tanto fácilmente hay que cuidar que el ajuste de los parámetros de los modelos sea tal que filtre el ruido proveniente del cálculo del valor esperado $CF(X)$ en base a un número finito de puntos de su distribución.

El trabajo de Bishop [7] es un enfoque que consideramos excelente en el sentido de reconocer que el modelo intenta dar una distribución de la salida condicionada al conocimiento de la entrada más que a dar un valor único. Con este planteo, la función a maximizar durante la calibración del modelo es la verosimilitud de las observaciones. Recordar que la minimización del error cuadrático es óptima en el sentido de maximizar la verosimilitud cuando la salida de un modelo está contaminada con ruido blanco gaussiano aditivo y por tanto es más limitada su aplicación que el planteo de las MDN.

Dado un conjunto de funciones de densidad de probabilidad $p_j(y/Z)$, observar que cualquier combinación lineal convexa es también positiva y de integral 1 (uno) por lo cual también es una función de densidad de probabilidad.

En la estructura planteada por Bishop, hay una NN, que en base a las entradas calcula en sus salidas los parámetros que determinan las características de un conjunto de distribuciones $p_j(y/Z)$ dado y los factores a aplicar para realizar la combinación lineal convexa de dichas distribuciones.

En la referencia [7] se desarrolla al detalle el modelo considerando distribuciones gaussianas para el conjunto $p_j(y/Z)$ (indicando que dichas funciones pueden aproximar con cualquier grado de exactitud cualquier distribución).

Este modelo, se implementó dentro del conjunto de herramientas desarrolladas en el presente trabajo para disponer de la posibilidad de experimentar con el mismo. En particular lo aplicamos para la estimación de la generación de parques eólicos a partir del conocimiento de la velocidad de viento en el anemómetro o del conocimiento del pronóstico de velocidad de viento obteniendo mejoras sustanciales respecto del comportamiento del modelo previamente disponible del tipo NN simple.

Espacios Grillados.

En la versión disponible de SimSEE en 2019, la función $CF(X, k)$ es representada por una discretización del espacio de estado para cada paso de tiempo k . La discretización utilizada se basa en considerar el producto cartesiano de las discretizaciones de cada una de las variables de estado en el rango especificado creando así un Grilla de cálculo. Es este grillado que produce la explosión combinatoria al aumentar la dimensión del espacio de estado. Por esta razón, no consideramos en este trabajo estrategias de aproximación de $CF(X, k)$ que signifiquen la confección de una grilla de cálculo del espacio de estado.

Kernels. Aproximación por cercanía.

En el modelado del comportamiento de un sistema, se asume generalmente que a estados “similares” las soluciones deben ser similares. Esta especie de continuidad, necesita de una noción de similitud entre puntos del espacio de estado. Esta similitud generalmente se expresa a partir de alguna noción de distancia o en forma más genérica a partir de lo que se conoce como un Kernel. Entonces, en el contexto de este trabajo entenderemos por un Kernel a un mapeo que asigna a cualquier par de puntos del espacio de estado, un número real positivo que es un indicador de similitud entre dichos puntos. Teniendo en cuenta que algunas dimensiones del espacio de estado pueden no ser números reales (algunas dimensiones son booleanas o números enteros) la adopción de un Kernel resulta necesaria para expresar en forma medible la idea de cercanía entre los estados. La noción de Kernel claramente resulta útil en otras disciplinas como la interpretación del lenguaje o reconocimiento de patrones donde la información de entrada no es un vector de reales como puede ser el volumen embalsado en las represas del sistema eléctrico.

Si todas las dimensiones del espacio de estado son reales, una medida de cercanía puede crearse a partir de la distancia euclídea entre los puntos. Por ej.: $K(X_1, X_2) = e^{-\|X_1 - X_2\|}$.

La misma noción de cercanía puede aplicarse sobre el espacio de características $Z = R(X, k)$ y generalizar la norma utilizada por ejemplo con una expresión del tipo:

$K(Z_1, Z_2) = e^{-(Z_1 - Z_2)^T A (Z_1 - Z_2)}$ dónde la matriz A debe ser definida positiva y puede usarse para darle más importancia a algunas características que a otras en la medida de similitud.

Las RBF mencionadas en la sección 2.2.2 son un ejemplo de uso de Kernels en una estructura de aproximación.

Otra forma de uso de Kernels, es definir una estructura de aproximación en base directamente a las muestras X_j, CF_j (en cada paso de tiempo) realizando una combinación ponderada en base a la similitud indicada por el Kernel de funciones específicas centradas en cada punto (por ejemplo una gaussiana centrada en cada punto X_j). Esta aplicación puede considerarse como la descrita para las RBF en la sección 2.2.2 agregando un neurona RBF para cada muestra con el centro en cada muestra. Este tipo de aproximaciones es útil cuando el conjunto de puntos de aproximación es grande y entonces se puede considerar que la aproximación por "los cercanos" es suficiente. En la Lucha contra la Maldición de Bellman, la situación es otra. Se dispone de poca información en base a "puntos observados" y debe ser complementada con conocimiento. Por este motivo, no incorporaremos explícitamente aproximaciones basadas en Kernels sobre el conjunto de muestras en nuestro desarrollo.

Redes Neuronales Recurrentes. (RNN)

Las Redes Neuronales Recurrentes, son estructuras como la mostrada en la Fig.3 pero en la que las entrada de las neuronas de una capa, además de contener el conjunto de salidas de las capas anterior, puede considerar las salidas de la misma capa o de capas posteriores. En definitiva se trata de crear una estructura recursiva en que las salidas de una capa pueden alimentar a las neuronas de la misma capa o anteriores. Si nuestro problema fuera de estructura constante, es posible que una RNN pudiera captar las periodicidades del sistema. Por ejemplo, en una simulación de paso horario, una RNN de 24 capas podría captar el ciclo diario realimentando la salida de la última capa a la primera. Nuestro sistema no es de estructura fija, presenta pseudo ciclos diarios y anuales además de un comportamiento expansivo. Por esta razón, en primera instancia no se implementaron estructuras del tipo RNN. Pero no se descarta que en un futuro se pueda probar este tipo de estructura para su aplicación en períodos cortos (por ejemplo 15 días) con paso horario encadenando el modelo con otro de más largo plazo.

2.3. Regularización.

En la calibración de un modelo el exceso de parámetros puede llevar a un sobre-ajuste (overfitting) perdiendo el modelo las propiedades de generalización. La idea es buscar métodos de "apagar" los parámetros que no agregan explicaciones al modelo. En el caso de una regresión lineal, "apagar" los parámetros consiste en llevar a CERO (cero) aquellos parámetros asociados a variables que no inciden en la salida. Para lograr este objetivo, se agrega a la función objetivo de la calibración ec.11 un término de regularización que penaliza el uso de parámetros que no reducen el error de la ec.11 en forma apreciable.

La ec.16 muestra lo que se conoce como regularización de Ridge (cuando es aplicada a la regresión lineal). El parámetro $\lambda \geq 0$ de regularización ponderar el peso de la regularización respecto de la minimización del error del modelo.

$$J = \frac{1}{N} \sum_{i=1}^N e_i^2 + \lambda \sum_j \rho_j^2 \quad (16) \text{ .Regularización de Ridge.}$$

La regularización de Ridge tiene la ventaja de ser diferenciable respecto de los parámetros y por tanto fácilmente implementable. Como desventaja, al tener los parámetros al cuadrado el peso de la regularización se extingue al reducir los parámetros y por tanto no se anulan por completo.

Como forma de evitar ese inconveniente se puede usar la regularización de Lasso que consiste en considerar el valor absoluto de los parámetros en lugar del cuadrado logrando así que no se reduzca su peso en la función objetivo en la medida en que tienden a cero. Como inconveniente, la función objetivo incluye el valor absoluto que dificulta el cálculo del gradiente.

$$J = \frac{1}{N} \sum_{i=1}^N e_i^2 + \lambda \sum_j |\rho_j| \quad \text{ec.(17) Regularización de Lasso}$$

Las ecuaciones 16 y 17 son directamente aplicables para el caso de la regresión lineal en que los valores de los parámetros nulos implican ignorar la dependencia de la salida con la entrada.

Lo mismo ocurre para las Neuronas con los pesos que ponderan las entradas de la Neurona, pero el peso asociado al offset (bias) tiene el propósito de "centrar" la neurona y depende de la función de activación si el valor de salida nulo se produce con las combinación lineal nula o no.

2.4. Alternativas para representar la información útil para el despacho óptimo de un paso de tiempo.

La información útil de la Política de Operación se encuentra en las derivadas del Costo Futuro respecto a los estados y no en el valor absoluto de dicha función. Por esta razón, lo importante es representar adecuadamente la variación δCF asociada a un movimiento del estado δX .

En esta sección se documentan las ideas manejadas para representar dicha información.

En un mismo paso de tiempo se dispone de un conjunto posiciones del estado $\{X_i\}$ y de estimaciones del costo futuro en dicho estado $\{y_i\}$

2.4.1 Aproximación a la función de Valor.

Si el objetivo es aproximar un modelo $y = CF(X, \theta)$ ajustando el conjunto de parámetros θ , problema se podría plantear como:

$$\min_{\theta} J_v(\theta) \quad \text{ec.(18) Aproximación a la función de Valor.}$$

Dónde: $J_v(\theta) = \sum_i e_i^2$, $e_i = CF(X_i, \theta) - y_i$

En el óptimo se debe cumplir:

$$\frac{\partial J_v(\theta)}{\partial \theta} = \sum_i 2e_i \frac{\partial e_i}{\partial \theta} \quad \text{ec.(19) .}$$

2.4.2 Aproximación a las diferencias espaciales desde uno de los estados visitados.

Si el objetivo es ajustar las diferencias espaciales, esto es cómo se ven las variaciones de $CF(X_j)$ desde el estado X_i

$$\min_{\theta} J_{di}(\theta)$$

$$J_{di}(\theta) = \sum_j ((CF(X_j, \theta) - CF(X_i, \theta)) - (y_j - y_i))^2 = \sum_j (e_j - e_i)^2$$

$$J_{di}(\theta) = \sum_j (e_j^2 - 2e_j e_i + e_i^2) = J_v(\theta) - e_i \sum_j (e_j - e_i)$$

2.4.3 Aproximación a las diferencias espaciales desde cualquiera de los estados visitados.

Si no se quiere dar un tratamiento especial la posición del estado X_i la función objetivo sería la suma de la anterior para cada uno de los puntos X_i posibles.

$$\min_{\theta} J_d(\theta)$$

ec.(20) Aproximación a las variaciones espaciales de CF.

Dónde: $J_d(\theta) = \sum_{ij} ((CF(X_j, \theta) - CF(X_i, \theta)) - (y_j - y_i))^2$

que puede escribirse en función del error de la función de valor como:

$$J_d(\theta) = \sum_{ij} (e_i - e_j)^2$$

$$\frac{\partial J_d(\theta)}{\partial \theta} = \sum_{ij} 2(e_i - e_j) \left(\frac{\partial e_i}{\partial \theta} - \frac{\partial e_j}{\partial \theta} \right) = \sum_{ij} 2 \left((e_i \frac{\partial e_i}{\partial \theta} - e_j \frac{\partial e_i}{\partial \theta}) - (e_i \frac{\partial e_j}{\partial \theta} - e_j \frac{\partial e_j}{\partial \theta}) \right)$$

$$\frac{\partial J_d(\theta)}{\partial \theta} = 4N \sum_i e_i \frac{\partial e_i}{\partial \theta} - 4 \sum_{ij} e_j \frac{\partial e_i}{\partial \theta} = 2N \frac{\partial J_v(\theta)}{\partial \theta} - 4 \sum_{ij} e_j \frac{\partial e_i}{\partial \theta}$$

Observar que la condición de optimalidad del problema de aproximación a la función de valor (ec.18) implica $\frac{\partial J_v(\theta)}{\partial \theta} = 0$ y por tanto $\frac{\partial J_d(\theta)}{\partial \theta} = -4 \sum_{ij} e_j \frac{\partial e_i}{\partial \theta}$ y no necesariamente es nulo. Por lo tanto los problemas no son equivalentes.

2.4.4 Aproximación a la función de Valor con nivel libre.

Si el objetivo es aproximar un modelo $y = CF(X, \theta) + \alpha$ ajustando el conjunto de parámetros θ y el parámetro α , problema se podría plantear como:

$$\min_{\theta, \alpha} J_v(\theta, \alpha) \quad \text{ec.(21) Aproximación a la función de Valor.}$$

Dónde: $J_v(\theta, \alpha) = \sum_i e_i^2$, $e_i = CF(X_i, \theta) + \alpha - y_i$

En el óptimo se debe cumplir:

$$\frac{\partial J_v(\theta, \alpha)}{\partial \theta} = \sum_i 2e_i \frac{\partial e_i}{\partial \theta} \quad \text{ec.(22) .}$$

y

$$\frac{\partial J_v(\theta, \alpha)}{\partial \alpha} = \sum_i 2e_i \frac{\partial e_i}{\partial \alpha} = \sum_i 2e_i \quad \text{ec.(23) .}$$

Las condiciones de optimalidad sobre la ec.23 implican:

$$\sum_i e_i = 0 \quad \text{ec.(24)}$$

y sobre la ec.22 implican:

$$\sum_i e_i \frac{\partial e_i}{\partial \theta} = \sum_i (CF(X_i, \theta) + \alpha - y_i) \frac{\partial e_i}{\partial \theta} = \sum_i (CF(X_i, \theta) - y_i) \frac{\partial e_i}{\partial \theta} + \sum_i \frac{\partial e_i}{\partial \theta} \alpha \quad \text{ec.(25)}$$

La ec.24 implica $\sum_i \frac{\partial e_i}{\partial \theta} = 0$

De la definición del error e_i se puede escribir $\frac{\partial e_i}{\partial \theta} = \frac{\partial (CF(X_i, \theta) - y_i)}{\partial \theta}$

Y sustituyendo ambas expresiones en la ec.25 se obtiene

$$\sum_i e_i \frac{\partial e_i}{\partial \theta} = \sum_i (CF(X_i, \theta) - y_i) \frac{\partial (CF(X_i, \theta) - y_i)}{\partial \theta} = 0 \quad \text{ec.(26)}$$

Observar que la condición de optimalidad expresada por la ec.26 es independiente de α por lo que se puede resolver el problema 21 para un valor de α fijo y luego calcular α de forma tal de cumplir con la ec.24 como se muestra en la ec.

$$\alpha = \frac{1}{N} \left(\sum_i y_i - \sum_i CF(X_i, \theta) \right) \quad \text{ec.(27)}$$

Para que los parámetros θ no sean desaprovechados en tratar de explicar lo que luego puede explicar el parámetro α se debería cumplir que

$$\frac{\partial \alpha}{\partial \theta} = \frac{1}{N} \sum_i \frac{\partial CF(X_i, \theta)}{\partial \theta} = 0 \quad \text{ec.(28)}$$

3. Incorporación a la plataforma SimSEE.

Se decidió realizar los cambios necesarios en SimSEE para permitir incorporar nuevas representaciones de $CF(X, k)$ (Costo Futuro del estado X en el paso de tiempo k).

Para ello se cambió la representación del problema de despacho de un paso de tiempo para que aparezca en forma explícita como variable de control el Costo Futuro esperado al final de la etapa CF en el estado en que quede el sistema al final de la etapa.

La ec.29 muestra la implementación original. Se puede apreciar que la función objetivo del problema es el costo de etapa ce más una aproximación del costo futuro en el estado de llegada representada por un desarrollo de primer orden alrededor de un estado proyectado \tilde{X}_{k+1} .

$$\min_{u, X_{k+1}} \left[ce(X_k, u_k, r_k, k) + CF(\tilde{X}_{k+1}, k) + \nabla_X CF(\tilde{X}_{k+1}, k)(X_{k+1} - X_k) \right] \quad \text{ec.(29) Despacho óptimo de un paso de tiempo. Versión original de SimSEE.}$$

$$\textcircled{a} \begin{cases} u \in \Omega(X_k, r_k, k) \\ X_{k+1} = f(X_k, u_k, r_k, k) \end{cases}$$

La ec.30 muestra la nueva formulación del problema. Como se puede apreciar desaparece en principio la necesidad de linealizar en el entorno de un estado proyectado y se representa el costo futuro en el estado de llegada a partir de un conjunto de restricciones lineales de desigualdad.

$$\min_{u, X_{k+1}, CF} \left[ce(X_k, u_k, r_k, k) + CF \right] \quad \text{ec.(30) Despacho óptimo de un paso de tiempo. Nueva implementación.}$$

$$\textcircled{a} \begin{cases} u \in \Omega(X_k, r_k, k) \\ X_{k+1} = f(X_k, u_k, r_k, k) \\ CF \geq a_{ik}^T X_{k+1} + c_{ik}; i = 1, 2 \dots n_h \end{cases}$$

Dónde

X_k es el vector de estado en al inicio del paso k .

u_k es el vector de control durante el paso y es resultado del despacho.

X_{k+1} es el vector de estado al final del paso y tal como está planteado sería una variable de optimización.

CF es el Costo Futuro del estimado (informado) para el estado X_{k+1} al final del paso k

$ce(X_k, u_k, r_k, k)$ Es el costo de etapa o Costo Directo del Paso y corresponde a los gastos de combustibles + fallas + importaciones - exportaciones resultantes de la operación.

$u \in \Omega(X_k, r_k, k)$ Representa el conjunto de restricciones sobre las variables de control (potencias despachadas, vertimientos, flujos por las interconexiones). Es el Dominio factible y como se puede apreciar puede depender del estado, de las variables no controlables y del tiempo.

$X_{k+1} = f(X_k, u_k, r_k, k)$ es la ecuación de evolución del estado o modelo de la dinámica del sistema o restricción dinámica.

$CF \geq a_{ik}^T X_{k+1} + c_{ik}; i = 1, 2 \dots n_h$ Restricciones de desigualdad que determinan que el valor de CF será el mayor de los valores calculado para cada uno de los n_h hiperplanos.

Observar que los multiplicadores de Lagrange de la restricción dinámica son las derivadas direccionales del costo al inicio del paso. Esto es utilizado en el algoritmo de Programación Dinámica Estocástica, para tener una nueva aproximación de un hiperplano tangente a $CF(X, k)$. Lo menciono solo por si se nos ocurre que dicha información pueda valer la pena exportarla junto con los resultados de la simulación.

3.1. Bucle de mejora de la política de operación (PO).

Se propuso realizó una primer implementación que incluye las formulaciones más simples en los diferentes pasos para disponer así de un Tractorcito (en el sentido de una máquina simple todo terreno aunque no sea eficiente).

Se seleccionó una Sala SimSEE correspondiente a la programación semanal actualmente en uso. Estas salas son sencillas en cuanto a su representación del estado pero como se dijo antes se trata de construir El Tractorcito. Las variables de estado representadas son los volúmenes almacenados en Bonete, Palmar y Salto Grande. La Sala es de paso horario, de horizonte de simulación 14 días y se simula 1000 crónicas.

El bucle de mejora de la PO es el que se muestra en la Fig.1. La caja verde “Simulación SimSEE” representa la simulación de la Sala y genera un conjunto de archivos de texto con las CRÓNICAS de evolución del estado, el costo directo del paso y el CF (de la función usada como PO para la simulación) por los estados recorridos. En la Fig.1 en lugar del estado original X se utilizó una variable transformada Z previendo la posibilidad de incorporar en SimSEE un traductor de estados que permita calcular $Z = R(X)$. El conjunto de crónicas es leído por un “Aproximador CF” (caja naranja) que realizará una nueva propuesta $CF(Z, k)$ a ser utilizada en la nueva Simulación.

Esta nueva propuesta es entonces leída por SimSEE y traducida a una representación como conjunto de hiperplanos para que sea posible la resolución del despacho como fue planteado en la ec.30. La caja “hiperplanador” se encargará de la tarea de traducir la descripción que se recibe de $CF(Z, k)$ a un conjunto de hiperplanos.

Las cajas “traductor” e “hiperplanador” no están implementadas y en el Tractorcito no son necesarias pues $Z=R(X)$ y la aproximación $CF(Z, k)$ planteada es un hiperplano para cada paso k .

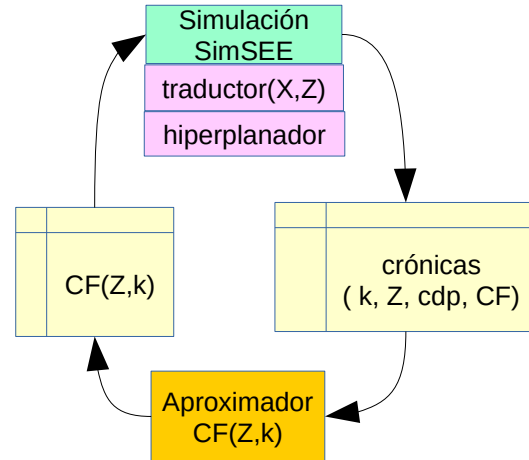


Fig. 1: Bucle de mejora de la PO.

3.2. Función de Valor y Función de Valor-Acción.

Explorada la producción científica sobre el tema, se puede concluir que hasta el momento, los métodos de ataque de la Maldición de la Dimensionalidad de Bellman están por el lado de la “exploración hacia adelante” con intentos de aprender la función de valor de Bellman (o función de costo futuro $CF(X, k)$) o aprender la función de Valor Acción (o Q-learning, $CF(X, u, k)$). En ambos casos se trata de obtener una estimación (mediante iteraciones y por eso utilizaremos la expresión “aprender la función”) del valor esperado del costo de la operación futura (El Costo Futuro).

En el primero de los casos $CF(X, k)$ para cada posición del estado X y para cada paso de tiempo k y en el segundo caso, $CF(X, u, k)$ para cada estado X y decisión (acción) tomada en el tiempo k . Entonces una diferencia importante entre las dos aproximaciones es que la segunda implica el mapeo de un espacio de mayor dimensión (la de X más la de u). En los casos en los que no se dispone de un modelo del sistema y el aprendizaje se realiza mediante la observación de las respuestas del mismo a las señal de control u no hay más remedio que aprender la función $CF(X, u, k)$ con el fin de poder construir una política de operación. En nuestro caso, se dispone del modelo del sistema y por tanto, conocido $CF(X, k)$ es posible calcular en todo momento $CF(X, u, k)$ mediante la resolución del problema ec. 31.

$$CF(X, u, k) = \langle ce(X_k, u_k, r_k, k) + CF(X_{k+1}, k) \rangle_r$$

$$@|X_{k+1} = f(X_k, u_k, r_k, k)$$

ec.(31) cálculo de la función de Valor-Acción a partir del modelo y la Función de Valor

El conocimiento de “el modelo” o de “un modelo” del sistema es el que permite calcular el costo de etapa $ce(X_k, u_k, r_k, k)$ y la evolución del estado $X_{k+1} = f(X_k, u_k, r_k, k)$ en la ec.31.

Además de que el conocimiento de un modelo del sistema implica la no-necesidad de intentar aprender la función de Valor-Acción en nuestro caso, la plataforma SimSEE está implementada de forma tal que no está disponible en forma explícita el vector de control u_k y por tanto resulta más sencillo no intentar manipular este vector en forma directa. Si fuese necesario, es posible y al igual que se hizo con el vector de estado X_k que se representa en forma explícita solicitando a todos los Actores que registren sus variables de estados en un “Administrador del Estado” habría que agregar un “Administrador del Control” en el cuál todos los actores registren sus variables de control y las restricciones sobre las mismas. En esta etapa del proyecto se ha concluido que el aprendizaje de la función de Valor-Acción no agregaría valor y por tanto se ha tomado la decisión de no cambiar la implementación de SimSEE en esa dirección. De surgir en el avance del proyecto la necesidad se reconsiderará el tema.

3.3. El Tratorcito.

La idea es que El Tratorcito, es una aplicación que ejecuta la simulación de una Sala dada en base a unos parámetros de entrada que le fijan los siguientes parámetros:

- Estados Iniciales a explorar. (archivo con el listado de valores del vector X al inicio)
- Semilla Aleatoria. (entero)
- Cantidad de Crónicas a Simular. (entero)
- Política de Operación (Archivo con la descripción)

Los resultados de la ejecución del Tratorcito (que puede ejecutar multi-hilo y distribuido) es una colección de archivos que contienen los resultados correspondientes a la evolución del estado X_k , el costo directo del paso cdp_k y el valor del Costo Futuro al final del paso $CF(X_{k+1}, k+1)$ por los estados que son explorados.

El Tratorcito no utiliza traductor de variables de estado, pero si un hiperplanador que transforma cualquier aproximador de CF en multiplanos.

El planteo es que el aproximador sea simplemente calcular en hiperplano que mejor aproxima en cada paso de tiempo k a la nube de puntos P_j donde el índice j identifica las crónicas simuladas con la definición dada por la ec.32.

$$P_j = (X_k, cdp_k + q CF_{k+1})_j \quad \text{ec.(32) Corrección de CF.}$$

La suma $cdp_k + q CF_{k+1}$ es la corrección del costo futuro informado en la iteración anterior a partir del conocimiento del costo incurrido en la etapa. Siendo q el factor de actualización utilizado en la simulación. La ec.32 es solo un ejemplo de la infinidad de posibles formas de corrección/actualización de CF y se propone para El Tratorcito solo por ser simple.

3.4. Bucle del Tractorcito

Como el Tractorcito puede ejecutarse multi-hilo y distribuido y como ya se mencionó genera una colección de archivos. El bucle del tractorcito puede retomar desde cualquier iteración dada, leyendo la colección de archivos con la información de las trayectorias para poder realizar otra iteración del bucle..

3.5. El Hiperplanador.

En base a los resultados leídos por el bucle del Tractorcito, el Hiperplanador, tiene la función de construir una nueva aproximación de la función $CF(X, k)$ a base de Hiperplanos.

4. Inicialización y Convergencia, Redes Neuronales.

En esta sección se documentan aspectos claves en la resolución eficiente de aproximar la función $CF(X, k)$ a partir de las trayectorias $\{X_i, CF_i\}_k$ donde el subíndice i identifica la trayectoria (esto es un estado de partida al inicio del horizonte de simulación y una “suerte” determinada por una semilla aleatoria) y el subíndice k identifica el paso de tiempo.

4.1. Valores iniciales, regularización y especialización de las neuronas.

La estructura de red neuronal utilizada es del tipo de una capa oculta de Neuronas del tipo Lineal o del tipo Tangente Hiperbólica seguida de una capa con una única neurona que toma el máximo de las salidas de la capa anterior como se ejemplifica en la Fig.2. Se realizaron pruebas con diferentes estructuras y esta estructura demostró ser adecuada para modelar la función de Costo Futuro dado que dicha función, por la característica del problema, presenta facetas planas.

Como es sabido, las Redes Neuronales presentan un desafío al momento de su entrenamiento dado que al tratarse de funciones no-lineales con muchos parámetros, el problema de minimizar el error cuadrático de la aproximación presenta

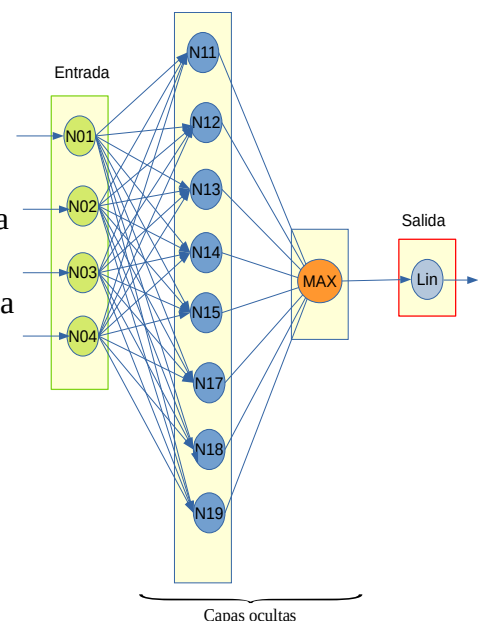


Fig. 2: Estructura de Red Elegida.

mínimos locales. Los diferentes algoritmos utilizan diferentes estrategias para lograr “escapar” de los mínimos locales. No hay entonces garantía de convergencia al mínimo y el resultado es dependiente de los valores de inicialización. En nuestro caso, como la idea es que el algoritmo está en un bucle ejecutándose en forma continua, tenemos por lo menos la esperanza que una vez alcanzada una buena aproximación, con el pasar del tiempo, cuando suceda cambios del sistema (o simplemente avancen las horas) los cambios en los parámetros para converger al nuevo óptimo serán menores. El asunto es entonces lograr en algún momento una buena solución y luego mantenerla.

Nuestro bucle de aproximación incluye una etapa de Simulación de un conjunto de trayectorias seguida de una etapa de mejora de CF (o de la aproximación a la Política de Operación contenida en CF para ser más precisos). La etapa de Simulación consiste en simplemente usar el SimSEE para simular la evolución del sistema a partir de un estado conocido en el instante inicial y una Semilla Aleatoria que determina la suerte del sistema en el horizonte temporal. A cada simulación identificada por el estado inicial y la semilla aleatoria es a lo que llamamos una Trayectoria. Como información de cada Trayectorias i se almacena el estado en cada paso de tiempo y el costo futuro estimado en dicho estado $(X_k, CF_k)_i$ y es en base a esta información que se mejora el modelo de aproximación formado por una red neuronal como la de la Fig.2 para cada paso de tiempo.

En la primer iteración, los parámetros de la red son inicializados en forma aleatoria. Para las siguientes iteraciones, se parte de la red previamente calibrada para dicho paso de tiempo y se recalibra en base a la información de las nuevas trayectorias. Entonces, existe una inicialización de los parámetros que tiene su impacto sobre todo en el éxito de la aproximación de la primer iteración y cuyo efecto se va diluyendo en la medida en que el bucle Simulación-Mejora se ejecuta muchas veces.

La Neurona MAX() usada para seleccionar el máximo de las salidas de las neuronas como se muestra en la Fig.2 presenta un problema en la convergencia de los algoritmos que utilizan en gradiente dado que aquellas neuronas cuya entrada no es seleccionada no aportan a la salida y por tanto el gradiente del error respecto a sus parámetros es nulo. Aquellas neuronas, cuyas salidas, quedan siempre mayoradas por alguna otra en el conjunto de muestra quedan así como “neuronas muertas” sin información para la calibración de sus parámetros. Son Neuronas que “no explican nada” y podrían quitarse del modelo. La suerte con la que una Neurona queda “muerta” o no la determinan los valores iniciales de los parámetros. En un caso extremo en que una neurona mayor a todas las otras por la simple razón de que todas las demás quedaron por debajo, la salida de la red sería simplemente un plano (o una tangente hiperbólica) y la aproximación sin duda sería muy pobre. Para evitar este efecto, se hizo uso de la regularización de los parámetros (al estilo de Ridge). Los parámetros que multiplican las entradas de una neurona muerta tienen a cero por el efecto de la regularización. Si la Neurona es lineal, que los pesos de las entradas tiendan a cero significa que “el plano” que representan tiende a ser horizontal. El Bias de la Neurona determina el nivel del plano. Para hacer que las neuronas se especialicen en explicar diferentes partes del costo futuro, se implementó que la regularización del Bias, en lugar de hacer que el mismo tienda a cero (que implicaría que todos los planos no usados tienden a ser horizontales y con altura cero, todos juntos) tiendan a un valor (nivel) dado en forma aleatoria al inicio. Con esto se logra que las neuronas “muertas” evolucionen a representar un plano

horizontal pero de niveles diferentes dentro del rango de variación de CF. De esa forma, tarde o temprano hay una neurona que es mejor que las otras para tratar de explicar lo que pasa en determinado nivel de CF. Este mecanismo se implemento con éxito mostrando ser determinante en que la complejidad asignada a la red sea efectivamente utilizada en la modelación de CF.

A modo de ejemplo se presenta al Fig.3 donde se muestra un espacio de estado de 2 variables (eje horizontal y oblicuo) y el Costo Futuro en el eje vertical. Los puntos amarillos corresponden a la discretización de una “vatea” (cuatro lados planos inclinados y un fondo plano). En enrejado fucsia es la salida de la red neuronal inicializada en forma aleatoria.

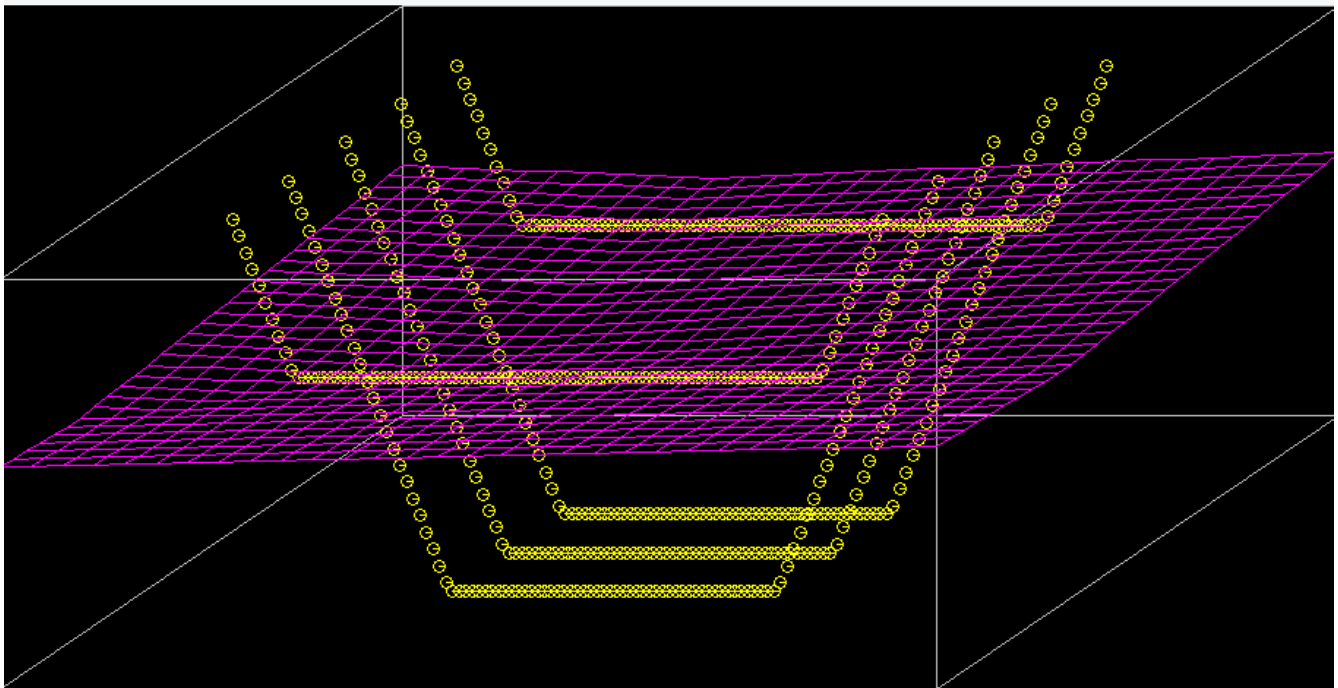


Fig. 3: Ejemplo de inicialización de 3 Tangentes Hiperbólicas con $\text{Max}()$.

Como la vatea tiene 5 caras planas, la red con 3 neuronas es limitada en su capacidad de representación. La Fig.4. Muestra el resultado luego del entrenamiento (con los puntos amarillos). Como se puede observar la red “hace lo que puede” pero las 3 neuronas están utilizadas (no quedó ninguna muerta).

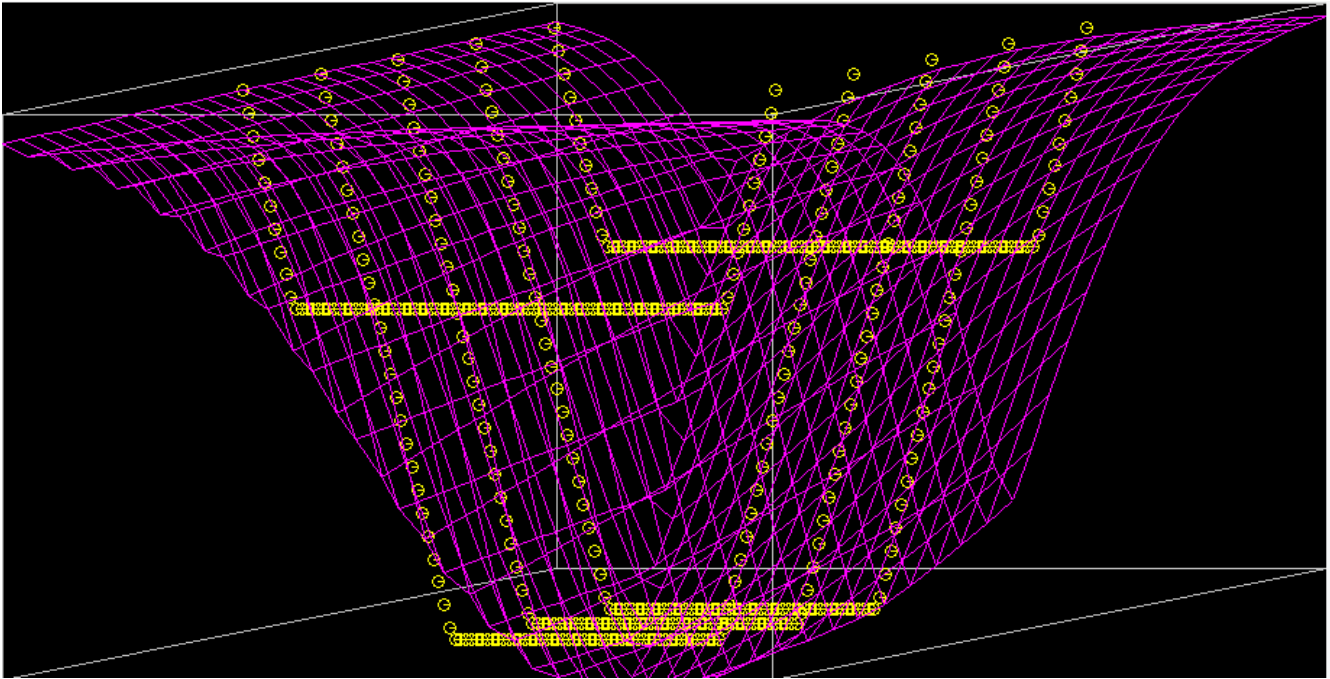


Fig. 4: Convergencia del ejemplo. Con 3 Neuronas Tanh + Max().

La Fig.5 corresponde a la red con 4 neuronas tangente hiperbólica y una Max() en la penúltima capa de la red. Como se puede apreciar la aproximación a la vatea ya es bastante buena.

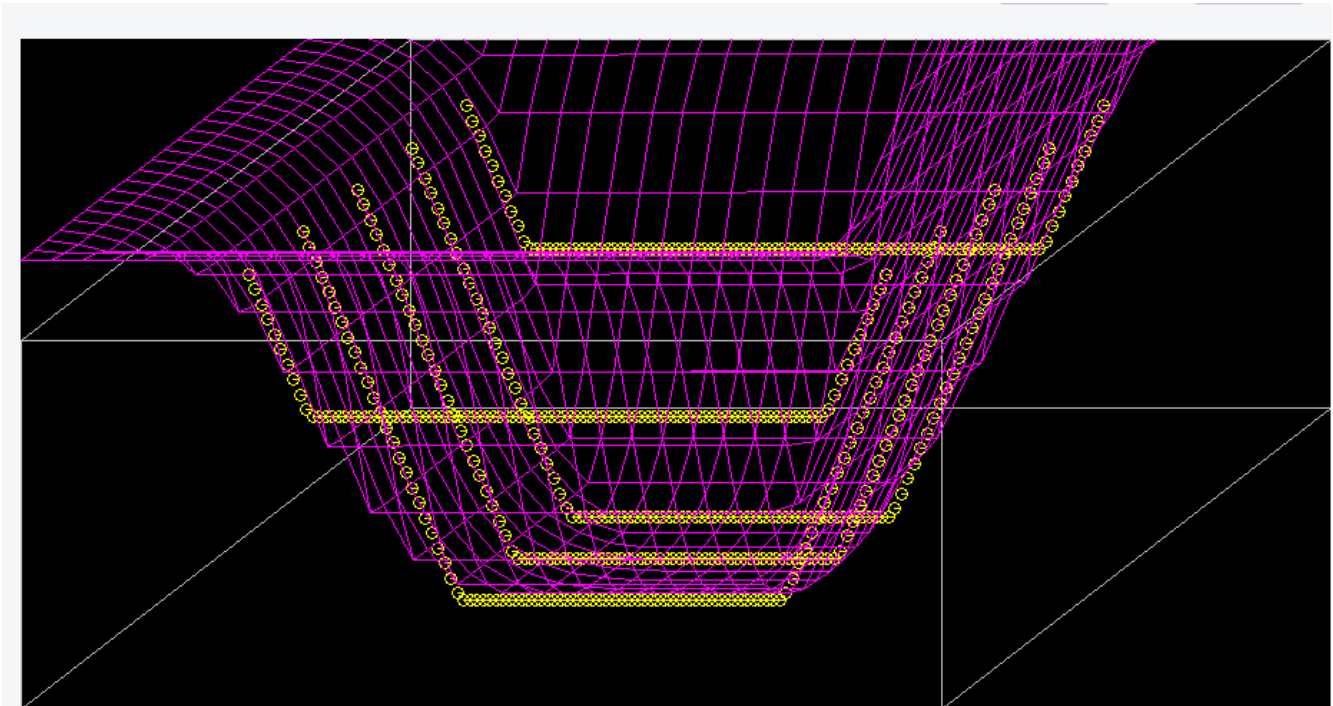


Fig. 5: Ejemplo 4 Neuronas Tanh + Max().

Se puede observar que en el fondo de la vatea la aproximación es un poco “redondeada” dado que con 4 neuronas no es posible representar fielmente los 5 planos.

La Fig.6 muestra la aproximación con 5 Neuronas del tipo Tangente Hiperbólica seguida de una Neurona Max. Del lado izquierdo se muestra en perspectiva igual que las anteriores y se puede observar ahora que “el fondo” es plano. Para visualizar mejor esto se muestra del lado derecho en una proyección sobre uno de los ejes de las variables de estado.

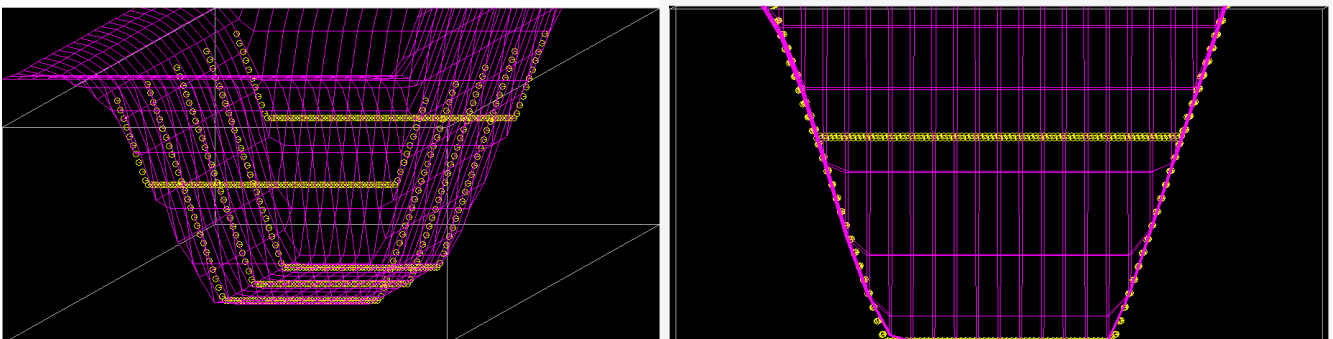


Fig. 6: Ejemplo, Vatea representada con 5 Neuronas Tanh + Max().

Las figuras anteriores corresponden a un ejemplo que consideramos más complejo de aproximar que la información obtenida de las simulaciones.

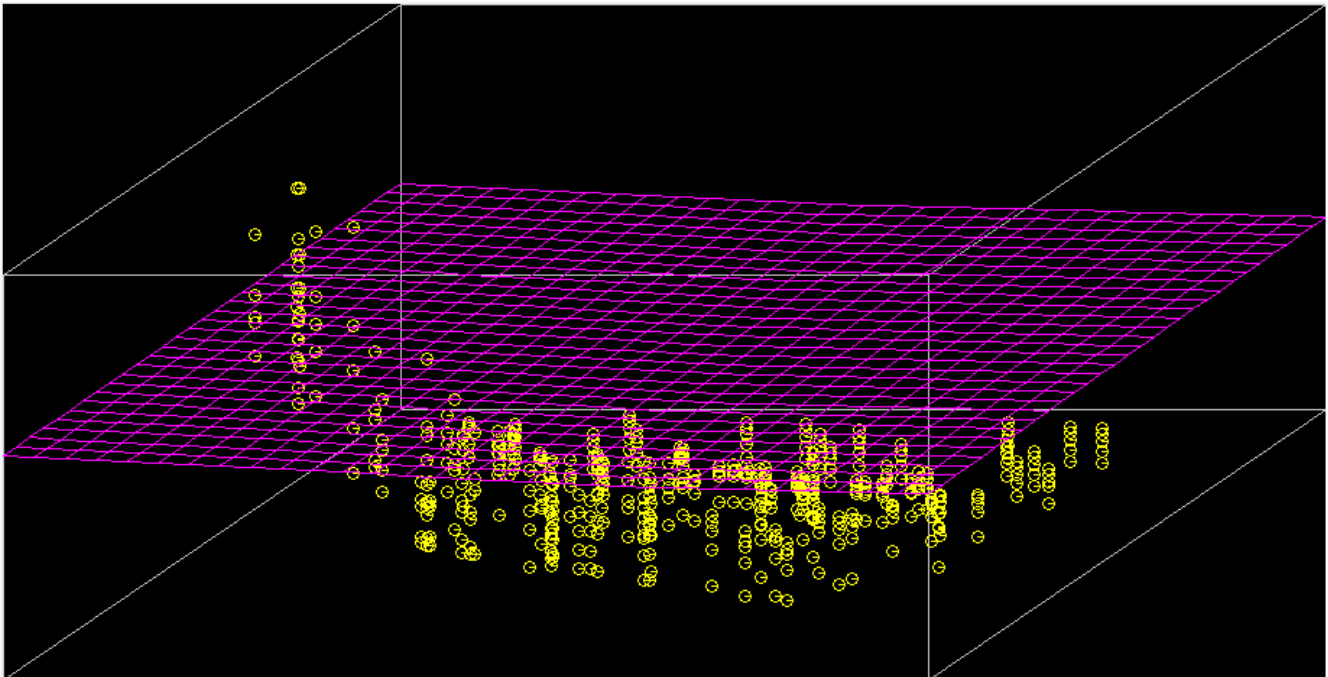


Fig. 7: Primer cuadro de una simulación de un sistema con dos variables.

La Fig.7 muestra la nube de puntos correspondientes al paso $k=0$ de una simulación.

4.2. Exploración vs. Dinámica.

Para la primer simulación, se parte de una aproximación equivalente a $CF(X,k)=0$ esto tiene la consecuencia de dar derivadas nulas (no hay información como para saber si aumentar o reducir el valor del estado sube o baja el costo del futuro). En el caso de los embalses de las centrales hidroeléctricas esto se traduce en valor el agua a 0 (Cero) y como consecuencia, en esa primer simulación, todas las trayectorias, sin importar del estado inicial tenderán a vaciar los lagos. La Fig.8 muestra como ejemplo, la distribución de los estados (Estrellas) en el paso $k=0$ de izquierda a derecha y bajando. En cada cuadro en horizontal está la variable de estado que corresponde al volumen embalsado en Rincón de Bonete y en vertical está la variable de estado que representa el “estado hidrológico” (sesgo de los aportes hidráulicos, mayor valor mayor probabilidad de lluvias y aportes menor valor menos probabilidad). Si partimos de ese conjunto de estados iniciales y con la política de operación contenida en $CF(X,k)=0$, el resultado será que se utilizará el agua embalsada como si la misma tuviera valor cero causando que todos los estados vayan transitando hacia la izquierda.

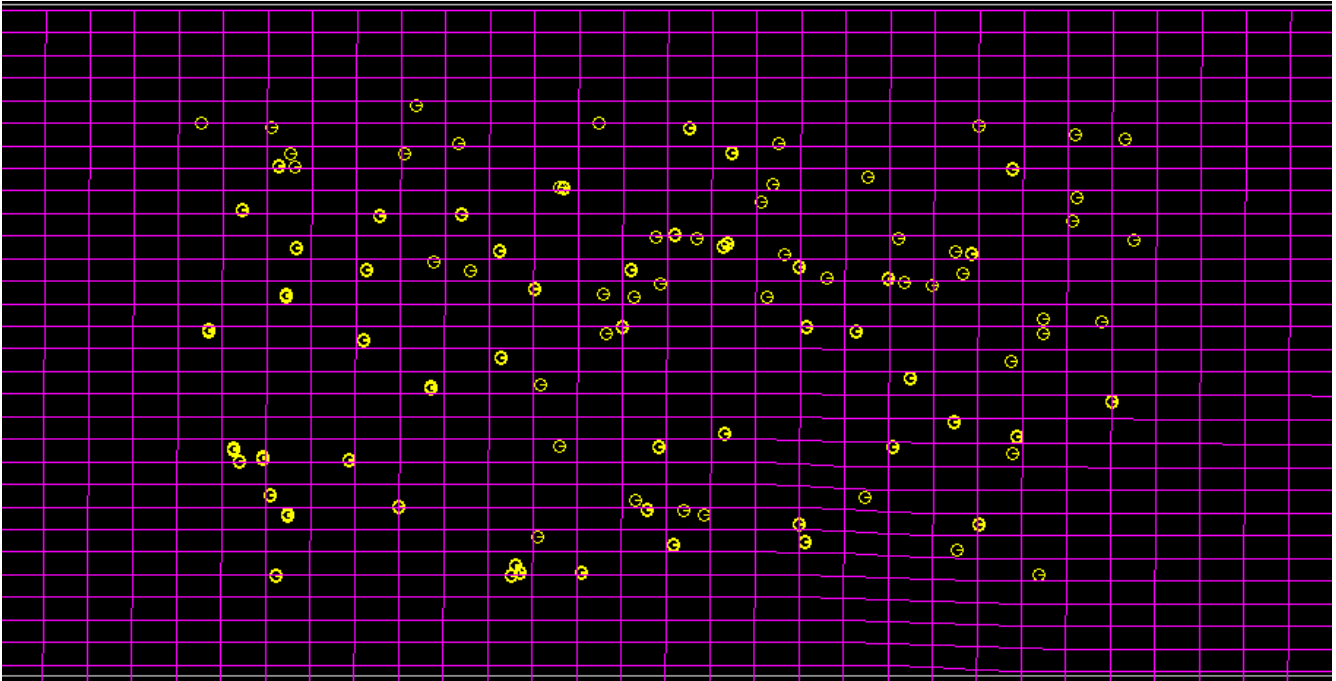


Fig. 8: Ejemplo de Estrellas del primer frame $k=0$. Sala LP. Horizontal VBon y Vertical H.

La Fig.9 muestra la evolución del conjunto de Estrellas (estados) de los primeros 18 frames correspondientes a pasos de simulación semanal. Como se puede apreciar, los estados tienden a evolucionar hacia la izquierda amontonándose en lo que representa el fondo del lago. Esta evolución del estado presenta el inconveniente de no permitir tener una buena representación de $CF(X, k)$ dado que quedan partes del espacio de estado sin explorar. Este comportamiento es especialmente indeseable al inicio de las aproximaciones, dado que no se tiene información sobre $CF(X, k)$ y por tanto no se sabe si la parte que no se explora es o no relevante.

Para solucionar este aspecto, se modificó el simulador SimSEE para permitir especificar durante la simulación el tipo de evolución de estado que se quiere aplicar. El Modo 0 (Cero) es el clásico y el estado evolución acorde a la dinámica del sistema (respetando la ecuación de evolución de estado). La Fig.9 se obtuvo con ese modo de evolución. En el Modo 1 (No evolución) los estados no evolucionan con lo cual se explora el conjunto de estados iniciales como si fuesen trayectorias hacia adelante.

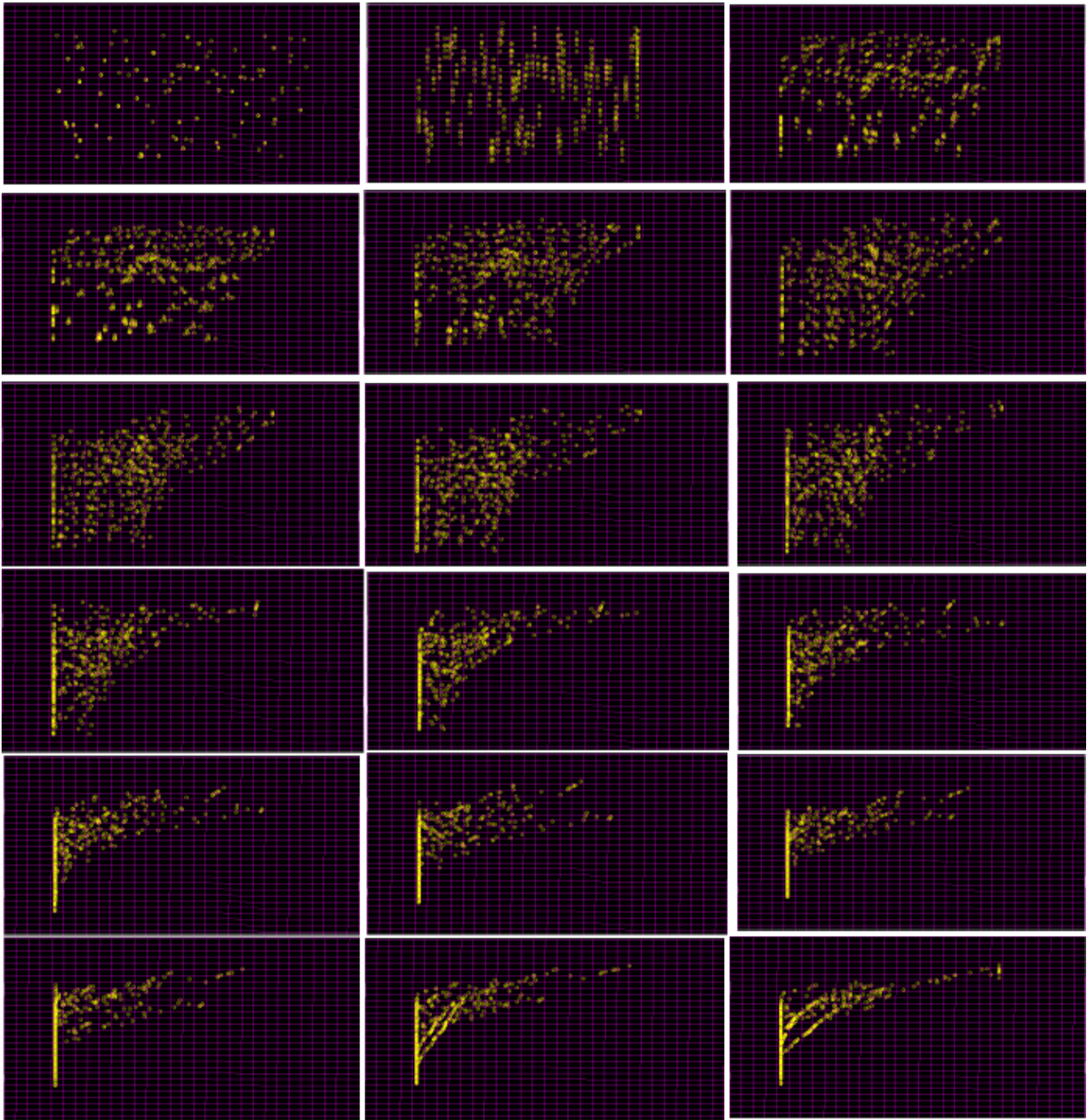


Fig. 9: Primeros 18 frames, evolucionando el estado según la dinámica del sistema.

Como se mencionó anteriormente, la necesidad de realizar una exploración del espacio de estados surge como consecuencia de que vamos a simular el modelo del sistema solo hacia adelante en el tiempo y el sistema no va poder pasar por todos los estados posibles del espacio de estados. Esto es inevitable que suceda ya que, aunque el sistema parte de un conjunto de estados iniciales (que puedes ser elegidos arbitrariamente), la evolución desde cada uno de esos estados está gobernada por funciones de transición, las cuales determinan los posibles estados de llegada en cada paso de tiempo. Esos estados que el sistema se está perdiendo de visitar podrían aportar información relevante para poder lograr una mejor aproximación a la función de CF, la cual depende de los estados y del tiempo.

Debido a lo anterior, los estados visitados en cada trayectoria que toma el sistema, se ubican en regiones del espacio de estados que pueden tener información más o menos relevante para el objetivo de aproximar la función de CF en dicho paso. En otras palabras, una buena elección de los estados a evaluar en cada paso de tiempo puede aportar más información para lograr una mejor aproximación del CF.

4.2.1 Exploración del espacio de estado

La simulación del sistema está sujeta a las restricciones dinámicas que rigen la interacción a lo largo del tiempo entre los distintos elementos que lo constituyen. Estas restricciones dinámicas implican que no sea posible que el sistema recorra trayectorias que entren en conflicto con dichas restricciones, en las transiciones de un paso de tiempo al siguiente. Lo anterior presenta una limitación si se requiere guiar al sistema por regiones del espacio de estados que no son alcanzables desde el estado inicial. En base a lo anterior se desarrolla la posibilidad de realizar la simulación sin la necesidad de respetar la ecuación de evolución de estado del sistema. Con ese objetivo se implementaron 5 modos de evolución (o exploración) para que en la simulación se pueda modificar los estados visitados en cada paso de tiempo y en cada iteración del bucle de optimización.

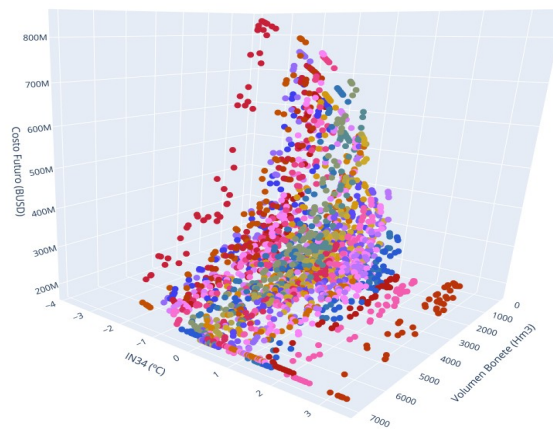
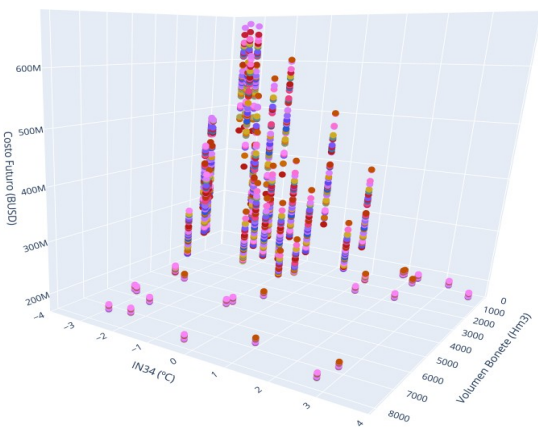
4.2.2 Distribución inicial de estados

Al comienzo de la simulación se realiza una distribución inicial de estados de partida del sistema, los cuales son llamados estrellas iniciales. Para definir estos estados iniciales se requiere especificar la cantidad de estados y la cercanía de los mismos con respecto al estado inicial definido en el modelo del sistema. Para esto existe una variable que representa la “distancia” de los estados a determinar con respecto al estado inicial del sistema. La variable varía entre cero y uno, donde el cero representa la distancia nula con respecto al estado inicial y el uno permite que las estrellas iniciales se distribuyan en todo el rango del espacio de estados. Una vez especificado la cantidad de estrellas y la distancia con respecto al estado inicial del sistema, las estrellas iniciales son distribuidas de forma uniforme en el espacio de estados, teniendo en cuenta que según la distancia especificada, los estados de partida puede estar más concentrados en un entorno del estado inicial o más distribuidos en el espacio de estados.

Modos de exploración

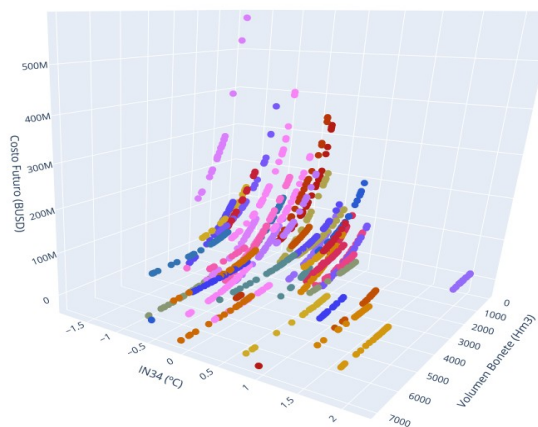
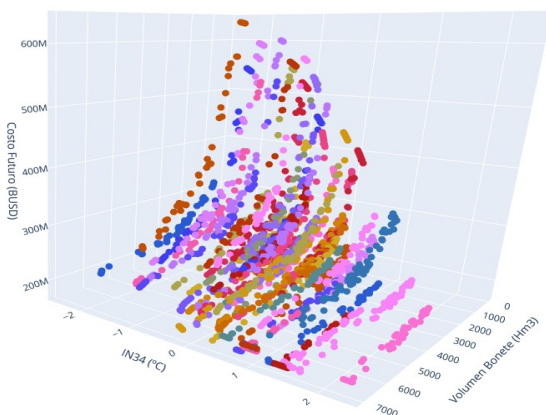
Modo 0: Este modo de evolución es el que tiene por defecto el módulo de simulación del SimSEE clásico. La evolución de los estados de un paso de tiempo al siguiente se obtiene en la resolución del simplex y se evoluciona el sistema al siguiente paso utilizando los estados de llegada de las variables según las funciones de transición del sistema y las restricciones correspondientes.

Las simulaciones realizadas sobre el *sistema dinámico simple para testeo* utilizando este modo de evolución muestran un comportamiento particular de los estados visitados en cada paso de tiempo. En el paso de tiempo inicial, los estados de partida son elegidos mediante la distribución inicial de estados explicada anteriormente. A medida que el tiempo avanza, los estados visitados en cada paso de tiempo van agrupándose en conjuntos que presentan la misma suerte. Dicho de otro modo, desde el paso inicial (donde están ubicado en el espacio de estado según la distribución elegida) los estados van juntándose y agrupándose en sub-grupos en cada paso de tiempo, hasta el punto en que para los pasos de tiempo finales, los estados están condensados (amontonados) en relativamente pocos lugares del espacio de estados. Esto implica que en los pasos de tiempo iniciales se realiza la simulación con una cantidad de estados distribuidos por una región del espacio de estado y a lo largo del tiempo los estados se amontonan entre sí para culminar hacia el final del horizonte de simulación, con una distribución de estados mucho menor que la seleccionada para el comienzo. Este comportamiento particular se muestra en las siguientes figuras y se debe a que las transiciones del sistema desde estados que son similares y bajo una misma suerte, va a tener transiciones hacia estados también similares. Este proceso se va a repetir en cada paso de tiempo, produciendo que desde estados de partida cercanos, el sistema evolucione hacia estados de llegada también cercanos, disminuyendo el grupo de estados visitados en cada paso de tiempo a medida que el sistema evoluciona.

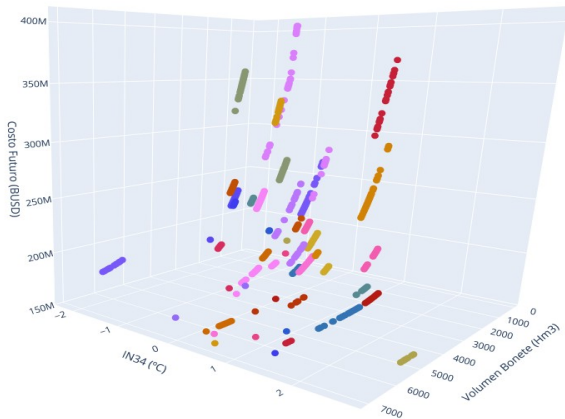


Paso 1

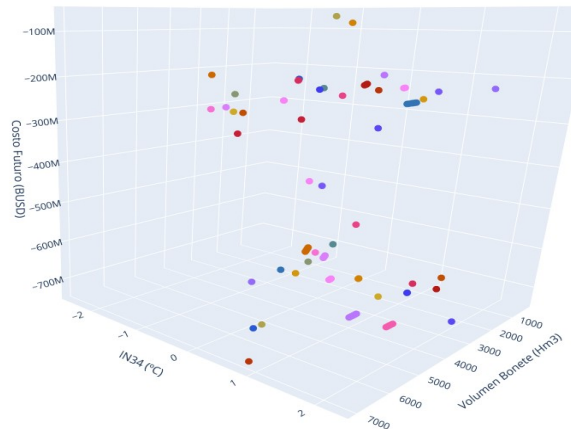
Paso 5



Paso 10



Paso 20

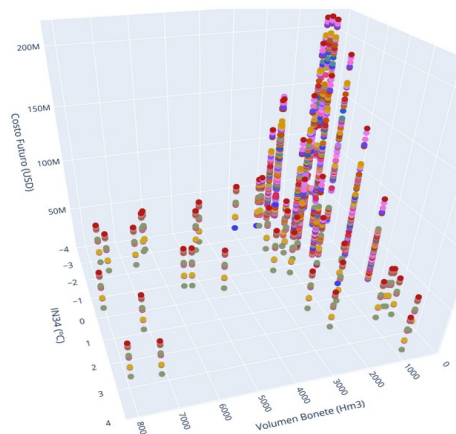
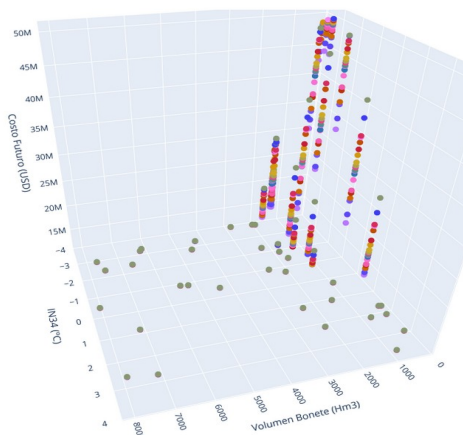


Paso 50

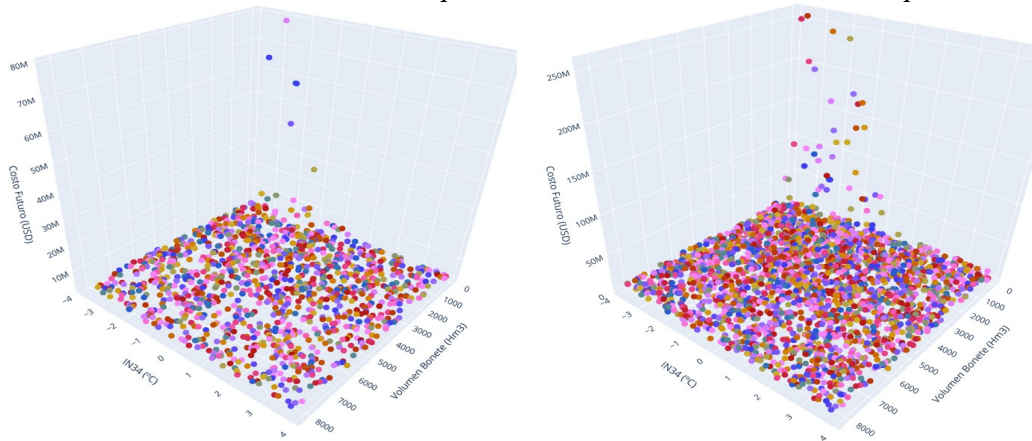
Paso 90

A partir de lo anterior resulta conveniente disponer de una metodología para poder asignar a cada paso de tiempo los estados que se quiere visitar. Como contrapartida, al seleccionar los estados en cada paso de tiempo sin respetar la función de transición del sistema surge el problema de no poder acumular, por más de un paso de tiempo, los costos directos del paso obtenidos en la resolución de cada paso de tiempo. Esto es debido a que al no respetar la dinámica del sistema en la evolución del estado, no se puede sumar costos incurridos por evolucionar el sistema de un estado a otro que posiblemente sería inalcanzable si el sistema estuviera evolucionando según su función de transición. Además de lo anterior, y de ser posible seleccionar estados en el paso de llegada que si pudieran ser alcanzados por el sistema desde los estados de salida del paso de inicio, quedaría la dificultad de definir la asignación de estados de salida y llegada para realizar la acumulación de los costos directos incurridos en la resolución del paso de tiempo.

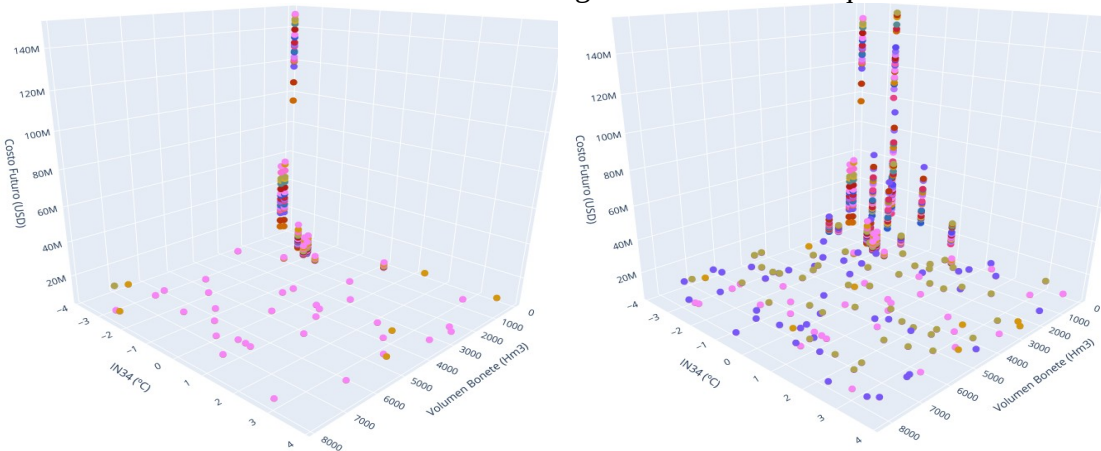
- Modo 1: En el modo de evolución 1 se comienza como en el caso anterior con una distribución de estados determinada y luego de la resolución de cada paso de tiempo se asignan los estados de llegada al paso siguiente simplemente copiando los estados de inicio del paso anterior. Este es el modo de evolución más simple y gráficamente se puede visualizar como un “tubo” en el cual se encuentran los estados en el espacio de estados a lo largo de los pasos de simulación, donde los estados se mantienen iguales.



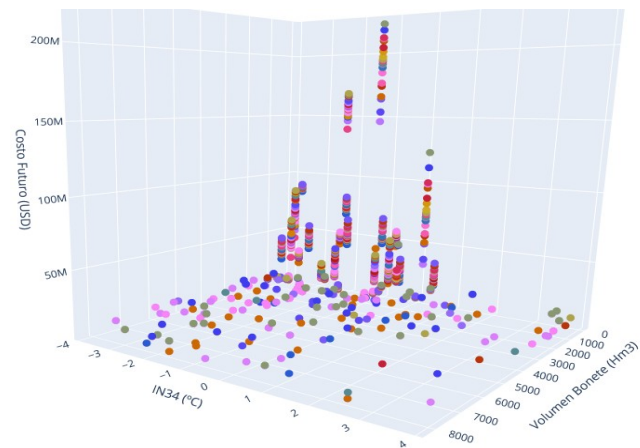
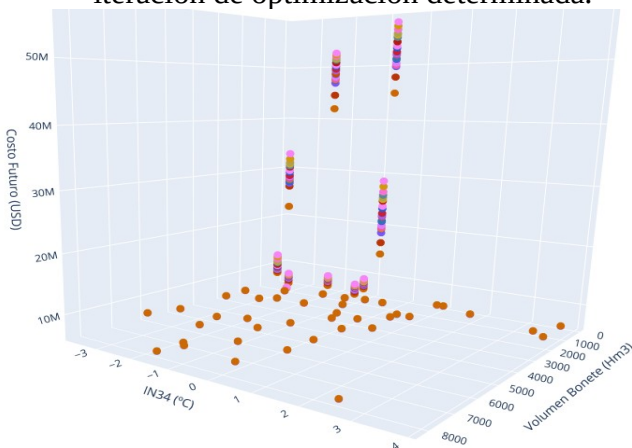
- Modo 2: En el modo de evolución 2 se comienza el primer paso de la simulación como en el caso anterior con una distribución de estados determinada y luego de la resolución de cada paso de tiempo se realiza un sorteo aleatorio con una distribución uniforme para seleccionar los estados de llegada (que son los de comienzo del paso de tiempo siguiente). El sorteo se realiza para cada paso de tiempo utilizando una fuente generadora de números pseudo-aleatorios inicializada con una semilla que cambia con el índice del estado que se está sorteando.



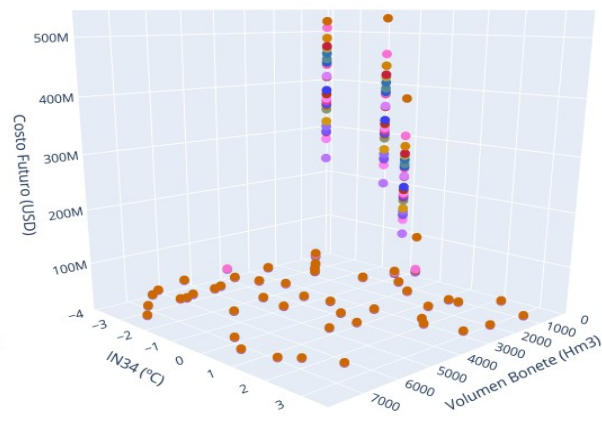
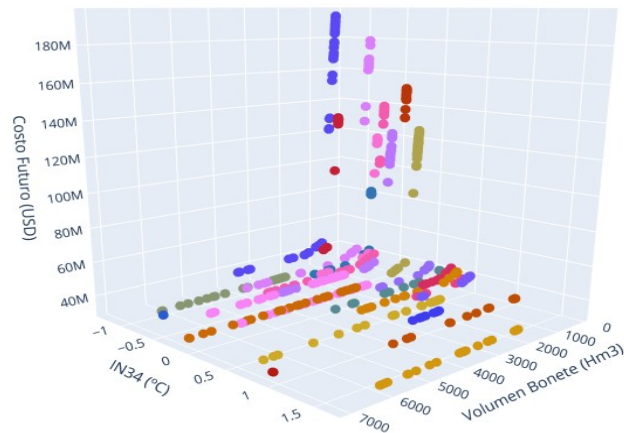
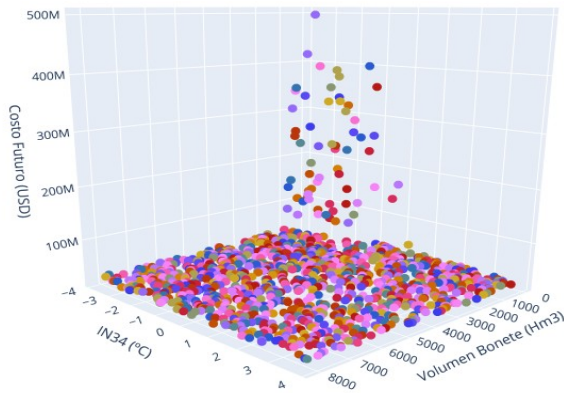
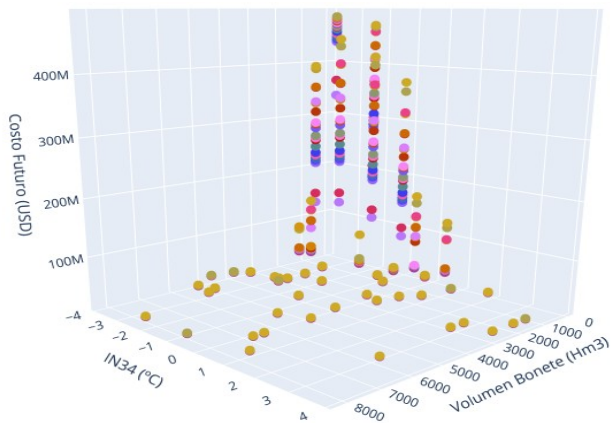
- Modo 3: El modo de evolución 3 es muy similar al modo 2, se comienza como en el caso anterior con una distribución de estados determinada y luego de la resolución de cada paso de tiempo se realiza un sorteo aleatorio con una distribución uniforme para seleccionar los estados de llegada (que son los de comienzo del paso de tiempo siguiente). El sorteo se realiza para cada paso de tiempo utilizando una fuente generadora de números pseudo-aleatorios inicializada con una semilla que cambia con el índice del estado inicial (el estado inicial es el estado correspondiente de los estados utilizados en el primer paso de tiempo para comenzar la simulación). Al tener una semilla diferente que genera los sorteos para obtener los estados, se obtienen una distribución de estados ligeramente diferente que la obtenida en el modo 2.



- Modo 4: El modo de evolución 4 también genera una distribución de estados para cada paso de tiempo, pero se diferencia de los modos de evolución anteriores en que los sorteos de los estados se obtienen de una fuente generadora de números pseudo-aleatorios que tiene una semilla que cambia con el número de la iteración del bucle de optimización. Esto produce que la distribución de estados para un paso de tiempo k en una iteración de optimización j sea única y diferentes a la del resto de la iteraciones de optimización para ese mismo paso de tiempo. Además otra funcionalidad que tiene el modo 4 es permitir utilizar una fuente generadora de números pseudo-aleatorios con sorteos según una distribución gaussiana. Esto se utiliza para generar distribuciones de estados con más densidad en cierta región del espacio de estados que sea de interés tener mejor muestreada. Para el *sistema dinámico simple para testeo* en el cual se tiene 2 variables de estado y se conoce el comportamiento de la función de costo futuro, en el sentido que se sabe que si el estado evoluciona en cierta dirección el costo sube o baja, puede ser de interés evaluar con más resolución (más cantidad de estados en una región específica del espacio de estados) la región del espacio de estados donde la función de costo sube más rápido. Entonces se puede usar la fuente gaussiana para generar una distribución de estados que, manteniendo la cantidad total de estados por paso de tiempo, tenga más cantidad de estados en esa región y menos en el resto del espacio. Al ser dependiente la semilla del número de la iteración del bucle de optimización y no del índice del estado en el bucle de simulación, la distribución de estados en el espacio de estados no cambia a lo largo de la simulación para una iteración de optimización determinada.



- Modo de evolución 5: En el modo de evolución 5 se combinan los modos de evolución anteriores dentro del bucle de optimización, permitiendo seleccionar en una secuencia predeterminada cual modo de evolución está activo para cada iteración del bucle de optimización. Esto permite que se vayan alternando los modos en iteraciones sucesivas para darle riqueza en la selección de los estados de cada paso de tiempo, permitiendo la incorporación de información potencialmente valiosa cuando el sistema evoluciona desde estados que presentan características especiales como puede ser una derivada grande de la función CF en la vecindad de ese estado.



4.2.3 CONVERGENCIA DEL ALGORITMO

Se presenta a continuación una propuesta para intentar asegurar la convergencia del algoritmo hacia una función CF solución que sea un óptimo local en el espacio de funciones posibles. La idea es que en cada paso del iteración del tractor se obtenga una función CF que sea más óptima que la obtenida en el paso anterior, de manera que se avanza en cada iteración con una política de operación mejor. La métrica para evaluar que tan buena es una función CF es calcular mediante la simulación del modelo del sistema el costo directo del paso acumulado (cdpAcum) en el horizonte de optimización para un número determinado de crónicas de simulación. Esta magnitud puede ser promediada en el conjunto de crónicas para condensar en un solo valor la información y también se puede calcular su varianza o dispersión, para tener noción del rango en el cual puede moverse.

Dentro de la implementación para obtener dicha convergencia se destacan la inicialización y el objetivo del bucle de convergencia.

Inicialización: Se inicializa el algoritmo con un CF base que es nulo y que representa que no se tiene ningún conocimiento sobre la política de operación. Si el operador del sistema sigue una política de operación obtenida de un CF nulo, el agua guardada en los embalses de las centrales de generación hidroeléctrica no tiene valor en ningún momento y es usada de forma discrecional siendo despachada lo más posible en todo momento (o compitiendo con otros recursos de costo variable nulo como la generación eólica y solar) en función de otras restricciones que puedan existir en el modelo del sistema (cantidad de maquinas disponibles, roturas, congestión de la red, etc). Como consecuencia de lo anterior una función de CF que es nula representa que no se tiene conocimiento alguno sobre en que momento es más conveniente utilizar generación hidroeléctrica o en que momento es conveniente guardarla y utilizar otro recurso de generación en su lugar, dando a lugar una operación el sistema muy costosa con un costo directo del paso acumulado en el horizonte de simulación que va a tomar el valor más grande posible.

Objetivo del bucle de convergencia: A partir de una función de CF nula el algoritmo debe ir obteniendo en cada iteración una mejor función de CF que produzca, al ser evaluada con la métrica del c_{dpAcum} , un costo menor que el que producía la función de la iteración anterior.

4.2.4IMPLEMENTACIÓN

Inicialización:

1. Se inicializa el CF base con un CF nulo.
2. Se crean las trayectorias a utilizar eligiendo un modo de evolución determinado.

Bucle de iteración:

1. Se simula el sistema utilizando esas trayectorias según el modo de evolución elegido utilizando la política de operación obtenida del CF nulo y se calcula el costo directo del paso para cada paso de tiempo de cada trayectoria. Una vez finalizado este proceso, se está en condiciones de calcular para cada estado visitado, el costo incurrido en la resolución del paso más el costo futuro del estado de llegada (aplicando el factor de descuento correspondiente), obteniendo la magnitud $c_{dp} + \gamma \cdot CF$ como aproximación para el valor esperado del CF en ese estado $\langle CF \rangle$. Esta es la información necesaria para aproximar la nueva función de CF por lo que se procede a utilizar el aproximador seleccionado (red neuronal, aproximador lineal, etc) para ajustar una función a los puntos (x, CF) en cada paso de tiempo.
2. Se vuelve a realizar este mismo procedimiento con cada uno de los otros modos de evolución que se elijan para utilizar.
3. Una vez terminado el paso anterior se dispone de un conjunto de funciones CF donde cada una fue aproximada sobre un conjunto de puntos del espacio de estados obtenido bajo un modo de evolución determinado.
4. En este momento se está en condiciones de evaluar dichas funciones CF y ver cual de ellas es mejor. Para lograr esto se simula el sistema en modo de evolución 0, utilizando como CF base cada uno de los CF's obtenidos en el paso anterior, y se calcula el c_{dpAcum} correspondiente en cada simulación. Luego al comparar estas magnitudes se puede

determinar cual de los CF's produce el cdpAcum menor siendo elegido como candidato para ser utilizado como CF_base en la próxima iteración.

5. En este momento es necesario verificar si el cdpAcum del CF seleccionado es menor que el cdpAcum del CF de la iteración anterior. Si lo es, entonces se tiene un CF válido para poder continuar con el bucle del algoritmo, ya que se obtuvo una función que produce una política de operación que es más óptima según la métrica del costo acumulado total. Si no se verifica que el cdpAcum del CF seleccionado es menor que el que se tenía anteriormente, entonces se debe volver al segundo paso de la inicialización para intentar encontrar un nuevo CF que cumpla lo anterior. Esto se realiza cambiando las semillas utilizadas en los modos de evolución y/o seleccionando otro modo de evolución que no había sido elegido y volviendo a obtener los candidatos a CF mediante simulación.
6. Una vez obtenido un CF que sea más óptimo que el del paso anterior, se lo selecciona como CF base y se comienza una nueva iteración. Mediante este procedimiento se va obteniendo en cada iteración un CF mejor y un cdpAcum menor.

4.2.5 Condiciones de convergencia:

7. Es necesario definir las condiciones de convergencia del algoritmo con algún criterio que asegure que ya se obtuvo una función CF lo suficientemente buena o que no es posible mejorar. En función de lo detallado en el bucle de iteración, las sucesivas iteraciones pueden continuar produciendo funciones CF hasta cierto momento en el cual no se pueda obtener un nuevo CF que sea mejor que el de la iteración anterior. Esto puede ser determinado por una máxima cantidad de cambios de semilla a especificar, según lo comentado en el punto 5. Sin embargo es posible que durante las sucesivas iteraciones se continúe obteniendo una mejora en la función CF pero a un ritmo muy lento (la disminución del cdpAcum de una iteración a la siguiente es muy pequeña), en tal caso se puede definir un porcentaje de cambio del cdpAcum mínimo, por debajo del cual se considere que la magnitud no está modificándose significativamente. Una vez obtenida alguna de las condiciones de convergencia, se termina el bucle de iteraciones y se da como resultado el CF obtenido.

4.3. Reducción de la Varianza. Coherencia de la Suerte.

La información recogida en las simulaciones está sujeta a la suerte que tenga el sistema en cada una de las trayectorias simuladas.

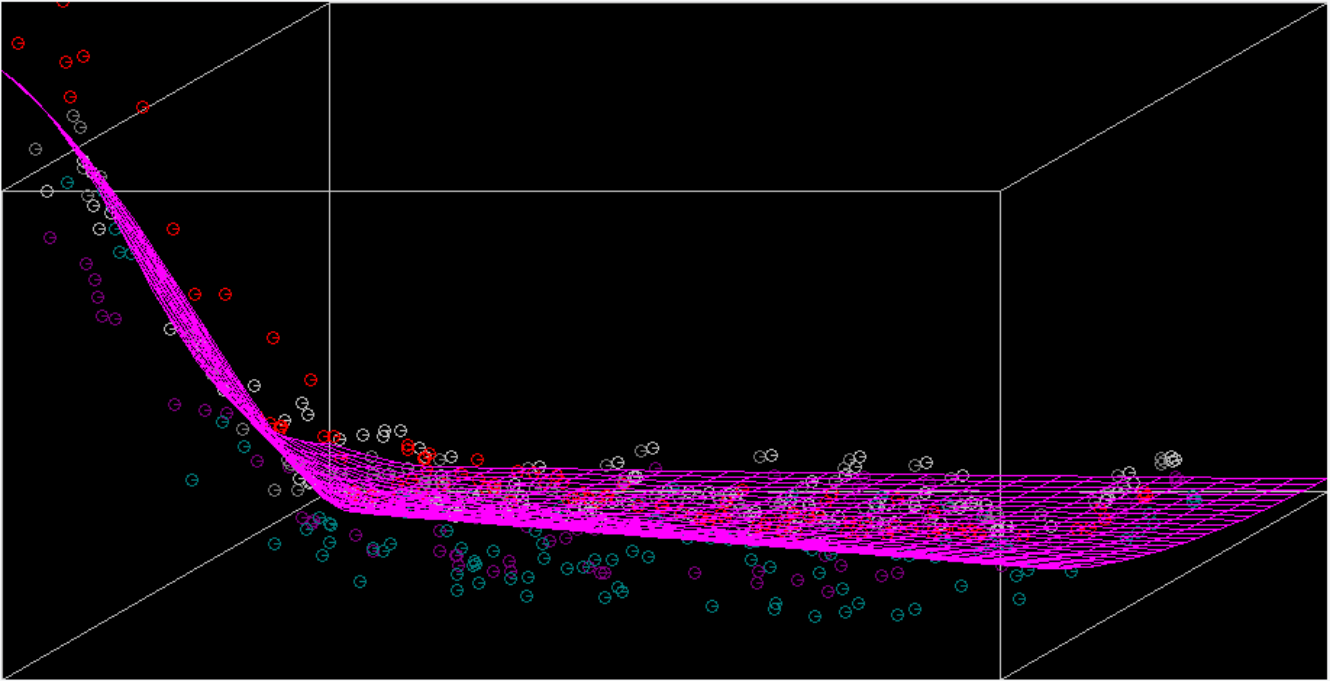


Fig. 10: Frame 0. 50 Estrellas x 5 Suertes. 3 tanh + Max().

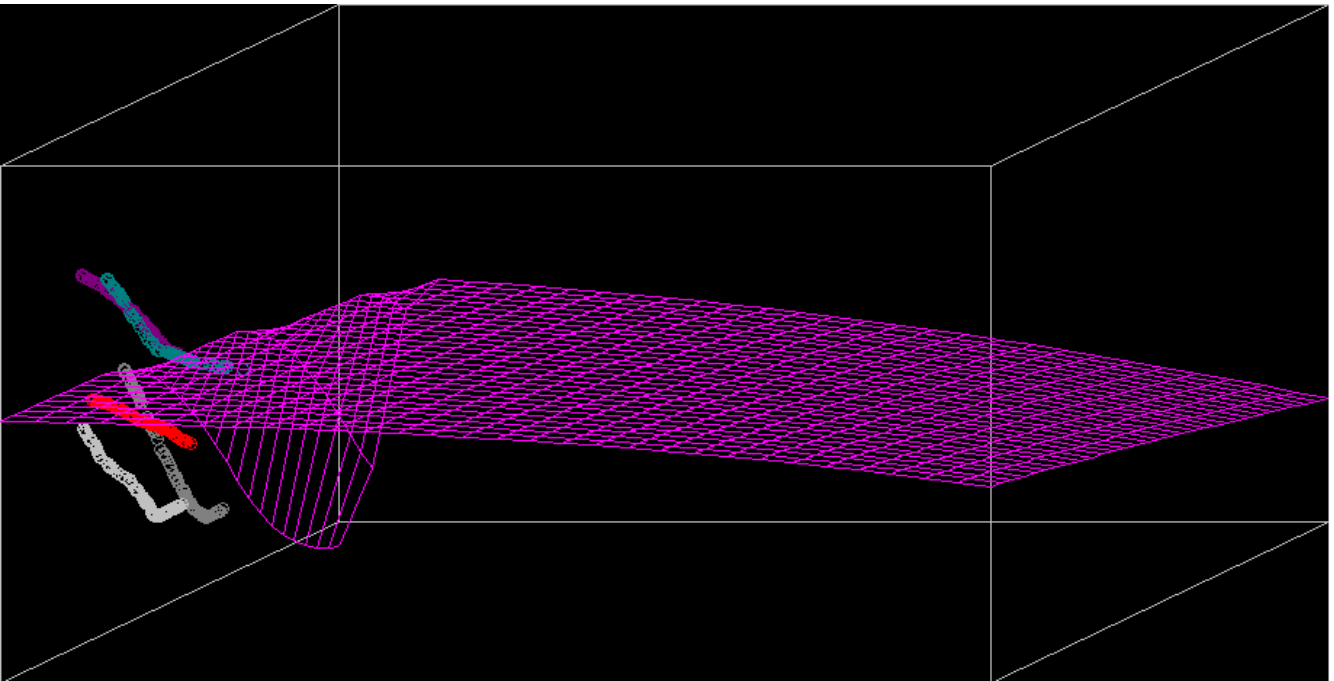


Fig. 11

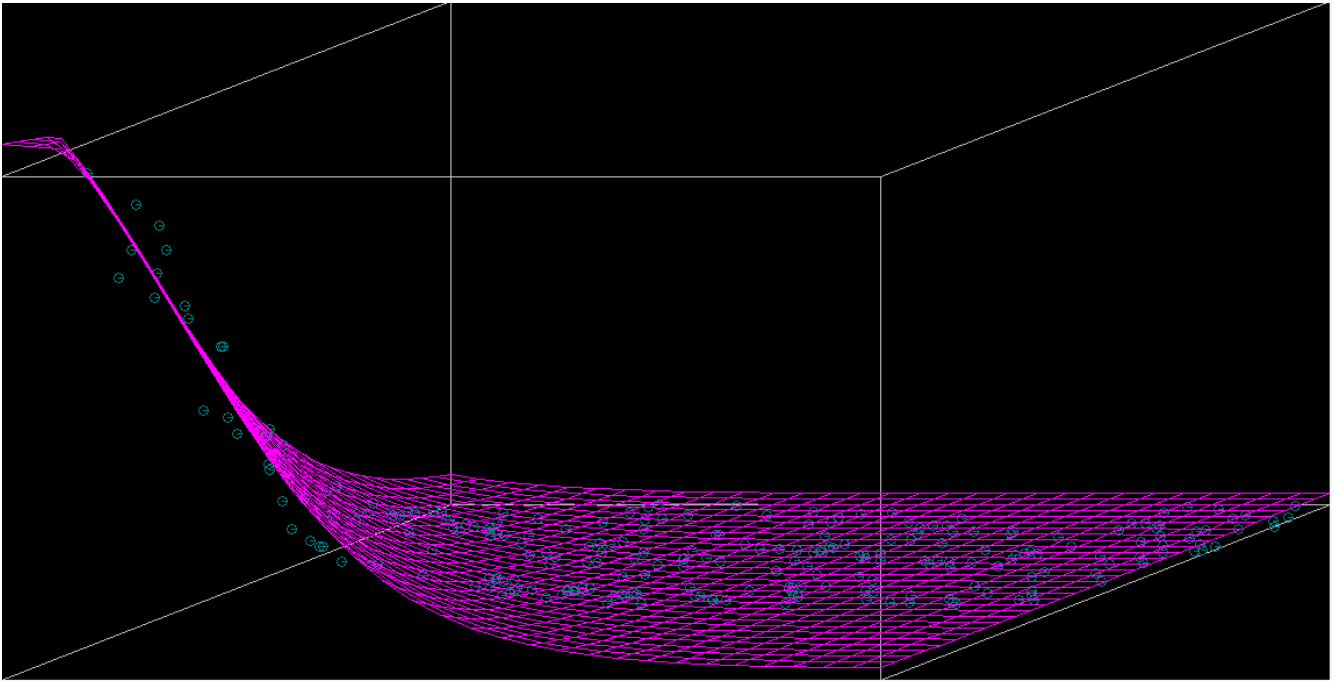


Fig. 12: Frame 0, 250 Estrellas x 1 Suerte. 3 Tanh + Max()

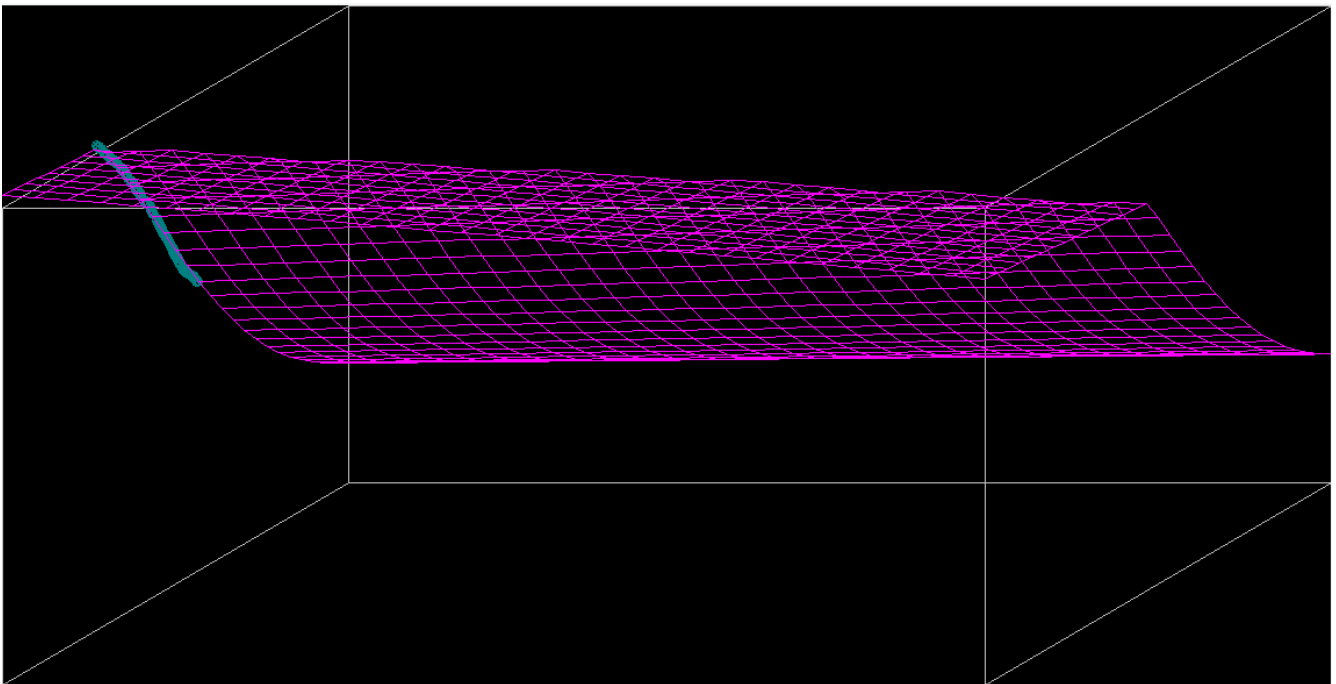


Fig. 13: Frame 62, 250 Estrellas x 1 Suerte. 3 TanH + Max().

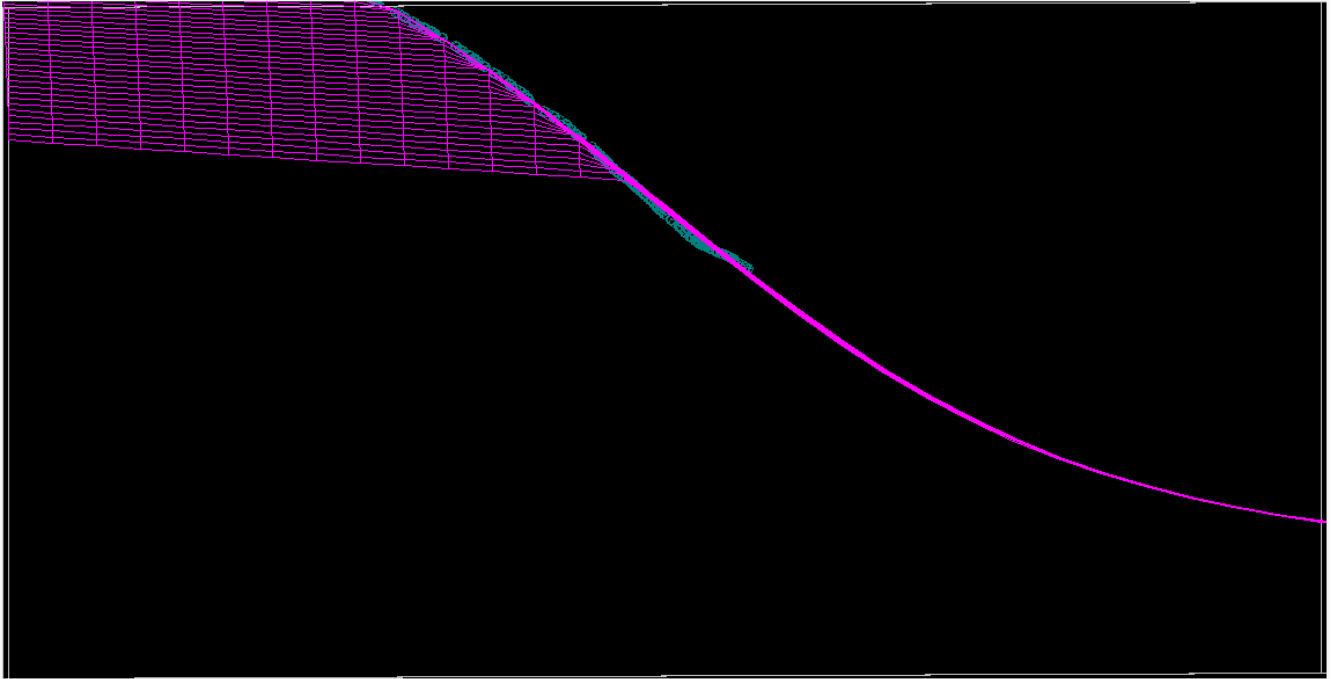


Fig. 14: Vista colineal con el Volumen de Bonete. En horizontal el estado H.

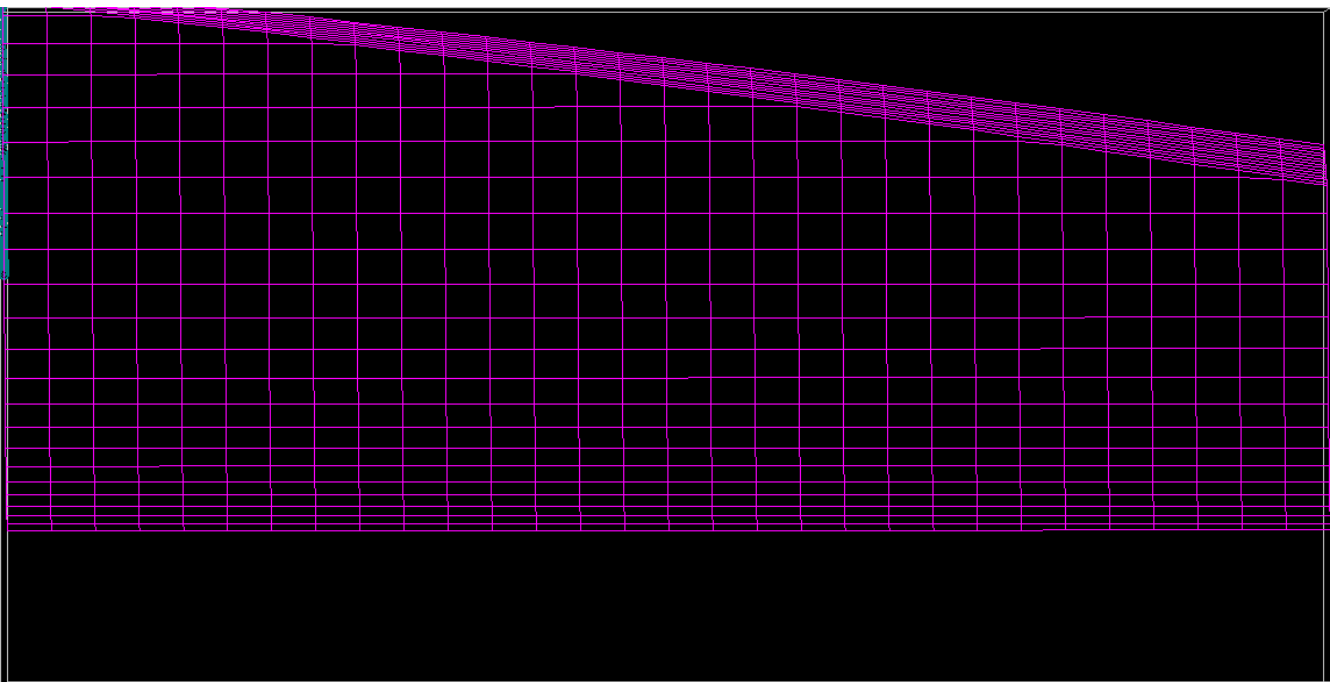


Fig. 15: Vista colineal con el estado H. En horizontal el Volumen de Bonete.

4.4. Inicialización de las Neuronas para permitir su especialización en la explicación de zonas del espacio de estado.

Cuando se selecciona como estructura de las RN una capa oculta seguida de una capa con una única neurona del tipo Max (o PonderSoftMax) se tiene como resultado que, dependiendo de la inicialización algunas neuronas pueden quedar totalmente "opacadas" por sus hermanas, no participando para nada de la explicación de la función $CF(X)$. En las primeras implementaciones que se realizan, se daban casos en que aunque los datos mostraban ser representables por dos o más planos y la aproximación se estaba realizando con múltiples neuronas, la solución que se obtenía era la de un solo plano. La explicación no era otra que dada la condición inicial, un plano quedaba mayorando a los demás en el conjunto de datos y por tanto solo la neurona que lo generaba mostraba aportaba al gradiente de calibración (calculado con el método de back-propagation). Esto se debe a que la neurona que pone a su salida el máximo de sus entradas, solo transmite el error (y por tanto el cálculo del gradiente) hacia la entrada que quedó mayorando para una muestra dada.

Para evitar este problema, se introdujo un parámetro de regularización del Bias de las neuronas, de forma tal que si la neurona "no participa de la explicación", sus pesos de las entradas tienden a cero y su bias a un valor establecido. Pensando en los Planos, las neuronas que no participan de la explicación, tienden entonces a planos horizontales en una posición determinadas por su bias.

Se definió entonces en las neuronas un parámetro b_{reg} al que tiende el bias de la Neurona si la misma no participa de la explicación de la salida. La idea es entonces que ubicando adecuadamente estos bias se logre que las neuronas se especialicen en la explicación de una parte del mapeo $CF(X)$.

Se implementaron 2 tipos de fijación de los parámetros de regularización que se pasan con el parámetro "trbr" (de Tipo Random Bias Regularización) que puede tomar los valores 0 o 1 con las definiciones siguientes:

4.4.1 trbr = 0. "Aleatorio radio 1.3 anidado."

El bias de la primer Neurona se selecciona con probabilidad uniforme en el rango -1.3 a 1.3, el de la segunda en el rango definido para la primera y 1.3 y así sucesivamente como se muestra en la ec.

$$b_{reg}(k) = U[b_{reg}(k-1), 1.3]; k = 1 .. N_{Neuronas} \text{ y } b_{reg}(0) = -1.3 \quad \text{ec.(33) Aleatorio radio anidado.}$$

4.4.2 trbr = 1. "Lineal entre 8 y 10."

Como a nivel de las neuronas está todo en por unidad, los valores entre 8 y 10 so valores "altos" y por tanto es de esperar que si los planos tienden a esos valores cuando no explican, tarde o temprano

terminen superando al conjunto de planos que si participan de la explicación y por tanto comiencen a colaborar en la explicación. A la primer neurona se le fija el valor $b_{reg}=8$ y a la última el valor $b_{reg}=10$ y a las intermedias el valor resultante de la interpolación lineal entre ambos valores.

El parámetro “trbr” debe ser especificado para la regularización de los parámetros de la red neuronal. Para la inicialización de la red neuronal desde 0 se puede fijar el parámetro “iRNpu4”. Si se especifica este parámetro en True se fijan los parámetros ro y los bias de la Capa0 como si fueran planos tangentes a una función CF monótona decreciente en todas sus variables de entrada x_i . Para la implementación se elige la función que se muestra en la ec.34, pudiendo ser cualquier otra monótona que aproxime razonablemente una función de costos futuros.

$$CF(X) = (\overline{CF}(x_1), \overline{CF}(x_2), \dots, \overline{CF}(x_n))$$

$$\overline{CF}(x) = a(x - x^*)^4 + b$$

ec.(34) Inicialización de neuronas a partir de Planos tangentes a una función CF.

Dado que tanto las entradas como las salidas se suponen normalizadas entre -1 y 1, se calculan las constantes a , b y x^* para que $\overline{CF}(-1)=1$ y $\overline{CF}(1)=-1$.

Para cada una de las N Neuronas de la Capa0 se imponen los ro y bias según las rectas tangentes a \overline{CF} calculadas en los puntos equiespaciados $x_k = \delta_x(k+1/2) - 1$, siendo $\delta_x = 2/N$ Neuronas. En la ec. 35 se muestra como se calculan los ro y bias de una neurona k .

$$ro_i^k = \frac{\partial \overline{CF}}{\partial x}(x_k) ; \forall i=1..NEntradas$$

$$bias^k = \overline{CF}(x_k) - \frac{\partial \overline{CF}}{\partial x}(x_k)x_k$$

ec.(35) Ro y bias para una neurona k a partir de planos tangentes a una función CF.

Se realizaron pruebas para comparar la convergencia de los parámetros de inicialización y regularización de la red neuronal. Para la comparación se utilizan datos de trayectorias de un determinado paso de tiempo que fueron obtenidos de una iteración avanzada del Tractorcito, por lo cual se supone que permiten generalizar los resultados obtenidos a partir de las distintas inicializaciones. Se evaluó la convergencia de un cerebro con 5 neuronas lineales y con activación Máx inicializados de manera aleatoria. Para el método “Aleatorio radio 1.3 anidado” se entrenó la red con 100 suertes o semillas distintas de modo que los resultados sean independientes de la inicialización aleatoria del cerebro. En la Fig. 16 se muestra los puntos $CF(X)$ a aproximar y en fucsia la salida de la red neuronal inicializada de manera aleatoria.

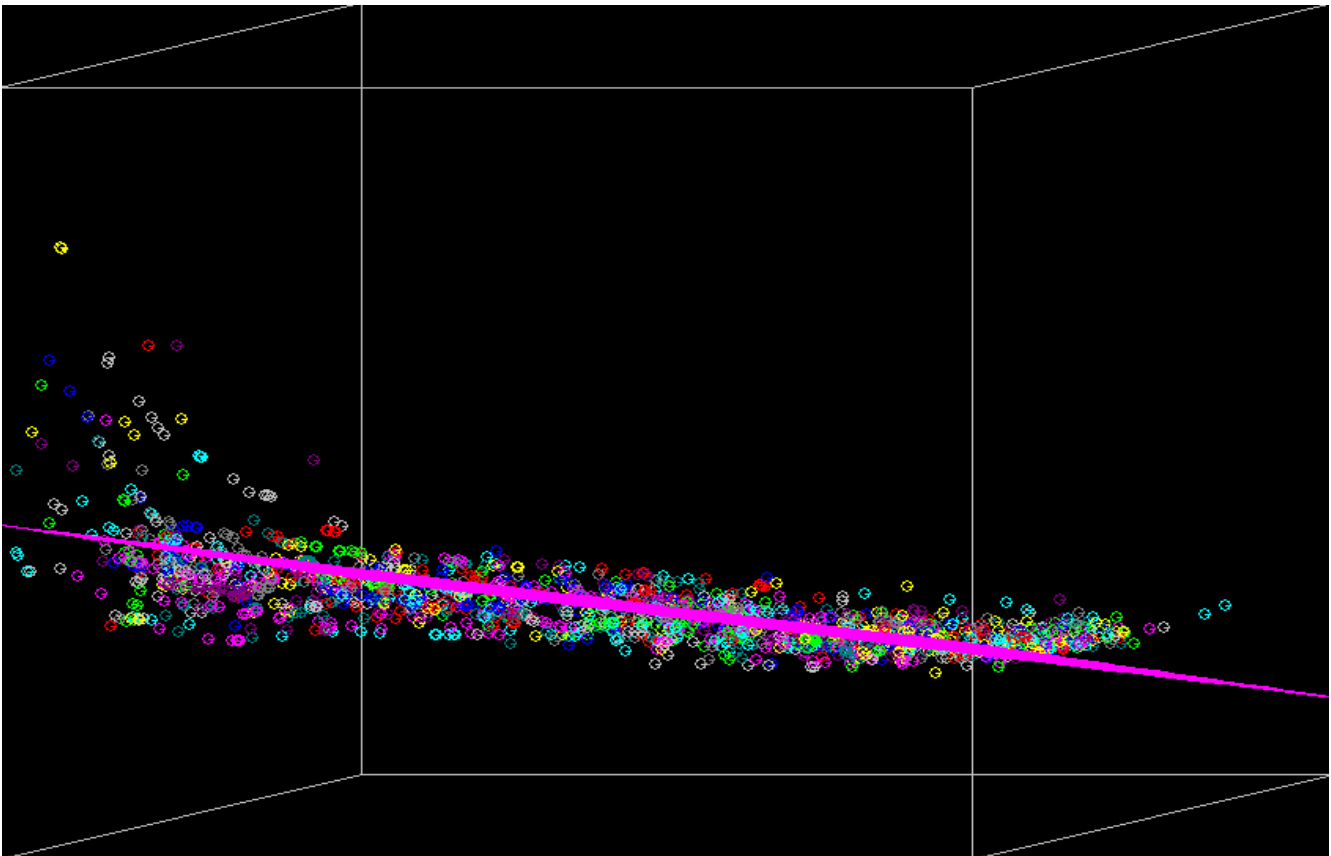


Fig. 16: $CF(X)$ a aproximar a partir de distintos métodos de inicialización.

En la Tabla 1 se comparan los resultados obtenidos a partir de cada método de inicialización de la red neuronal. Se muestra el valor medio del error cuadrático medio y el valor medio de la cantidad de iteraciones obtenidas en el total de semillas simuladas, como estimador del tiempo de convergencia de cada método.

Tabla 1: VE error RMS y VE cantidad de iteraciones según tipo de inicialización para un caso de prueba.

tipo_rnd_bias_reg	VE error RMS [p.u]	VE Cantidad iteraciones	Cantidad Semillas
Aleatorio radio 1.3 anidado	0,28	5466	100
Lineal entre 8 y 10	0,24	2157	1
Planos en u^4	0,23	496	1

Se observa que la inicialización de los parámetros mediante “Planos en u^4 ” arroja el menor tiempo de convergencia entre los 3 métodos analizados. Se observa también que el método de inicialización “Aleatorio radio 1.3 anidado” alcanza un error cuadrático medio un 17 % mayor que el resto.

Cabe destacar que si bien la inicialización de los parámetros mediante “Planos en u^4 ” arroja los mejores resultados, no permite utilizar los parámetros de la red obtenidos en la iteración anterior del Tractorcito como semilla inicial para comenzar el entrenamiento de la siguiente iteración. Por tal razón, solo se puede concluir que el método “Planos en u^4 ” presenta una mayor velocidad de convergencia si la red que se pretende entrenar parte de un estado aleatorio.

5. Enganche de las Políticas de Operación

El enganche o encadenamiento de corridas permite ir agregando detalle al modelado en el corto plazo y realizar optimizaciones en el largo plazo con menor nivel de detalle que permitan obtener resultados en un tiempo de cálculo razonable. En general se enganchan salas de corto plazo (CP) a salas de mediano plazo (MP) y estas a su vez a salas de largo plazo (LP). El tractorcito permite especificar un archivo “CF_Enganche” que contiene los valores de CF ($CF_E(Y)$) a ser considerados en el enganche entre las salas. Los valores del CF especificados en el archivo pueden venir dados por una función que devuelve un valor de CF para cada estado Y del sistema (p.ej. Red Neuronal) o de forma explícita un valor para cada discretización del espacio de estados.

El uso del Intermediario de Costo Futuro implica que se considere el costo futuro al final de la etapa como variable de control en el simplex. El costo futuro en el estado de llegada se representa a partir de un conjunto de restricciones lineales de desigualdad. Estas restricciones ingresan en el SIMPLEX como Hiperplanos, por lo que las distintas representaciones del Costo Futuro tienen que tener la capacidad de devolver un hiperplano tangente para cada estado del sistema. Para disponer de un hiperplano tangente para cada estado del sistema es necesario calcular las derivadas parciales con respecto a cada variable de estado (gradientes) y el CF para cada valor de la variables de estado.

Por otro lado, en SimSEE se permite especificar una función de mapeo $Y = F(X) = (f_1(X), f_2(X), \dots, f_M(X))$ que transforma las variables de estado $X = (x_1, x_2, \dots, x_N)$ (de la sala cuya PO se pretende calcular) a las variables de estado $Y = (y_1, x_2, \dots, y_M)$ de la sala de enganche (sala cuya PO fue calculada con anterioridad).

La derivada parcial del CF en el enganche (paso de tiempo \tilde{k}) con respecto a la variable x_i se calcula mediante la regla de la cadena (ec.36) a partir de las derivadas parciales del costo futuro del enganche ($\frac{\partial CF_E(Y)}{\partial f_j}$) y de las funciones de mapeo.

$$\frac{\partial CF(X, \tilde{k})}{\partial x_i} = \sum_{j=1}^M \frac{\partial f_j(X)}{\partial x_i} \cdot \frac{\partial CF_E(Y, \tilde{k})}{\partial f_j} \quad \forall i=1..N$$

ec.(36) Derivada parcial del costo futuro en el enganche.

Las derivadas parciales del costo futuro en el enganche se corresponden con los elementos del vector gradiente al plano tangente a CF_E en $Y(X)$.

El valor del CF en el enganche se calcula de acuerdo a la ec. 37.

$$CF(X, \tilde{k}) = CF_E(Y(X, \tilde{k}))$$

ec.(37) Costo Futuro en el enganche evaluado en el estado X.

A partir del gradiente y valor del CF en el enganche se tiene una representación del hiperplano que se agrega al Simplex durante la optimización del último paso de tiempo. En el caso en que no se disponga de un archivo que permita el enganche de las salas, se imponen planos nulos al final del paso del último paso de optimización.

En los modos de ejecución 1 a 5 la información del CF en el enganche llega al paso k ($k < k_{Enganche}$) al cabo de $\lceil (k - k_{Enganche}) / learningRate \rceil$ iteraciones del Tractorcito. Para acelerar la propagación de la información del enganche hacia los pasos de tiempo anteriores se puede fijar el parámetro "icfae" en True. Si se activa el parámetro anterior, en el paso de iteración 1 del tractorcito y para todos los pasos de tiempo, se agregan al Simplex los hiperplanos que se obtendrían en el Enganche para cada estado de fin del paso. Solo para la primer iteración del Tractorcito se fija $learningRate = 1$ de modo que el sistema logre asimilar el 100 % de la información del enganche en una sola iteración. A partir del segundo paso de iteración se fija el learning-rate de acuerdo al especificado por el usuario como parámetro y se calcula el CF al final de cada paso según lo obtenido en la iteración anterior, de acuerdo al flujo normal del programa.

6. Parsimonia Temporal

Es de esperar que la Política de Operación presente cierta continuidad (ausencia de saltos abruptos) y que presente también un comportamiento estacional anual. También es esperable que debido a la expansión del sistema y a su cambio de estructura, se experimenten cambios en el largo plazo.

Los ajustes de los CF's sin Parsimonia se realiza paso a paso de tiempo calibrando los parámetros en base a las muestras disponibles del paso resolviendo el problema de la ec.38

$$\min_{\theta_k} \sum_i (M(X_{ki}, \theta_k) - CF_{ki})^2 \quad \text{ec.(38) Calibración paso a paso.}$$

Si se considera que existe una Parsimonia en los pasos de tiempo se puede determinar una parsimonia vinculando los θ_k para los diferentes pasos de tiempo k por la ec.39.

A los efectos de la notación, la representación nos permite pasar del conjunto de parámetros $\{\theta_1, \dots, \theta_{nPasosT}\}$ al conjunto de parámetros $\alpha = \{\alpha_1, \dots, \alpha_{npp}\} = \{a_0 \dots a_{np}; A_q \dots\}$ donde, para que tenga sentido la imposición de la parsimonia, se debe cumplir $npp \ll \text{Dim}(\theta) \times nPasosT$.

Sea θ_k^j la componente j del vector θ_k y α_h^j la componente j del vector α_h .

Supongamos que la parsimonia se representa por un conjunto de funciones que permiten el cálculo de θ_k^j en función del conjunto de parámetros $\{\alpha_{1j}, \dots, \alpha_{npp}^j\}$ como se muestra en la ec.39.

$$\theta_k^j = P(\alpha_1^j, \dots, \alpha_{npp}^j, k) \quad \text{ec.(39).}$$

En particular supondremos que la parsimonia se impone como la suma de una serie truncada de potencias y una serie truncada de Fourier como se muestra en la ec.40.

$$\theta_k^j = a_0^j + \sum_{h=1}^{np} a_h^j k^h + \sum_{h=1}^{na} (A_q^j \cos(w_0 h k) + B_q^j \sin(w_0 h k)) \quad \text{ec.(40).}$$

Al imponer la parsimonia, el problema a resolver, en lugar de el de la ec.38 sería el de la ec.41

$$\min_{\alpha_1, \dots, \alpha_{npp}} \frac{1}{2} \sum_k \sum_i (M(X_{ki}, P(\alpha_1, \dots, \alpha_{npp}, k)) - CF_{ki})^2 \quad \text{ec.(41) Calibración Global con Parsimonia.}$$

Para calcular el gradiente del error de aproximación respecto del parámetro α_i basta con aplicar la regla de la cadena como se muestra en la ec.42.

$$\sum_k \sum_i e_{ki} \frac{\partial e_{ki}}{\partial \alpha_h} = \sum_k \sum_i \sum_{j=1}^{Dim(\theta)} e_{ki} \frac{\partial e_{ki}}{\partial \theta_{kj}} \frac{\partial}{\partial \alpha_h} P_j(\alpha_1, \dots, \alpha_{npp}, k) \quad \text{ec.(42).}$$

Si utilizamos la aproximación del CF a partir de Redes Neuronales, se tiene que los Cerebros saben calcular las derivadas $e_{ki} \frac{\partial e_{ki}}{\partial \theta_{kj}}$ utilizando el método de Backpropagation para cada muestra (X_{ki}, CF_{ki}) .

Observar que $\frac{\partial}{\partial \alpha_h} P_j(\alpha_1, \dots, \alpha_{npp}, k)$ no depende del valor de las muestras.

Como las muestras con las que se entrena la Red Neuronal presentan un valor esperado distinto de cero, que a efectos de determinar la parsimonia de los datos no aporta ninguna información, por el contrario, puede incorporar error en la aproximación, se optó por independizarse del valor esperado. Se le resta el VE a todas las muestras con la que se entrena la RN.

También se implemento un “Colchón”, que lo que intenta es construir una parsimonia sin las muestras de los bordes, para que no interfiera en el entrenamiento.

A continuación se muestran algunos resultados aplicando la parsimonia en una sala de Largo plazo del sistema Uruguayo (del 1/1/2019 al 31/12/2046, sala con paso semanal), dicha sala tiene 2 variables de estado y se construyó el CF a partir de 1 Hiperplanos por paso de tiempo.

Se aplicó la parsimonia en los coeficientes de los Cerebros con distintas configuraciones de parsimonia y utilizando un colchón al final del periodo. También se muestran los resultados comparándolos con el método convencional para el CF implementado en SimSEE y también comparándolo con el CF de RN sin parsimonia.

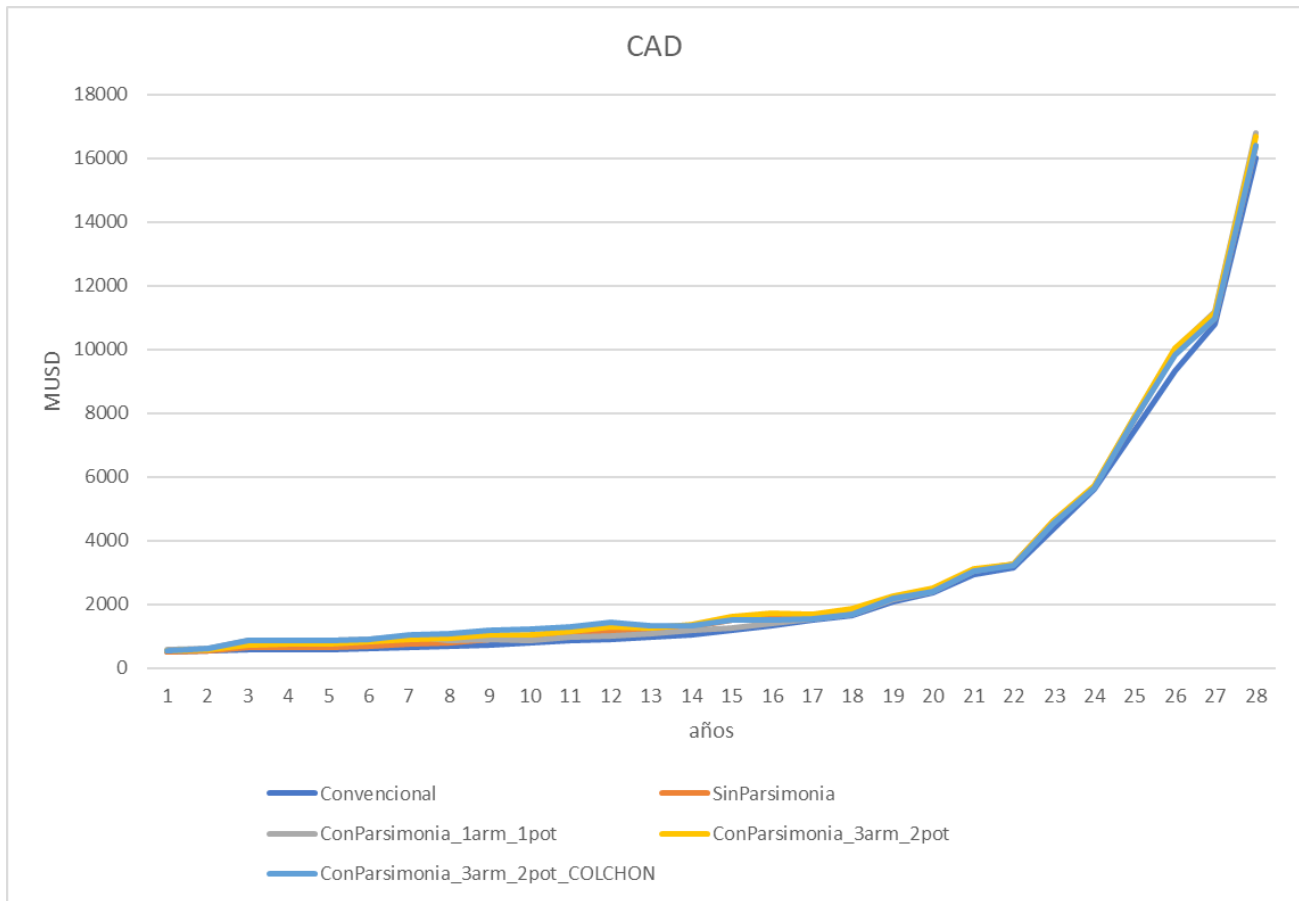


Figura 5: Costo de abastecimiento de la demanda para los distintos métodos de optimización.

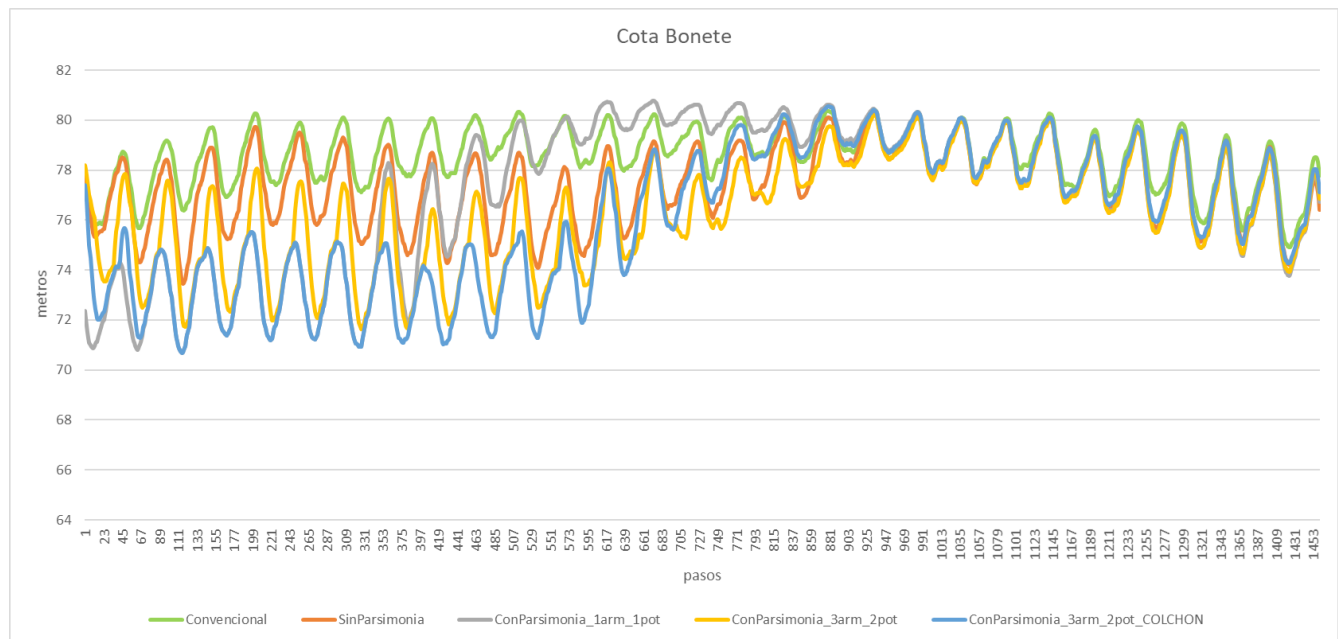


Figura 6: Cota de Bonete para los distintos métodos de optimización.

En la siguiente tabla se muestra el error relativo con respecto al método convencional.

SinParsimonia	Con Parsimonia		
	1arm_1pot	3arm_2pot	3arm_2pot_COLCHON
2%	5%	4%	2%

A partir de las figuras 6, 5 y de la tabla se puede concluir que el CF construido a partir del método convencional del SimSEE sigue siendo el mas óptimo por el momento, aunque la diferencia no es demasiado significativa. Se observa que la cota de la represa fue aproximadamente igual sobre el final del horizonte de simulación, lo que significa que operan la represa de igual manera. Al inicio el modo convencional es mas conservador y guarda agua, mientras que el caso con parsmonia lleva la cota a valores mas bajos lo que resulta en mayores costos de operación.

Para ir un poco mas al detalle, observemos los coeficientes de los planos de las trayectorias para los distintos casos.

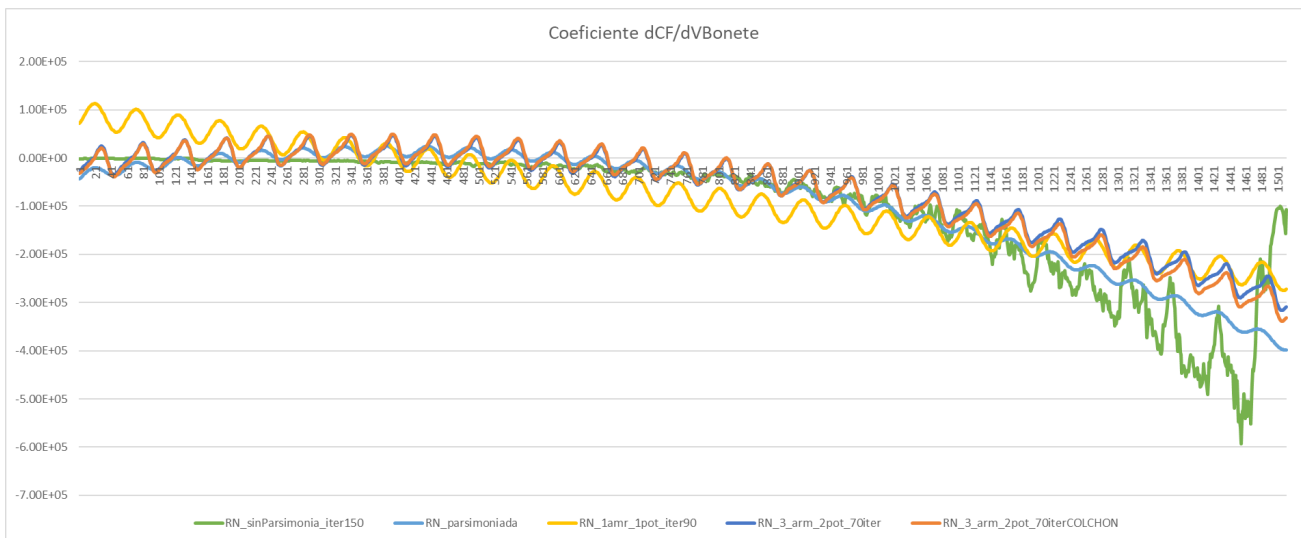


Figura 7: Coeficiente del plano correspondiente al volumnes de Bonete

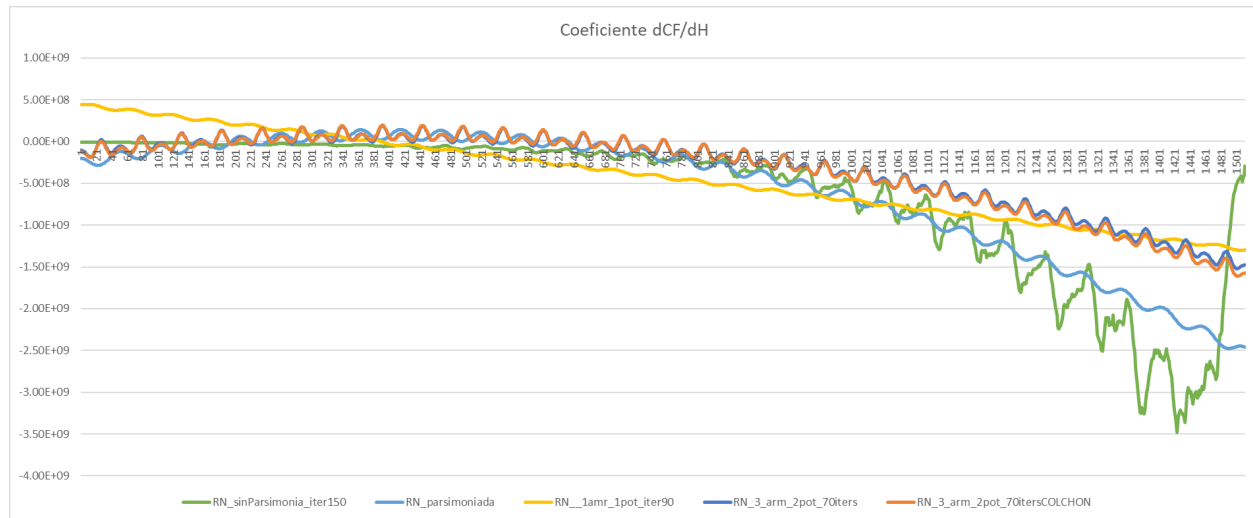


Figura 8: Coeficiente del plano correspondiente al estado hidrológico

Observar que la información que se “pierde” por aproximar por una curva parsimonioso no es mucha, lo que se condice con los resultados obtenidos para el CAD y la evolución de la cota de Bonete.

Es importante destacar la cantidad de variables que disminuye al usar las parsimonias. En el caso sin parsimonia tenemos la cantidad de parámetros de la Red Neuronal por la cantidad de pasos de tiempo (1513), en cambio aplicando parsimonia tenemos la cantidad de parámetros de la Red Neuronal por la cantidad de coeficiente de la parsimonia ($2 \cdot n_{\text{Armonicos}} + n_{\text{Potencias}} + 1$)

7. Comparación con Política de Operación Clásica.

7.1. Consideraciones del efecto de la cantidad de discretizaciones del volumen embalsado y de la imposición de valor del agua positivo sobre la Política de Operación del lago de Rincón de Bonete obtenida con el algoritmo clásico.

En este capítulo se documentan dos aspectos de la Optimización Clásica que es necesario tener en cuenta al realizar comparaciones con el nuevo algoritmo. Se recuerda que dicho algoritmo consiste en resolver la recursión de Bellman con linealización de la ecuación de evolución de estado y de la función de Costo Futuro en el entorno del estado del sistema.

En particular, se espera que el tema de la estabilidad numérica de la nueva metodología sea mejor que la clásica por la sustitución del modelo lineal de CF por un conjunto de hiperplanos de aproximación que permiten modelar la concavidad de la función. En igual sentido, la condición impuesta en la resolución clásica respecto del valor del agua positivo debiera también ser resuelta

automáticamente al poder representar que un valor negativo implica la apertura del vertedero (por la resolución lineal del problema) sin visualizar el costo que ello implica en los siguientes pasos.

7.1.1 Resumen ejecutivo.

Hasta la fecha se utiliza una discretización de 10 cotas entre 70 m y 81 m del lago de Rincón de Bonete para el cálculo de la Política de Operación y se impone tanto en Optimización como en Simulación que el valor del agua sea positivo. Esta imposición se realiza simplemente imponiendo su valor a 0 (cero) en caso de que menos la derivada del Costo Futuro respecto del Volumen Embalsado resulte negativa.

Se probó aumentar a 100 niveles la discretización y se confirma que esto no significa una mejora sino que la PO obtenida implica un sobre costo del 1% respecto del caso anterior. Se probó no imponer la restricción de valor de agua positivo y se confirma que esto no significa una mejora sino que la PO obtenida implica un sobre costo de 3% respecto del caso anterior.

7.1.2 Introducción.

La Plataforma SimSEE permite la simulación de la operación del sistema de generación de energía eléctrica para un Política de Operación (PO) dada por intermedio de la función de Costo Futuro (CF). El mismo simulador, permite obtener una PO cuasi-óptima en base a resolver la recursión de Bellman. Para ello, se realiza una discretización del espacio de estado en un conjunto de puntos $\{X_i\}$ y se barre dicha discretización calculando la función $CF(X_i, k)$ resolviendo el problema de optimización que se expresa en la ec.43.

$$CF(X_i, k) = \left\langle \min_u (ce(X_i, u, r, k) + q CF(\tilde{X}, k+1)) \right\rangle_r \quad \text{ec.(43)}$$

$$@ \tilde{X} = f(X_i, u, r, k)$$

En donde X es el vector de estado, u es el vector de control, r es el vector de entradas no controlables (aleatorias y exógenas), $q \in (0, 1)$ es un factor de actualización y k es un ordinal que identifica la etapa o paso de tiempo. La función $\tilde{X} = f(X_i, u, r, k)$ es conocida como ecuación de evolución de estado, o restricción dinámica y es la que captura cuál será la evolución del estado del sistema a lo largo del paso de tiempo k partiendo del estado X_i si es sometido a las entradas u, r . La función $ce(X, r, u, k)$ es una función dada que permite calcular el costo de la etapa (gastos en combustibles + importaciones - exportaciones + costos de falla).

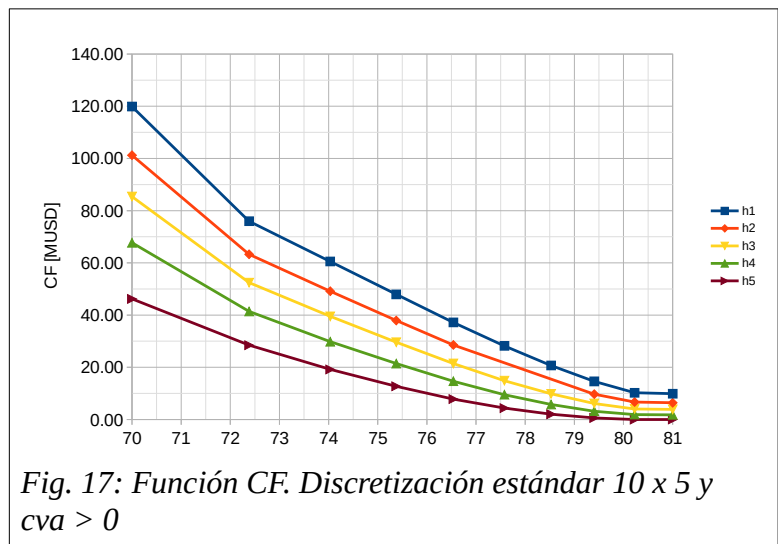
La ec.43 (recursión de Bellman) puede verse como un algoritmo recursivo que transforma el vector $CF(X_i, k+1)$ en el vector $CF(X_i, k)$.

Para reducir la ec.43 a la resolución de un problema lineal, se suele linealizar $CF(\tilde{X}, k+1)$ y la ecuación de evolución de estado.

7.1.3 Efecto de la cantidad de discretizaciones.

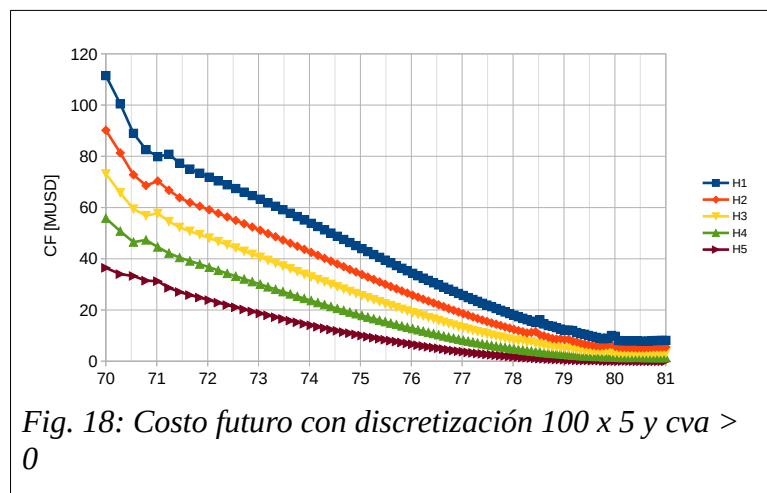
En el trabajo: R. Chaer and P. Monzon, "Stability conditions for a stochastic dynamic optimizer for optimal dispatch policies in power systems with hydroelectrical generation," 2008 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America, Bogota, 2008, pp. 1-5, doi: 10.1109/TDC-LA.2008.4641717." se muestra que el algoritmo de la ec.43 cuando se aplica sobre una discretización del espacio de estado y cuando se aproxima la derivada del costo futuro respecto a la variación del estado como la variación del costo en la grilla de cálculo dividido por el desplazamiento del estado entre los puntos, presenta problemas de estabilidad numérica si el movimiento del estado del sistema en un paso de tiempo supere al ancho de discretización del estado en la grilla. Esto implica que si se sube la discretización del estado, manteniendo el paso de tiempo fijo, entonces se llegará a que el movimiento posible del estado en un paso de tiempo supere la discretización y por tanto el algoritmo presentará problemas de estabilidad numérica.

En el caso particular de las simulaciones de paso semanal del SIN, se utilizan 10 puntos de



discretización del volumen embalsado en Bonete.

La Fig.17 muestra un cuadro de la función de CF para una sala de paso semanal clásica con 10 puntos de discretización del volumen del lago y 5 puntos de discretización de la variable de estado hidrológico.

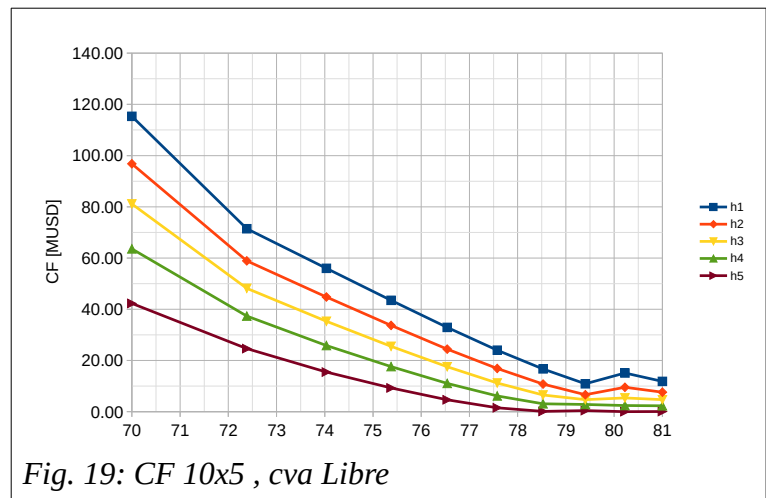


La Fig.18 muestra el mismo paso de tiempo de la función CF habiendo subido la cantidad de discretizaciones del lago de Bonete de 10 a 100. Como se puede apreciar, aparecen cambios de monotonía.

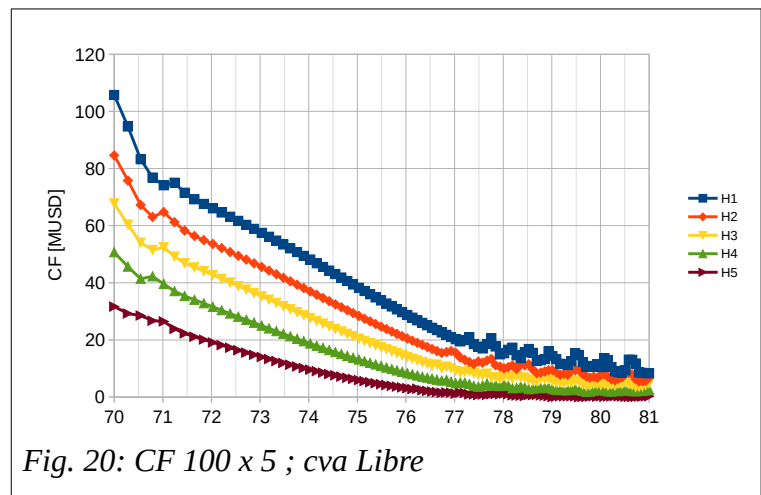
En ambos casos se impuso que el valor del agua sea positivo. Para ello, si al estimar menos la derivada de CF respecto del volumen embalsado (el valor del agua) da negativo, se considera 0 (Cero).

7.1.4 Efecto de imponer $cva > 0$.

La Fig.19 muestra el equivalente a la Fig.17 (discretización 10 x 5) pero sin imponer $cva > 0$. Como se puede apreciar, al liberar dicha imposición aparece un cambio de concavidad para los estados $h1$ y $h2$ entre los 79.5 y 80 m.

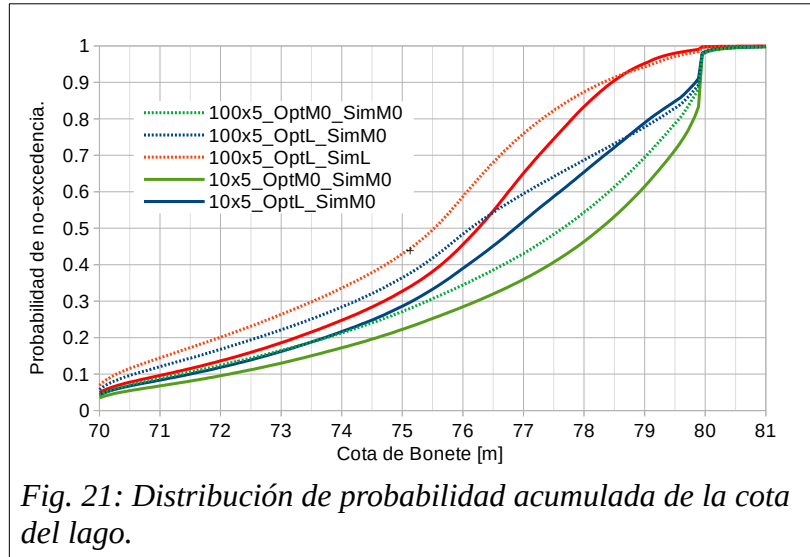


La Fig.20 es la correspondiente a discretización 100 x 5 sin imponre $cva > 0$. Como se puede apreciar es ruidosa.



7.1.5 Estadística de la cota del lago.

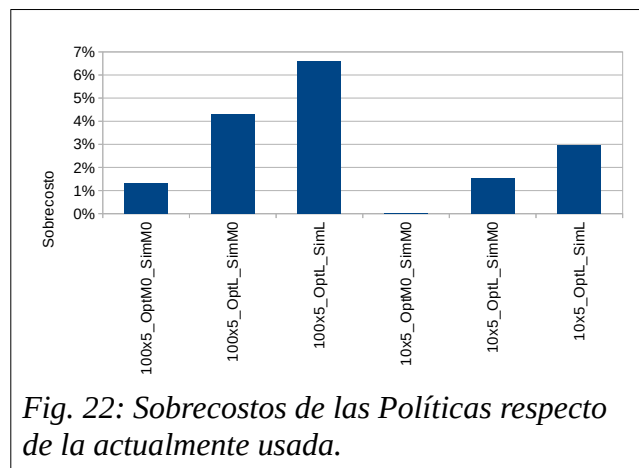
La Fig.21 muestra la función de probabilidad acumulada de la cota del lago. Como se puede observar la metodología actualmente en uso es la que mantiene la cota más alta. Esto de por si no es ni bueno ni malo, por lo cual a continuación se comparan los costos esperados de la operación futura.



En estas simulaciones no se utilizó ningún control de cota para preservar el fondo del lago. Mirando lo que pasa por debajo de los 72 m queda claro que la política actual asegura que la probabilidad de estar por debajo de esa cota es 10% mientras que las demás tienen probabilidades mayores llegando a 20% en caso de la 100x5 sin imponer $cva > 0$ ni en optimización ni en simulación.

7.1.6 Comparación de los Costos Futuros

La Fig.22 muestra el sobre costo de las diferentes políticas en valor esperado respecto de la obtenida con discretización 10x5 e imponiendo $cva > 0$ tanto en Optimización como en Simulación (método usado actualmente). Esto muestra que por lo menos en la Sala de Largo Plazo (paso semanal,



con variables dos variables de estado: Volumen de Bonete y Estado Hidrológico) la discretización

actual de 10 discretizaciones del lago es mejor que pasar a 100 y que es mejor imponer $cva > 0$ que no imponerla.

Los horizontes de optimización y simulación utilizados en todas las corridas son los que se muestran en la Fig.23. Los costos para el cálculo de los sobrecostos de la Fig.22 corresponden al valor presente al inicio de la simulación del costo directo del paso actualizado al 5%.

Horizonte de tiempo

Fecha de inicio: 1/1/2018 Fecha de fin: 1/1/2049 (18/8/2020 08:48:32) Huso horario: -3

Optimización: 1/1/2018 31/12/2046

Simulación: 1/1/2018 31/12/2046 Horizonte de guarda para simulación: 1/1/2020

Fig. 23: Horizontes de Optimización y Simulación.

7.1.7 Intermediario de Costo Futuro

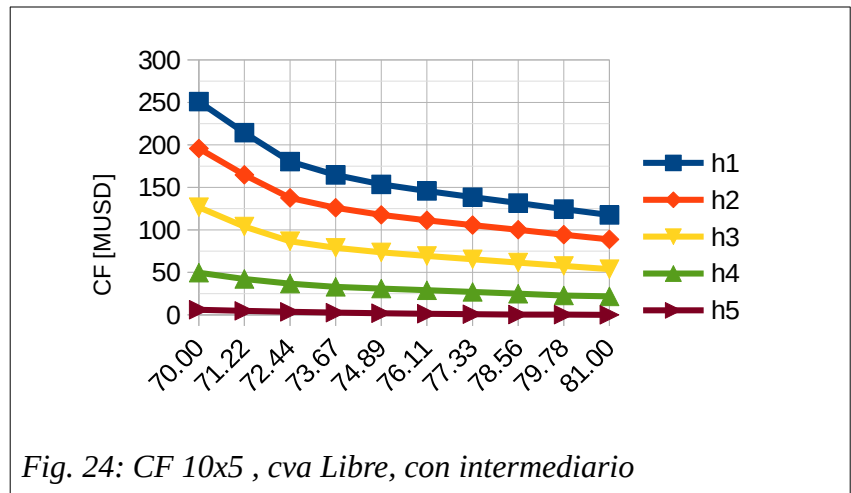
Se probó utilizar varios hiperplanos para aproximar la PO en cada punto agregando como restricción al simplex el desarrollo de Taylor de orden 1 de la función de costo en un entorno del punto de operación. Por ser el problema convexo, se cumple que la función de costo no solo se puede aproximar por este desarrollo sino que además es mayorada por él según se indica en la ec. (44)

$$CF(X_s) \geq CF(X_0) + \nabla CF(X_0) \cdot (X_s - X_0) \quad (44)$$

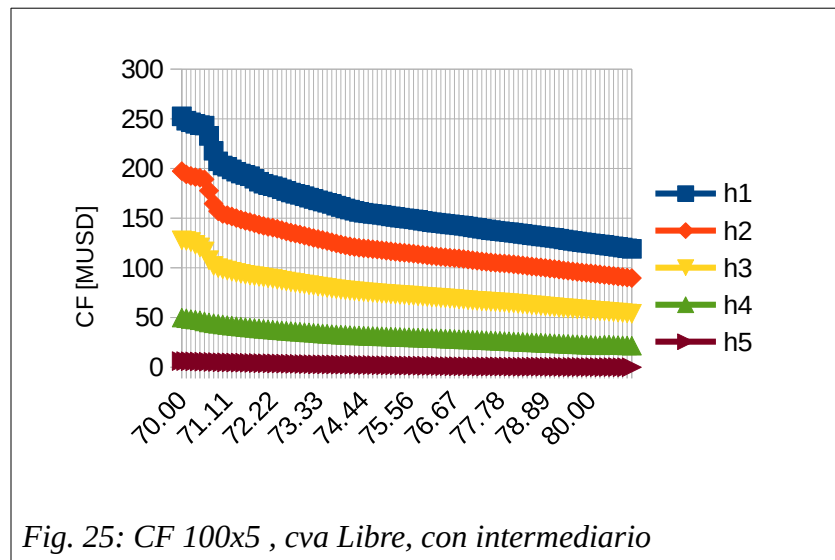
Donde X_s es el estado solución del paso y X_0 es el estado proyectado al inicio del paso. En el caso de que el paso se resuelva en una sola iteración, resolver este problema es equivalente a resolver el problema original. Pero con esta nueva forma de plantear el problema, en cada iteración de un mismo paso de tiempo, X_0 es reemplazado por X_s de la iteración anterior y se obtiene un nuevo hiperplano. Con esto se pretende lograr *facetar* la función de costo y obtener una PO más precisa. Esto también tiene la ventaja de eliminar oscilaciones en la iteraciones ya que la información es agregada y no reemplazada.

Para esta implementación se agrega como variable al simplex el costo futuro del estado al final del paso $CF(X_s)$, una restricción como la de la ec. (44) por cada iteración dentro del paso y la función de costo como el costo directo del paso más el costo futuro del estado de llegada afectado por el factor de actualización q .

Con esta modalidad, se volvieron a calcular las funciones de costo en las mismas condiciones que los puntos anteriores.



La Fig. 24 es análoga a la Fig. 19, la diferencia es que en esta se utiliza el intermediario de costo futuro. Se puede observar que la dispersión de los costos a diferentes valores de la variable h es mucho mayor.



Lo mismo pasa con la Fig. 25 y la Fig. 20. Otra cosa que puede observarse es que desaparece el problema de estabilidad numérica que sucedía en las cotas más altas. Si bien parece que la PO con intermediario es más costosa, los resultados de costos incurridos en simulación son similares al caso tradicional. Esto puede observarse en los resultados de la Tabla 2.

	Costos [MUSD]							
	Con intermediario				Sin intermediario			
	Con cva > 0		Sin cva > 0		Con cva > 0		Sin cva > 0	
	10 disc	100 disc	10 disc	100 disc	10 disc	100 disc	10 disc	100 disc
Costo esperado	8060	8084	8102	8480	8122	8199	8405	8765
Costo con riesgo 5% de ser excedido.	8760	8811	8825	9259	8938	9066	9282	9712
Costo CVaR(5%).	9000	9038	9044	9527	9230	9356	9581	10042

Tabla 2: Comparativo de los costos de simulación utilizando intermediario de costo futuro

Según esta tabla se puede ver que el mejor resultado se obtuvo utilizando el intermediario de costo futuro y la restricción de costo variable positivo.

8. Resultados sobre el Sistema dinámico simple.

Se realizaron pruebas con distintas configuraciones de redes neuronales para aproximar la función de costo futuro sobre el sistema dinámico de prueba. Se presenta a continuación los resultados comparativos entre la optimización clásica por recursión de Bellman y la obtenida con el nuevo algoritmo de optimización forward utilizando una red neuronal no secuencial de 3 neuronas con función de activación TANH y con una neurona con función MAX en la capa final. Las funciones de costo futuro obtenidas son el resultado de 250 iteraciones del nuevo algoritmo. Las comparaciones se realizaron sobre un conjunto de casos distintos, creados utilizando 4 cotas iniciales del embalse de Bonete para la simulación y contra 3 números de crónicas de optimización en la optimización clásica.

Para obtener las funciones de costo futuro se realizó la optimización clásica y la optimización del nuevo algoritmo sobre el conjunto de casos definidos. Una vez obtenida la función de costo futuro, la misma se utilizó para realizar una simulación de 1000 crónicas para cada caso correspondiente.

8.1. Configuración del sistema de prueba

Horizonte de optimización: 01/01/2018 – 01/01/2020.

Horizonte de simulación: 01/01/2018 – 01/01/2020.

Horizonte de guarda para simulación: 01/06/2018.

Cantidad de crónicas de optimización: Se realizaron pruebas con 5, 25 y 500 crónicas.

Cantidad de crónicas de simulación: 1000.

Discretización del embalse de Bonete: 50.

Discretización del la variable de estado hidrológico H: 50.

Costos hidro positivos activado en la optimización clásica y en la simulación.

8.2. Resultados

Se presentan en las tablas a continuación los resultados obtenidos para varias cotas de inicio del embalse de Bonete. La magnitud utilizada para comparar el rendimiento del algoritmo es el costo directo del paso de tiempo acumulado en todo el horizonte de simulación, afectado por su correspondiente factor de descuento y promediado sobre las 1000 crónicas de simulación. Se muestran 3 tablas, donde cada una se corresponde con un número determinado de crónicas de optimización para el caso que utiliza la optimización clásica.

COTA BONETE INICIAL (m)	CDP ACUMULADO (MUSD)		DIF %
	CLÁSICO 5 CRON OPT	RN 3 TAN, 1 MAX	
72	558.37	511.80	91.7%
75	547.88	500.47	91.3%
78	534.30	494.49	92.5%
80	521.79	491.12	94.1%

Tabla 3: Comparativo nuevo algoritmo vs clásico con 5 crónicas de optimización.

COTA BONETE INICIAL (m)	CDP ACUMULADO (MUSD)		DIF %
	CLÁSICO 25 CRON OPT	RN 3 TAN, 1 MAX	
72	515.74	511.80	99.2%
75	506.28	500.47	98.9%
78	494.43	494.49	100.0%
80	486.31	491.12	101.0%

Tabla 4: Comparativo nuevo algoritmo vs clásico con 25 crónicas de optimización.

COTA BONETE INICIAL (m)	CDP ACUMULADO (MUSD)		DIF %
	CLÁSICO 500 CRON OPT	RN 3 TAN, 1 MAX	
72	511.16	511.80	100.1%
75	501.38	500.47	99.8%
78	490.69	494.49	100.8%
80	483.17	491.12	101.6%

Tabla 5: Comparativo nuevo algoritmo vs clásico con 500 crónicas de optimización.

A partir de los resultados anteriores se puede observar que el nuevo algoritmo produce un descenso en el costo directo acumulado de entre un 6% a un 8% aproximadamente, en el caso donde la optimización clásica se realiza con 5 crónicas de optimización.

Para los otros casos, con 25 y con 50 crónicas de simulación, el costo directo del paso acumulado no presenta una reducción significativa en comparación con la optimización clásica, quedando en magnitudes muy similares para el conjunto de cotas de inicio utilizadas.

8.3. Resultados en algoritmo clásico

Se muestra a continuación un conjunto de pruebas sobre el algoritmo de optimización clásica sobre el sistema dinámico de prueba. Se realizan diferentes configuraciones de discretización sobre las variables de estado (10 y 50 discretizaciones) y se prueba la opción de imponer los costos hidro positivos en la etapa de optimización y en la de simulación. Nuevamente la magnitud utilizada para evaluar el rendimiento es el costo directo del paso acumulado en todo el horizonte de simulación.

A partir de los resultados se puede concluir que los menores costos se obtienen cuando la opción de imponer costos hidro positivos está activada en optimización y en simulación. En dicho caso no hay una diferencia significativa en el costo obtenido con respecto a la cantidad de discretizaciones de las variables de estado.

Por otro lado, los costos más altos se obtienen con la opción de costos hidro positivos desactivada en optimización y en simulación, y con una discretización de 50 puntos por variable de estado. Estos costos son aproximadamente entre un 50% a un 60% más caros que los correspondientes al mejor caso.

SIMSEE CLÁSICO CDP ACUMULADO PROMEDIO DE 1000 CRÓNICAS DE SIMULACIÓN SOBRE 500 CRÓNICAS DE OPTIMIZACIÓN (MUSD)				
cva > 0 en OPTIMIZACIÓN				
COTA BONETE INICIAL (m)	dis 50*50 cva activo en SIMULACIÓN	dis 10*10 cva activo en SIMULACIÓN	dis 50*50 cva inactivo en SIMULACIÓN	dis 10*10 cva inactivo en SIMULACIÓN
72	511.16	511.59	539.53	554.98
75	501.38	499.23	529.66	544.50
78	490.69	490.45	520.41	537.11
80	483.17	485.20	518.61	534.55
SIMSEE CLÁSICO CDP ACUMULADO PROMEDIO DE 1000 CRÓNICAS DE SIMULACIÓN SOBRE 500 CRÓNICAS DE OPTIMIZACIÓN (MUSD)				
cva libre en OPTIMIZACIÓN				
COTA BONETE INICIAL (m)	dis 50*50 cva activo en SIMULACIÓN	dis 10*10 cva activo en SIMULACIÓN	dis 50*50 cva inactivo en SIMULACIÓN	dis 10*10 cva inactivo en SIMULACIÓN
72	547.50	552.32	763.05	723.76
75	541.77	545.22	762.79	720.90
78	533.18	537.45	765.38	726.15
80	524.76	529.90	765.38	726.15

9. Convergencia de la aproximación lineal de orden 1 para el caso de la Sala de Largo Plazo (Planificación de Inversiones).

En esta sección se analizan los resultados de utilizar la iteración propuesta para aprendizaje de una aproximación lineal de orden 1 (un hiperplano) para la Sala de Largo Plazo (LP) del sistema de generación eléctrica de Uruguay.

La referida Sala LP tiene un horizonte de simulación del 1/1/2018 al 31/12/2046, es de paso de simulación semanal (168 horas) dividido en 4 Postes (o bandas horarias) de duración 5, 30, 91 y 42 horas del Poste 1 al 4 respectivamente. La Sala utiliza el sistema de “postizado dinámico” del paso de tiempo en base a la Demanda Neta implementado por SimSEE. El Poste 1 corresponde al Poste de mayor requerimiento de Potencia (mayor Demanda Neta) y el poste 4 al de menor requerimiento de Potencia.

Las variables de estado sobre las que se debe generar una Política de Operación son el Volumen del embalse de Rincón de Bonete (V_{Bon}) y la variable de Estado Hidrológico (H).

La variable de estado hidrológico, está definido dentro de la fuente sintetizador que genera las series de caudales de aportes propios a las centrales de Bonete (Q_{Bon}), Palmar (Q_{Palmar}) y Salto Grande (Q_{SG}), de la anomalía de la temperatura superficial del Océano Pacífico (i_{N34}) y los costos marginales operativos, en las bandas horarias Pesada (cmo_1) y Leve (cmo_3) de la región sur de Brasil. La ec.45 muestra la definición de la variable H como una combinación lineal de las variables del CEGH (ponderando por cero cmo_1 y cmo_1) en el espacio gaussiano.

$$H = 0.25 g Q_{Bon} + 0.05 g Q_{Pal} + 0.2 g Q_{SG} + 0.5 g i_{N34} \quad \text{ec.(45) Definición de la reducción del espacio de estado en el CEGH de Aportes.}$$

Con esta definición del espacio de estado (X) la Política de Operación tendría la forma que se muestra en la ec.46.

$$u = PO(V_{Bon}, H) \quad \text{ec.(46) Política de Operación para la Sala de Largo Plazo utilizada.}$$

En la implementación de SimSEE para la resolución de un paso de tiempo, la PO queda determinada por la resolución de un minimizar costo de transición (o costo de etapa) más el Costo Futuro del estado en que queda el sistema al final del paso. Con esta implementación, la información necesaria para el cálculo de la PO está contenida en la función $CF(X)$. El propósito de esta sección es mostrar cuál es el resultado de la operación del SIN cuando se utiliza para aproximar la función $CF(X)$ una aproximación lineal de orden 1 como se muestra en la ec.47.

$$CF(V_{Bon}, H) = \frac{\partial CF}{\partial V_{Bon}} V_{Bon} + \frac{\partial CF}{\partial H} H + C_0 \quad \text{ec.(47) Aproximación lineal de orden 1.}$$

La Fig.26 muestra la evolución de las primeras 25 iteraciones de los coeficientes del hiperplano de aproximación (ec.47) durante el horizonte de la simulación. El punto de partida son las líneas horizontales comenzando con azul puro y cada curva (cambiando gradualmente de azul puro a rojo puro) muestra una iteración. La curva en rojo puro con trazo más grueso corresponde a la última iteración.

Estas primeras 25 iteraciones son realizadas con coeficiente de aprendizaje 1 (uno) y realizando la actualización de CF observando solamente el costo de cada etapa y la estimación del CF en el estado de llegada de la misma etapa. En el algoritmo esto corresponde a PesosForward = [q].

Como se puede apreciar, los coeficientes se van “formando” mostrando rápidamente la estacionalidad anual.

Se observa que C_0 muestra una tendencia creciente. Siendo esta figura construida en base a las primeras 25 iteraciones (con un vector de PesosForward de dimensión 1), el valor de C_0 refleja la suma de 25 valores de costo directo del paso (de diferentes iteraciones) y el que esa suma sea creciente en el horizonte de tiempo seguramente se deba a que al ir creciendo la Demanda el costo directo del paso también va creciendo.

Es notoria la estacionalidad anual de las tres variables graficadas. También se observa que las componentes del gradiente $\nabla_x CF = \left[\frac{\partial CF}{\partial V_{Bon}}, \frac{\partial CF}{\partial H} \right]^T$ no presenta la misma tendencia creciente que el valor C_0 .

Llama la atención la diferencia del comportamiento de las tres variables en los primeros 3 años respecto del resto. Para tratar de tener una explicación del comportamiento se muestra en la Fig.27 superpuesto al gráfico de la Fig.26 el Plan de Expansión de la generación. Como se puede observar, lo que podría explicar el comportamiento diferente en los primeros años es la entrada de la generación de UPM2 (segunda planta de celulosa de UPM). Como la entrada en operación de dicha planta impone una restricción al manejo del lago de Rincón de Bonete para asegurar un erogado mínimo de 65 m3/s (valor usado en la simulación).

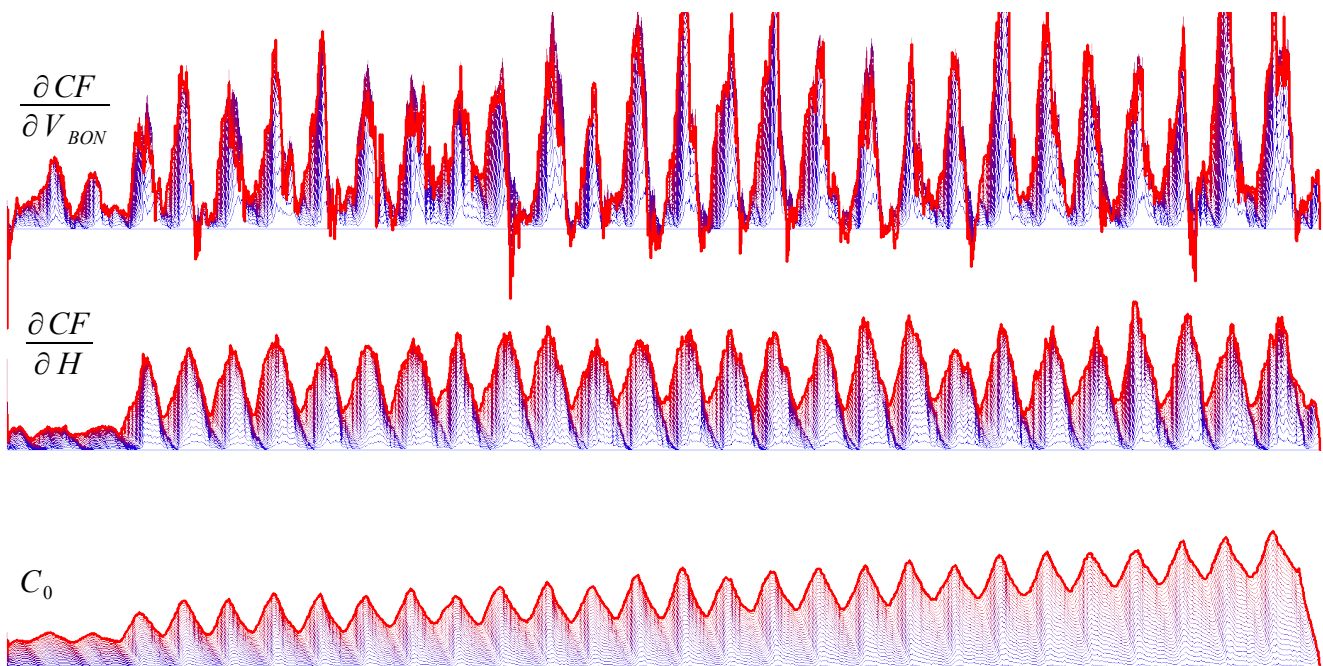


Fig. 26: Convergencia de las primeras 25 iteraciones en Sala de Largo Plazo.

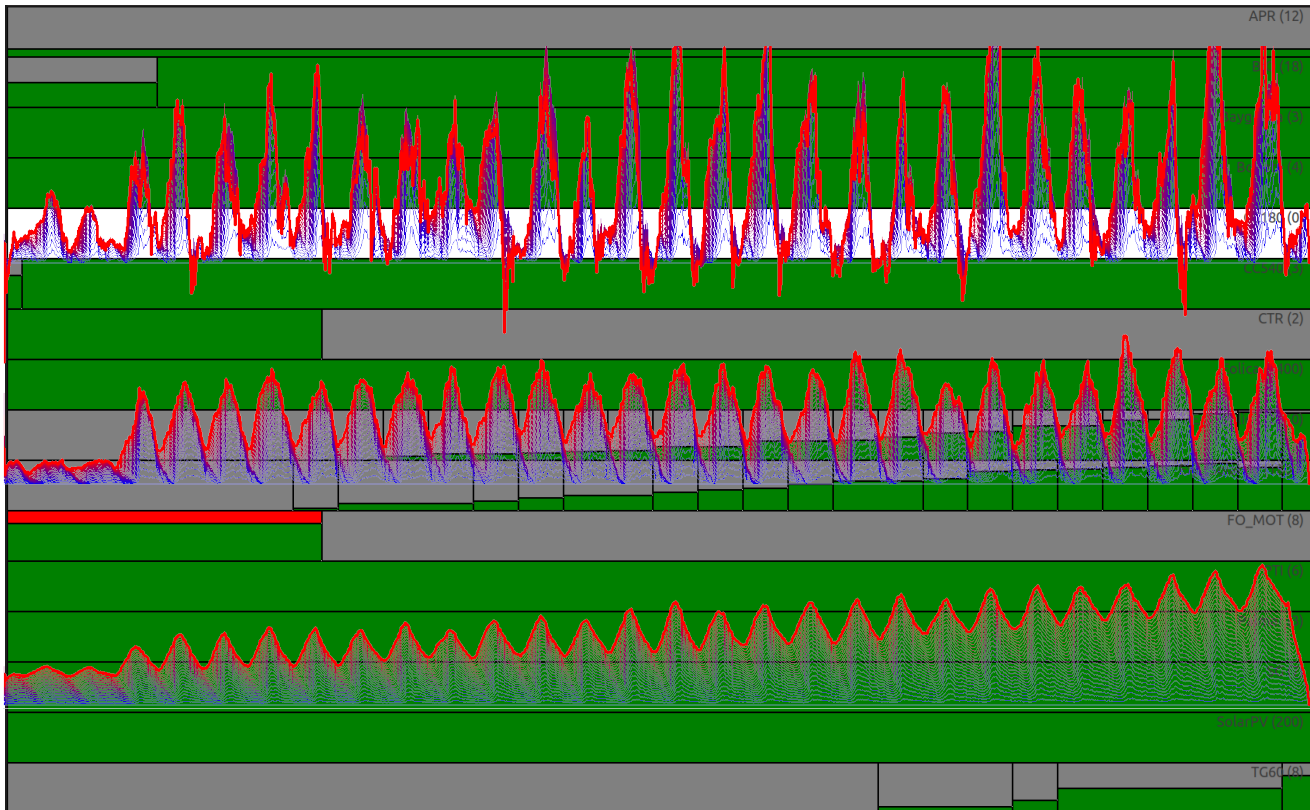


Fig. 27: Política de operación comparada con el plan de expansión.

9.1. Tipo de aproximador.

La ec.47 muestra la aproximación por un único hiperplano de la función de Costo Futuro. Como se puede observar, las variaciones del estado hidrológico (variable H) no tiene consecuencias directa sobre la valorización del agua de Rincón de Bonete ($\frac{\partial CF}{\partial V_{Bon}}$). En forma indirecta, un valor mayor de H implica una probabilidad superior de recibir aportes mayores lo que implica una probabilidad mayor de subir la cota del lago.

Como con la aproximación utilizada, el coeficiente $\frac{\partial CF}{\partial V_{Bon}}$ lo que lleva a que expresado el valor del agua en USD/MWh

Analizar utilidad del coeficiente de H al no tener impacto directo sobre el valor del agua.

Capaz que un mejor aproximador con igual complejidad sería:

$$CF(V_{Bon}, H) = A e^{b V_{Bon} + c H}$$

Para lograr ganar intuición sobre el tipo de aproximación que mejor representaría la función CF se realizó la optimización clásica, descretizando el espacio de estado con los resultados que se muestran en la Fig.28. para una semana en particular solo a modo de ejemplo.

Como se puede apreciar, las curvas iso-nivel son cuasi-rectas en casi todo el área y la distancia entre las curvas decrece en la zona roja respecto de la distancia en la zona azul.

La Fig.29 muestra el valor del agua de Bonete ($-\frac{\partial CF}{\partial V_{Bon}}$) en función del volumen embalsado para diferentes valores del estado hidrológico H . Como se puede apreciar, el valor del agua es sensible al valor de H relación que no logra captar la aproximación por un hiperplano. También se observa que el valor del agua sería bastante bien aproximado para cada valor de H por dos rectas, una para volúmenes inferiores a los 6300 Hm^3 y otra para los volúmenes superiores.

De lo anterior, se deduce que una aproximación como se muestra en la ec.48

$$CF = (a H + b V_{Bon} + c)^2 + d \quad \text{ec.(48)}$$

podría ser adecuada para cada tramo lineal.

Este es solo un ejemplo, sobre un caso concreto, pero sirve para identificar el tipo de estructura que puede presentar el sistema y que dependiendo de la zona del espacio de estado, la representación (o la aproximación eficiente puede cambiar). En el desarrollo del proyecto comenzamos con

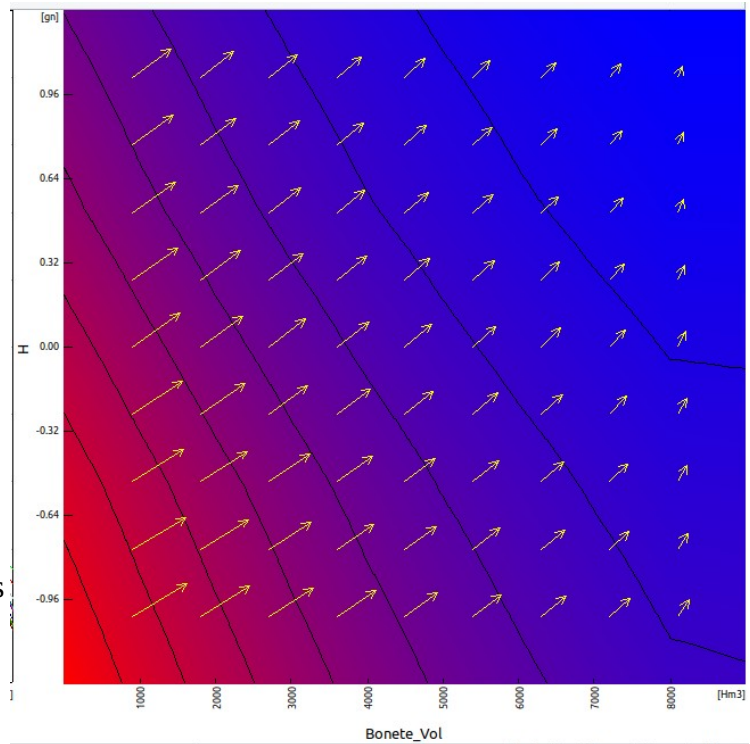


Fig. 28: Costo Futuro para una semana en particular. Iso-niveles y gradiente.

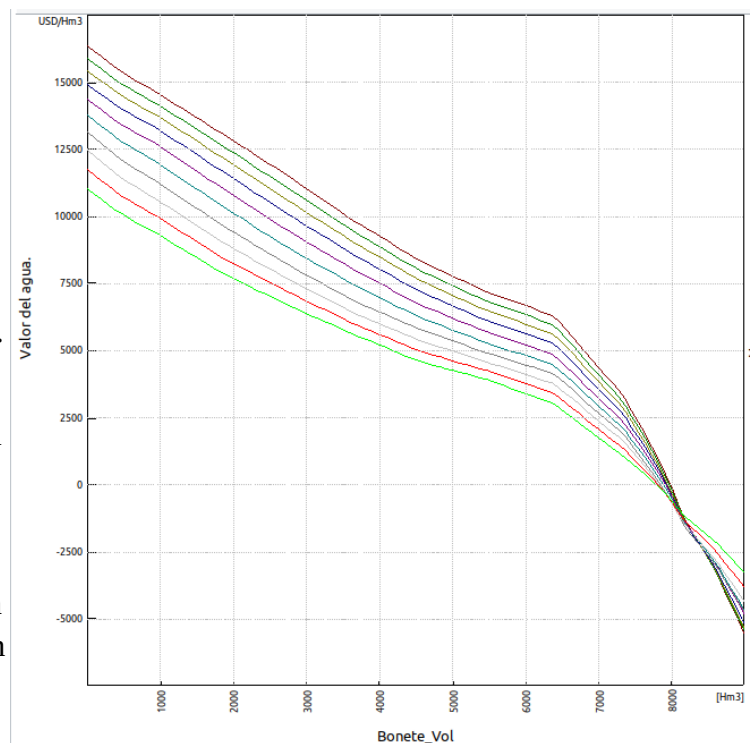


Fig. 29: Valor del agua de Bonete en función del volumen embalsado para diferentes valores de H .

un aproximador por único hiperplano en todo el espacio, y como vemos la Sala LP ese tipo de aproximación no es capaz de captar la relación entre el valor del agua y la variable H.

9.2. Parsimonia temporal.

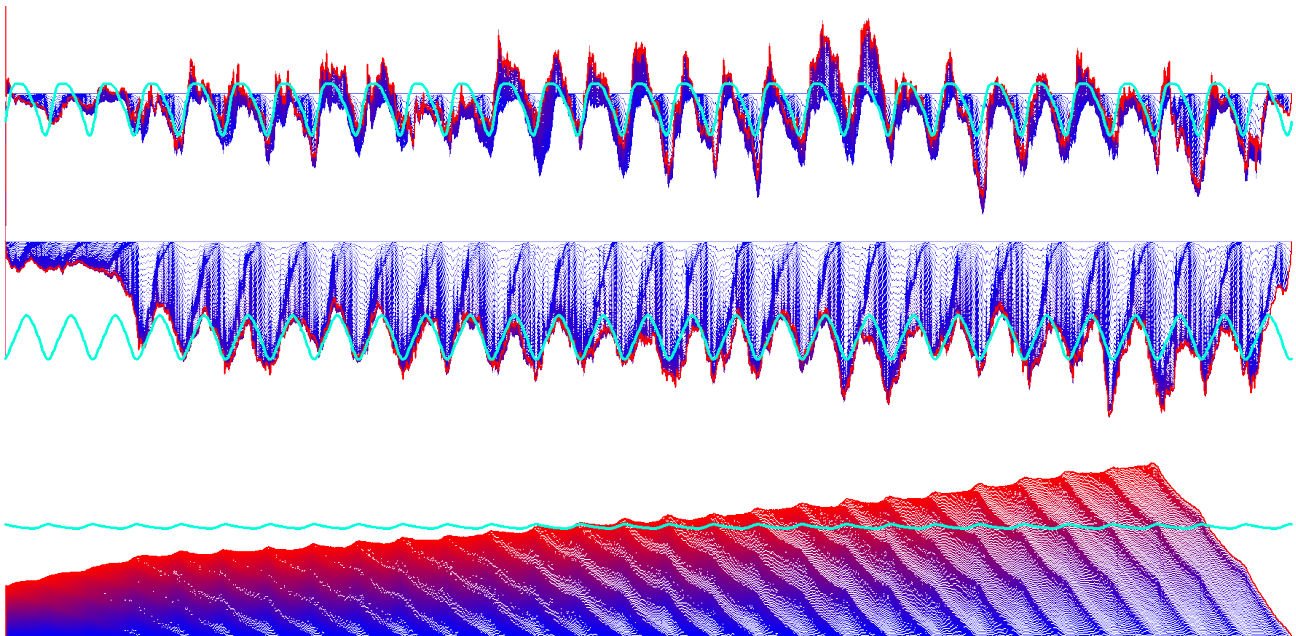


Fig. 30: Aproximador lineal de orden 1 y aproximación con 3 armónicos anuales.

La Fig.31 muestra la curva de permanencia del costo de abastecimiento actualizado al 1/1/2018, con tasa de descuento 10% de los 29 años de operación en MUSD de 1/1/2018.

La curva etiquetada “Clásica” corresponde a la simulación de 100 crónicas utilizando la PO obtenida por el método clásico (esto es discretización del espacio de estado e iteración de Bellman desde el futuro hacia el presente). La curva etiquetada “kiter165hp1” corresponde a la iteración 165 de la aproximación de la PO con un único hiperplano para cada paso temporal (con un solo paso de consideración de los costos de etapa y con coeficiente de aprendizaje 1) La curva etiquetada “kiter165hp1F3” corresponde a la política de operación que se obtiene al cambiar los coeficientes de los hiperplanos de la kiter165hp1 por su desarrollo de Fourier considerando los armónicos correspondientes a periodicidad anual y sus siguientes dos múltiplos (o sea semestral y trimestral).

Observar que aproximar por un hiperplano como la ec.47 implica utilizar 3 coeficientes para cada paso de tiempo. Al ser 1514 pasos de tiempo, la cantidad de coeficientes involucrados es: $3 \times 1514 = 4542$. En la simplificación con un desarrollo de Fourier de orden 3, se tiene el término constante más dos coeficientes por armónico lo que da un total de: $1 + 2 \times 3 = 7$ coeficientes, lo que implica sin lugar a dudas una reducción drástica de la cantidad de coeficientes.

Como era de esperar, una aproximación tan sencilla como un hiperplano lleva a una PO de mayor costo que la obtenida con el método clásico. Y como se puede observar, la simplificación del

modelo de un hiperplano determinado para cada paso de tiempo resulta en una política de operación un poco mejor que la que resulta de utilizar hiperplanos cuyos coeficientes son ajustados en forma global como un desarrollo de Fourier de orden 3.

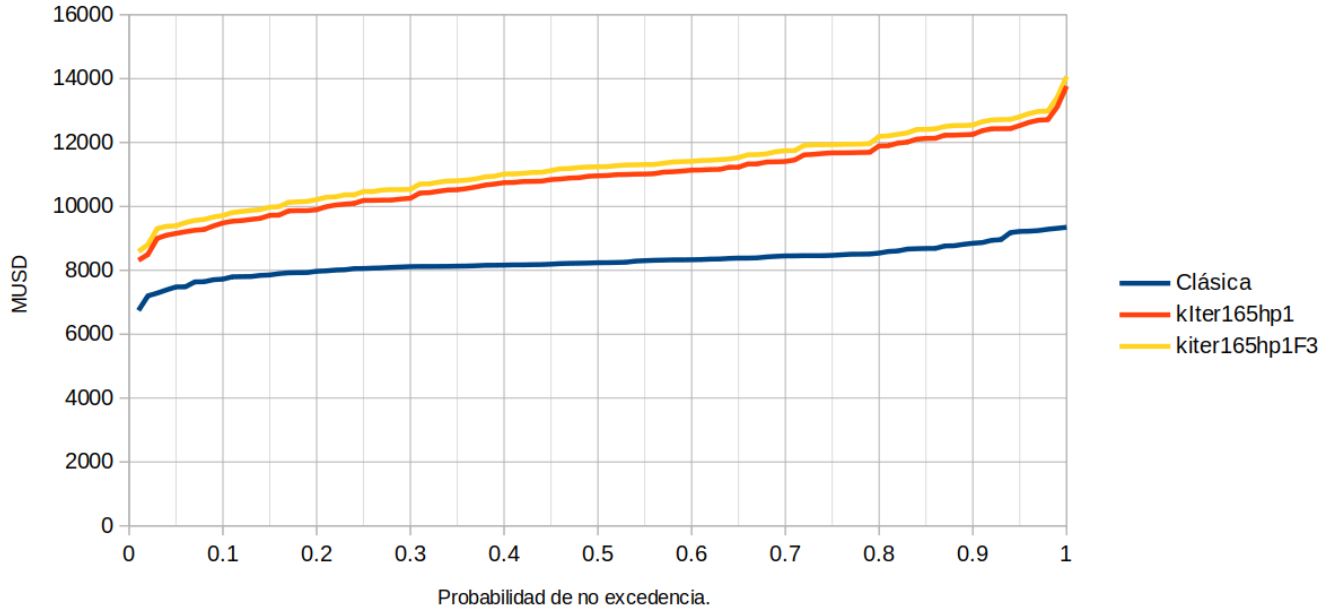


Fig. 31: Comparación entre los costos de 100 crónicas usando la PO-Clásica, a aproximación lineal y su simplificación con 3 armónicos.

Observar que la aproximación con 3 armónicos sobre todo el período no es buena sobre todo en los primeros años. Sería conveniente utilizar un tipo de simplificación (imposición de parsimonia) diferente para los primeros años que para el resto. También es notorio que el valor constante C_0 (canal inferior de la Fig.30) tiene un comportamiento creciente en el tiempo, comportamiento que no logra captar un desarrollo de Fourier que supone una señal en estado estacionario. Como el valor absoluto de la función de costo futuro no tiene consecuencias en la optimización, la mala aproximación de C_0 , no tiene consecuencias. Si los canales correspondientes a las derivadas parciales de CF respecto de los estados mostraran tendencias, además de estacionalidades, las mismas deberán ser captadas por modelos que no necesiten la suposición de señal estacionaria. Por ejemplo podría ser un modelo que combinara el producto de una envolvente por un desarrollo de Fourier cómo se ejemplifica en la ec.

$$C_0(kT) = (\alpha + \beta kT) \left(A_0 + \sum_{i=1}^3 \left[A_i \cos\left(2i\pi \frac{kT}{T_{\text{año}}}\right) + B_i \sin\left(2i\pi \frac{kT}{T_{\text{año}}}\right) \right] \right) \quad \text{ec.(49)}$$

9.3. Propagación de la información y Pasos Hacia Delante.

En cada iteración de la PO, se vuelve a tener una nueva aproximación de la PO mediante la adición de los costos directos del paso (cdp) resultantes de la simulación al Costo Futuro de llegada al

final de un conjunto de pasos. Si bien simulamos crónicas (o realizaciones) de muchos pasos, al momento de hacer la actualización del CF podemos elegir la cantidad de pasos hacia delante (phd) que utilizaremos para realizar la actualización y para considerar el CF en el estado de llegada al final de esa serie de pasos. En lo comentado anteriormente, se utilizó un solo paso hacia delante por lo cual la actualización tiene la forma de la ec

$$CF_{i+1}(X_k, k) = cdp_k + q CF_i(X_{k+1}, k+1) \quad \text{ec.(50) Actualización con un único paso hacia delante.}$$

Dónde q es el actualizador correspondiente a la tasa de actualización utilizada.

Si en lugar de utilizar un solo paso hacia adelante se utilizan n la actualización de la función CF sería como se muestra

$$CF_{i+1}(X_k, k) = \sum_{j=0}^{n-1} q^j cdp_{k+j} + q^n CF_i(X_{k+n}, k+1) \quad \text{ec.(51) Actualización con } n \text{ pasos hacia delante.}$$

El mirar más pasos hacia adelante, favorece la transferencia de la información en el sentido inverso del tiempo posibilitando así apurar la convergencia de las aproximaciones o incluso evitar la convergencia a mínimos locales cuando las funciones son no-convexas; pero también puede ser una fuente de aumento de la varianza de los valores de CF que puede comprometer la convergencia de las aproximaciones.

En la implementación realizada, se permite especificar el cálculo de varias actualizaciones como la ec.51 e indicar cómo serán ponderadas las mismas para obtener el nuevo valor de CF . Habrá que implementar un algoritmo capaz de analizar las diferentes opciones y con la capacidad de adaptarse a las características de cada problema. Seguramente no es lo mismo una Sala con paso de tiempo semanal que una con paso horario.

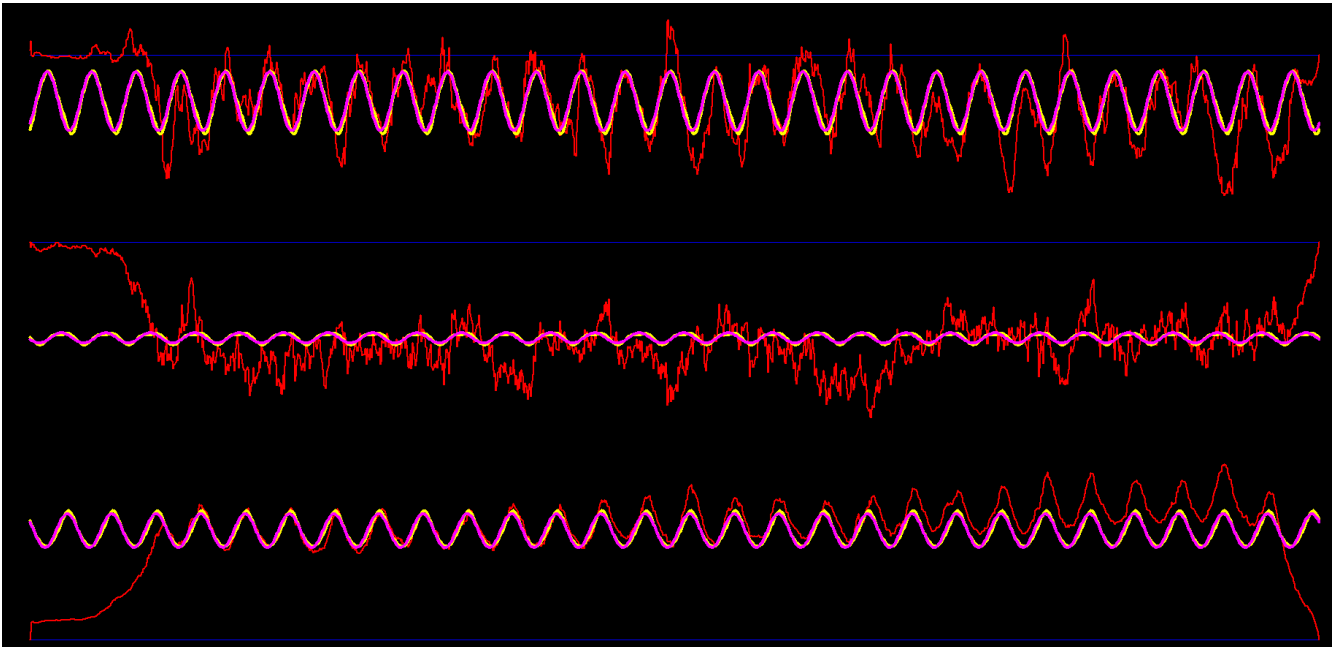


Fig. 32: Primer iteración con 104 pasos hacia delante (rojo). Y aproximación con 1 (púrpura) y 3 armónicas (amarillo).

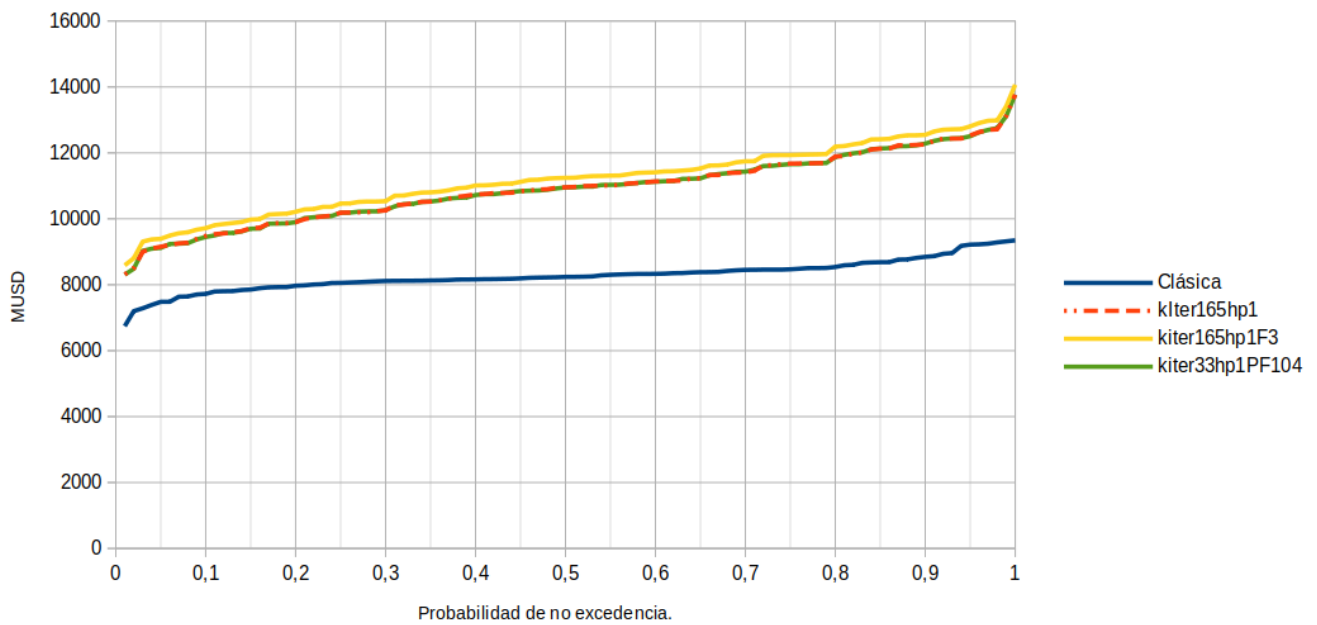


Fig. 33: Permanencia del costo total en las 100 crónicas para las diferentes aproximaciones.

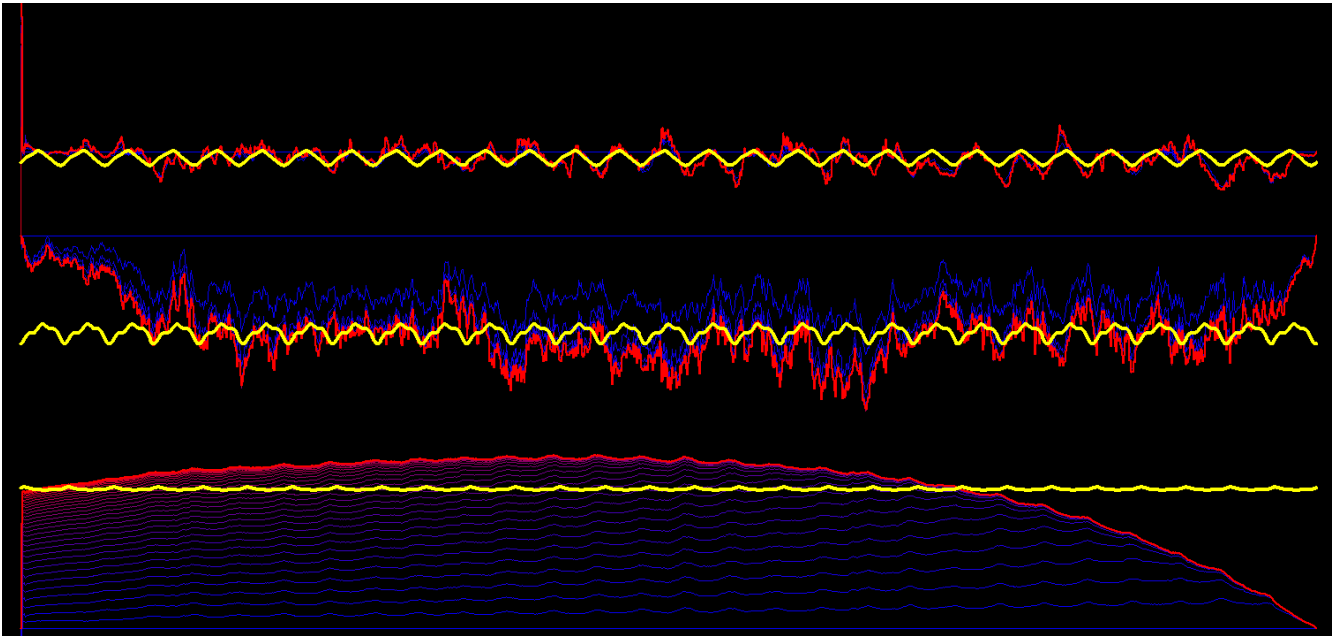


Fig. 34: Primeras 33 iteraciones con 104 pasos hacia delante.

10. Sistema Hidro-térmico con Aportes con Estado y costo térmico exponencial.

Se un problema de operación óptima de un sistema dinámico con dos variables de estado V_k y A_k

Las ecuaciones que rigen la dinámica del estado son:

$$0 = h_v(V_{k+1}, V_k, A_k, H_k, G_k) = -V_{k+1} + V_k - 6.12 H_k + 1.028 e^{1.0378 A_k + 6}. \quad \text{ec.(52) Restricción dinámica del embalse.}$$

$$0 = h_A(A_{k+1}, A_k, R_k) = -A_{k+1} + 0.7 A_k + 0.7141 R_k \quad \text{ec.(53) Restricción dinámica del proceso estocástico de los aportes en el espacio gaussiano.}$$

El costo de etapa se calcula como:

$$ce_k = 20 e^{0.023(100 - H_k)} \quad \text{ec.(54) Costo de Etapa.}$$

Dónde:

$H_k \in [0, 99.12]$ y $G_k \geq 0$ son las variables de control,

$V_k \in [0, 8208]$ y $A_k \in (-\infty, +\infty)$ son las variables de estado,

R_k es ruido blanco $N(0,1)$.

Suponiendo que se le suministra un CF NULO, el problema de despacho se puede plantear como la minimización de la Lagrangiana (dónde por comodidad de la notación hemos eliminado el subíndice k y hemos sustituido el subíndice $k+1$ por s (de “siguiente”).

$$CF(V, A) = \min_{H, G} (ce(H) + CF_s(V_s, A_s) + \lambda_v h_v(V_s, V, H, A, G) + \lambda_A h_A(A_s, A, R) + \beta V)$$

Haciendo variaciones en la dirección de V

$$\frac{\partial}{\partial V}(ce(H)+CF_s(V_s, A_s))+\lambda_V \frac{\partial}{\partial V} h_v(V_s, V, H, A)+\lambda_A \frac{\partial}{\partial V} h_A(A_s, A)+\beta=0$$

$$\frac{\partial}{\partial V} h_v(V_s, V, H, A)=1 \quad , \quad \frac{\partial}{\partial V} h_A(A_s, A)=0 \quad \text{de dónde}$$

$$\frac{\partial}{\partial V}(ce(H)+CF_s(V_s, A_s))=-\lambda_V-\beta$$

Ahora, haciendo variaciones en la dirección de A

$$\frac{\partial}{\partial A} ce(H)+\lambda_V \frac{\partial}{\partial A} h_v(V_s, V, H, A)+\lambda_A \frac{\partial}{\partial A} h_A(A_s, A, R)+\beta \frac{\partial}{\partial A} V=0$$

$$\frac{\partial}{\partial A}(ce(H)+CF_s(V_s, A_s))=-\lambda_V \frac{\partial}{\partial A} h_v(V_s, V, H, A)-\lambda_A \frac{\partial}{\partial A} h_A(A_s, A, R)-\beta \frac{\partial}{\partial A} V$$

$$\frac{\partial}{\partial A} h_v(V_s, V, H, A)=1.028 e^{1.0378 A_i+6.4781} 1.0378$$

$$\frac{\partial}{\partial A} h_A(A_s, A, R)=0.7$$

$$\frac{\partial}{\partial A} V=0$$

Derivando respecto de las variables de estado de llegada obtenemos:

$$\frac{\partial}{\partial A_s}(ce(H)+CF_s(V_s, A_s))=-\lambda_V \frac{\partial}{\partial A_s} h_v(V_s, V, H, A)-\lambda_A \frac{\partial}{\partial A_s} h_A(A_s, A, R)$$

$$\frac{\partial}{\partial A_s} CF_s(V_s, A_s)=\lambda_A \quad \text{ec.(55)}$$

y

$$\frac{\partial}{\partial V_s}(ce(H)+CF_s(V_s, A_s))=-\lambda_V \frac{\partial}{\partial V_s} h_v(V_s, V, H, A)-\lambda_A \frac{\partial}{\partial V_s} h_A(A_s, A)$$

$$\frac{\partial}{\partial V_s} CF_s(V_s, A_s)=\lambda_V \quad \text{ec.(56).}$$

Las ecs. 55 y 56 imponen que si $\nabla CF_s=0$ los multiplicadores de Lagrange λ_V y λ_A son nulos. Y entonces la única forma de que $\nabla CF \neq 0$ es que se active la restricción $V \geq 0$ para que $\beta \neq 0$ y entonces se cumple:

$$\frac{\partial}{\partial V} CF(V, A)=\frac{\partial}{\partial V}(ce(H)+CF_s(V_s, A_s))=-\beta$$

Entonces en la primer iteración en que $\nabla CF_s = 0$,

$$\frac{\partial}{\partial V} CF(V, A) = -\beta \neq 0 \quad \text{para aquellos estados de partida que terminen en } V_s = 0$$

y

$$\frac{\partial}{\partial V} CF(V, A) = 0$$

En la segunda iteración, para aquellos estados en que $\frac{\partial}{\partial V} CF(V, A) = -\beta \neq 0$ se tendrá un $\lambda_v \neq 0$

y entonces:

$\frac{\partial}{\partial A} (ce(H) + CF_s(V_s, A_s)) = -\lambda_v \frac{\partial}{\partial A} h_v(V_s, V, H, A)$ lo que se traduce en un valor $\lambda_A \neq 0$ para la tercer iteración.

10.1. Sala Caso 1. Resolución clásica.

Se construyó una Sala SimSEE de paso semanal con una representación que se intenta ajustar a las ecuaciones del Caso 1. Para ello, se definió una demanda plana de 1200 MW y se construyó un

CEGH que genera los aportes hidráulicos $1.028 \frac{10^6}{168 * 3600} e^{1.0378 A_k + 6.4781} [m^3/s]$ y con una central hidráulica con embalse de forma de representar la dinámica ec.52.

Esta Sala se optimizó con el método clásico con 10 discretizaciones de la variable de estado V_k que representa el volumen del embalse y 5 discretizaciones de la variable de estado A_k que representa el estado del proceso estocástico que genera el aporte hidráulico a la represa. Las diez discretizaciones del Volumen corresponden a divisiones de igual volumen (912 Hm3) entre el fondo del lago (cota 70 m) y el máximo del lago (cota 81 m). Las cinco discretizaciones corresponden al centro (el punto que deja igual probabilidad de ambos lados) de las cinco bandas de probabilidad de 20%. Los valores son los que se muestran en la tabla 6 y los estados se pueden clasificar como Super Seco, Medio Seco, Normal, Medio Húmedo y Super Húmedo según el sesgo que imprimen en la proceso que genera los aportes.

Tabla 6: Valores de las cinco discretizaciones de A.

Como el sistema representando está en estado estacionario (no cambia su estructura con el tiempo) la Política de Operación debería ser constante. Como la optimización se realiza en un horizonte utilizando la recursión de Bellman sobre la función de Valor (Costo Futuro), partiendo de una función nula para el fin de la última etapa de tiempo, se tiene que la función de Valor se va aproximando a una función dada que define la Política de Operación óptima.

En total se ejecutaron 1618 pasos (semanas) de optimización. La Fig. muestra la estimación del Valor del Agua ($-\frac{\partial CF(V_k, A_k)}{\partial V_k}$) expresada en USD/MWh para lo cual se dividió la derivada respecto al volumen por el coeficiente energético calculado como

$$c_e(h_{Bonete}) = 1000 * 9.8 * (h_{Bonete} - 7.05) * 0.87 / 3600 [MWh/Hm^3] \quad \text{ec.(57) Coeficiente energético Río Negro.}$$

Dónde h_{Bonete} es la cota del lago de Rincón de Bonete y la relación entre la cota y el volumen embalsado está dada por la ec

$$V = 42.1818(h_{Bonete} - 70)^2 + 282.1818(h_{Bonete} - 70) \quad \text{ec.(58) Volumen embalsado en función de la cota.}$$

En la resolución clásica (recursión hacia atrás en el tiempo calculando la función de valor de Bellman) en cada paso de tiempo k , desde el último hasta el primero, se recorren todos las estrellitas (así denominamos a los estados correspondiente al producto cartesiano del as discretizaciones seleccionadas) aplicando en cada una una realización del proceso estocástico (ec.53) y resolviendo el problema de encontrar la generación hidráulica H_k que minimiza la suma entre el costo de etapa (ec.54) y el Costo Futuro en el estado llegada al fin del paso $CF_{k+1}(V_{k+1}, A_{k+1})$. Resolviendo el mismo problema en cada estrellitas para diferentes realizaciones de los procesos estocásticos y tomando el valor esperado se obtiene así la estimación de $CF_k(V_k, A_k)$.

En el problema de minimizar costo de etapa más costo futuro de llegada, la información relevante de $CF_{k+1}(X)$ (o sea aquella que forma la política de operación) es su variación al cambiar el estado de llegada X (en el ejemplo $X = [V, A]^T$). Entonces, lo importante más que la estimación de $CF_{k+1}(X)$ en si, es la estimación de $CF_{k+1}(X_a) - CF_{k+1}(X_b)$ para tener así una justa comparación entre dos opciones de control H_a , H_b . Al estimar mediante simulaciones, la diferencia de dos variables aleatorias, es conveniente utilizar “sorteos coherentes” en el sentido de que los diferentes estados sean “probados” (o puestos a prueba para determinar el costo de la operación futura partiendo de ese estado) con las mismas “suertes”. De otra forma podría pasar por ejemplo comparando dos estados iniciales uno con mayor volumen embalsado que otro, la estimación resultara en que es más costoso el futuro del estado de mayor volumen embalsado que el de menos solo porque al realizar los sorteos, en ese estado se evolucionó a uno de menores lluvias que en el segundo. Estos errores serían eliminados si la cantidad de realizaciones con que se testea cada estado tendiera a infinito. Para reducir la variancia de la estimación de la diferencia de costos entre los estados, en SimSEE se aplican las mismas realizaciones (R_k en este ejemplo) sobre todos los estados del paso k . Esta es una técnica clásica de reducción de variancia aplicable a métodos de Monte Carlo cuando se está estimando la diferencia entre dos variables aleatorias (en nuestro caso la diferencia de costos entre los estados). Esta técnica permite obtener una función $CF(X)$ cuyas derivadas espaciales (en las direcciones de los estados) son aproximadas con razonable precisión con pocos sorteos R_k . La Fig.35 muestra los tres primeros pasos de tiempo el valor del agua calculado como se muestra en la ec.59.

$$cva = \frac{-1}{c_e(V)} \frac{\partial CF_k(V, A)}{\partial V} \quad \text{ec.(59) Valor del agua embalsada.}$$

La Fig.35 muestra cinco paneles correspondientes a los cinco estados de la variable A . En cada panel, se muestra el valor del agua (ec.59) para cada una de los 10 valores del volumen embalsado V . Como se puede apreciar, el valor del agua crece a menor volumen y también crece a menor valor de A . La Fig.35 muestra los tres primeros pasos de tiempo, pero también se están mostrando 3 estimaciones diferentes con 1 (una), 10 (diez) y 100 (cien) realizaciones de R_k en cada paso de tiempo. Como se puede apreciar, las figuras correspondientes a 10 y 100 realizaciones (o crónicas) están prácticamente superpuestas lo que muestra que por pasar de 10 a 100 realizaciones no se logra una precisión mayor significativa. Observar que las curvas correspondientes a una sola realización están apartadas de las otras lo que implica que puede ser relevante utilizar más de una realización. De cualquier forma, observar que esto estaría cambiando a qué nivel del embalse se utilizan otros recursos, pero que en líneas generales se tienen el mismo andamio. Este comportamiento se logra gracias a lo que hemos llamado “sorteos coherentes”.

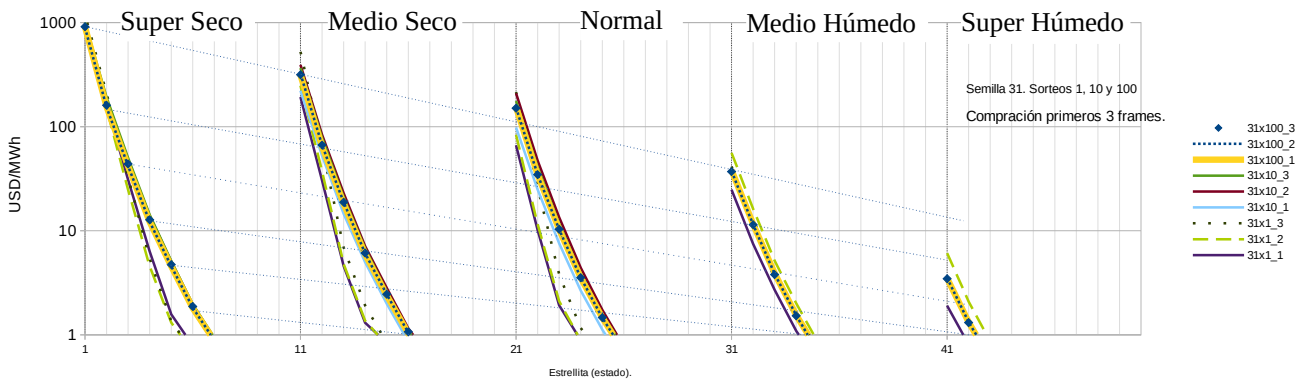


Fig. 35: Menos derivada de CF respecto de V , expresada en USD/MWh. (Valor del agua)

Para reflejar la “varianza” que puede tener una estimación con sorteos no-coherentes, en la Fig.36 se muestra la evolución de $CF(X_1)$ (esto es de la estrellita que representa el embalse vacío y en el estado Super Seco) para 1, 10 y 100 realizaciones por paso de tiempo. Como se puede apreciar, el “ruido” de la trayectoria de 1 realización es comparable al propio valor estimado y también que en la medida en que se utilizan más realizaciones, la varianza de la estimación de se reduce.

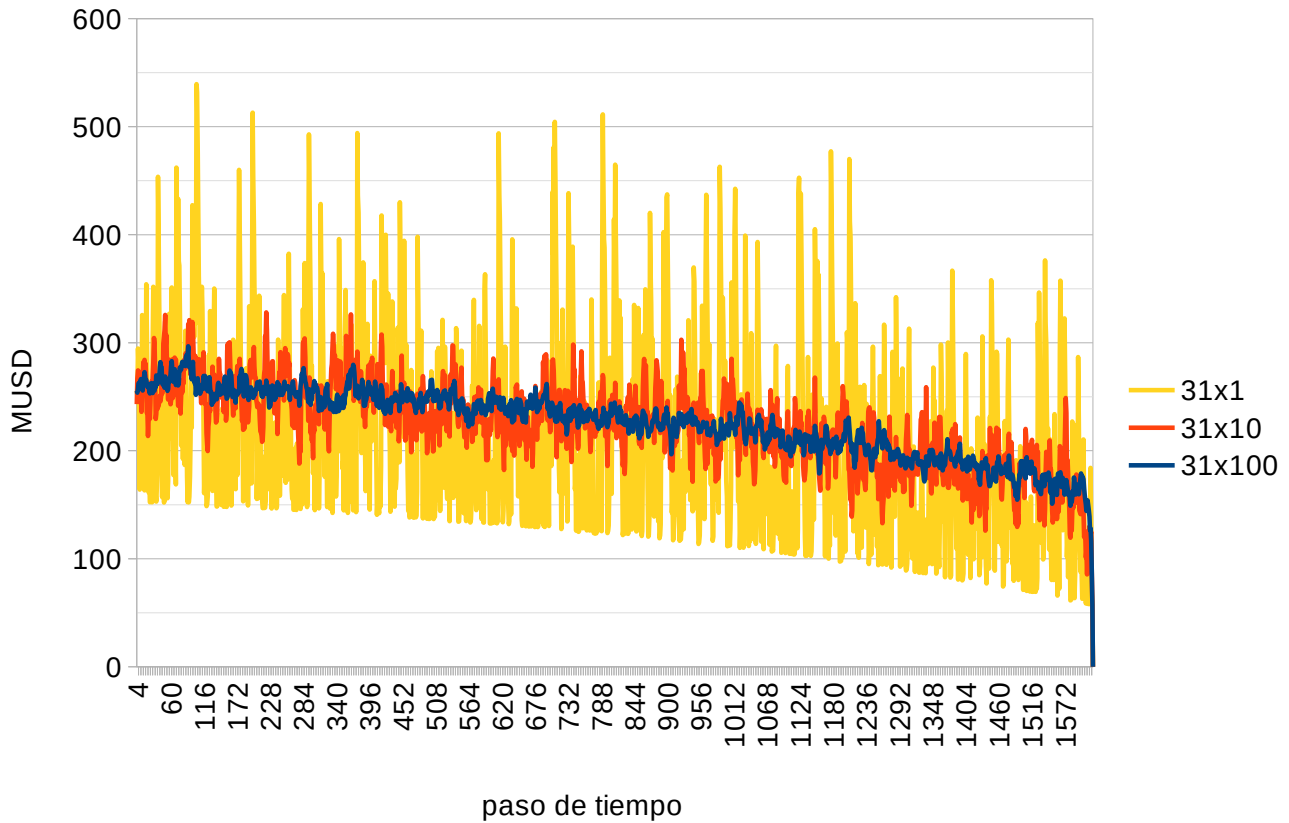


Fig. 36: Costo Futuro de la estrella $V=0$ (Fondo del lago), $A=-1.28156178489703$ (SuperSeco)

La Fig.35 nos es útil también para ganar intuición sobre el tipo de función de costo futuro que debemos esperar en los sistemas de generación de energía. En particular observar que en este caso que se consideró escala logarítmica, las curvas son bastante lineales en la variación de V . También lo son en la variación de A como lo muestran las líneas punteadas que unen los puntos de igual V entre los paneles de estado hidrológico, salvo para el estado “Super Húmedo”. Este comportamiento se ha observado en otras configuraciones del sistema de generación. Muestra que existen unos costos que crecen más que lineal (en el ejemplo exponencial) en la media en que escasean los recursos y que para el lado de la abundancia de recursos hay un límite a partir del cual no existe posibilidad de bajar aún más el costo. Esta observación nos puede guiar en el tipo de funciones de aproximación a utilizar.

11. Caso de prueba sobre la operación de Corto Plazo del Sistema.

Para testear la validez de lo implementado se ejecutó la Sala Vates (Programación semanal, de paso horario bajada de la web de ADME) con el algoritmo clásico y con el algoritmo de auto-aprendizaje desarrollado con una política de operación aproximada por un conjunto de redes neuronales. La sala en cuestión tiene 3 variables de estado consistente en el volumen embalsado de los lagos de Rincón de Bonete, Palmar y Salto Grande. La sala es de paso horario y como principal característica tiene en consideración los pronósticos de generación eólica, de aportes hidráulicos a las represas y de la Demanda.

La intención de este caso de test, es solo mostrar que los algoritmos implementados aprenden una política de operación y que se cumple que la misma va mejorando con las diferentes iteraciones. Para la prueba, se consideraron dos estructuras de aproximación a CF. Una consistente en una capa de 3 neuronas del tipo Tangente Hiperbólica seguida de una capa con una neurona también del tipo Tangente Hiperbólica (estructura 3th+th) y la otra con una capa de 5 neuronas lineales seguida de una capa que considera el máximo de dichas neuronas (estructura 5lin+max). Estas estructuras se repite para cada una de las 168 horas del horizonte de simulación.

La Fig. 37 muestra la evolución del sobre costo de cada Política de Operación, aproximada con las dos estructuras de aproximador consideradas, respecto de la Política de Operación clásica (recursión de Bellman). En la iteración (cero) se parte de la Política de Operación Nula que consiste en suponer un Costo Futuro de gradiente nulo en todo el espacio. Las primeras 90 iteraciones están realizadas con el Modo de Evolución 4 (las estrellitas del estado se randomizan en cada paso y se usa Time Difference 1) y a partir de la 91 inclusive se pasa al Modo de Evolución 0 (las estrellitas del estado evolución según la dinámica del sistema) utilizando la acumulación de los costos directos hasta el final del horizonte en cada paso (Time difference 168).

Al utilizar el Modo4, la información del futuro "viaja hacia el presente" a razón de un paso de tiempo por iteración (por usar Time Difference 1) y por tanto, para que la información del enganche (al final del horizonte de simulación) sea considerada en la primer hora del horizonte, se deben ejecutar por lo menos 168 iteraciones. Esta "lentitud" en transmitir la información se nota en la lentitud de las mejoras de la PO en la Fig.37. En la misma figura se aprecia que la velocidad de variación aumenta al cambiar al Modo0 con acumulación de costos hasta el final. Como se explica en la sección 4.2.1 cada modo de evolución tiene sus pros y contras. Recordar que los algoritmos desarrollados son para su aplicación en forma continua sobre el conjunto de Salas en uso para la programación del despacho óptimo del Sistema Interconectado Nacional y por tanto, la lentitud de la convergencia no es tan importante, dado que una vez alcanzado una buena representación, las iteraciones continuas siempre la van mejorando (no se descarta lo aprendido). De cualquier forma, dependiendo del horizonte de estudio, de la existencia o no de pronósticos (de generación, hidráulica, etc.) puede valer la pena combinar los modos de evolución de estado. La utilización del Modo 0 (evolución con la dinámica del

sistema) tiene el atractivo de reflejar las probabilidades de los estados, pero tiene el inconveniente, que las trayectorias pueden "unificarse" y a partir de ese paso de tiempo se pierda la capacidad de "senear" las variaciones de CF por el movimiento del Estado. Sobre todo al inicio de las iteraciones, comenzando con la Política de Operación Nula, la tendencia es a vaciar los lagos y por tanto todas las trayectorias tienden al (0, 0, 0). Para evitar esto se recomienda comenzar el aprendizaje con un modo de evolución del estado > 0 y luego de tener cierto grado de exploración del espacio de estado, perfeccionar la Política de Operación con iteraciones en el Modo 0 de forma de considerar las probabilidades de ser visitados de los diferentes estados y poder considerar acumulaciones de costos directo de más de un paso (Time difference > 1) permitiendo acelerar la convergencia. Sin duda que el desarrollo de estrategias que combinen los modos de evolución será parte de las investigaciones a realizar en la aplicación directa de los algoritmos.

Para presentar los resultados se muestran la salida de la generación por fuente en valor esperado para las primeras 168 horas (una semana) sobre la base de simulaciones de 1000 crónicas.

Se utilizó factor de aprendizaje (learning-rate) de 0.3. Esto significa que con cada iteración, los parámetros de la aproximación obtenida son ponderados por 0.3 y sumados a los de la estimación anterior multiplicados por 0.7. Se usaron 100 crónicas de simulación en cada iteración formadas por 100 "estrellitas" y 1 sola suerte. El modo de evolución seleccionado fue el 4, por lo que durante el entrenamiento las 100 estrellitas son movidas en forma aleatoria en cada paso de simulación.

El espacio de estado de dimensión 3, considerando los volúmenes de los embalses de Bonete, Palmar y Salto.

La representación 3th+th implica $(3 + 1) * 3 + (3 + 1) = 16$ parámetros por paso de tiempo.

La representación 5lin+max implica $(3 + 1) * 5 = 20$ parámetros por paso de tiempo.

La representación Clásica implica $20 * 15 * 15 = 4500$ parámetros por paso de tiempo, donde 20, 15 y 15 corresponden a las discretizaciones consideradas en los lagos de Bonete, Palmar y Salto respectivamente.

Como se puede apreciar, en la Fig.37 las aproximaciones logradas con las dos estructuras probadas de 16 y 20 parámetros, en 108 iteraciones, tienen un sobrecosto del orden de 2% respecto de la PO-Clásica. A nuestro entender estos resultados son promisorios, dada la relación de complejidad (16/4500 y 20/4500) de los modelos. Es posible que el error se pueda reducir aún más si se continua el aprendizaje, pero a los efectos de este trabajo no se considera relevante, pues un 2% de error en el valor esperado, en los resultados de simulaciones de 1000 crónicas está dentro del error de estimación con una confianza de 95%.

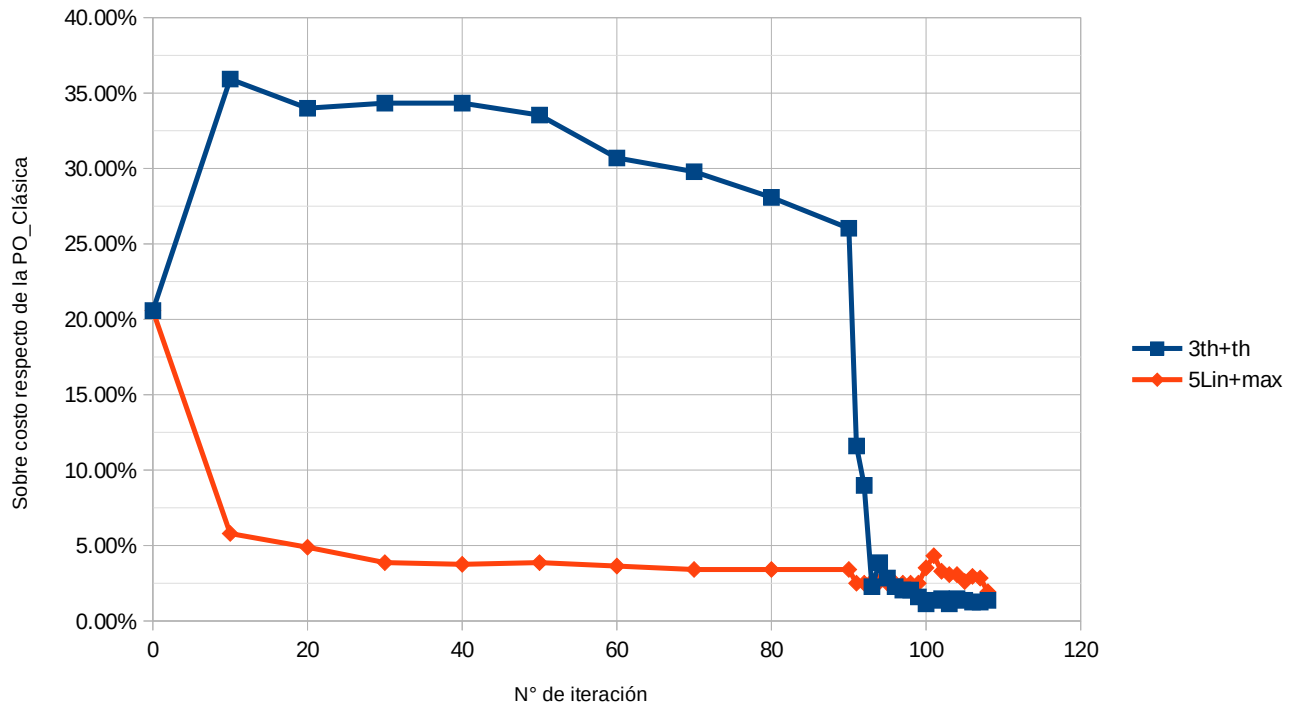


Fig. 37: Sobrecosto de la operación para las Políticas de Operación aproximadas según la iteración.

Para completar los resultados del caso de comparación, en las Fig.38 y 39 se muestra la Generación por Fuente en valor esperado (gráfico clásico mostrado en la web de ADME). Como se puede observar, si bien hay diferencias, el despacho en valor esperado es similar. Es un poco más "ruidosa" la operación con la PO_3th+th que la clásica. Este aspecto seguramente se mejoraría de continuar con las iteraciones o si se impusiera cierto requerimiento de parsimonia en los parámetros de la aproximación. Las Figs.40 y 41 muestran la valorización del agua de Rincón de Bonete (menos la derivada del costo futuro respecto del volumen embalsado) para la PO_Clásica y la PO_3th+th respectivamente. Como se puede apreciar hay diferencias, sobre todo en que la PO_Clásica presenta una dispersión mayor por paso de tiempo (bandas de probabilidad separadas) que los valores de la PO_3th+th mientras que esta segunda, si bien es menos dispersa por paso de tiempo, tiene mayor variación durante el horizonte de tiempo. Las Figs. 42 y 43 muestran los costos marginales mostrando mayor dispersión en la PO_3th+th que en la PO_Clásica.

Se concluye que las herramientas de aprendizaje desarrolladas tienen buen poder de aproximación para la resolución de programación semanal con pronósticos de generación (Salas de Corto Plazo). Se observa (cuestión ya conocida que también afecta a la PO_Clásica) que al tratarse el problema de despacho óptimo de un problema en que generalmente hay muchas soluciones muy parecidas (por ej. puedo despachar en una hora un embalse y en la siguiente otro, o viceversa, hay muchas combinaciones que terminan sumando el mismo costo. Es una condición de estar resolviendo un balance de energía paso a paso y en el horizonte) lo que ameritaría agregar requerimientos de parsimonia para seleccionar entre las equivalentes las Políticas de Operación "más suaves".

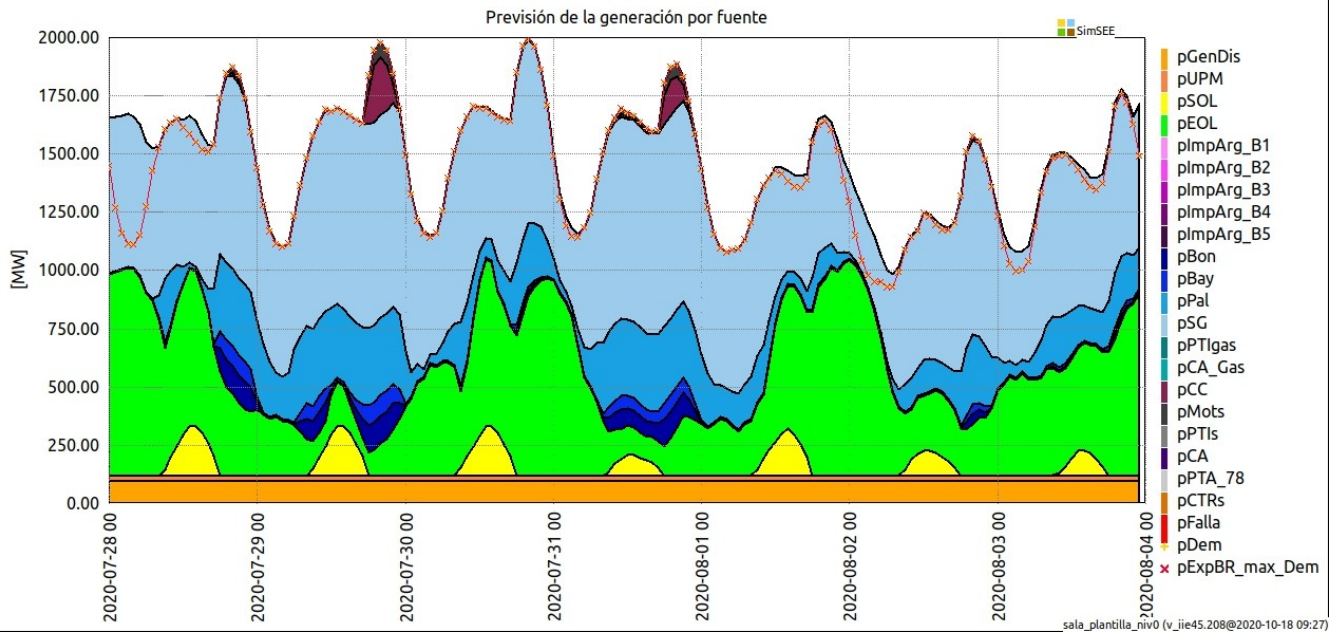


Fig. 38: Generación por fuente, Clásica (recursión de Bellman).

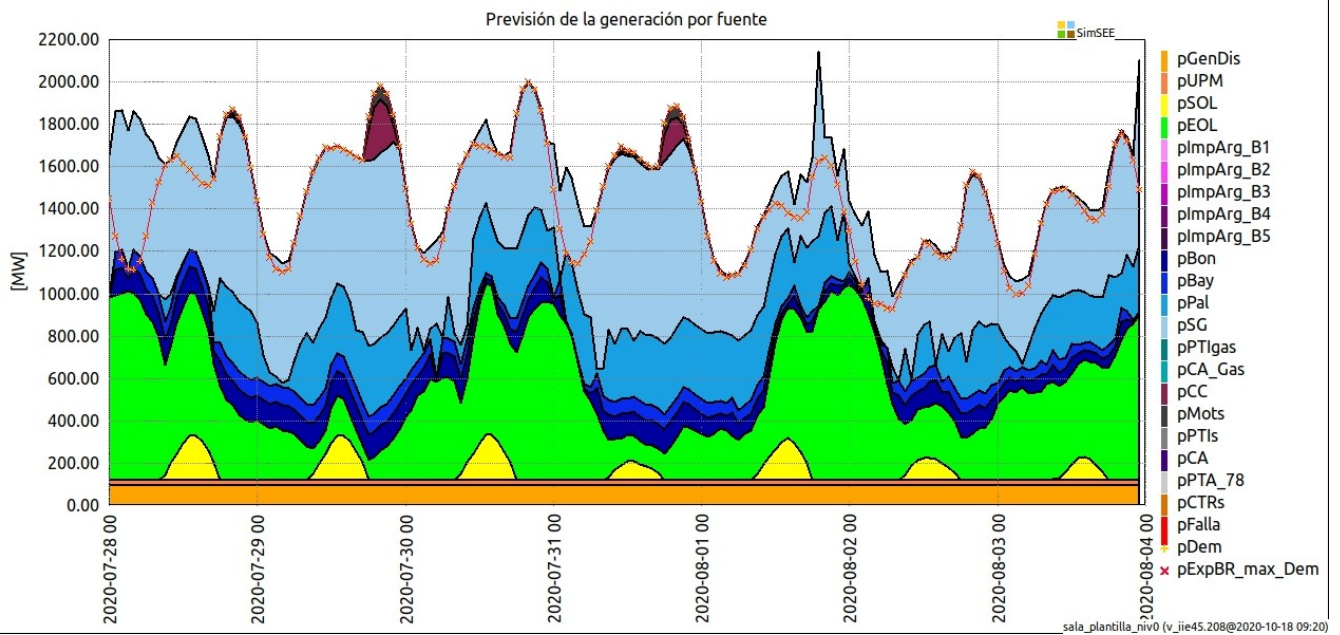


Fig. 39: Generación por fuente, 3th+th, 90 iter. M4 + 18 iter. M0+td(168)

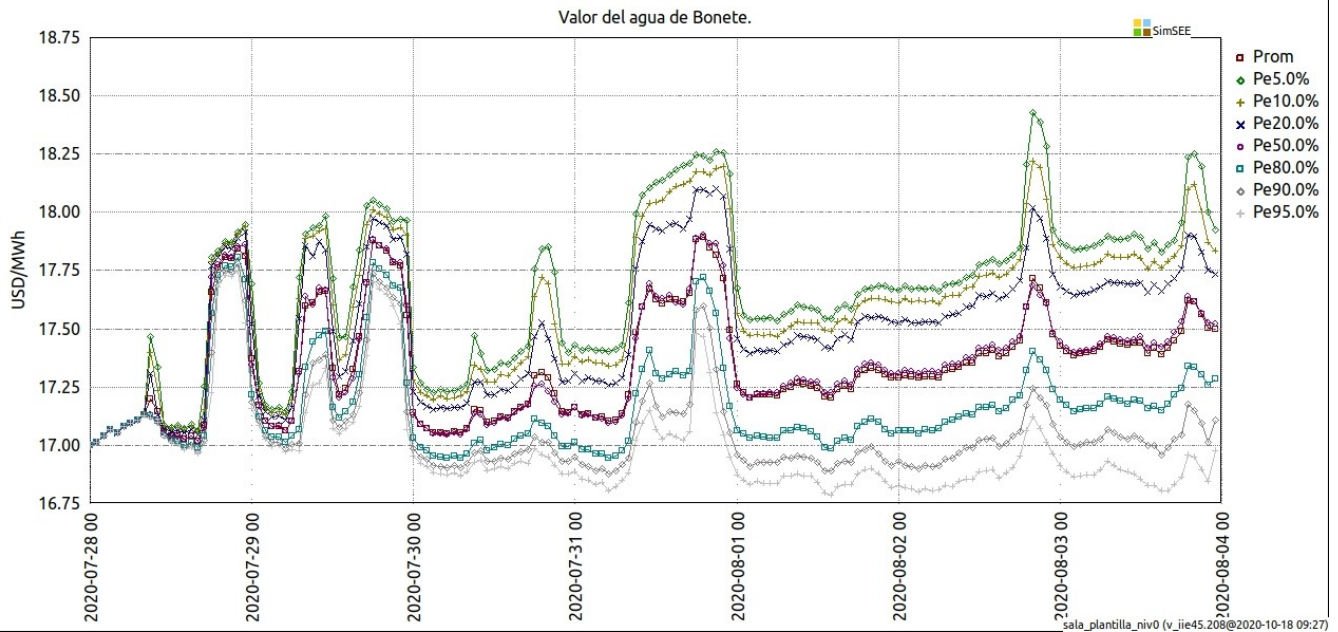


Fig. 40: Valor del agua de Rincón de Bonete, Clásica (recursión de Bellman).

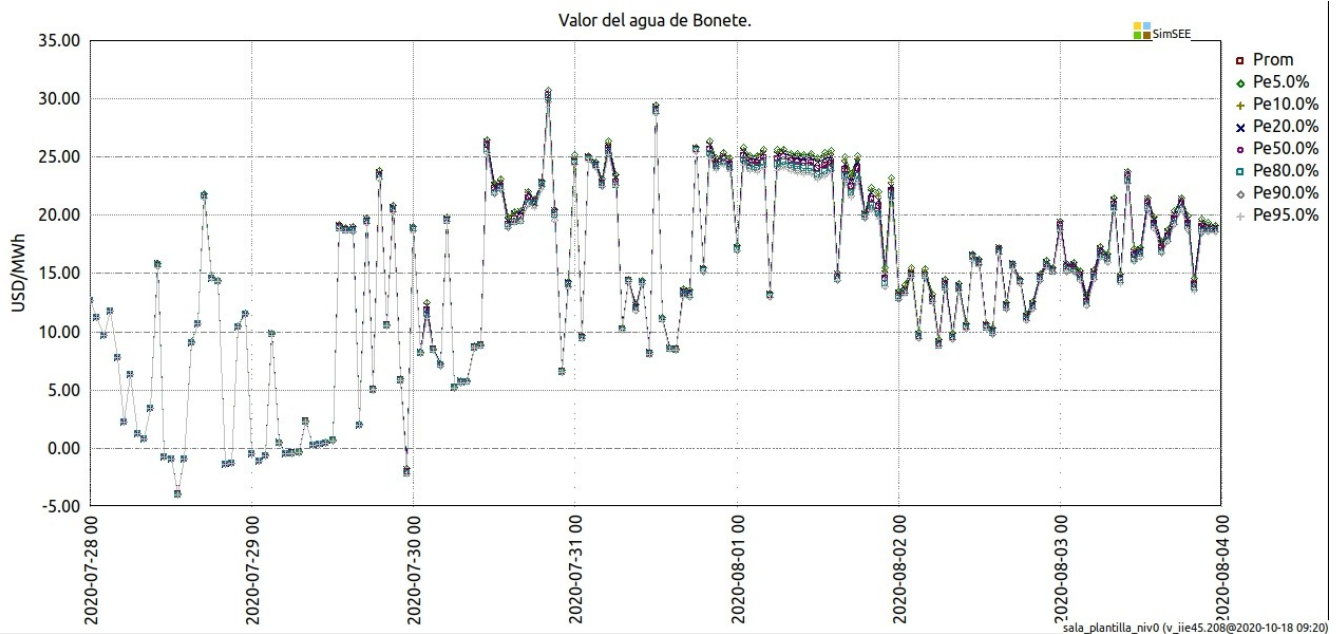


Fig. 41: Valor del agua de Rincón de Bonete, 3th+th, 90 iter. M4 + 18 iter. M0+td(168).

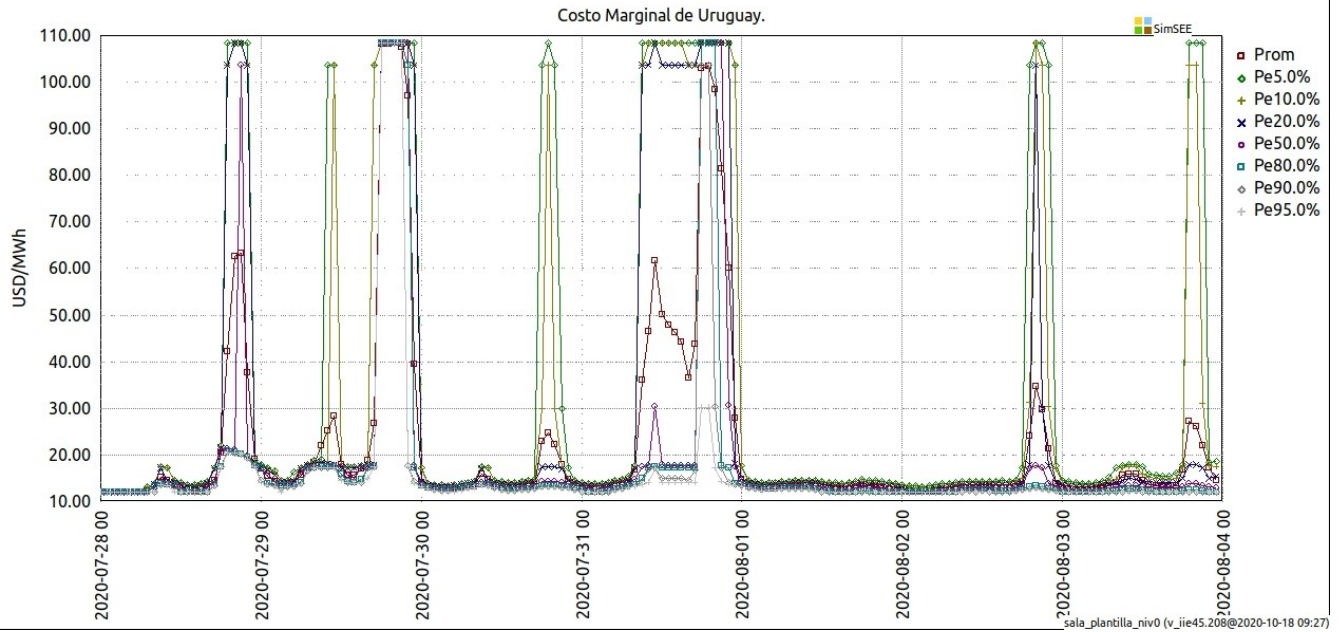


Fig. 42: Costo Marginal. Simulación con Política de Operación Clásica.

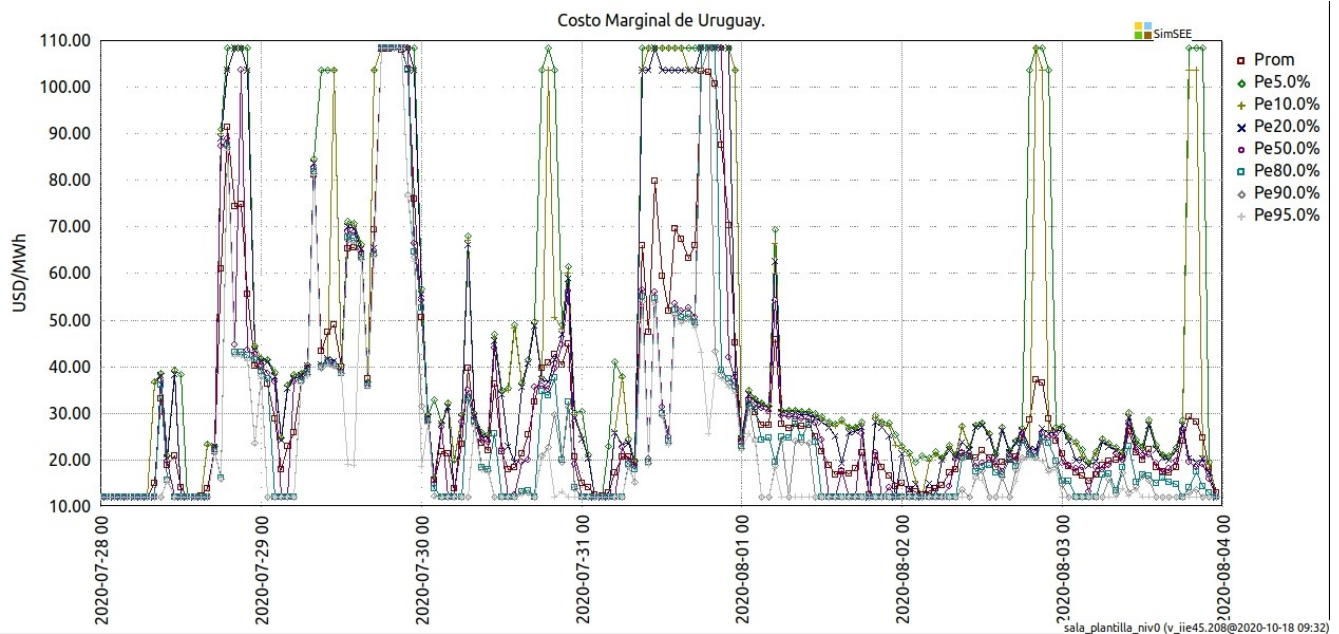


Fig. 43: Costo Marginal. Política de Operación: 3th+1th, 90 iter. M4 + 18 iter. M0+td(168).

12. Revisión bibliográfica.

Las principales lecturas realizadas en el proyecto fueron:

12.1. Abstract Dynamic Programming (Bertsekas)

Este libro tiene como, su título lo indica, una descripción formal de las ecuaciones asociadas al problema de programación dinámica estocástica. Puede servir de base para dar formalidad al lenguaje.

12.2. Learning and Soft Computing Support Vector Machines, Neural Networks, and Fuzzy Logic Models, Vojislav Kecman, 2001 MIT.

archivo: Learning And Soft Computing - Support Vector Machines, Neural Networks, And Fuzzy Logic Models (2002).pdf

Este libro tiene un resumen de las diferentes técnicas de aproximación de funciones.

12.3. DYNAMIC PROGRAMMING

Bellman , Princenton University Press. 1957

archivo: Bellman Dynamic Programming.djvu

Libro clásico de Bellman.

12.4. Iterative Gaussianization: From ICA to Random Rotations

Valero Laparra, Gustavo Camps-Valls, Senior Member, IEEE, and Jesús Malo

archivo: 05720319.pdf

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 22, NO. 4, APRIL 2011 537

Este trabajo presenta un método de gaussianización de un mapeo que puede resultar una vía de exploración para los aproximadores de $CF(X,k)$ en el sentido de introducir un mapeo previo que permita distribuir en form uniforme la información contenida en la función.

12.5. RADIAL GAUSSIANIZATION WITH CLUSTER-SPECIFIC BIAS COMPENSATION

Shuai Huang, Damianos Karakos , Daguang Xu. Center for Language and Speech Processing
Human Language Technology Center of Excellence
Department of Electrical and Computer Engineering
Johns Hopkins University, Baltimore, MD
email: {shuaihuang,damianos,dxu5}@jhu.e
2011 IEEE Statistical Signal Processing Workshop (SSP)

archivo: 05967640.pdf

Este trabajo presenta una estrategia de eliminación del "bias" para mejorar el procedimiento de gaussianización de un conjunto de vectores.

12.6. A DAM : A M ETHOD FOR S TOCHASTIC O PTIMIZATION

archivo: ADAME_OPTIMIZER_1412.6980.pdf
Diederik P. Kingma *
University of Amsterdam, OpenAI Jimmy Lei Ba *
University of Toronto
dpkingma@openai.com jimmy@psi.utoronto.ca

Presenta el algoritmo ADAM de optimización utilizado para el entrenamiento de redes neuronales en Tensor Flow. De este trabajo podemos sacar la estrategia de modificación del paso (learning-rate) en el algoritmo actualmente disponible en SimSEE.

12.7. Analysis of stochastic dual dynamic programming method

Alexander Shapiro
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA
archivo: Analysis of stochastic dual dynamic programming method.pdf
European Journal of Operational Research 209 (2011) 63–72
Contents lists available at ScienceDirect

European Journal of Operational Research
journal homepage: www.elsevier.com/locate/ejor
Stochastics and Statistics

Este trabajo es de "las piezas" claves en la temática de aproximaciones a la programación dinámica estocástica. En particular intenta mostrar el tratamiento de lo estocástico en los algoritmos SDDP (Stochastic Dual Dynamic Programming). El enfoque dado en nuestro proyecto no es el de SDDP porque nos hemos orientado hacia una aproximación de CF por otras vías, que podrían incluir la acotación sucesiva por hiperplanos inferiores de CF como en SDDP, pero que pretendemos no se limite a ello dado que creemos que la posibilidad de introducir PARCIMONIA TEMPORAL en los parámetros debe ser explotada.

12.8. Reinforcement Learning: An Introduction

Second edition, in progress
February 28, 2018
Richard S. Sutton and Andrew G. Barto
archivo: [bookdraft2018feb28.pdf](#)

Este libro presenta diferentes estrategias de aprendizaje de la función de Valor y de la función de Valor-Acción. Está más bien orientado a la resolución de juegos (como el go, el ajedrez) que revisten complejidades diferentes a las enfrentadas por nosotros. En nuestro caso, la dinámica del sistema sobre variables de estado continuas y la representación de lo estocástico hace que se necesiten técnicas diferentes a las planteadas en el libro. Igualmente se considera que este libro aportó a una generalización del lenguaje utilizado y a obligar al grupo de trabajo a pensar un poco diferente sobre algunos de los temas tratados. Solo a modo de ejemplo el análisis sobre la conveniencia o no de aprender la función de valor o la de valor-acción.

12.9. Variance Reduction in Population-Based Optimization: Application to Unit Commitment

Jean-Joseph Christophe, Jérémie Decock, Jialin Liu, Olivier Teytaud
Bat 660 Claude Shannon Univ. Paris-Sud, 91190 Gif-sur-Yvette, France
{firstname.lastname}@inria.fr
<https://tao.lri.fr>
archivo: [CRN_Aplicacion_UnitComitment_EAsource.pdf](#)

Uno de los temas claves para la implementación de cualquier función, por muestreo, en un ambiente estocástico es el tratamiento de Lo Aleatorio y las técnicas de reducción de varianza que se utilicen. Tenemos la experiencia de primera mano de la importancia de este tema en la implementación actual de SimSEE siendo el tratamiento de la reducción de la varianza uno de los puntos claves para que hoy

SimSEE sea una herramienta útil de uso industrial. Este artículo presenta el tema en forma interesante con casos artificiales en los cuales la técnica de Common Random Numbers (CRN, que es lo actualmente usado en SimSEE) son buenos y en casos donde los resultados son malos. También presenta la técnica de Antithetic Variables, Importance Sampling, Quasi Monte Carlo, Control Variates, Stratified sampling. De todas estas técnicas, entendemos que debemos mantener el uso de CRN para una exploración inicial y luego de disponer de aproximaciones de la función de valor seleccionar realizaciones específicas en base a sus consecuencias sobre la función de valor (lo que sería una mezcla entre CRN y stratified sampling o también podría visualizarse como un importan sampling).

12.10. SOME GUIDELINES AND GUARANTEES FOR COMMON RANDOM NUMBERS*

PAUL GLASSERMAN AND DAVID D. YAO

Graduate School of Business, Columbia University, New York, New York 10027

IEI OR Department, Columbia University, New York, New York 10027

archivo:CRN_glasserman_yao_guidelines.pdf

MANAGEMENT SCIENCE

Vol. 38, No. 6, June 1992

Printed in U.S.A.

Tiene algunas demostraciones sobre los CRN. Puede aportar a entender un poco mejor la técnica, pero en si no contribuye a nuestro proyecto.

12.11. Dynamic Programming and Optimal Control - Vol I

archivo: Dynamic Programming and Optimal Control - Vol I Bertsekas 3rd Ed.pdf

Libro general de control óptimo y programación dinámica. Util para entender las bases teóricas.

12.12. Dynamic Programming and Optimal Control 3rd Edition, Volume II Approximate Dynamic Programming

archivo: Dynamic Programming and Optimal Control 3rd Edition, Volume II Approximate Dynamic Programming.pdf

Libro general de control óptimo y programación dinámica. Util para entender las bases teóricas.

12.13. Improving the performance of Stochastic Dual Dynamic Programming

Vitor L. de Matos a, * , Andy B. Philpott b , Erlon C. Finardi c

a Plan4 Engenharia, Rua Maria Luiza Agostinho, n 92, Itacorubi, Florianopolis, Brazil

b Electric Power Optimization Centre, University of Auckland, New Zealand
c Laboratório de Planejamento de Sistemas de Energia Elétrica, Universidade Federal de Santa Catarina, Brazil

archivo: Improving the performance of Stochastic Dual Dynamic Programming.pdf

Journal of Computational and Applied Mathematics 290 (2015) 196–208

Contents lists available at ScienceDirect

Journal of Computational and Applied
Mathematics

journal homepage: www.elsevier.com/locate/cam

Trabajo útil para entender un poco mejor la SDDP y contiene estrategia de recorrido del árbol de escenario y de selección de PLANOS de aproximación que pueden resultar útiles.

12.14. Optimal Long-Term Distributed Generation Planning and Reconfiguration of Distribution Systems: An Accelerating Benders' Decomposition Approach

Salman Khodayifar 1 · Mohammad A. Raayatpanah 2 · Abbas Rabiee 3 ·

Hamed Rahimian 4 · Panos M. Pardalos 5

Received: 26 June 2018 / Accepted: 28 July 2018 / Published online: 10 August 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

archivo: Khodayifar2018_Article_OptimalLong-TermDistributedGen.pdf

[ps://doi.org/10.1007/s10957-018-1367-5](https://doi.org/10.1007/s10957-018-1367-5)

Este trabajo aplica la SDP a un caso de generación distribuida y considera (en forma simplificada) las restricciones de red. Si bien no aporta para la etapa actual del proyecto seguramente nos pueda servir para la segunda etapa cuando hagamos pruebas con restricciones de transporte.

12.15. On variance reduction of mean-CVaR Monte Carlo estimators

Václav Kozmík

archivo: Kozmík2015_Article_OnVarianceReductionOfMean-CVaR.pdf

Comput Manag Sci (2015) 12:221–242

DOI [10.1007/s10287-014-0225-7](https://doi.org/10.1007/s10287-014-0225-7)

ORIGINAL PAPER

Este trabajo presenta el análisis de técnicas de reducción de varianza para la estimación de la cola de riesgo de una distribución. En la etapa actual del proyecto, estamos avocados a la obtención de la

política óptima en base a la estimación de la función de valor (valor esperado del costo futuro de operación). En SimSEE existe también la posibilidad de considerar la aversión al riesgo mediante combinación del valor esperado con medidas de riesgo como son VaR y CVaR. Una vez que tengamos operativos los nuevos algoritmos, talvez como trabajos futuros, tendríamos que habilitar la posibilidad de introducir en los nuevos algoritmos el concepto de aversión al riesgo y este trabajo puede resultar útil para ello.

12.16. Multi-stage stochastic optimization applied to energy planning

archivo: Multi-stage stochastic optimization applied to energy planning.pdf
Mathematical Programming.
M. V. F. Pereira L. M. V. G. Pinto
May 1991, Volume 52, Issue 1–3, pp 359–375 | Cite as

Este es clásico. Tiene la aplicación de la Dual Dynamic Programming al sector eléctrico.

12.17. SDDP vs. ADP: The Effect of Dimensionality in Multistage Stochastic Optimization for Grid Level Energy Storage.

Tsvetan Asamov, Daniel F. Salas and Warren B. Powell
Department of Operations Research and Financial Engineering, Princeton
archivo: SDDP-ADP.pdf

Interesante comparación entre SDDP y ADP. Me parece que resultará interesante al final del proyecto comparar nuestros resultados con estos.

12.18. Neuro-Dynamic Programming

Neuro-Dynamic Programming (D P Bertsekas, J N Tsitsiklis).djvu
Este libro presenta la aplicación de Redes Neuronales a la programación dinámica estocástica aproximada. El propio autor comienza indicando que "no cree mucho en las RN" pero que dado que ya no tiene con qué luchar contra la Maldición de Bellman nada pierde con probar.

12.19. On the solution variability reduction of Stochastic Dual Dynamic

Programming applied to energy planning

Murilo Pereira Soares a , * , Alexandre Street a , Davi Michel Valladão b

a Electrical Engineering Department, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro, RJ 22451-900, Brazil

b Industrial Engineering

archivo: On the solution variability reduction of Stochastic Dual Dynamic Programming applied to energy planning.pdf

European Journal of Operational Research 258 (2017) 743–760

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Este artículo presenta dos técnicas para "regularización de variables" una basada imponer costo a las soluciones de despacho con mayor varianza y otra basada en imponer sobrecostos a las soluciones de varianza alta en el costo futuro (midiendo las colas de riesgo). No tengo claro que nos pueda servir, pero puede ser que cuando se quiera imponer "parsimonia en la aproximación" (entendiendo por esto baja variabilidad de los parámetros en el tiempo) haya que considerar los efectos sobre la variabilidad de los resultados o incluso que la parsimonia deba realizarse con una ponderación sobre las colas de riesgo de las aproximaciones de CF.

12.20. Approximate Dynamic Programming

Solving the Curses of Dimensionality

Second Edition

Warren B. Powell

Princeton University

The Department of Operations Research and Financial Engineering

Princeton, NJ

archivo:

Powell_ApproximateDynamicProgrammingSolvingTheCursesOfDimensionality2edWiley2011.pdf

Resume la problemática de la programación dinámica estocástica aproximada y sirve como base conceptual para el trabajo de este proyecto.

12.21. Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling

Timo Lohmann ^a, Amanda S. Hering ^b, Steffen Rebennack ^a, *

Division of Economics and Business, Colorado School of Mines, Golden, Colorado, USA
Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, Colorado, USA
archivo: Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling.pdf
European Journal of Operational Research 255 (2016) 243–258
Contents lists available at ScienceDirect
European Journal of Operational Research
journal homepage: www.elsevier.com/locate/ejor
Innovative Applications of O.R.

Este trabajo muestra como introducir el pronóstico de aportes hidráulicos en el caso de la SDDP. No es directamente aplicable a nuestro caso dado que este tema lo tenemos ya resuelto dentro de los CEGHs pero puede resultar interesante conocer esta solución.

12.22. Stochastic dual dynamic programming applied to nonconvex hydrothermal models

Santiago Cerisola [↑], Jesus M. Latorre, Andres Ramos
Instituto de Investigación Tecnológica, ICAI, Universidad Pontificia Comillas, Alberto Aguilera 23,
28015 Madrid, Spain
archivo: Stochastic dual dynamic programming applied to nonconvex hydrothermal models.pdf
European Journal of Operational Research 218 (2012) 687–697
Contents lists available at SciVerse ScienceDirect
European Journal of Operational Research
journal homepage: www.elsevier.com/locate/ejor
Stochastics and Statistics
archivo: Stochastic dual dynamic programming applied to nonconvex hydrothermal models.pdf

Este trabajo muestra una técnica para aplicar SDDP en el caso de una función de costo no convexa. La técnica no es otra cosa que dividir la función en zonas convexas y agregar variables booleanas que determinen la Zona activa.

En nuestro proyecto dado que no vamos a imponer que los aproximadores sean convexas tendríamos que poder resolver problemas no-convexas. Este trabajo igual puede ser de utilidad dado que una posible estrategia es combinar aproximadores convexas con variables enteras que

12.23. Programación Dinámica Estocástica Revisión Bibliográfica

Se adjunta informe con resumen (archivo: InformeSergio_PDE_resumen.pdf) realizado por Sergio Martinez como parte de su trabajo en este proyecto.

13. Conclusiones y trabajos futuros.

Se finaliza el proyecto con los objetivos cumplidos y con los algoritmos operativos en modo de testeo. Los resultados obtenidos para las salas de prueba y las salas utilizadas en las programaciones energéticas del sistema uruguayo dieron acordes a lo esperado.

Se demostró que la herramienta SimSEE es flexibles y potentes al momento incorporar nuevas mejoras en la política de operación

La idea es incorporar complejidad a la red neuronal que modela la función CF en función de los requerimientos de precisión que se necesiten en cada paso de tiempo. Debido a que la función de CF condensa la información de la política de operación óptima en función de las variables de estado del sistema, dicha información tiene una determinada dependencia con respecto a cada variable en particular, pudiendo estar más influida por las variaciones en el estado en una variable determinada y menos en otra. Además, la función de CF es representada mediante un modelo que tiene cierta complejidad, la cual puede ser cuantificada como la cantidad de parámetros de dicho modelo. A partir de lo anterior se propone administrar, utilizando un mecanismo automatizado, cuánta complejidad se asigna a la representación de la función de CF en cada paso de tiempo. Esto permitiría tener una representación de la función de CF de complejidad variable a lo largo del horizonte de simulación, teniendo mejor resolución en los pasos de tiempo en donde se necesita realizar una operación más precisa del sistema, y una representación más simple en donde existan múltiples soluciones de operación equivalentes que conducen al sistema por trayectorias similares que no presentan diferencias significativas en términos de costos.

En el caso en donde el modelo de la función de CF es una red neuronal, la propuesta consiste en tener una neurona “vacante” en el modelo, e intentar incorporarla cuando el algoritmo detecte que dicha incorporación produce beneficios en cuanto a una mejor operación del sistema.

En cuanto a la parsimonia, se llegó a resultados aceptables, pero identificamos que aun presenta algunos errores, los cuales seguiremos trabajando para encontrarlos. Además como trabajo a futuro también se quiere incorporar la idea parsimonia a nivel de etapas de programación, esto es, en el corto plazo tener una parsimonia con detalle horario donde se identifique parsimonia entre los momentos del día. Y luego a medida de que se aumentan las etapas de programación al mediano y largo pargo plazo perder el detalle horario y pasar a una parsimonia diaria o semanal para captar la estacionalidad. De esta manera se tendría una única sala pero separada por la parsimonia de cada etapa.

14. Anexo: Cronograma del proyecto.

El proyecto comenzó el 1/9/2019 y finalizó el 1/9/2020 de acuerdo a lo previsto.

FSE_1_2017_1_144926

Actividad	Hito	T1		T2			T3			T4			T5			T6			T7			T8			T9			Objetivo	Descripción		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24						
contratación de personal	no	1	1	1																								todos	Realización de llamado a interesados, selección y contratación.		
revisión bibliográfica	no	1	1	1	1																							1	cuanto su importancia para el caso de aplicación (sector eléctrico y operación permanente)		
elaboración de primeras propuestas	no	1	1	1	1																							3	Selección de metodologías de optimización aproximada a implementar y adaptación de la plataforma SimSEE para este objetivo		
Implementación de las propuetas	no					1	1	1	1	1																		1	Implementación de las metodologías seleccionadas y testeo de la operativa en conjunto con SimSEE		
Generación de caso de aplicación a la Planificación de inversiones											1	1	1															1	de inversiones en generación y se compararán los resultados con los de la herramienta actualmente en uso.		
Documentación de la propuesta y resultados del caso	si										1	1	1																de inversiones en generación y se compararán los resultados con los de la herramienta actualmente en uso.		
Depurado e implemetnación piloto en paralelo con VATES.													1	1	1	1	1											5	información que el VATES (en ejecución en ADME) para permitir la comparación de resultados.		
Evaluación comparada de resultados y documentación	si															1	1	1	1	1	1	1						5	ejecución en paralelo de implementado con VATES y generación de la documentación correspondiente.		
Ajustes y mejoras de la implementación.																												2	Revisión de lo implementado, mejora de escritura del código, correcciones y ajustes ssi corresponde y mejora de la documentación		
Informe final y taller de divulgación de resultados	si																									1	1	2	2	todos	Generación del informe final y de una presentación para divulgación de resultados. Realización de un taller de divulgación.

15. Anexo: Introducción al lenguaje de optimización.

Esta sección tiene por objetivo introducir el lenguaje utilizado en la descripción de las técnicas de optimización y resumir los conceptos básicos.

Problema de Minimización :

$$P): \begin{matrix} \min J(X) \\ @ X \in \Omega \end{matrix} \quad \text{ec.(60).}$$

Se dice que $J(X): \Omega \rightarrow \mathbb{R}$ es La **Función de Costo** del Problema, que X son Las **Variables Libres** del Problema (o **parámetros de búsqueda**) y que Ω es el **Dominio** de X o **Espacio de Búsqueda**.

Si \hat{X} es solución del problema se debe cumplir

$$J(X') \geq J(\hat{X}) \quad \forall X' | (X' \in \Omega) \quad \text{ec.(61) Condición necesaria de optimalidad.}$$

Si la función objetivo es diferenciable en punto \hat{X} la ec.61 implica que en un entorno $\varepsilon > 0$, suficientemente pequeño, de \hat{X} se debe cumplir

$$J(\hat{X} + \varepsilon \vec{u}) \simeq J(\hat{X}) + \nabla_x J(\hat{X}) \cdot \varepsilon \vec{u} \geq J(\hat{X}); \quad \forall \vec{u} | ((\hat{X} + \varepsilon \vec{u}) \in \Omega) \quad \text{lo que es equivalente a la ec.62.}$$

$$\nabla_x J(\hat{X}) \cdot \vec{u} \geq 0; \quad \forall \vec{u} | ((\hat{X} + \varepsilon \vec{u}) \in \Omega) \quad \text{ec.(62) Condición necesaria de optimalidad si es diferenciable J.}$$

En palabras, la ec.62 expresa que para que \hat{X} sea un optimo local, la proyección del gradiente respecto a vectores de movimiento suficientemente pequeños dentro de la zona factible debe ser positiva. Observar que si el conjunto de vectores de movimiento factible expanden el espacio de X entonces se debe cumplir directamente: $\nabla_x J(\hat{X}) = 0$. La única razón para que los vectores de movimiento factible para un $\varepsilon > 0$ suficientemente pequeño no expandan el espacio es que el punto \hat{X} se encuentre en la frontera del dominio. La ec.62 también implica, que si el punto está en la frontera del dominio, y si la frontera es diferenciable (se puede calcular la normal a la frontera y queda así definido un hiperplano tangente) entonces si \vec{t} es una dirección tangente a la frontera se deberá cumplir

$\nabla_x J(\hat{X}) \cdot \vec{t} = 0$ dado que los movimientos factibles son en los dos sentidos de \vec{t} . En forma más genérica, dado un punto de la frontera del dominio, se define el Sub-gradiente de la frontera en el punto al conjunto de vectores que permiten movimientos factibles de largo menor o igual a un $\varepsilon > 0$ suficientemente pequeño.

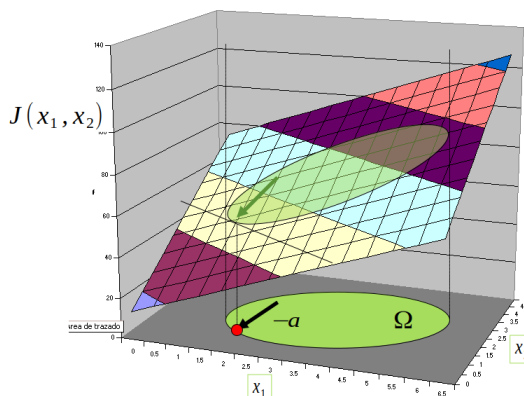


Fig. 44: Función de costo lineal.

Abusando de la notación escribiremos $\Omega(X) \leq 0$ para indicar que $X \in \Omega$ y $\Omega(X) = 0$ si y solo si el punto X se encuentra en la frontera de Ω .

Dado un punto \hat{X} y una dirección (vector) \vec{u} , decimos que es una dirección de movimientos factibles si $\exists \varepsilon > 0 / \forall \mu \in (0, \varepsilon); (\hat{X} + \mu \vec{u}) \in \Omega$. Esto nos permite definir el conjunto de direcciones factibles dado un punto \hat{X} como $F_\Omega(\hat{X}) = \{ \vec{u} / (\exists \varepsilon > 0 / \forall \mu \in (0, \varepsilon); (\hat{X} + \mu \vec{u}) \in \Omega) \}$.

Llamaremos **Sub Gradiente** (de la frontera del dominio en un punto dado) de $\Omega(X)$ y lo anotaremos como: $S_\Omega(\hat{X})$ al conjunto de vectores (direcciones) que proyectan un valor negativo sobre todos los vectores de direcciones factibles.

$$S_\Omega(\hat{X}) = \{ \vec{v} / (\vec{v} \cdot \vec{u} \leq 0; \forall \vec{u} \in F_\Omega(\hat{X})) \}$$

Observar que con esa definición, $S_\Omega(\hat{X})$ es convexo. Dados dos vectores, su combinación lineal convexa pertenece al conjunto.

$$\vec{v}_1 / (\vec{v}_1 \cdot \vec{u} \leq 0; \forall \vec{u} \in F_\Omega(\hat{X})) \quad \vec{v}_2 / (\vec{v}_2 \cdot \vec{u} \leq 0; \forall \vec{u} \in F_\Omega(\hat{X})) \quad \text{y} \quad \vec{v} = \alpha \vec{v}_1 + (1 - \alpha) \vec{v}_2 \quad \text{con} \quad \alpha \in [0, 1]$$

$$\vec{v} \cdot \vec{u} = (\alpha \vec{v}_1 + (1 - \alpha) \vec{v}_2) \cdot \vec{u} = \alpha \vec{v}_1 \cdot \vec{u} + (1 - \alpha) \vec{v}_2 \cdot \vec{u} \leq 0 \quad \text{dado que si} \quad \alpha \in [0, 1] \quad \text{se cumple} \quad \alpha > 0 \quad \text{y} \quad (1 - \alpha) \geq 0$$

Si la frontera del dominio, está descrita por una función diferenciable $\vec{n} = -\frac{\partial}{\partial X} \Omega(X)$ es un vector normal saliente en la frontera y $S_\Omega(\hat{X}) = \{ \vec{n} \lambda; \forall \lambda > 0 \}$.

Si la solución \hat{X} de P es interior al conjunto Ω entonces se debe cumplir $\nabla_x J(\hat{X}) = 0$ para que \hat{X} sea solución de P . En palabras, el gradiente de la función de costo es nulo en el punto solución si el mismo es interior al dominio. Si el punto solución se encuentra en la frontera del dominio, en ese punto el gradiente de la función de costo tiene que ser tal que no tenga componente sobre el subespacio tangente a la frontera del dominio en el punto o lo que es equivalente

$\nabla_x J(\hat{X}) \in \frac{\partial}{\partial X} \Omega(\hat{X})$. O expresado de otra forma, el gradiente debe ser co-lineal con una normal a la frontera en el punto $\nabla_x J(\hat{X}) = \lambda \vec{n}_\Omega(\hat{X})$.

Esta explicación no intenta ser rigurosa, la frontera podría no ser diferenciable en el punto entonces no habría un único vector normal a la frontera. Lo que importa aquí es reforzar el concepto de que $-\nabla_x J(\hat{X})$ es la dirección en que por lo menos en un entorno de \hat{X} , en caso de poder mover \hat{X} en una dirección que proyecte sobre esa dirección, se obtendría un valor menor de la función de costo y por tanto para que \hat{X} sea un mínimo, no debe ser posible realizar tal movimiento y para ello el gradiente del costo tiene que ser colineal con una normal a la frontera. En la literatura se conoce como **subgradiente**, al conjunto de vectores normales a los planos que dejan al dominio del mismo lado en un entorno del punto.

Como el dominio está generalmente descrito por la intersección de varias condiciones que se deben cumplir, es común

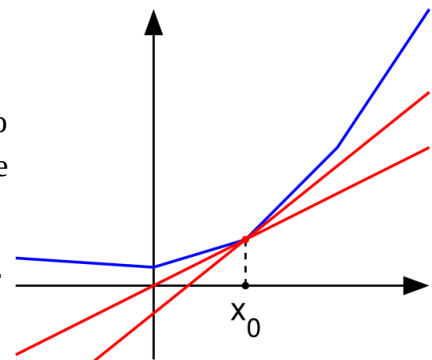


Fig. 45: Subgradiente en la frontera.

que las fronteras del dominio presenten quiebres y por lo tanto sean no diferenciables como se ejemplifica en la Fig.45.

Si la función de costo $J(X)$ tiene la forma de la ec.63 se dice que el problema tiene **función de costo lineal** (ver ej. Fig.44). Observar que $\nabla_x J(X)=a$. Si el vector $a=0$ la solución es trivial y cualquier punto de Ω tiene el mismo valor y es solución. Si el vector $a \neq 0$, entonces el gradiente de la función de costo no se anula en el interior del dominio por lo cual necesariamente la solución del problema está en la frontera de Ω . Sin pérdida de generalidad podemos decir que el óptimo se alcanza en la frontera en ambos casos.

$$y = a^T X + c \quad \text{dónde: } y, c \in \mathbb{R}, a, X \in \mathbb{R}^n \quad \text{ec.(63) Función lineal.}$$

Se define **Supremo del Problema** al conjunto de puntos que queda por encima de la función de costo sobre el Dominio del Problema. Para el problema P de la ec.60 la definición del Supremo sería la mostrada en la ec.64. En matemáticas se llama Epigraph o Supergraph de una función al conjunto de puntos que queda por encima de la función. El Supremo, es el subconjunto del Epigraph que cumplen $X \in \Omega$. (solo para complementar la terminología, a la parte inferior de la función se le llama Hypograph o Subgraph).

$$\text{Sup}(P): \{(X, y) / X \in \Omega, y \in \mathbb{R}, y \geq J(X)\} \quad \text{ec.(64) Supremo de } P.$$

Observar que el Supremo de un Problema tiene toda la información que define el problema. La solución del Problema, puede pensarse como la búsqueda del punto de menor y en el supremo con la definición de y dada en la ec.64.

Hay diferentes técnicas para buscar el óptimo. Muchos algoritmos son del tipo de **búsqueda local** que consiste en partir de un punto y buscar localmente (en un entorno del punto) otro punto en que la función de costo sea menor, avanzar a ese punto y volver a aplicar el algoritmo. Cuando la búsqueda se realiza seleccionando una dirección y buscando en esa dirección se dice que el algoritmo es de **búsqueda local lineal**. Cuando la dirección seleccionada en cada punto es menos el gradiente de la función de costo se tiene el algoritmo conocido como **búsqueda lineal por máximo descenso**. Los algoritmos de búsqueda local, pueden llegar a un **mínimo local**. Es decir un punto que en su entorno es la mejor solución, pero que no es La Solución.

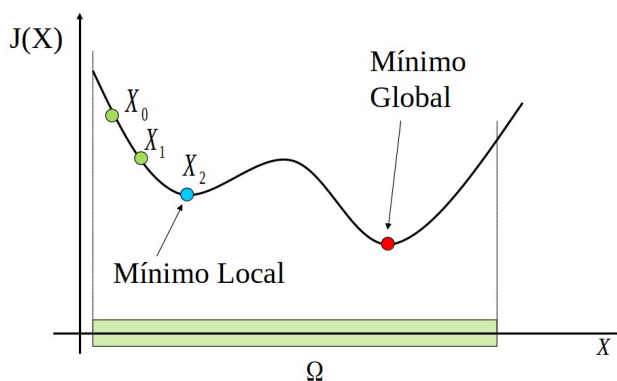


Fig. 46: Mínimo Local.

Se dice que un **Conjunto es Convexo** cuando dados cualquiera dos puntos del conjunto, el segmento que los une

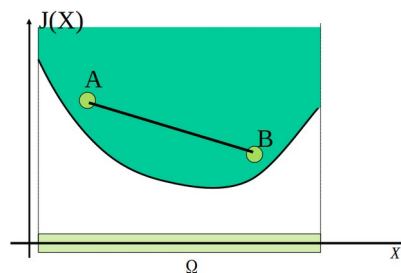


Fig. 47: Problema Convexo.

pertenece al Conjunto. Respecto de un Problema de Minimización, se dice que El Problema es Convexo, cuando su Supremo es Convexo como se ejemplifica en la Fig.47. Al ser el Supremo Convexo, todos los puntos “se ven entre si” en el sentido de que dados dos puntos cualesquiera la línea recta que los une también pertenece al Supremo. Esta propiedad permite asegurar, que los **algoritmos de búsqueda local** son capaces de encontrar la solución al Problema de Minimización. La convexidad del problema entonces asegura, que si se obtiene un mínimo con un algoritmo de búsqueda local, ese mínimo es El Mínimo del Problema. En la literatura matemática, también se define la convexidad de un problema a partir de exigir la convexidad de la función de costo sobre el dominio y de la convexidad del dominio en sí, condiciones que son equivalentes a exigir la convexidad del Supremo. Algunas propiedades útiles de los conjuntos convexos son:

a) La intersección de convexos es convexa. La Fig.48 se muestra solo para fijar la idea.

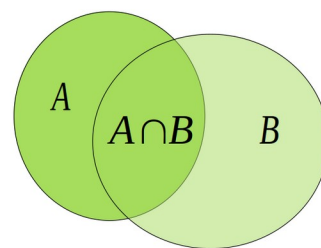


Fig. 48: La intersección de convexos es convexa.

b) Dado un convexo y un punto en su frontera existe un hiperplano que deja todos los puntos del convexo del mismo lado y que pasa por el punto dado. Decimos que el plano es tangente al convexo en el punto dado.

c) Dado un punto de $V \in \mathbb{R}^{n+1}$ y un convexo $C \subset \mathbb{R}^n$ el cono de proyección formado por los puntos de todas las semirectas con origen en V y que pasan por los puntos $(X, 0) \in \mathbb{R}^{n+1}$ (es decir puntos de C con cero en la dimensión agregada)

$[P = V + ((X, 0) - V)\lambda / X \in C, \lambda \in \mathbb{R}^+]$ es un convexo. En el extremo en que el punto V es alejado al infinito, se tiene la proyección cilíndrica del convexo C .

La definición del dominio Ω se suele realizar en base a un conjunto **restricciones** sobre el vector de variables libres $X \in \mathbb{R}^n$ dadas por funciones como la ec.65 **restricciones de desigualdad** y como la ec.66 **restricciones de igualdad**.

$$g_i(X) \leq 0 \quad ; \quad \text{dónde: } g_i: \mathbb{R}^n \rightarrow \mathbb{R} \quad \text{ec.(65) Restricción de desigualdad.}$$

$$h_i(X) = 0 \quad ; \quad \text{dónde: } h_i: \mathbb{R}^n \rightarrow \mathbb{R} \quad \text{ec.(66) Restricción de igualdad.}$$

Cada restricción define un conjunto Ω_i y el cumplimiento de todas las restricciones define el conjunto $\Omega = \bigcap_i \Omega_i$. Si cada uno de los Ω_i es convexo, aplicando que la intersección de convexos es convexa, se tiene que Ω es convexo.

Cada restricción de igualdad se puede representar como dos de desigualdad por lo que obviaremos cuando sea posible hacer la distinción entre ambas.

Observar que en el caso particular de **restricciones lineales** (las g_i como la ec. 63), cada $g_i(X) = 0$ define un hiperplano en \mathbb{R}^n que divide \mathbb{R}^n en dos subespacios, el que cumple la restricción Ω_i y su complemento en \mathbb{R}^n . Como dados dos puntos cualesquiera de un mismo lado

de un hiperplano, el segmento que los une está del mismo lado respecto del hiperplano los Ω_i así definidos son convexos y por tanto Ω es convexo.

Si un problema, tiene función de costo lineal y dominio definido por un conjunto de restricciones lineales, decimos que es un **Problema Lineal**. Como ya se mencionó, por ser la función de costo lineal, un punto de la frontera de Ω es óptimo. También se cumple que como la frontera de Ω es lineal a tramos, la función de costo o bien toma el mismo valor sobre un tramo lineal o toma valores decrecientes en un sentido y por tanto la solución está en la intersección de los tramos (vértices del poliedro Ω , ver Fig.49) lo que da lugar a lo que se conoce como el método Simplex.

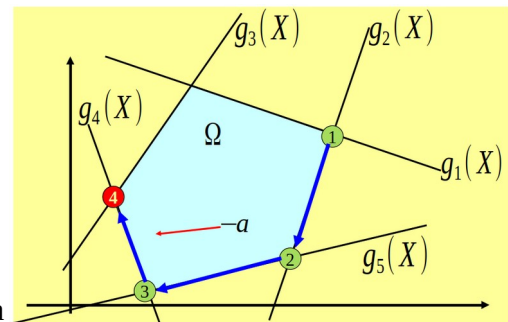


Fig. 49: Método Simplex.

La búsqueda del mínimo puede ser más o menos compleja según la forma del Supremo y su convexidad o no. Una técnica para abordar problemas en que la exploración del Supremo resulta compleja, se han desarrollado diferentes técnicas. Una familia de esas técnicas consisten en lo que se conoce como **Técnicas de Relajación**. Estas técnicas consisten en “aflojar” (relajar) el Problema (o su Supremo que es lo mismo) con el fin de facilitar la exploración y luego ir “ajustando” (desrelajando) el Problema y mejorando la estimación de la solución.

Se dice que un Problema de minimización P_B es una **Relajación** de otro problema P_A si el Supremo de P_B contiene al Supremo de P_A . (en notación de conjuntos sería $\text{Sup}(P_B) \supseteq \text{Sup}(P_A)$). La Fig. 50 muestra un ejemplo. Como se puede apreciar, la solución del problema P_B (círculo celeste en la Fig.50) es necesariamente una cota inferior de la solución del problema P_A (círculo rojo en la Fig.50) dado que la solución del problema relajado, es el punto extremo del Supremo en la dirección de costos decrecientes y como el Supremo del problema original está contenido en ese conjunto, su punto extremo en la misma dirección necesariamente es superior o igual al óptimo del problema relajado.

Adicionalmente, si el valor de X correspondiente a la solución del problema relajado, está dentro del dominio del problema original (decimos que el punto es **factible**) es posible entonces calcular la función de costo en el problema original en ese valor de X (círculo amarillo en la Fig.50) que será una cota superior del óptimo del problema original. Esta propiedad de obtener una cota inferior al resolver un problema relajado y eventualmente una superior (si resultó que la solución es factible en el original) permite adoptar técnicas de relajaciones sucesivas, comenzando por problemas “flojos” e ir ajustando la relajación (ajustando el Supremo del problema relajado al Original) obteniendo sucesivas cotas mínimas y máximas. Si en cualquier iteración, la cota mínima iguala a la máxima se ha obtenido la solución del problema. Adicionalmente, si la diferencia entre las cotas es pequeña, puede resultar suficiente la aproximación

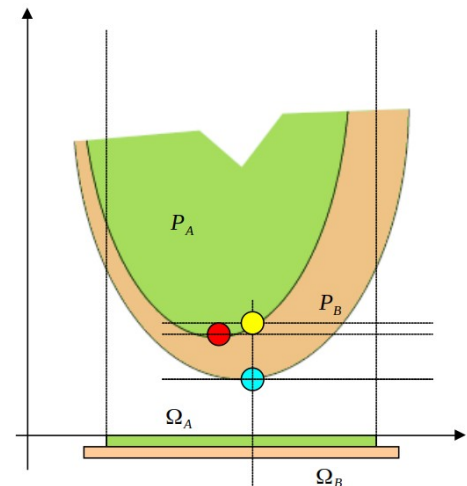


Fig. 50: Solución del problema relajado y cotas del original.

obtenida factible (la menor de las cotas superiores logradas durante las iteraciones). A modo de ejemplo si el problema es buscar un despacho factible de las centrales de generación en una hora, y la cota superior es 10020 USD y la inferior 10000 USD seguramente no resulte de interés continuar “desrelajando” el problema para obtener el óptimo pues se sabe que a los sumo habrá un ahorro de 20 USD que seguro está por debajo del error con que se plantea el problema de despacho.

Un ejemplo de técnica de relación paramétrica, es la **Relajación Lagrangina**. Consiste en crear un problema relajado sumando a la función de costo aquellas funciones que definen las restricciones que se desean relajar mutiplicadas por un parámetro de la relajación. La ec.68 muestra El Problema Original, en el que se ha separado una restricción de las que definen el dominio de forma tal que el Dominio original es: .

$$\Omega = \{X / (g(X) \leq 0)\} \cap \Omega' \quad \text{ec.(67) Dominio Original.}$$

La ec.69 muestra el problema relajado con el parámetro $\beta > 0$.

$$P): \quad \min_X J(X) \quad \text{ec.(68) Problema Original.}$$

$$\text{@} \left| \begin{array}{l} g(X) \leq 0 \\ X \in \Omega' \end{array} \right.$$

$$P_\beta): \quad \min_X J(X) + \beta g(X) \quad \text{ec.(69) Problema Relajado } P_\beta .$$

$$\text{@} \left| \begin{array}{l} \beta \geq 0 \\ X \in \Omega' \end{array} \right.$$

Se suele llamar Lagrangina a la función objetivo del problema relajado.

$$L(X, \beta) = J(X) + \beta g(X) \quad \text{ec.(70) Lagrangiana.}$$

Observar que $\Omega = \{X / (g(X) \leq 0)\} \cap \Omega'$ implica que el dominio de P_β incluye al original dado que la restricción $\beta \geq 0$ es respecto de una nueva dimensión del Dominio y no impone restricciones sobre X y el conjunto $\Omega' \supset \Omega$ por ser Ω . Además, para los puntos del dominio original, se cumple que $g(X) < 0$ y siendo $\beta > 0$ se cumple entonces que: $J(X) + \beta g(X) \leq J(X)$ y por tanto el Supremo de P_β incluye el Supremo de P y por tanto P_β es un problema relajado de P

Sea X_β la solución de P_β , $L(X_\beta, \beta)$ es cota inferior de El Óptimo (del problema original). Observar que en la zona en que $g(X) < 0$ la forma de aumentar la cota inferior (ajustar la relajación, o desrelajar) es disminuir β y por tanto, si la solución del problema es un punto interior

del Dominio, en las sucesivas desrelajaciones $\beta \rightarrow 0$. Por otra parte, si $g(X_\beta) > 0$, la forma de lograr que X_β se mueva hacia la zona factible es aumentando β dando así más peso en la función de costo del problema relajado al costo asignado al incumplimiento de la restricción.

La solución del problema relajado X_β debe cumplir la condición de optimalidad ec.62

$$\nabla_x L(X_\beta, \beta) \cdot \vec{u} = (\nabla_x J(X_\beta) + \beta \nabla_x g(X_\beta)) \cdot \vec{u} \geq 0 \quad \text{ec.(71)}$$

Para todo \vec{u} para los que exista movimientos factibles en un entorno de X_β

Observar que la ec.71 implica que si X_β es interior a Ω' entonces el conjunto de direcciones de movimiento factible \vec{u} expande el espacio de X_β entonces se debe cumplir

$(\nabla_x J(X_\beta) + \beta \nabla_x g(X_\beta)) \cdot \vec{u} = 0$ dado que es posible moverse en ambas direcciones de cualquier vector \vec{u} . En particular, si \vec{u} es tangente a la restricción relajada se cumple $\beta \nabla_x g(X_\beta) \cdot \vec{u} = 0$ y por tanto se cumplirá $\nabla_x J(X_\beta) \cdot \vec{u} = 0$ lo que implica que $\nabla_x J(X_\beta)$ es ortogonal a los vectores tangentes a la restricción relajada o lo que es lo mismo es colineal con la normal a dicha restricción (o con algún vector del subgradiente de la misma).

Observar que si la solución del problema original está en la frontera definida por $g(X) = 0$ es posible encontrar un valor de β que hace la solución del problema relajado coincida con la del original haciendo que se cumpla que el gradiente de la Lagrangiana calculado en el punto es nulo ec.

Por el absurdo, si X_β

16. Anexo: Caso base para prueba de administrador de estados multiplano.

16.1. Introducción.

En el marco del proyecto **ANII_FSE_1_2017_1_144926** se busca la construcción de un Administrador de Estados a partir de un conjunto de hiperplanos para la representación del Costo Futuro en la aplicación SimSEE. Los coeficientes de los hiperplanos son calculados a partir de iterar los resultados de simulación. Se busca verificar la metodología implementada para la construcción de hiperplanos. Para la realización de las pruebas se construyó una sala base y una plantilla SimRes3 descriptas a continuación.

Descripción de la sala:

- Versión del SimSEE: v_ii5
- Horizonte de simulación 29/10/2018 16:00:00 a 12/11/2019 16:00:00
- Sala horaria (9096 pasos de simulación)
- Actores:
 - Bonete:
 - 7 metros de cota de descarga
 - Potencia total instalada 155.2 MW (4 máquinas de 38.8 MW)
 - Fuente de aportes CEGH (SintetizadorCEGH_BPS_iN34_cmoBR_Horario_SVE)
 - Cota inicial 80.125 m
 - Demanda:
 - Plana de 100 MW
 - Un escalón de falla de 200 USD/MWh
 - Sumidero a 0.1 USD/MWh

Descripción de la plantilla SimRes3:

- Variables crónicas:
 - Bonete_Vol: Volumen embalsado en el lago del Rincón del Bonete
 - CFIni: Costo futuro al inicio del paso
 - cdp: Costo del paso
 - CF_s_q: Estimación del costo futuro de la siguiente iteración (PostOper combinarDespCronVars)

$$CF_{k+1}(t, X(t)) = cdp + q \cdot CF_k(t + \Delta t, X(t + \Delta t))$$

Donde:

- q coeficiente de actualización
- X vector de variables de estado (Volumen de Bonete)
- CF_k costo futuro de la iteración actual

- CoeficientesHiperplano: Elementos de la matriz A que aproxima de forma lineal la relación entre X y CF_{k+1} (PostOper AproLin)

$$AX = C F_{k+1}$$
- PDem: Potencia de la demanda (100 MW planos)
- PBon: Potencia generada por Bonete
- PFalla: Potencia de falla
- cotaBon: Cota de Bonete
- Bon_ce: Coeficiente energético de Bonete (MWh/m³)
- Bon_VVert: Volumen vertido por Bonete (Hm³)
- CFVECT: Crónicas del costo futuro vectorizadas
- BONVOLLECT: Crónicas del volumen de Bonete vectorizadas

16.2. Pruebas Realizadas

Primera prueba

- Descripción:
 - Se simulan 100 crónicas y se observa la relación entre el costo futuro y la variable de estado partiendo de un conjunto de hiperplanos con un hiperplano idénticamente nulo para cada paso de tiempo.
- Observaciones:
 - El costo futuro es “independiente” de la variable de estado
Se distinguen varios casos:
 - Todas las máquinas disponibles y con recurso disponible (la central es capaz de generar a pleno). La generación es de 155.2 MW, de los cuales 100 MW cubren la demanda y los restantes se exportan a 0.1 USD/MWh por lo que el CF es - 5.52 USD.
La dependencia con la variable de estado es no lineal si el volumen embalsado es suficiente para generar a pleno.
 - Lineal en el caso de que la central no pueda generar a pleno, pero logre abastecer la demanda. A menor volumen embalsado mayor será el CF.
 - En el caso que la generación sea menor a la demanda el costo futuro tomará valores de falla, dependiendo los aportes ocurridos. En este caso la central se comporta como una hidráulica de pasada.

Los dos últimos ítems ocurrieron en una sola crónica de 100, pero dicha crónica distorsiona los coeficientes de los hiperplanos ya que el Costo de Falla es de varios órdenes de magnitud mayor que el beneficio de exportación.

Por esto, los coeficientes se encuentran distorsionados al final del horizonte de simulación debido a que el sistema falló por no haber recurso hidráulico. En la Fig.1 se puede observar el efecto de haber vaciado el lago sobre el final de la tira de datos.

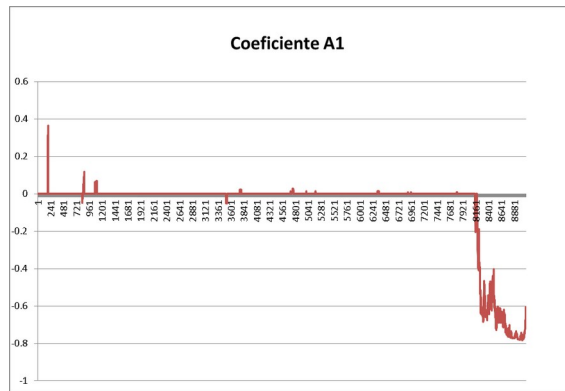


Figura 1 – Coeficiente A1, factor del hiperplano que multiplica al volumen de Bonete

Los coeficientes se encuentran distorsionados al principio de la simulación debido a que se parte del mismo estado para todas las crónicas, la aproximación por mínimos cuadrados no está bien condicionada.

En el primer punto el problema de mínimos cuadrados tiene infinitas soluciones, dado que es un único punto el que se quiere linealizar, por lo que se optó por agregar un término de ajuste, basado en la regresión Ridge. Al principio de la Fig.2 se puede observar el efecto del mal condicionamiento de mínimos cuadrados.

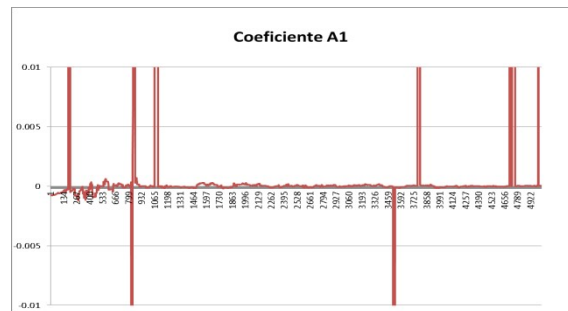


Figura 2 – Primeros valores del Coeficiente A1

Fuera de estas condiciones se presentan valores constantes cercanos a cero, lo que significa que el CF no presenta variaciones, a excepción de cuando hay roturas de máquinas que aparecen fallas nuevamente y los coeficientes toman valores positivos o negativos, dependiendo de los valores tomados por las crónicas.

Caso con coeficiente positivo con presencia de falla (En paso de tiempo 192):

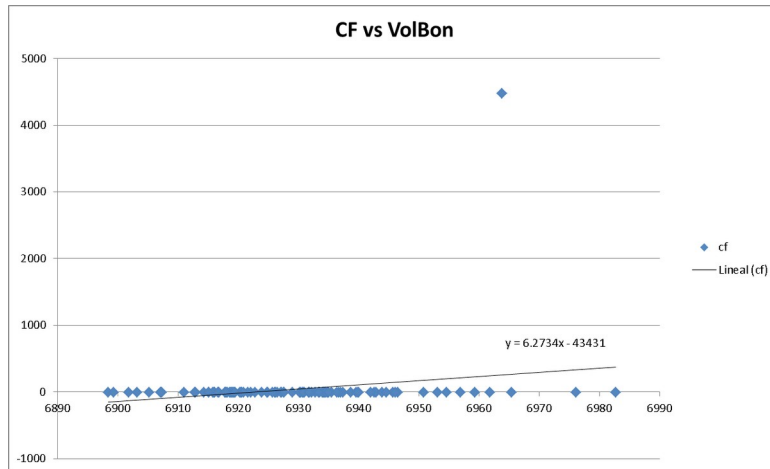


Figura 9 - Coeficiente positivo con presencia de rotura

Caso con coeficiente negativa con presencia de falla (En paso de tiempo 830):

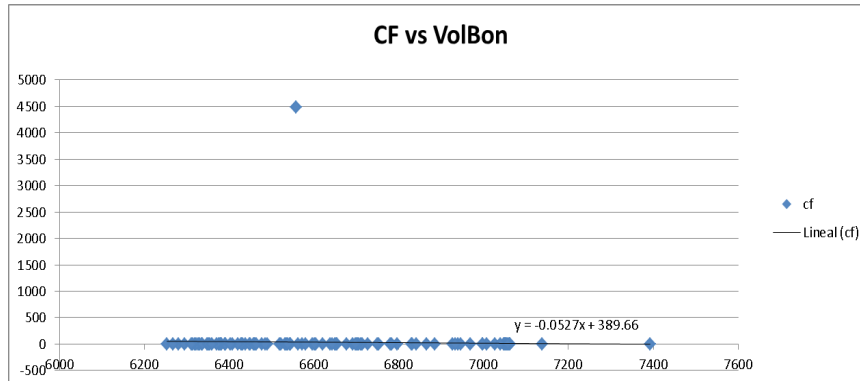


Figura 10 - Coeficiente negativo con presencia de rotura

El signo del coeficiente va a estar dado por el volumen al cual ocurrió la rotura. Si ocurrió a un volumen menor al punto medio del conjunto de volúmenes de ese paso de tiempo el signo será negativo y si ocurrió en un punto mayor, será positivo.

Esto no es deseable ya que este coeficiente es el opuesto al valor del agua en el caso de las centrales hidráulicas y puede hacer la diferencia en decisiones de vertimiento o exportación a sumidero.

- Al variar la cota inicial, y en particular al llevarla cerca del fondo del lago, se dio que todas las crónicas llegaron a volumen 0. Esta situación es similar al inicio donde el estado está condensado en un solo punto con la contrapartida de que el costo no es igual para todas. Como se mencionó anteriormente, al estar con el lago vacío, la central se empieza a comportar como una hidráulica de pasada, esto hace que los aportes que

reciba cada crónica sean turbinados y el resto de la demanda se cubra con falla. Esta situación genera una serie de puntos con volumen igual a 0 y con costos del paso distintos. En ese caso no estaría definida la pendiente del hiperplano en ese estado (en este caso la aproximación lineal implementada toma por convención fijar costo 0). En el caso de que en alguna de las crónicas haya aportes suficientes como para aumentar el volumen del embalse, la pendiente del hiperplano en esa dirección será muy grande en valor absoluto y de signo negativo. Esto se traduce en que el valor del agua termina siendo mayor al costo de falla y en el despacho de la próxima iteración se termina decidiendo por fallar en lugar de generar con la central hidráulica. En las siguientes iteraciones se realimenta esta situación a los pasos de tiempos anteriores y se obtiene una política de operación inestable y lejos de ser óptima.

▪ Conclusiones:

- El hecho de que las simulaciones partan de un mismo estado perjudica no solo el primer paso de tiempo, sino los primeros hasta que el ramillete se abre y se tiene mejor explorado el espacio de estados. Esto puede mejorar si las simulaciones parten de distintos estados iniciales.
- En situaciones límite, como es el lago vacío o cualquier situación donde el estado esté condensado alrededor de un punto, no está bien representado el espacio de estados ni la función de costo futuro y esto afecta la convergencia del algoritmo iterativo.

Segunda prueba

▪ Descripción:

- En base a los resultados de la prueba anterior, se decide tomar medidas para facilitar la convergencia del algoritmo y que las crónicas exploren mejor el espacio de estados.
 - Para la convergencia se opta por implementar un filtrado del tipo:

$$Y_{k+1} = \alpha Y_k + (1 - \alpha) X_{k+1}$$

Donde Y_k es el conjunto de hiperplanos utilizado por la iteración anterior y X_{k+1} es el conjunto de hiperplanos calculado en la iteración actual con el método descrito en la introducción. El conjunto utilizado en la iteración actual será Y_{k+1} . El valor de α comienza en 0.5 pero se acerca a 0 o a 1 si se cumple que el algoritmo iterativo representa una contracción o no respectivamente. Con esto se logra asimilar información de manera más lenta pero menos ruidosa.

- Para mejorar la exploración y por lo tanto la representación del espacio de estados se decide que las crónicas comiencen en distintos estados. Para esto se modifica el actor Bonete para tener una altura inicial de 76.5 m, un error de ± 6.5 m y se activa el checkbox `cbIniErrorSim`. Se eligen estos números ya que las cotas máximas y mínima de Bonete son 83 m y 70 m respectivamente y esto permite que las crónicas partan de cualquier parte del espacio de estados. También se modifica el factor de disponibilidad a 1 para evitar el problema de la prueba anterior y verificar la convergencia del algoritmo.

- Se implementa un script en R que permite graficar la evolución de las crónicas y la aproximación de la función de costo futuro, con el volumen de Bonete y el tiempo como variables independientes y el costo futuro como variable dependiente.
 - Se reduce el horizonte de simulación a 14 días para acortar el tiempo del experimento.
- Observaciones:
- Se logra que converja el algoritmo en 630 iteraciones. En la Figura 5 se puede observar la evolución del conjunto de hiperplanos al correr de las iteraciones. En el eje Y está el tiempo, en el x el volumen de Bonete y en el z el costo. Los puntos azules son todas las triadas (tiempo, volumen, costo futuro) y las curvas rojas son algunas crónicas en particular, la recta verde es una aproximación lineal para un paso dado y la superficie es la unión de todas las aproximaciones para todos los pasos de tiempo.

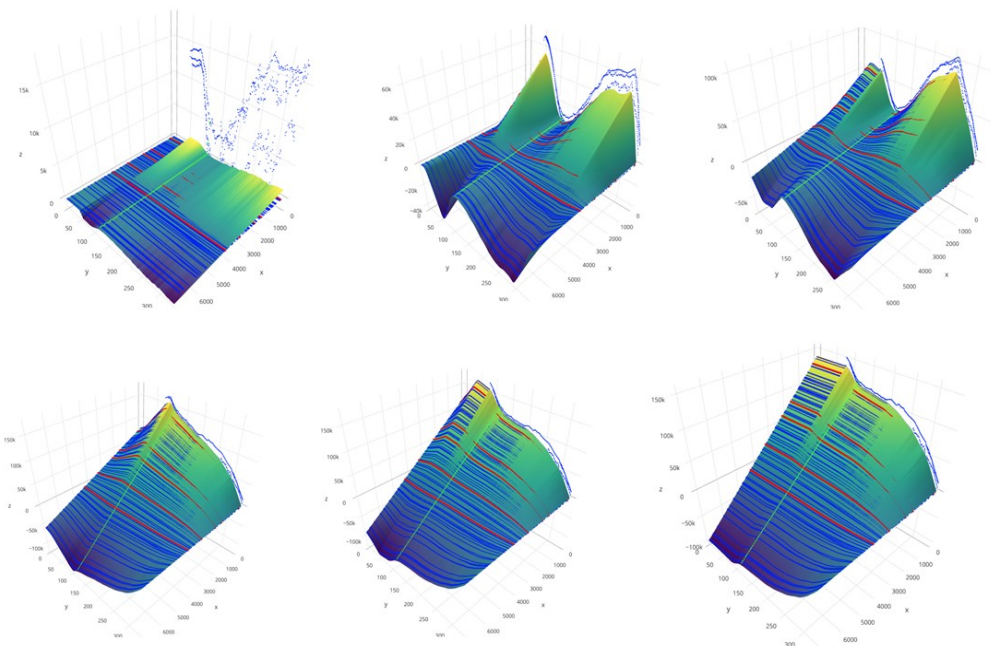
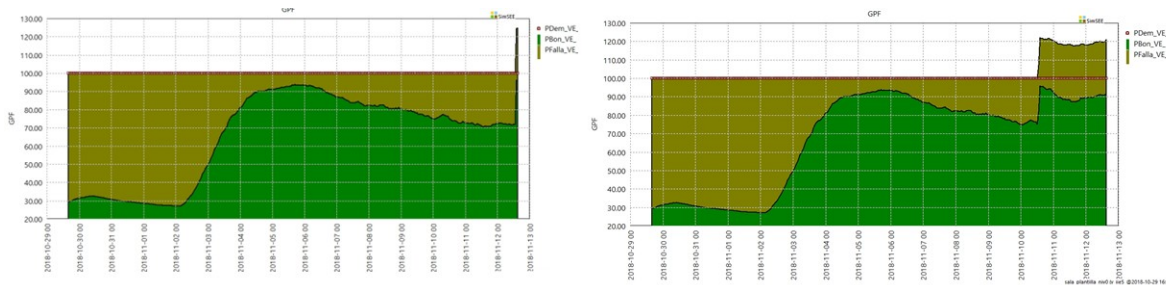


Figura 5 - Evolución de los conjuntos de hiperplanos en las iteraciones

- Se realiza la comparación con la misma sala utilizando la optimización estándar contra el conjunto de hiperplanos y se analizan los resultados. Se elige como cota inicial 71 m sin error al inicio de la simulación.
 - Se utiliza la optimización tradicional y se simulan 100 crónicas. En todas las crónicas se cubre la demanda sin necesidad de falla, se exporta al sumidero en los 2 últimos pasos con lo que el costo de abastecimiento como la integral del costo del paso sin aplicar el factor de actualización es de -10.99 USD en todas las crónicas. Se utiliza la optimización con los conjuntos de hiperplanos. El despacho es el mismo pero se exporta los últimos 52 pasos de tiempo con un beneficio mayor, -283.15 USD.
 - Se realiza la misma prueba con cota 80 m. Se exporta durante todo el período en el caso de la optimización tradicional, con un costo de -1850.71 USD. En el caso del conjunto de hiperplanos el resultado es igual.
 - Se realiza una prueba con cota 70 m. En este caso hay bastante despacho de falla en ambos casos. Se pueden observar los despachos en la Figura 6. El costo esperado promedio en el caso tradicional es de 2,338,788.79 USD, en el caso de los hiperplanos 2,348,173.82 USD. Hay una diferencia de 0.4 % a favor del caso tradicional.



Cota (m)	Costo esperado multiplanos (USD)	Costo esperado método tradicional (USD)	Diferencia relativa al método tradicional (%)
70	2,348,173.82	2,338,788.79	-0.40
71	-283.15	-10.99	2476.43
80	-1,850.71	-1,850.71	0.00

▪ Conclusiones:

- En el caso de las cotas 70 y 80 la diferencia entre los dos métodos fue mínima. En el caso de cota 71 el método de multiplanos dio un beneficio mayor ya que comenzó a exportar antes que el método tradicional, la diferencia relativa es grande pero esto se debe a que el costo del método tradicional fue bajo. De todas maneras se puede decir que ambos métodos dan resultados similares.

Tercera prueba

▪ Descripción:

- Se toma la misma sala que la prueba anterior pero ampliando el horizonte de simulación y se realizan las mismas pruebas.
- Horizonte de simulación: 29/10/2018 a 29/01/2019, 2208 pasos de simulación.

- Observaciones:
 - Se logra que converja el algoritmo en 3064 iteraciones. El criterio de convergencia es que los archivos de conjuntos de hiperplanos generados por dos iteraciones consecutivas sean idénticos. Esto llevó a que haya más iteraciones de las necesarias ya que en los archivos se imprimen valores con una cantidad de cifras significativas del orden de 14. En futuras pruebas se considerará un criterio menos estricto (e.g. que la máxima diferencia de los valores en p.u. sea menor que un 0.02).
 - Se realizan las mismas comparaciones que en la prueba anterior. Los resultados se pueden observar en la siguiente tabla.

Cota (m)	Costo esperado multiplanos (USD)	Costo esperado método tradicional (USD)	Diferencia relativa al método tradicional (%)
70	13,826,015.23	13,824,903.54	-0.01
71	7,665,611.23	7,662,467.72	-0.04
80	-12,010.56	-12,015.81	-0.04

- Conclusiones:
 - Las diferencias entre los métodos son mínimas pero siempre a favor del método tradicional con un menor costo o mayor beneficio. En esta prueba se observaron mejores resultados debido a que el horizonte de simulación fue mayor por lo que los efectos de borde tuvieron menor peso que en el caso anterior.

17. Anexo: Informe con estudio de caso de Planificación de Inversiones.

Igual procedimiento que el seguido para la Sala SimSEE de Corto plazo analizado en la sección 16 se siguió con una sala SimSEE de Largo Plazo correspondiente a una Planificación de Inversiones. El objetivo primario del análisis de este caso, era disponer de una base testigo de la Política de Operación que se puede obtener con un nivel bajo de complejidad (un hiperplano). Este caso base nos tendrá que servir para comparar en las etapas posteriores del proyecto las mejoras realizada.

Para nuestra sorpresa, la aproximación con un solo hiperplano por paso de tiempo ya resultó en una Política de Operación mejor que aquella obtenida por el método clásico. Esto se explica porque en el caso del método clásico, la PO se obtiene por la iteración de Bellman (reversa en el tiempo) lo que necesariamente tiene asociada simplificaciones del modelado que luego al simular (con el tiempo hacia adelante) no se realizan. La tabla 7 se muestra el resultado de la simulación. Corresponde al valor esperado, costo con riesgo 5% de excedencia (VaR(5%)) y valor esperado del costo condicionado a una excedencia de 5% (CVaR(5%)) del costo de la operación futura actualizado 5%.

Tabla 7: Comparación Caso Planificación de Inversiones - Clásico vs. aprox. 1 hiperplano por paso.

Costo esperado	Costo con riesgo 5% de ser excedido	Costo CVaR(5%)

	[MUSD]	[MUSD]	[MUSD]
Aprox. 1HP/paso	10927.11	12973.69	13249.75
Clásico SimSEE	11135.66	13184.18	13323.83
Diferencia	-208.55	-210.49	-74.08

Como se puede apreciar, el costo en valor esperado de la simulación con la Política de Operación obtenida con la aproximación de 1 hiperplano por paso de tiempo, resultó en 208 MUSD (doscientos ocho millones de dólares) inferior que el obtenido con la Política de Operación obtenida con el SimSEE con la metodología clásica. Esto implica una reducción en el costo de $208/11135 = 1.86\%$

Como se dijo antes es un resultado no esperado en esta etapa del proyecto, pero sin duda alentador en cuanto a que se está mejorando la herramienta disponible.

Al realizar la aproximación de la PO con un solo hiperplano por paso de tiempo, la misma es representable por el vector normal al hiperplano. En este caso las variables de estado son el volumen embalsado en el lago de Rincón de Bonete (VBon) y una variable de H que representa el estado hidrológico. La variable H en este caso captura la dependencia de las lluvias con la anomalía de la temperatura superficial del Océano Pacífico en la zona

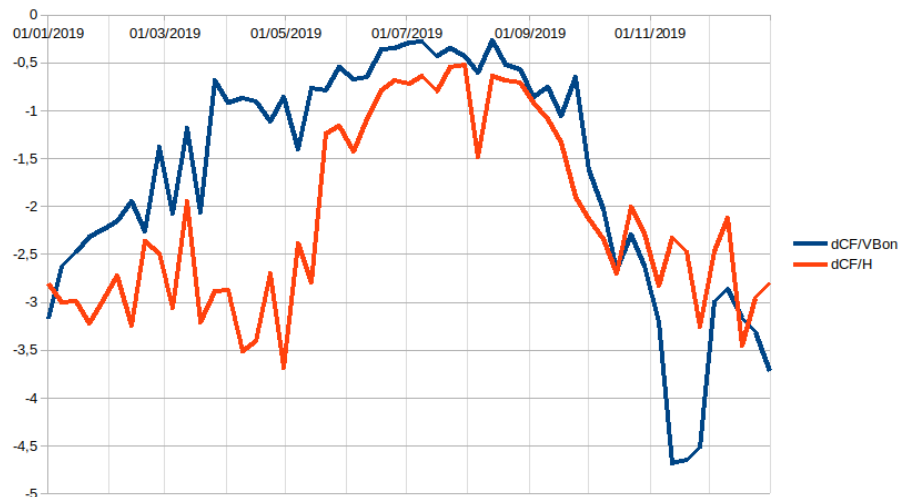


Fig. 51: Evolución anual de los coeficientes del hiperplano al inicio del período.

identificada como 3.4 y en la misma variable captura el agua que se encuentra escurriendo en las cuencas a los lagos. Las derivadas direccionales de CF respecto de ambas variables en por unidad de 8943 USD/Hm³ y 14324483 USD/u se representan en la Fig.51. Como se puede apreciar, en ambos casos las derivadas son negativas indicando que a mayor volumen (o mayor hidrulicidad) el costo futuro decrece. Se hace notar, que las derivada son mayores durante el verano (indicando que el agua tiene mayor valor en ese período) y decrecen significativamente durante el invierno lo que es consistente con la operación tradicional del lago.

La Fig.52 muestra la evolución de los coeficientes del hiperplano en todo el horizonte (largo plazo). Como se puede observar, la variabilidad anual mostrada en la Fig.51 se mantiene. También se observa que en el largo plazo van decreciendo los valores lo que puede tener dos explicaciones. 1) Al final del horizonte “es el fin del mundo” y por tanto hay que vaciar los lagos y 2) En la medida en que la demanda crece y el sistema hidráulico permanece constante la importancia del mismo en el sistema decrece. Seguramente el punto más fuerte es el 1).

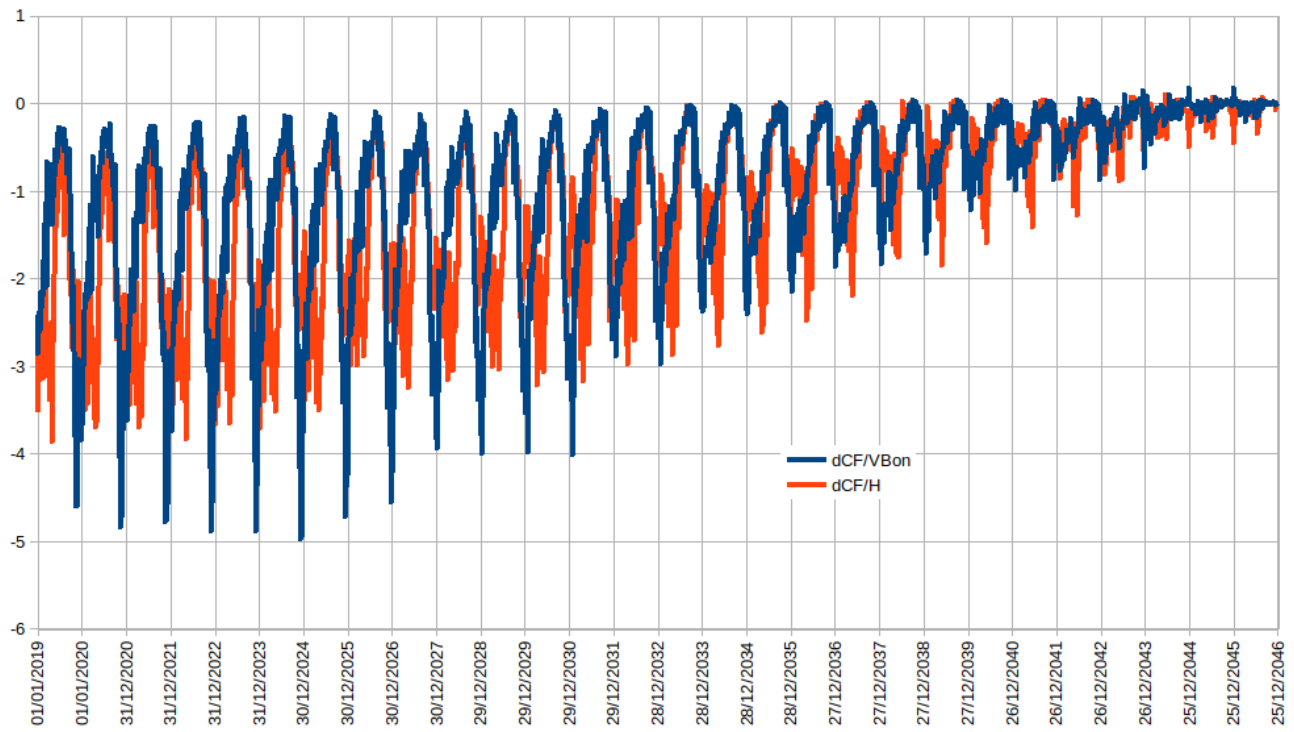


Fig. 52: Evolución de los coeficientes del hiperplano en todo el horizonte.