

**SEGUNDO INFORME SOBRE LA
INCORPORACIÓN DE ACUÍFEROS EN EL
MÓDULO DE OPTIMIZACIÓN DE LA GESTIÓN
DE SISTEMAS DE RECURSOS HÍDRICOS**

-

**ANÁLISIS DE ALGORITMOS DE
RESOLUCIÓN DE REDES DE FLUJO
PARA LA OPTIMIZACIÓN DE
ESQUEMAS DE RECURSOS HÍDRICOS**

Índice

1. Análisis de algoritmos de resolución de redes de flujo para la optimización de esquemas de recursos hídricos.	11
1.1. Algoritmos de optimización de redes de flujo	12
1.1.1. Out-of-Kilter.....	13
1.1.2. Relax-IV	15
1.1.3. NETFLO (SIMPLEX)	17
1.1.4. Generalized Network Algorithm (EMNET y NETPD).....	18
1.1.5. Otros algoritmos.....	19
1.2. Comparación según la bibliografía	19
1.3. Limitaciones de las redes de flujo.....	21
1.4. Consideraciones adicionales.....	21
2. Aplicación de los algoritmos de resolución de redes de flujo escogidos en la optimización de un sistema de recursos sencillo.	23
2.1. Caso de estudio: La cuenca del río Mijares simplificada	24
2.2. Optimización del sistema sin iteraciones	26
2.3. Introducción de no linealidades en la optimización del sistema.....	31
2.3.1. Optimización del sistema teniendo en cuenta los retornos de las demandas.	32
2.3.2. Optimización del sistema teniendo en cuenta la evaporación en los embalses.....	35
2.3.3. Optimización del sistema teniendo en cuenta evaporación y retornos. .	38
2.4. Análisis de resultados.	39
2.5. Consideraciones finales del capítulo	43

Lista de figuras

Figura 2.1 Situación de la cuenca del río Mijares.....	24
Figura 2.2. Modelo del sistema del Mijares creado con AQUATOOL para su ejecución con el programa SIMGES.....	25
Figura 2.3. Evolución del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo.....	27
Figura 2.4. Volumen total regulado en el sistema sin iteraciones con los tres algoritmos para el periodo de optimización correspondiente a 720 meses. No se observa ninguna diferencia entre ninguno de ellos.....	29
Figura 2.5. Evolución del volumen almacenado en el embalse de Arenós, durante el periodo de optimización de 720 meses, para cada uno de los tres algoritmos comparados.....	30
Figura 2.6. Evolución del volumen almacenado en el embalse de Sichar, durante el periodo de optimización de 720 meses, para cada uno de los tres algoritmos comparados.....	31
Figura 2.7. Modelo simplificado del sistema Mijares modificado para la inclusión de elementos de retorno.....	33
Figura 2.8. Evolución del valor de la función objetivo con el número de iteraciones para el horizonte de optimización de 720 meses.....	33
Figura 2.9. Tiempo de ejecución del modelo de optimización con cada algoritmo para el horizonte de optimización de 720 meses.	35
Figura 2.10. Tiempo de ejecución de cada iteración del modelo de optimización con cada algoritmo para el horizonte de optimización de 720 meses.....	35
Figura 2.11. Evolución del valor de la función objetivo con cada iteración para un horizonte de optimización de 720 meses.	37
Figura 2.12. Evolución del valor de la función objetivo con cada iteración para un horizonte de optimización de 720 meses.	38
Figura 2.13. Número de iteraciones según el valor de la variable NEPSI.....	41
Figura 2.14. Evolución del valor de la función objetivo a lo largo de 200 iteraciones con OPTIGES con RELAX-IV.....	43

Lista de tablas

Tabla 2.1 Tamaño de la red correspondiente a cada horizonte de optimización.....	27
Tabla 2.2. Evolución del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo.....	28
Tabla 2.3. Relación entre los tiempos de ejecución de los algoritmos según el tamaño de la red de flujo.....	28
Tabla 2.4. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.	34
Tabla 2.5. Déficit total en cada demanda para el horizonte de optimización de 720 meses.....	34
Tabla 2.6. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.....	34
Tabla 2.7. Tasa de evaporación en los embalses del modelo.	36
Tabla 2.8. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.....	37
Tabla 2.9. Déficit total en cada demanda para el horizonte de optimización de 720 meses.....	37
Tabla 2.10. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.	37
Tabla 2.11. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.....	39
Tabla 2.12. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.	39
Tabla 2.13. Déficit total en cada demanda para el horizonte de optimización de 720 meses.....	39
Tabla 2.14. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.	39
Tabla 2.15. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.....	41
Tabla 2.16. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.	41
Tabla 2.17. Déficit total en cada demanda para el horizonte de optimización de 720 meses.....	42
Tabla 2.18. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.	42

1. Análisis de algoritmos de resolución de redes de flujo para la optimización de esquemas de recursos hídricos.

Las redes de flujo son una herramienta de amplia aplicación en numerosos aspectos de la ciencia y la ingeniería. La similitud, esquemática, de una gran variedad de sistemas, como pueden ser las redes de telecomunicaciones, las redes eléctricas o los mismos sistemas de recursos hídricos, con la propia definición de las redes de flujo ayuda a que se opte por estos modelos matemáticos a la hora de resolver los problemas asociados al óptimo funcionamiento de los sistemas de estudio.

La utilización de redes de flujo para obtener el funcionamiento óptimo de un sistema tiene las mismas ventajas que las de la programación lineal, por ser las redes de flujo una configuración especial de ésta. Estas ventajas, ya comentadas en el anterior capítulo son: (1) la flexibilidad de ésta técnica para ser empleada eficientemente en problemas con un tamaño considerable; (2) la convergencia a un óptimo global; (3) no requerir una solución inicial por parte del usuario; y (4) el desarrollo de una teoría dual efectiva para el análisis de sensibilidad. A éstas han de añadirse las específicas asociadas a la especial configuración de una red de flujo como es su estructura fácilmente representable en matrices, lo que permite una manipulación muy eficiente de la información contenida en las mismas.

Por supuesto, esta metodología no está exenta de ciertos problemas, siendo el mayor de ellos la dificultad, si no imposibilidad, de representar aspectos no lineales, más habituales en la realidad que los lineales. Aun así, estas no-linealidades pueden solventarse, en cierto modo, mediante aproximaciones e iteraciones.

Desde encontrar el camino más corto, hasta cuál es el máximo flujo que puede circular por una red, los problemas que se pueden resolver mediante una red de flujo, con el algoritmo apropiado a cada caso, son muchos y variados (Ahuja, 1993 y Bazaraa, 1977). Muchos de estos problemas son casos especiales del problema de flujo a coste mínimo (MCNFP, "Minimum Cost Network Flow Problem", en la literatura en inglés). Este problema puede definirse, matemáticamente, de la siguiente manera (Ahuja, 1993):

Sea un grafo dirigido $G = (N, A)$ consistente en un conjunto N de nodos y un conjunto A de m arcos dirigidos. Cada arco $(i, j) \in A$ tiene un coste asociado c_{ij} que denota el coste por unidad de flujo en dicho arco. Además, asociamos a cada arco $(i, j) \in A$ una capacidad u_{ij} , que representa el máximo flujo que puede atravesar ese arco, y un límite inferior l_{ij} , que represente el mínimo flujo que debe atravesar el arco. También asociamos a cada nudo $i \in N$ un número entero b_i representando su

suministro/demanda. Si $b_i > 0$, el nudo i es un nudo de suministro; si $b_i < 0$, el nudo i es un nudo de demanda con una demanda $-b_i$; y si $b_i = 0$, entonces el nudo i es un nudo de transferencia. Las variables de decisión en el problema de flujo a coste mínimo son los flujos en los arcos y representamos el flujo en cada arco $(i, j) \in A$ como x_{ij} . El problema de flujo a coste mínimo es un modelo de optimización formulado como sigue:

$$\text{Min} \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (1.1)$$

$$\text{s.a.} \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = b_i \quad i = 1, \dots, m \quad (1.2)$$

$$x_{ij} \geq 0 \quad i, j = 1, \dots, m \quad (1.3)$$

Según Andreu (1992): *Resulta evidente que los sistemas de recursos hídricos, de una forma esquemática, son asimilables a una red de flujo (especialmente en el caso de sistemas de aguas superficiales), pues hay en ellos una serie de cauces y canales (arcos) que unen una serie de puntos (nodos) como son los embalses, las tomas, los puntos de conjunción de afluentes, etc., y en los que el agua circula siempre en un sentido predeterminado. Por lo tanto, es lógico plantear la optimización de un sistema de recursos hídricos mediante la resolución de un problema de flujo a coste mínimo.*

En el siguiente apartado se presentarán algunos de los algoritmos que más se han utilizado en los últimos tiempos para la resolución de redes de flujo orientadas a la optimización de sistemas de recursos hídricos, resolviendo el problema de flujo a coste mínimo asociado a las mismas. Además, se ha estudiado, también, las diferencias documentadas de funcionamiento de unos y otros, aspecto que se expondrá posteriormente.

Algunos de estos algoritmos son los que, actualmente, se encuentran implementados en varios Sistemas de Soporte a la Decisión (Andreu, 1996; y Labadie, 2000), de uso habitual en estudios de análisis de sistemas de recursos hídricos.

1.1. Algoritmos de optimización de redes de flujo

La optimización de una red de flujo conlleva, principalmente, el reparto de flujos a través de la misma de forma que se minimice el valor de una función objetivo que, generalmente, tendrá la forma de la ecuación 1, además de cumplirse las restricciones 2 y 3 en todos los nudos y conducciones respectivamente. Esto no es algo que pueda llevarse a cabo de forma analítica y habrá de recurrirse a algún, o algunos, algoritmos para resolver el problema de forma numérica (Bersetkas, 1998).

Pese a que la red de flujo podría resolverse mediante algún algoritmo general de programación lineal o no-lineal, resulta más ventajoso aprovecharse de la particular estructura de la red de flujo para acelerar la convergencia a la solución mediante la adaptación de un algoritmo general o la utilización de un algoritmo específico para redes de flujo.

Bersetkas (1998) divide los algoritmos para optimización de redes de flujo en 3 categorías: (1) Mejora del coste primal; (2) mejora del coste dual; y (3) subasta. La primera categoría se corresponde con los algoritmos que resuelven el problema de flujo a coste mínimo a partir de una solución factible inicial y que generan una secuencia de vectores de flujo factibles, cada uno con un coste mejor que el anterior, hasta alcanzar el óptimo. Los algoritmos pertenecientes a la segunda categoría se basan en la optimización de un problema relacionado con el original llamado problema dual, cuyas variables toman el nombre de precios. Posteriormente se trata de alcanzar el mejor coste dual mediante la construcción sucesiva de vectores de precios. Por último, los algoritmos del tipo subasta resuelven la red de flujo mediante un procedimiento que recuerda, en cierta manera, a las subastas reales. Aunque pueden considerarse aproximadamente como procesos de optimización dual, en realidad las iteraciones tienen en cuenta tanto los precios (dual), como los flujos (primal), llegando a poder empeorar ambos en una sola iteración, aunque finalmente se alcance una solución primal óptima.

Los algoritmos seleccionados en este estudio forman una muestra de las familias de algoritmos presentadas en el párrafo anterior.

1.1.1. Out-of-Kilter

Formulado por Ford y Fulkerson (1962), el algoritmo Out-of-Kilter es/ha sido uno de los algoritmos más utilizados en la optimización de redes de flujo conservativas. Además, resulta un algoritmo de exposición común en textos generales sobre redes de flujo debido a su sencillez de exposición (Bazaraa, 1977; Ahuja, 1993).

El problema específico que resuelve este problema es:

$$\text{Min} \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (1.4)$$

$$\text{s.a.} \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = 0 \quad i = 1, \dots, m \quad (1.5)$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad i, j = 1, \dots, m \quad (1.6)$$

Obsérvese que, a diferencia con el problema formulado en las ecuaciones (3.1) a (3.3), este problema de coste mínimo exige que todos los nudos del mismo sean de

transferencia. Además, se impone que flujo en los arcos se mantenga dentro de unos límites inferior y superior donde $l_{ij}, u_{ij} \geq 0$.

Este algoritmo, perteneciente a la segunda familia de algoritmos, basa el mecanismo de resolución de la red de flujo en la resolución sucesiva e iterativa de los problemas primal y dual. La estrategia de resolución incluye la comprobación del estado de “sincronización” (In-Kilter) o “desincronización” (Out-of-Kilter) de cada arco de la red, y el intento de asignar el flujo de forma que todos los arcos se encuentren “sincronizados” (In-Kilter).

A modo de resumen, el esquema general de resolución del algoritmo Out-of-Kilter es el que sigue (Bazaraa, 1977):

- (1) Comenzar con un flujo conservativo y una solución factible para el problema dual. Se identifican los estados de “sincronización” y se calculan los números de “sincronización”.
- (2) Si la red tiene un arco “desincronizado” (Out-of-Kilter), se lleva a cabo una fase primal del algoritmo. Durante esta fase se selecciona un arco “desincornizado” (Out-of-Kilter) y se intenta construir una nueva red de flujo conservativa de forma que no se empeore el número de “sincronización” de ningún arco y se mejore el del arco seleccionado.
- (3) Cuando ya no se puede mejorar ningún flujo en la fase primal, el algoritmo construye una solución dual de forma que no se empeore ningún número de “sincronización” y se repite el paso (2).

Iterando los pasos (2) y (3), el algoritmo termina construyendo una solución óptima o determinando que no existe ninguna solución factible.

Las principales ventajas de este algoritmo pueden encontrarse en Jensen (1987):

- No requiere ninguna estructura de memoria especial para la representación de la red. Los parámetros se introducen y almacenan en listas de orden arbitrario
- No utiliza parámetros externos. Todo se describe en los parámetros de los arcos.
- El algoritmo puede inicializarse con cualquier conjunto de flujos que satisfaga la conservación de flujo. Por ello, es especialmente útil para realizar análisis de sensibilidad cuando se varían los parámetros de los arcos.

Por supuesto, el algoritmo tiene sus desventajas:

- El empleo de arcos para especificar flujos en los nudos a menudo supone un alto requerimiento de memoria

- Las soluciones no son necesariamente básicas. Esto puede resultar en una convergencia lenta hacia el óptimo.
- El procedimiento de resolución puede requerir numerosas lecturas de la lista de arcos, por lo que no es demasiado eficiente, computacionalmente, para problemas grandes.

1.1.2. Relax-IV

Este algoritmo, presentado por Bersetkas (1994), une el algoritmo RELAX (Bersetkas, 1988) con un método de inicialización basado en un algoritmo de subasta que puede elegirse utilizar, o no. El problema que se resuelve es el siguiente:

$$\text{Min} \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (1.7)$$

$$\text{s.a.} \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = b_i \quad i = 1, \dots, m \quad (1.8)$$

$$0 \leq x_{ij} \leq u_{ij} \quad i, j = 1, \dots, m \quad (1.9)$$

Este problema es resultado de realizar ciertas operaciones que permitirán un ahorro de memoria de almacenamiento respecto al algoritmo Out-of-Kilter. El valor de l_{ij} se sustituye por cero y, a cambio, se emplea el vector $b(i)$, de menor tamaño. Para alcanzar esta formulación basta realizar el cambio de variables $x_{ij} = x_{ij} - l_{ij}$ en el problema enunciado en las ecuaciones (3.4) a (3.6), obteniéndose el mostrado encima de este párrafo.

Al igual que el algoritmo Out-of-Kilter, Relax-IV es un algoritmo perteneciente a la segunda familia, cuya estrategia de resolución de la red de flujo consiste en resolver de forma iterativa y alternativa la optimización de los problemas primal y dual. En el caso de este algoritmo no se habla de arcos sincronizados y desincronizados. El enfoque que se plantea en este caso se basa en determinar si un arco se encuentra activo, equilibrado o inactivo mediante la comparación de la diferencia de los precios (dual) asociados a cada uno de los nudos en los extremos de cada arco con el coste (primal) del flujo a través de este. Según el estado de "actividad" de cada arco, se definen una serie de condiciones de óptimo de forma que una solución primal-dual (x, p) será óptima sí y solo sí, se cumplen las siguientes condiciones en todos los arcos de la red simultáneamente (Bersetkas, 1988):

$$x_{ij} = l_{ij} \quad \forall (i, j) / t_{ij} < c_{ij} \quad (\text{el arco está inactivo}) \quad (1.10)$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i, j) / t_{ij} = c_{ij} \quad (\text{el arco está equilibrado}) \quad (1.11)$$

$$x_{ij} = u_{ij} \quad \forall (i, j) / t_{ij} > c_{ij} \text{ (el arco está activo)} \quad (1.12)$$

$$d_i = 0 \quad \forall i \in N \text{ (el déficit es cero en todos los nudos)} \quad (1.13)$$

siendo:

$$t_{ij} = p_i - p_j \quad \forall (i, j) \in A \text{ (vector tensión)} \quad (1.14)$$

La forma de alcanzar estas condiciones consiste en incrementar de forma iterativa el funcional dual. El vector de precios se incrementa mientras se mantienen las condiciones anteriores para el vector de flujo. El algoritmo finaliza cuando el déficit en cada nudo es cero. La forma de llevar a cabo los incrementos en el vector de precios consiste en, comenzando desde un solo nudo con déficit diferente a cero, comprobar si puede mejorarse la función dual si se incrementa su precio. Si no es posible, se construye gradualmente, según un proceso de etiquetado, bien un camino incremental de flujo o un conjunto asociado con una dirección de ascenso. La principal característica es que el proceso de etiquetado se detiene en cuanto se encuentra la primera dirección de ascenso, en vez de hacerlo para todos los nudos. A este proceso se le llama método de relajación.

Además de emplear este proceso de resolución, el algoritmo Relax-IV incluye la posibilidad de ser inicializado de dos formas distintas. La primera es similar a la de otros métodos duales, como el Out-of-Kilter, mediante cualquier conjunto de flujos que satisfaga la conservación de flujo, generalmente un vector de ceros lo hará. La segunda se corresponde con la ejecución de un algoritmo de subasta inicial para determinar un buen conjunto de flujos y precios iniciales, para una resolución más eficiente de problemas grandes, para los cuales el algoritmo RELAX sólo resultaba poco eficiente.

Al tratarse de un algoritmo similar, en esencia, al algoritmo Out-of-Kilter, las ventajas de Relax-IV son similares, aunque no iguales, a las del anterior:

- No requiere ninguna estructura de memoria especial para la representación de la red. Los parámetros se introducen y almacenan en listas de orden arbitrario
- El algoritmo puede inicializarse con cualquier conjunto de flujos que satisfaga la conservación de flujo. Por ello, el algoritmo es especialmente útil para realizar análisis de sensibilidad cuando se varían los parámetros de los arcos.
- Al no realizar lecturas exhaustivas de toda la lista de arcos y detenerse, normalmente, en el primero que cumple las condiciones de ascenso, la resolución de la red es más rápida.

Por supuesto, el algoritmo tiene sus desventajas:

- El procedimiento de resolución sin la inicialización por subasta se encuentra con dificultades para la resolución de problemas grandes. Aunque se aprecia una gran mejoría si se utiliza la inicialización.

1.1.3. NETFLO (SIMPLEX)

El método simplex para resolver problemas de programación lineal es un algoritmo muy poderoso para resolver problemas de optimización con restricciones. Dado que las redes de flujo suponen un tipo especial de programación lineal, se podría esperar que el método simplex fuese una alternativa atractiva para la resolución de las mismas. Sin embargo, el método general no resulta competitivo frente a otros algoritmos especialmente preparados para la resolución de redes de flujo aprovechando su particular estructura (Ahuja 1993).

NETFLO es el nombre dado a la implementación del algoritmo simplex para redes de flujo realizada por Kennington y Helgason (1980). Se trata de un algoritmo de la primera familia, que se basa en el incremento sucesivo del vector de flujo hasta que se alcanza un óptimo en la función objetivo.

El problema que se resuelve es el enunciado para el problema del flujo a coste mínimo en las ecuaciones (1.1) a (1.3).

Como ya se ha comentado previamente, NETFLO lleva a cabo la optimización de la red de flujo mediante el algoritmo simplex para redes de flujo que aprovecha la estructura especial de la red para adaptar el método simplex general. Este algoritmo se basa en la obtención de un tipo de soluciones llamadas “árboles de cobertura” (spanning trees). En estas soluciones se fija el flujo de los arcos que no pertenecen al “árbol de cobertura” bien a cero o al límite de su capacidad. El valor óptimo se alcanza avanzando de una solución a otra generando, en cada paso de cálculo, un árbol factible con su vector de flujos correspondiente. El coste de cada vector de flujos nunca ha de ser menor que en el paso anterior hasta que se alcanza el óptimo. Cada iteración, llamada pivote como en el algoritmo SIMPLEX de la programación lineal, opera de la siguiente forma Bersetkas (1998):

- (a) Se añade un nuevo arco al árbol de forma que se cree un único ciclo con coste negativo. El arco entrante se suele seleccionar en base al coste reducido asociado al mismo.
- (b) Se asigna al ciclo tanto flujo como sea posible sin violar la factibilidad del mismo.
- (c) Se descarta uno de los arcos del ciclo, lo que genera un nuevo árbol para ser utilizado en la nueva iteración. El arco descartado se elige de entre aquellos arcos del ciclo que aporta un menor incremento de flujo a través del mismo.

1.1.4. Generalized Network Algorithm (EMNET y NETPD)

El algoritmo que se analiza a continuación constituye la solución a un tipo de red de flujo distinta a las vistas hasta ahora. Una red de flujo con ganancias/pérdidas se trata de una red de flujo que, además de los parámetros habituales asociados a una red de flujo normal, tiene asociado un término multiplicador $\mu_{ij} > 0$ en cada arco (i, j) . De esta forma, el problema a resolver resulta (Jensen 1987):

$$\text{Min} \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (1.15)$$

$$\text{s.a.} \sum_{j=1}^m x_{ij} - \sum_{k=1}^m \mu_{ij} x_{ki} = b_i \quad i = 1, \dots, m \quad (1.16)$$

$$0 \leq x_{ij} \leq u_{ij} \quad i, j = 1, \dots, m \quad (1.17)$$

Se asume que el multiplicador μ_{ij} es un número racional que puede expresarse como $\mu_{ij} = p_{ij}/q_{ij}$, para p_{ij} y q_{ij} dos números enteros. Cuando enviamos 1 unidad de flujo a través del arco (i, j) , al nudo j llegan μ_{ij} unidades. Si $\mu_{ij} < 1$, el arco es perdedor; y si $\mu_{ij} > 1$, el arco es ganador.

Nótese que se asume que la capacidad de arco u_{ij} es un límite superior del flujo que se envía desde el nudo i , no del flujo disponible en el nudo j . De forma similar, c_{ij} es el coste por unidad de flujo que enviamos desde el nudo i , y no el coste por unidad de flujo que llega al nudo j . Los flujos mínimos están resueltos de la misma manera que en la formulación del problema para el algoritmo Relax-IV.

La resolución de esta red de flujo puede realizarse de forma similar a la resolución de los anteriores problemas adaptando los algoritmos. Ahuja (1993) y Jensen (1987) desarrollaron adaptaciones de algoritmos de tipo primal (simplex) para resolver el problema de flujo a coste mínimo en redes de flujo generalizadas.

EMNET (McBride, 1985), es el nombre de la implementación de un algoritmo tipo simplex (familia 1) para resolver una red de flujo generalizada. La estrategia general del algoritmo se divide en tres fases: En la fase inicial (0), se resuelve una red de flujo pura, quitando los aspectos complicados, lo que permite obtener una solución aproximada de una forma rápida y eficaz; en la siguiente fase (1) se añaden los aspectos complicados de la red y se obtiene una solución factible a partir de la obtenida en la fase 0; por último, la fase (2) consiste en variar los flujos, como en el algoritmo simplex, hasta alcanzar el óptimo.

NETPD (Curet, 1994), es una implementación de un algoritmo de tipo primal-dual (familia 2) para redes de flujo generalizadas. El procedimiento de resolución de la red de

flujo de este algoritmo es similar al expuesto en los algoritmos Out-of-Kilter y Relax-IV. De hecho, se realiza una mezcla de ambos procedimientos, se mantiene el incremento de la factibilidad de las soluciones durante la fase primal, como en Out-of-Kilter, pero se reduce la cantidad de trabajo necesaria para determinar qué variable dual debe incrementarse, como en Relax-IV.

Un aspecto interesante de los algoritmos para redes generalizadas es la posibilidad de resolver, también, redes de flujo puras de forma eficiente sin necesidad de realizar ningún tipo de cambio ni en códigos ni en la introducción de los datos.

1.1.5. Otros algoritmos

Aparte de los tratados, existen otros tantos algoritmos que permiten optimizar redes de flujo, aunque se ha optado por no mostrarlos en detalle debido a que no aportan un valor añadido a la exposición, bien por ser menos comunes en la bibliografía, o por ser implementaciones similares a las realizadas por alguno de los algoritmos mostrados. Ejemplos de estos algoritmos son:

- RNET (Grigoriadis y Hsu, 1978) es una implementación del algoritmo simplex para redes de flujo anterior a NETFLO aunque, básicamente, tienen los mismos principios.
- NEPSA (Paparrizos, 2009), es el nombre de un algoritmo en desarrollo, basado en una modificación del método simplex para redes de flujo, que aún no ha sido implementado y cuyo funcionamiento frente a otros algoritmos más desarrollados está aún por comprobar.

1.2. Comparación según la bibliografía

Para no perdernos dentro del espeso bosque que supone la existencia de tal cantidad de algoritmos orientados a realizar un trabajo similar, y con la firme intención de enfocar los esfuerzos de la investigación únicamente sobre los más adecuados, se ha resuelto realizar una primera estimación de la eficacia de los diferentes algoritmos en problemas reales de optimización de los recursos hídricos mediante la consulta de parte de la bibliografía existente al respecto para obtener una primera impresión del funcionamiento de los diferentes algoritmos presentados.

Una ventaja remarcable de todos los artículos consultados es la utilización, en la mayoría de ellos, de un generador aleatorio de problemas de redes de flujo común, NETGEN (Klingman, 1974), lo que, en cierta medida, asegura cierta similitud en los problemas resueltos por unos y otros.

Una buena parte de los autores que tratan en sus artículos el funcionamiento de unos y otros algoritmos (Bersetkas, 1985, 1988 y 1994; Rani, 2010; y Kuczera, 1993), coinciden

en que los códigos basados en el método de relajación (RELAX-IV e implementaciones anteriores), generalmente, funcionan de manera mucho más eficiente que el resto de algoritmos similares implementados hasta la fecha (Out-of-Kilter y Simplex), siendo, aproximadamente, alrededor de un orden de magnitud más rápidos.

De especial interés son los resultados aportados en (Kuczera, 1993), donde se realiza una comparación entre los tiempos de ejecución de los algoritmos NETFLO (simplex) y RELAX (primal-dual) para la resolución de una red de distribución de agua en superficie, con embalses y demandas. En este caso, se concluye que RELAX llega a ser entre, aproximadamente, dos y siete veces más rápido en la resolución de la red que NETFLO. Además, se muestra, en ese mismo artículo, que la información con la que se inicializa la resolución de RELAX afecta de forma bastante importante al tiempo de ejecución de éste.

No se ha encontrado ningún artículo en que se realizara una comparación explícita entre los algoritmos Out-of-Kilter y NETFLO. Aun así, en (Bersetkas, 1985) ya se comprueba el bajo rendimiento del Out-of-Kilter respecto a una de las primeras versiones de RELAX y de RNET (que implementa el método simplex al igual que NETFLO), lo que nos ayudaría a concluir que el algoritmo que Ford y Fulkerson crearon en 1962 se ha quedado, relativamente, anticuado.

Cabe comentar que según (Curet, 1998), el código RELAX es el método que más rápido resuelve los problemas de NETGEN.

Existen ciertas excepciones, como el caso de Paparrizos (2009), que aclama al método simplex como la mejor opción para resolver MCNFP, aunque no se aporta ninguna prueba al respecto.

Por otro lado, se encuentran los algoritmos dedicado a la resolución de redes de flujo generalizadas.

En (Sun, 1995) se realiza una comparación entre EMNET y MINOS, un software de optimización lineal y no lineal con diversos algoritmos, en la resolución de una serie de problemas tipo que representan esquemas de optimización de sistemas de recursos hídricos a largo plazo de acuerdo con los objetivos de planeamiento del Metropolitan Water District of Southern California (MWD). Aunque no especifica el tipo de optimización que se emplea con MINOS, en (Labadie, 2004) se comenta que es un algoritmo de programación lineal de tipo simplex, el cual resulta entre 11 y 117 veces más lento que EMNET en la resolución de la red. McBride (1985) también muestra que EMNET resulta alrededor de 5 veces más rápido que MINOS.

Se observa (Curet, 1994 y 1998) también que, al igual que para redes de flujo puras, los algoritmos del tipo primal-dual (NETPD) para redes de flujo generalizadas son, por lo general, más eficientes que los algoritmos de tipo simplex (EMNET).

Se ha comentado anteriormente que los algoritmos de optimización de redes de flujo generalizadas pueden, perfectamente, resolver redes de flujo puras. A este respecto, en (Curet, 1997 y 1998), se muestra que la resolución de una red de flujo pura con NETPD y con RELAX lleva, aproximadamente, el mismo tiempo de computación, siendo ambos dos algoritmos del tipo primal-dual. Por otro lado, comparado con un algoritmo de tipo simplex (NETFLO), se muestra como NETPD es capaz de resolver la red en mucho menos tiempo.

1.3. Limitaciones de las redes de flujo

Las redes de flujo resultan un método muy interesante para el desarrollo de modelos de optimización de sistemas de recursos hídricos. Sin embargo, su utilización no carece de desventajas. Un problema de redes de flujo, al fin y al cabo, no deja de ser un problema de programación lineal de resolución más eficiente por la característica formulación del mismo y, por tanto, adolece de los mismos problemas. El más importante de todos es la incapacidad de tratar con aspectos no lineales, lo que implica que deben realizarse simplificaciones que, en ocasiones, pueden resultar excesivas para la representatividad de los modelos desarrollados, o puede aproximarse la solución mediante iteraciones sucesivas hasta alcanzar la convergencia de los resultados (Fredericks 1998).

Se ha criticado la utilización de iteraciones en redes de flujo (Ilich 2009) por considerarse que las sucesivas iteraciones no alcanzan el resultado óptimo. Sin embargo, como se mostrará más adelante, en este estudio se ha considerado que la aproximación de las no linealidades mediante proceso iterativo sigue siendo válida ya que lo que se hace es resolver sucesivamente una red de flujo que cambia según las condiciones de flujo resultantes de la iteración anterior, cuya solución sí es la óptima, lo que implica que la solución final, cuando se alcanza la convergencia, también debe ser óptima mientras se cumplen unas condiciones adicionales a las intrínsecas a la formulación de la red de flujo.

1.4. Consideraciones adicionales

En este capítulo se ha realizado un estado del arte de algoritmos de optimización de redes de flujo y de su aplicación en la gestión/optimización de sistemas de recursos hídricos. También se han comentado los problemas que supone la utilización de las redes de flujo en la resolución de sistemas que presentan cierta dificultad de

representación, o precisión, especialmente cuando se incluyen aspectos de carácter no lineal, y la forma de enfocar dicha resolución.

Teniendo en cuenta el deseo expreso de mejorar y ampliar el modelo OPTIGES, y teniendo en cuenta que lo más acertado pasa por mejorar, en primer lugar, la capacidad de cálculo, la primera consideración que se debe realizar es si el algoritmo Out-of-Kilter, empleado en la resolución del modelo actualmente, es el más adecuado para hacerlo. Por lo visto hasta el momento, aparentemente, el algoritmo se ha quedado relativamente anticuado, ya que hay algoritmos que, al menos según la bibliografía, resuelven el mismo problema de forma más eficiente y que permiten incluir más elementos. Un algoritmo más eficiente permitiría añadir mayor complejidad a los modelos y, por tanto, a la red, pero este es un aspecto que se verá posteriormente.

Ahora bien, la duda que se despierta es, desde el punto de vista de escoger un algoritmo de optimización u otro, ¿qué resulta más eficiente, tanto en velocidad de cálculo como posibilidades de representación, escoger un algoritmo que sea capaz de resolver redes de flujo puras rápidamente y solucionar la inclusión de no linealidades mediante iteraciones, o escoger uno que permita introducir más elementos en el proceso de cálculo, a costa de incrementar el tiempo de cálculo, pero reduciendo el número de iteraciones? Esta pregunta viene a colación de la existencia de dos posibilidades. La primera sería emplear alguno de los algoritmos de resolución de redes de flujo puras presentados para sustituir el que actualmente se utiliza. La segunda sería implementar un algoritmo de resolución de redes de flujo generalizadas.

En principio se va a optar por la primera opción. Por un lado, porque el modelo de optimización en que se van a basar las pruebas (OPTIGES) genera una red de flujo que se adapta casi perfectamente a dichos algoritmos y no se requerirían demasiados cambios y, por otro, porque ya se dispone de un mecanismo bien estructurado de iteraciones que tampoco habría que modificar. Por último, aunque relativamente menos importante, también resulta una razón el hecho de que, hasta la fecha, ha resultado imposible localizar un algoritmo de resolución de redes de flujo generalizadas disponible para su utilización pública, por lo que la segunda opción se dejará para investigaciones posteriores.

Para seleccionar, entonces, el algoritmo más adecuado para va a realizar una comparación de primera mano entre las diferentes posibilidades. Para ello, en el próximo capítulo, se llevará a cabo la resolución de una serie de ejemplos sencillos, con la inclusión de ciertas particularidades en cada uno, de forma que nos permita comparar el funcionamiento de cada uno de los algoritmos y seleccionar el más adecuado.

2. Aplicación de los algoritmos de resolución de redes de flujo escogidos en la optimización de un sistema de recursos sencillo.

En el anterior capítulo se ha concluido que los algoritmos más adecuados para la resolución de las redes de flujo generadas por modelos de optimización como podría ser OPTIGES son Out-of-Kilter, RELAX-IV y NETFLO.

Existen dos razones por las cuáles se han escogido estos algoritmos y no otros. En primer lugar, por el hecho de que los dos nuevos algoritmos resuelven prácticamente el mismo problema, lo que evitaría realizar grandes cambios en el planteamiento del problema dentro del código del programa OPTIGES, modelo empleado como base para las pruebas. La segunda razón, y pese a que puede resultar una excusa, es porque pese a la multitud de algoritmos que se enuncian de forma teórica, muy pocos de ellos acaban teniendo una implementación pública, lo que dificulta enormemente su obtención.

Así, se crearon tres versiones del modelo de optimización empleando cada uno de los tres algoritmos como motor de cálculo en cada caso y sin realizar ningún otro cambio adicional. Las versiones de los tres algoritmos se obtuvieron de códigos públicos de los mismos, como ya se ha comentado.

Se realizaron algunas modificaciones a los códigos de los algoritmos para hacer coincidir nombres de variables o para calcular algunas variables adicionales en el caso de alguno de ellos. RELAX-IV y NETFLO consideran el límite inferior de los arcos siempre nulos y emplean una variable adicional para recrear los requerimientos de flujo mínimos llamada b_i . Las adaptaciones que se realizaron a ambos algoritmos consistieron simplemente en realizar la sustitución de los límites inferiores del flujo en los arcos de la red de flujo (l_{ij}) por el vector de demandas en los nudos (b_i), como se explica en los apartados correspondientes a estos algoritmos en el capítulo anterior. Esta sustitución se realiza de manera sencilla sumando los límites inferiores en diferentes términos según el nudo de procedencia de los arcos con dicho límite definido. Si bien esta sustitución plantea algunos problemas en puntos como los caudales mínimos en conducciones o volúmenes mínimos en embalses, se ha optado por no incluir estas posibilidades todavía en el código del modelo OPTIGES ya que no suponen una gran diferencia de cara a la comparación de la eficiencia de los algoritmos implementados.

Con las tres versiones del modelo compiladas se resolvieron diversos supuestos basados en el caso de estudio que se presentará en el siguiente apartado. Dichos supuestos incluyen:

- La optimización del sistema de estudio sin ningún tipo de iteración, es decir, excluyendo la evaporación de los embalses y los retornos de riego y resolviendo, por lo tanto, una red de flujo pura.
- La optimización del sistema incluyendo la evaporación y los retornos por separado.
- La optimización del sistema incluyendo evaporación y retornos a la vez.

2.1. Caso de estudio: La cuenca del río Mijares simplificada

El río Mijares se encuentra ubicado en la vertiente oriental de la península Ibérica (ver figura 2.1). La superficie total comprendida por este sistema es de 5.466 km². Con una población total en la zona de 363.578 habitantes su suministro se realiza a partir de la explotación de pozos y el aprovechamiento de manantiales. La superficie total cultivada asciende a 124.310 Has, de las cuales 43.530 Has (35 %) corresponden a regadíos, mientras que el resto (65 %) son de secano. Los cítricos constituyen el cultivo predominante, con un porcentaje próximo al 87 % de la superficie en regadío de la citada zona. La longitud del curso principal es de unos 156 Km y tienen una aportación media de la cuenca de 380 hm³/año. En cuanto a la infraestructura de almacenamiento la cuenca cuenta con tres embalses: el de mayor capacidad es el de Arenós (95 hm³ útiles), aguas abajo se encuentra el embalse de Sichar (49 hm³) y ubicado en el afluente “rambla de la Viuda” el de María Cristina (19.7 hm³).



Figura 2.1 Situación de la cuenca del río Mijares

La problemática de la cuenca se centra en el reparto del recurso entre las demandas agrícolas. En la parte baja de la cuenca se ubican los riegos denominados como “Riegos tradicionales”. Estos riegos datan de más de un milenio de antigüedad por lo que mantienen un derecho predominante sobre el resto de usos agrícolas. Por otro lado, los riegos de la parte media de la cuenca conforman los regadíos modernos, denominados también como mixtos debido a la posibilidad de utilizar tanto aguas superficiales como subterráneas. El incremento del uso de aguas subterráneas ha llevado a que el acuífero sufra problemas de intrusión marina en épocas de sequía en donde se produce una mayor explotación del acuífero. Bajo esta situación es necesario el establecimiento de una regla de gestión que proteja los derechos de los regadíos tradicionales restringiendo el uso superficial a los riegos más modernos.

Previamente a la aplicación de la metodología explicada al inicio de este capítulo se creó un modelo simplificado de la cuenca del río Mijares basado en un modelo de simulación representativo del sistema del Mijares creado para el programa SIMGES de simulación de sistemas de recursos hídricos (Andreu et al, 1996). Para ello se ha utilizado el sistema de ayuda a la Decisión AQUATOOL. Como el modelo de simulación resultaba demasiado complejo para una primera aproximación a la optimización del sistema, el modelo simplificado incluye únicamente los elementos principales de la cuenca, como son los embalses de Arenós y Schar y las demandas agrícolas más importantes (Cota 220, Cota 100 y Riegos Tradicionales). La siguiente figura muestra la topología del modelo de simulación desarrollado para su utilización con el programa SIMGES y el modelo simplificado creado para emplear con el modelo OPTIGES.

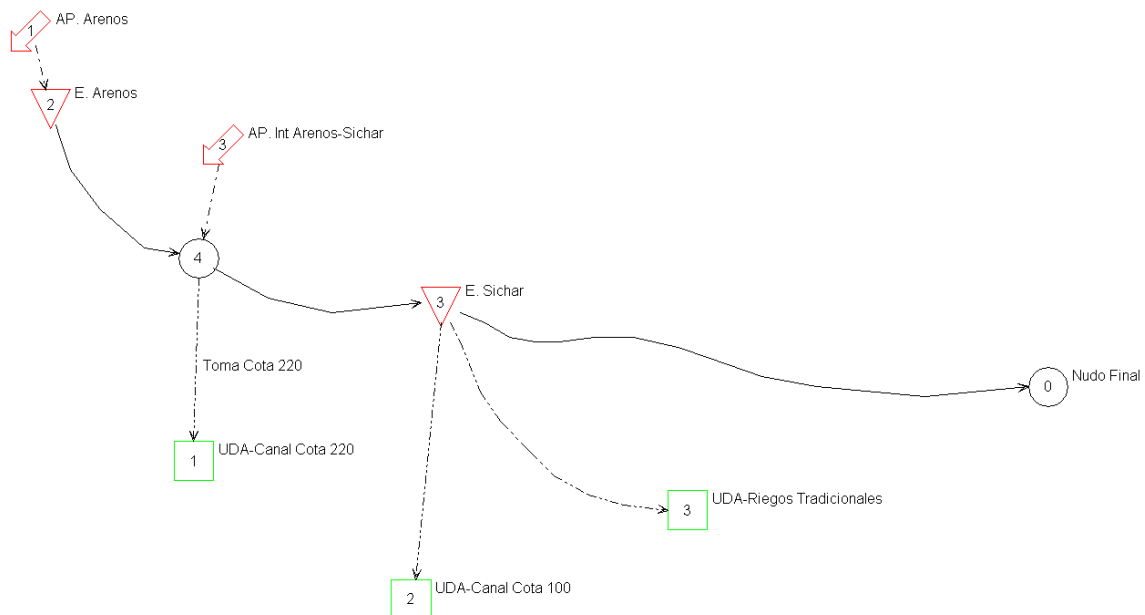


Figura 2.2. Modelo del sistema del Mijares creado con AQUATOOL para su ejecución con el programa SIMGES.

Como se puede ver en la figura el modelo de optimización consta de un curso principal que representa al río Mijares en donde se ubican los embalses de Arenós (E.Arenós) y Sichar (E. Sichar).

Las aportaciones consideradas son: el aporte de la cuenca aguas arriba del embalse de Arenós(Ap. Arenós), y el aporte de la cuenca media del río Mijares (Apo.Intermedia).

Los riegos del sistema se tienen en cuenta en el modelo con 3 elementos de demanda: “UDA-Riegos Tradicionales”; “UDA-Riegos Canal Cota-220”; y “UDA-Riegos Canal Cota 100”. No se han considerado los suministros urbanos en el modelo de optimización debido a que éstos se suplen, principalmente, desde el acuífero de la Plana de Castellón, el cual no puede incluirse, de momento, en esquemas de OPTIGES.

El orden de prioridad de las demandas es el siguiente: en primer lugar la demanda agrícola de riegos tradicionales y después las demandas de riego restantes. En cuanto a los embalses, se define el embalse de Arenós como prioritario frente al de Sichar aunque esto no tendrá efecto en el reparto de flujos salvo en el último mes del periodo de optimización considerado cada vez.

No se han tenido en cuenta en el modelo simplificado ni caudales ecológicos ni volúmenes mínimos de embalses por las razones comentadas al inicio del capítulo.

2.2. Optimización del sistema sin iteraciones

En este apartado se llevará a cabo una optimización del modelo del Mijares simplificado creado sin tener en cuenta elementos que puedan introducir no linealidades o, lo que es lo mismo, que hagan que sea necesaria ninguna iteración. De esta forma, se puede estudiar el funcionamiento de los diferentes algoritmos considerados desde el punto de vista puramente computacional.

Se ejecutaron las diferentes versiones de OPTIGES para varios horizontes de optimización crecientes, lo que implica un tamaño de la red de flujo creciente. En la tabla 2.1 se muestra el tamaño de la red de flujo resuelta para cada horizonte de optimización. En la figura 2.3 se muestra la evolución de los tiempos de ejecución de cada uno de los algoritmos según el tamaño de red resuelta. A la figura 2.3 le acompaña la tabla 2.2 con los valores numéricos incluidos en la misma para un mayor detalle.

Tamaño de la red	Número de nudos	Número de arcos
12 meses	119	317
120 meses	1091	3017
240 meses	2171	6017

360 meses	3251	9017
480 meses	4331	12017
600 meses	5411	15017
720 meses	6491	18017

Tabla 2.1 Tamaño de la red correspondiente a cada horizonte de optimización

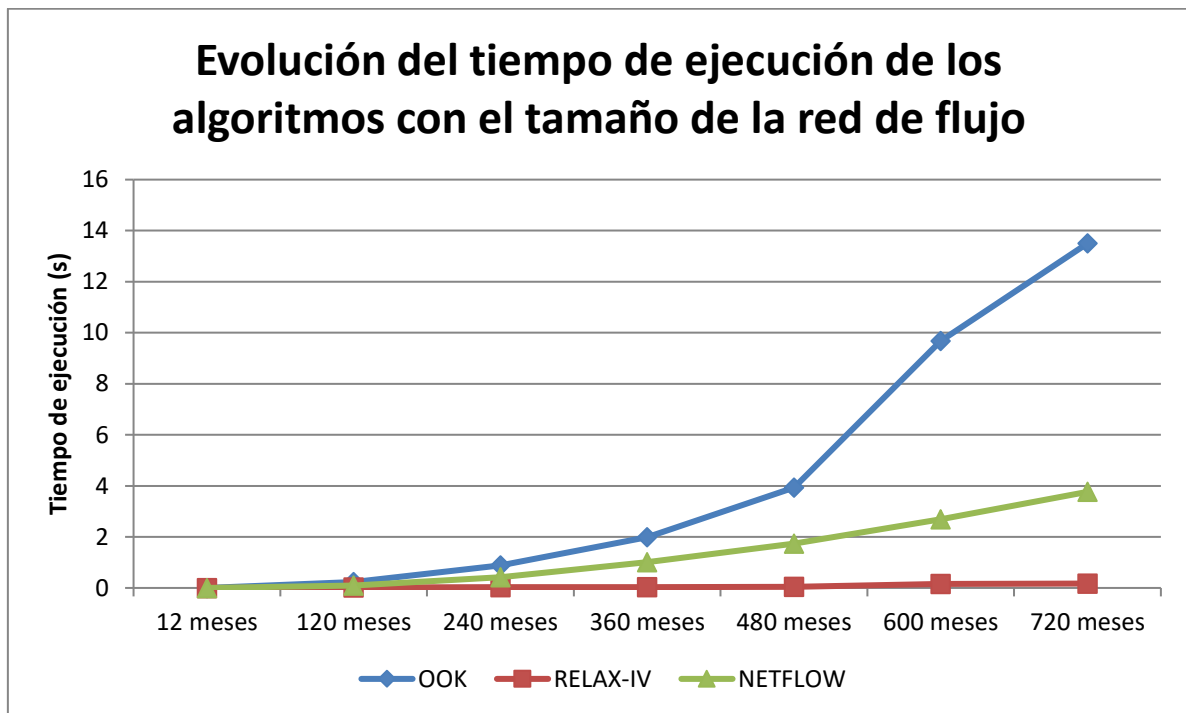


Figura 2.3. Evolución del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo

Evolución del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo							
	12 meses	120 meses	240 meses	360 meses	480 meses	600 meses	720 meses
OOK	0	0.234	0.891	1.984	3.938	9.672	13.5
RELAX-IV	0	0.016	0.031	0.031	0.047	0.156	0.172

NETFLOW	0	0.094	0.422	1.016	1.734	2.688	3.766
---------	---	-------	-------	-------	-------	-------	-------

Tabla 2.2. Evolución del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo.

Se puede observar como el algoritmo RELAX-IV supera notablemente en velocidad a los otros dos. En la tabla 2.3 se muestra la relación de velocidades entre ellos.

Relación del tiempo de ejecución de los algoritmos con el tamaño de la red de flujo							
	12 meses	120 meses	240 meses	360 meses	480 meses	600 meses	720 meses
OOK/RLX	0	15.00	28.50	63.50	84.00	61.90	78.55
OOK/NF	0	2.50	2.11	1.95	2.27	3.60	3.59
NF/RLX	0	6.00	13.50	32.50	37.00	17.20	21.91

Tabla 2.3. Relación entre los tiempos de ejecución de los algoritmos según el tamaño de la red de flujo.

Como indicadores de la bondad de los resultados, es decir, para comprobar que los resultados de los diferentes algoritmos son los mismos, se emplearon:

- El valor final de la función objetivo, la cual coincide siempre se emplee el algoritmo que se emplee.
- El valor del volumen embalsado, en cada embalse por separado, correspondiente al último mes del periodo de optimización, el cual, debido a que se trata del único mes en que se tiene en cuenta la prioridad de un embalse respecto a otro, debería ser el mismo para los tres algoritmos como, efectivamente, ocurre.
- El volumen embalsado total, que coincide, igualmente, para los tres algoritmos y que puede observarse en la figura 2.4.

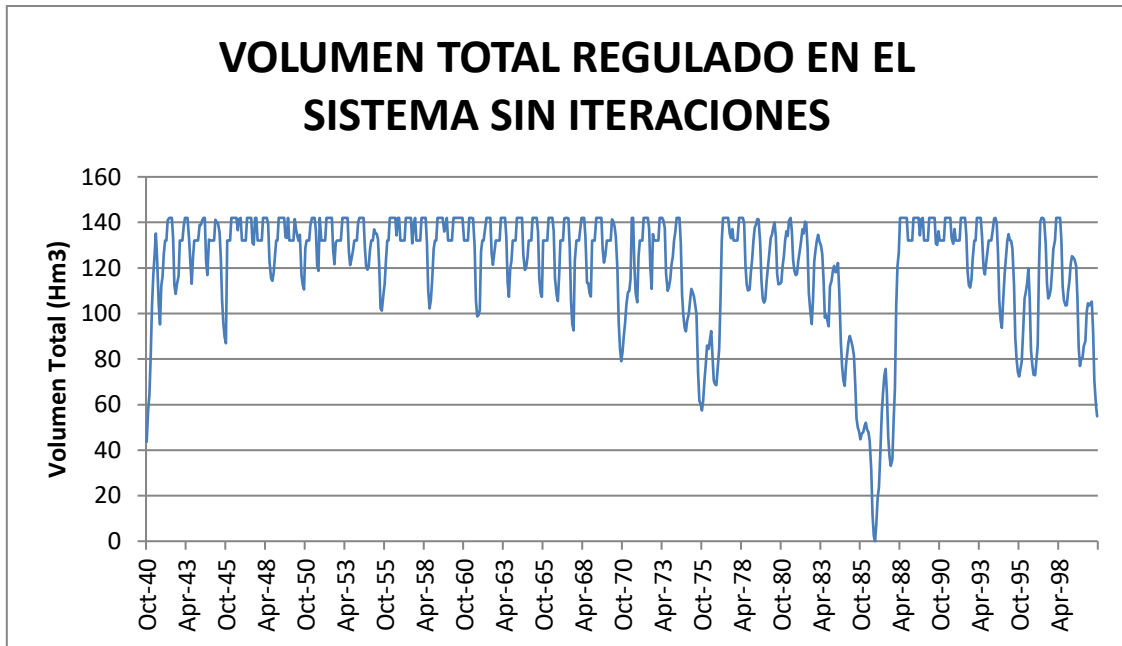


Figura 2.4. Volumen total regulado en el sistema sin iteraciones con los tres algoritmos para el periodo de optimización correspondiente a 720 meses. No se observa ninguna diferencia entre ninguno de ellos.

Cabe destacar que, pese a que se obtiene el mismo volumen total regulado con los tres algoritmos utilizados, no ocurre lo mismo si estudiamos la evolución de los volúmenes embalsados en cada embalse por separado. En las figuras 2.5 y 2.6, correspondientes al volumen regulado por los embalses de Arenós y Sichar respectivamente, se pueden observar las diferencias en los resultados obtenidas con cada algoritmo de resolución de la red de flujo. De todas formas, como ya se ha comentado anteriormente, se puede observar cómo el volumen correspondiente al último mes del periodo de optimización es el mismo en los tres casos e igual a 54.87Hm³ en el caso del embalse de Arenós y 0 Hm³ en del embalse de Sichar.

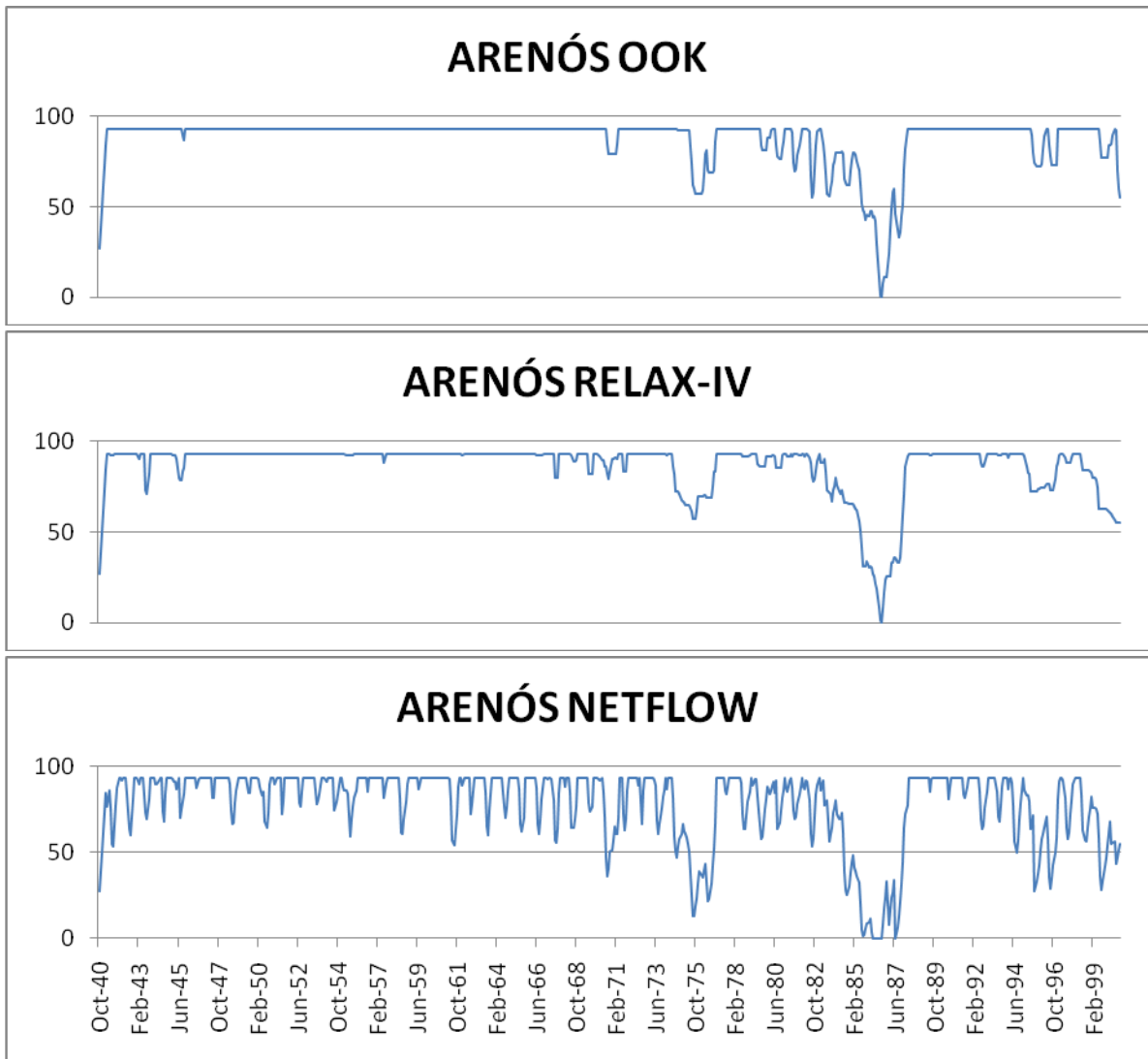


Figura 2.5. Evolución del volumen almacenado en el embalse de Arenós, durante el periodo de optimización de 720 meses, para cada uno de los tres algoritmos comparados.

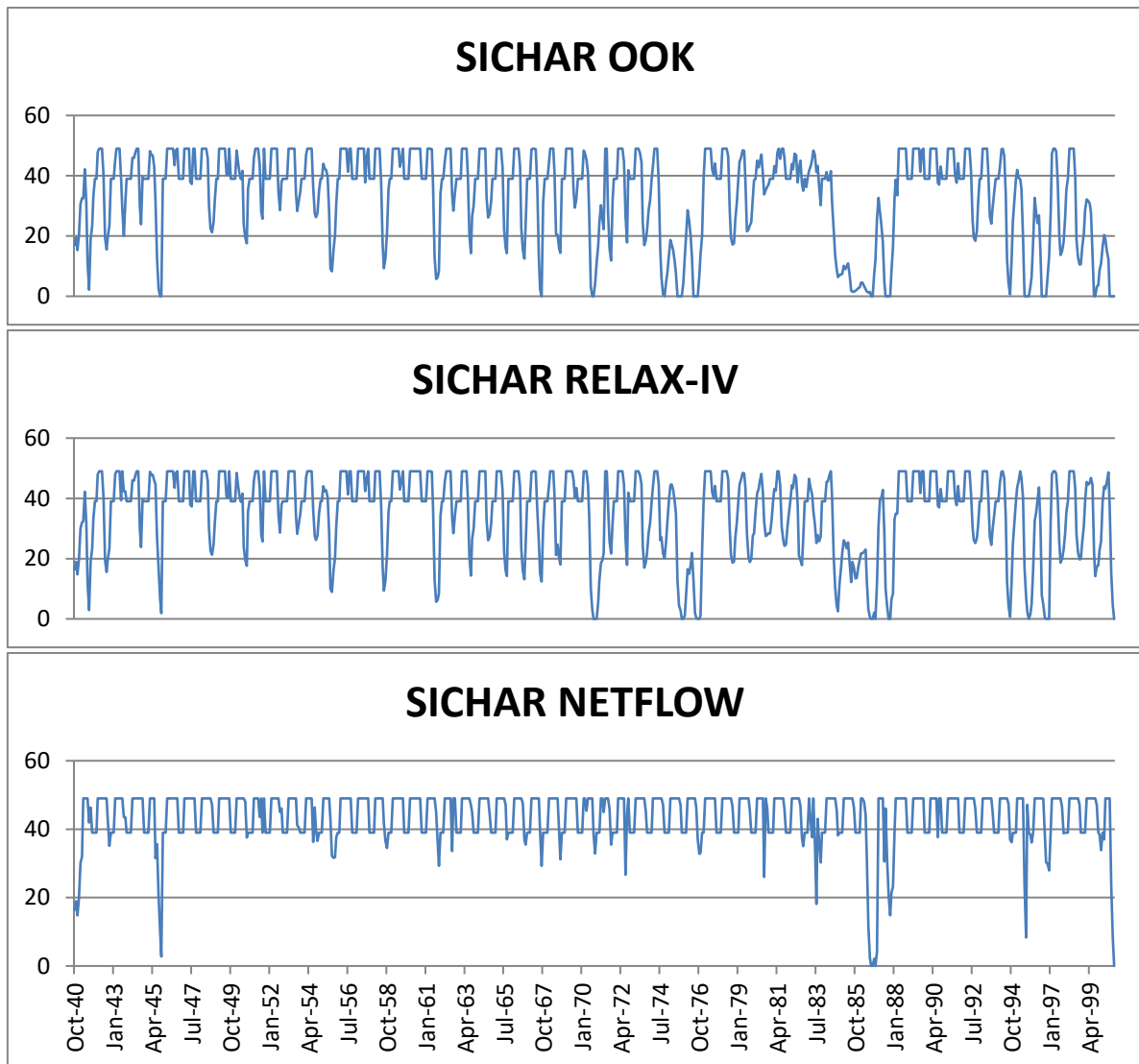


Figura 2.6. Evolución del volumen almacenado en el embalse de Sicchar, durante el periodo de optimización de 720 meses, para cada uno de los tres algoritmos comparados.

La diferencia en los resultados obtenidos en cada embalse por separado es debida, principalmente a que el óptimo del sistema no se corresponde con un único punto, sino que se corresponde con un hiperplano en cuyos puntos la función objetivo tiene el mismo valor. Este aspecto, sumado a la diferente metodología de búsqueda empleada por cada algoritmo, hace que se alcancen soluciones diferentes en los resultados pormenorizados del sistema aunque iguales en conjunto.

2.3. Introducción de no linealidades en la optimización del sistema

A continuación se trata la inclusión de aspectos en el sistema cuyo comportamiento no es lineal y, por tanto, no pueden resolverse directamente con la red de flujo. Para resolver este problema sin dejar de utilizar redes de flujo se recurre a las iteraciones.

El procedimiento para realizar las iteraciones que se sigue en OPTIGES consiste en resolver una red de flujo que inicialmente tiene asignados flujos nulos a través de los

arcos correspondientes a los elementos que representan las no linealidades. Posteriormente, y en base al resto de flujos circulantes en el sistema, se calcula el valor del flujo que debería circular a través de los arcos anteriormente descritos. El valor obtenido en cada arco se compara con el valor anterior (que en el caso de la primera iteración es cero) y si la diferencia es menor que un valor previamente definido, que en adelante llamaremos criterio de convergencia, se considera que no es necesario seguir realizando iteraciones y el último valor calculado se corresponde con el valor solución. Cabe notar que el criterio de convergencia debe cumplirse a la vez en todos los arcos del sistema y que si no se cumple, aunque sea sólo en uno, el modelo realizará una nueva iteración.

En el caso de la evaporación en los embalses, el procedimiento de cálculo consiste en obtener la superficie media del embalse en el mes a partir de los valores del volumen almacenado al inicio y al final del mismo y la curva superficie-volumen asociada al embalse, que el usuario debe definir como datos al crear el modelo de optimización.

El cálculo de los retornos de las demandas se realiza mediante la multiplicación del suministro a cada demanda por el porcentaje de retorno que le corresponda en cada mes, que también es un dato que el usuario debe definir durante la creación del modelo de optimización.

En los siguientes apartados se presentan los resultados de la inclusión de iteraciones en el modelo de optimización¹. Se comenzará por introducir los aspectos no lineales por separado y posteriormente se incluirán de forma conjunta. Además de a los indicadores de bondad de las soluciones observados en el apartado anterior, se prestará especial atención al número de iteraciones realizadas por cada algoritmo antes de alcanzarse la convergencia, así como al tiempo de ejecución total del modelo de optimización (lectura de datos, inicialización, resolución de las iteraciones y escritura de resultados).

2.3.1. Optimización del sistema teniendo en cuenta los retornos de las demandas.

Se definieron dos elementos de retorno, uno conectado a la demanda "Cota 220" y otro conectado a las otras dos demandas. Los coeficientes de retorno se definieron como el 10% del suministro en las demandas "Cota 220" y "Cota 100" y del 20% en la demanda "Riegos Tradicionales". El retorno de la demanda "Cota 220" se envía al embalse de Sichar y el de las otras dos demandas al nudo anterior al nudo final (se incluyó un nudo intermedio en la conducción entre el embalse de Sichar y el nudo final para poder introducir el segundo elemento de retorno ya que AQUATOOL no permite la llegada de

¹ Se mostrarán los resultados correspondientes al horizonte de optimización de 720 meses por considerarlos más ilustrativos, debido al mayor tamaño de la red de flujo resuelta, que horizontes menores.

elementos de retorno al nudo final). El esquema resultante de la inclusión de los nuevos elementos de retorno se muestra en la figura 2.7.

La evolución del valor de la función objetivo a lo largo de las distintas iteraciones, así como el número de estas, que realiza cada algoritmo se muestra en la figura 2.8. Se puede observar como el valor final de la misma es prácticamente el mismo para los tres algoritmos empleados.

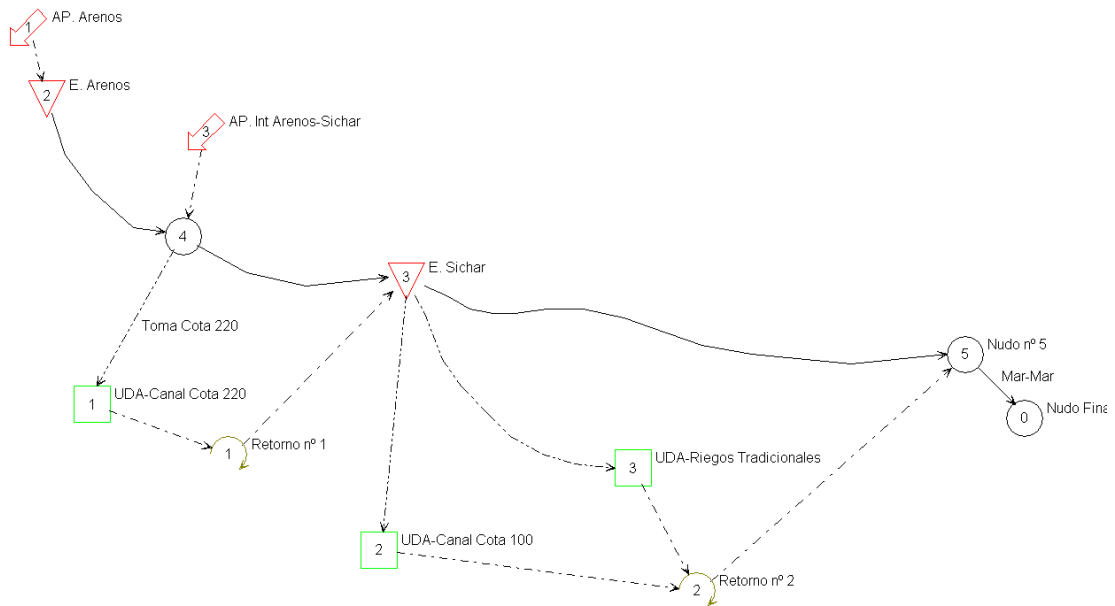


Figura 2.7. Modelo simplificado del sistema Mijares modificado para la inclusión de elementos de retorno.

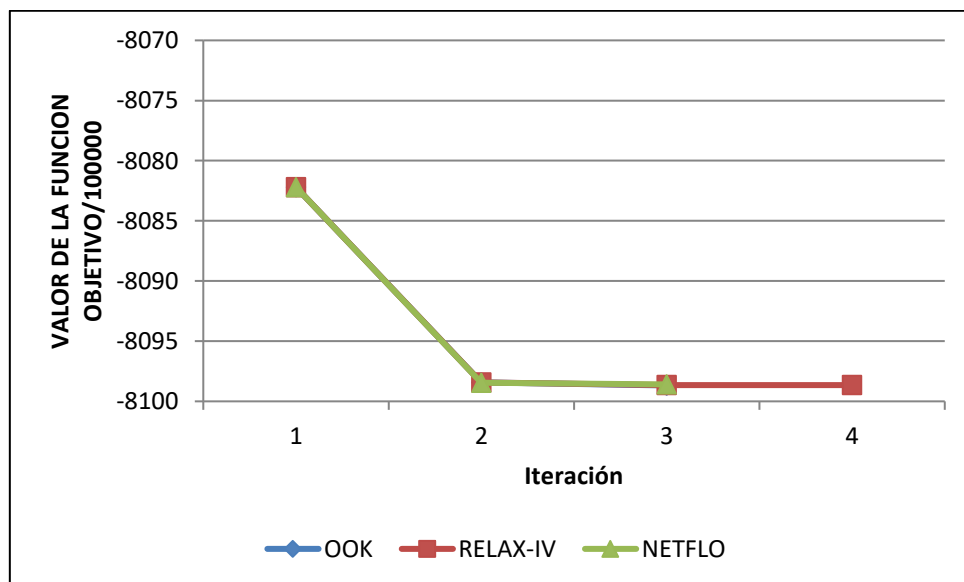


Figura 2.8. Evolución del valor de la función objetivo con el número de iteraciones para el horizonte de optimización de 720 meses.

Al igual que para el valor de la función objetivo, el resto de indicadores que reflejan el funcionamiento adecuado de los tres algoritmos presentan valores muy cercanos entre sí, siendo las pequeñas diferencias debidas principalmente al proceso iterativo. En las tablas que se incluyen a continuación se presentan los valores de los retornos, déficits de las demandas y volumen de agua vertido al mar a través de la última conducción del sistema.

RETORNOS TOTALES			
	OOK	RELAX-IV	NETFLOW
RETORNO 1	139.46	139.43	139.37
RETORNO 2	1177.08	1177.27	1177.33
TOTAL	1316.54	1316.7	1316.7

Tabla 2.4. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.

DÉFICITS TOTALES			
	OOK	RELAX-IV	NETFLOW
COTA 220	32.67	32.98	33.56
COTA 100	44.53	45.11	44.89
R. TRAD	85.91	85.05	84.75
TOTAL	163.11	163.14	163.2

Tabla 2.5. Déficit total en cada demanda para el horizonte de optimización de 720 meses.

VOLUMEN VERTIDO AL MAR			
	OOK	RELAX-IV	NETFLOW
TOTAL	6237.07	6237.26	6237.32

Tabla 2.6. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.

En cuanto al tiempo de ejecución del programa OPTIGES empleando cada uno de los diferentes algoritmos, se puede observar, en la figura 2.9, como el tiempo de ejecución cuando se emplea el algoritmo RELAX-IV, que anteriormente se había visto que es el más rápido, sigue siendo el algoritmo con menor tiempo de ejecución. Además, cabe destacar, respecto al tiempo de ejecución de cada iteración, que, dependiendo del tipo de solución factible inicial que se proporcione al algoritmo, se pueden alcanzar tiempos de ejecución sucesivos cada vez menores. En la figura 2.10 se puede comprobar cómo el tiempo de ejecución de cada iteración de Out-of-Kilter disminuye progresivamente debido a que, en cada nueva iteración, se toma como solución factible inicial la solución obtenida en la iteración anterior mientras que, con los otros dos algoritmos, la solución factible inicial se inicializa siempre como cero² y, por tanto, el tiempo de resolución de la red de flujo es aproximadamente el mismo cada vez.

² En sucesivas versiones de OPTIGES empleando RELAX-IV y NETFLO se implementará la inicialización de cada iteración con la solución de la iteración anterior.

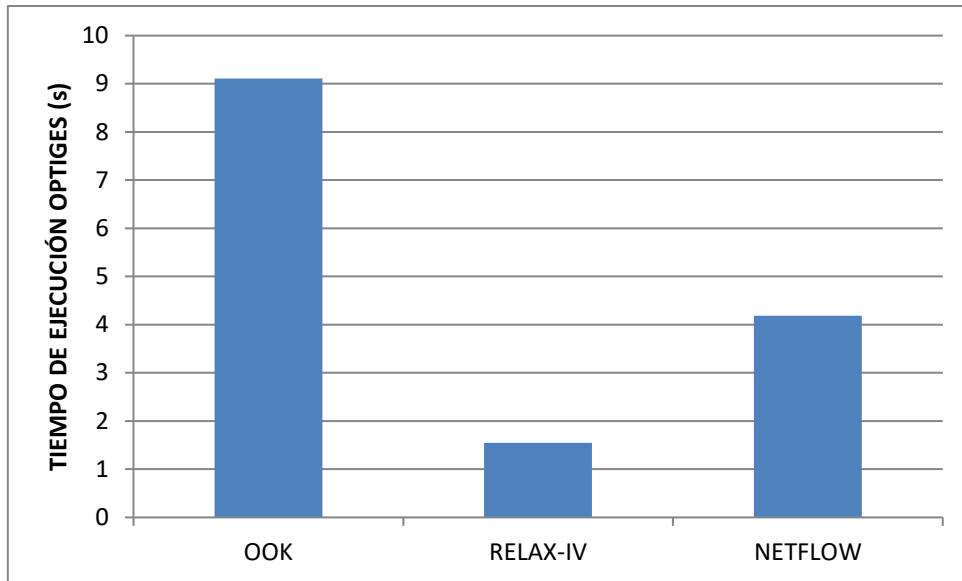


Figura 2.9. Tiempo de ejecución del modelo de optimización con cada algoritmo para el horizonte de optimización de 720 meses.

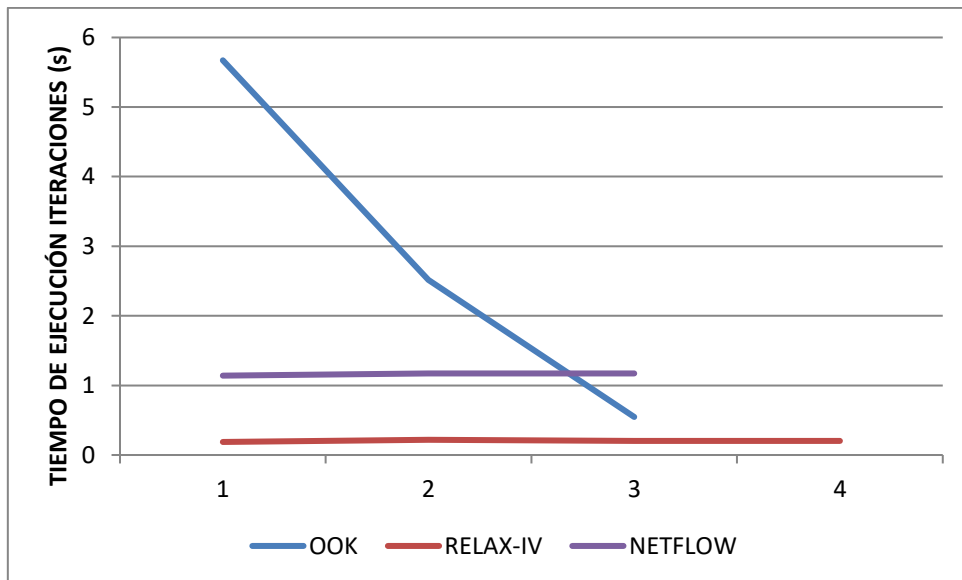


Figura 2.10. Tiempo de ejecución de cada iteración del modelo de optimización con cada algoritmo para el horizonte de optimización de 720 meses.

2.3.2. Optimización del sistema teniendo en cuenta la evaporación en los embalses.

En este apartado se introduce, como aspecto no lineal, la evaporación en los embalses. La tasa de evaporación es un dato que el usuario debe definir en la ventana correspondiente en el programa AQUATOOL al definir las características del modelo. Para posibilitar una reproducción del modelo presentado se incluye en la tabla 2.7 la tasa de evaporación correspondiente a cada embalse.

	TASA DE EVAPORACIÓN (mm/mes)	
MES	ARENÓS	SICCHAR
Octubre	57.4	65.6
Noviembre	36.4	41.6
Diciembre	30.1	34.4
Enero	34.3	39.2
Febrero	52.5	60
Marzo	63.7	72.8
Abril	83.3	95.2
Mayo	109.2	124.8
Junio	126.7	144.8
Julio	153.3	175.2
Agosto	127.4	145.6
Septiembre	90.3	103.2

Tabla 2.7. Tasa de evaporación en los embalses del modelo.

En este caso, observamos comportamientos distintos de cada uno de los algoritmos que, aun así, pueden justificarse sin problemas. En primer lugar, en la figura 2.11, sorprende encontrarse con unos resultados de la función objetivo dispares, sobre todo por parte del algoritmo NETFLO, cuyo valor difiere bastante respecto del obtenido con los otros dos. También cabe observar la cantidad de iteraciones que se realiza en el caso de RELAX-IV.

La diferencia en el valor de la función objetivo es debida a la diferente gestión de los embalses que realiza cada algoritmo, como ya se comentó en el apartado 3.2. Debido a estas diferencias en el volumen almacenado en cada embalse cada mes, el valor calculado para la evaporación a través de las iteraciones será diferente en cada caso. Esto, unido a que el coste de circulación del flujo a través de los arcos asociados a la evaporación es muy alto, para forzar así la circulación del flujo correspondiente, hace que el valor de la función objetivo difiera bastante de un algoritmo a otro. Aún así la diferencia no es significativa dado el orden de magnitud de los valores numéricos manejados.

El caso del número de iteraciones se analizará más en profundidad en el apartado 4.4.

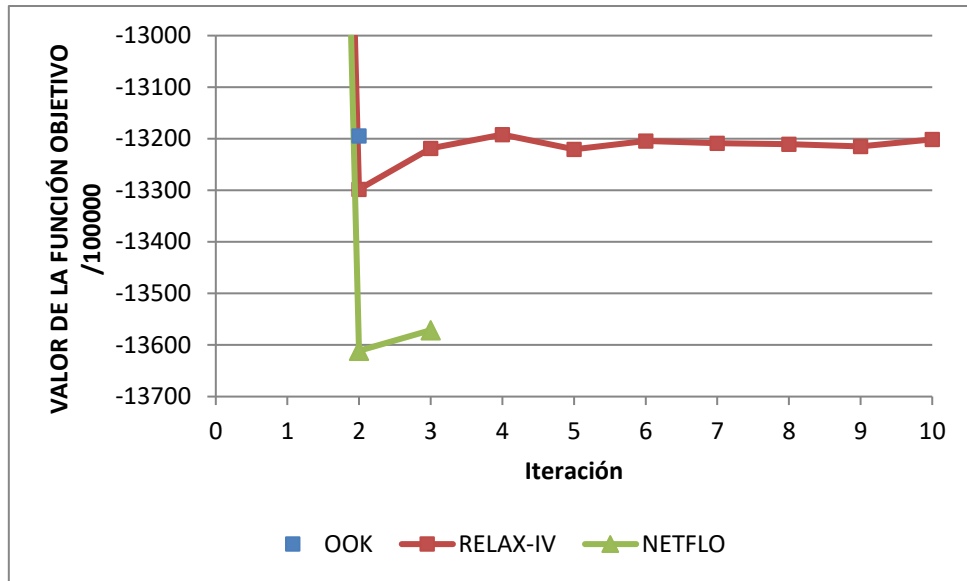


Figura 2.11. Evolución del valor de la función objetivo con cada iteración para un horizonte de optimización de 720 meses.

En cuanto a los resultados del resto de indicadores de bondad, se observan pequeñas diferencias debidas, principalmente a lo ya comentado previamente en este apartado y en el anterior.

EVAPORACION TOTAL			
	OOK	RELAX IV	NETFLO
ARENÓS	177.56	173.28	155.12
SICHAR	166.05	171.33	213.92
TOTAL	343.61	344.61	369.04

Tabla 2.8. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.

DÉFICITS TOTALES			
	OOK	RELAX-IV	NETFLO
COTA 220	44.31	44.41	44.93
COTA 100	61.44	61.38	61.49
R. TRAD	118.22	119.76	122.33
TOTAL	223.97	225.55	228.75

Tabla 2.9. Déficit total en cada demanda para el horizonte de optimización de 720 meses.

TOTAL AL MAR			
	OOK	RELAX IV	NETFLOW
TOTAL	4654.91	4656.38	4636.97

Tabla 2.10. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.

Puede observarse como, debido a la mayor evaporación calculada con NETFLO, el déficit del sistema es mayor y las salidas al mar son menores.

En cuanto al tiempo de ejecución, se siguen manteniendo los bajos tiempos de resolución de cada iteración pero debido a que RELAX-IV alcanza el máximo de

iteraciones (200) antes de que se alcance el criterio de convergencia, el tiempo que OPTIGES tarda en ejecutarse con este algoritmo es muy superior al empleado con los otros algoritmos, que sí alcanzan una rápida convergencia. Cuando se trate el tema de la convergencia de RELAX-IV más adelante se realizarán unas comparaciones de tiempos de ejecución y se retomará este punto. Se adelanta ya que, con un número adecuado de iteraciones, RELAX-IV sigue siendo el algoritmo más prometedor.

2.3.3. Optimización del sistema teniendo en cuenta evaporación y retornos.

Una vez observado el comportamiento de los diferentes algoritmos con las no linealidades que permite incluir el programa OPTIGES por separado, en el apartado que sigue se combinan ambas posibilidades.

Los datos empleados para llevar a cabo este apartado son los mismos que los empleados para realizar los apartados anteriores sobre el esquema modificado para incluir los elementos de retorno.

Los resultados obtenidos esta vez no suponen nada nuevo respecto a los obtenidos en los dos apartados anteriores salvo por ser una combinación de ambos. Se observa una diferencia más notable del valor de la función objetivo así como del número de iteraciones, que siguen alcanzando el máximo definido en OPTIGES con el algoritmo RELAX-IV. Mientras los resultados asociados a los retornos son muy similares con los tres algoritmos, el valor de la evaporación sigue dependiendo mucho de la gestión de los embalses que realiza cada uno y, como ya se observó anteriormente, resulta mayor para OPTIGES con el algoritmo NETFLO. Debido a las diferencias en las evaporaciones se tienen, de igual forma, diferencias en los déficits y en el agua vertida al mar. En las siguientes figuras se muestran los resultados obtenidos para una resolución del modelo para un horizonte de optimización de 720 meses.

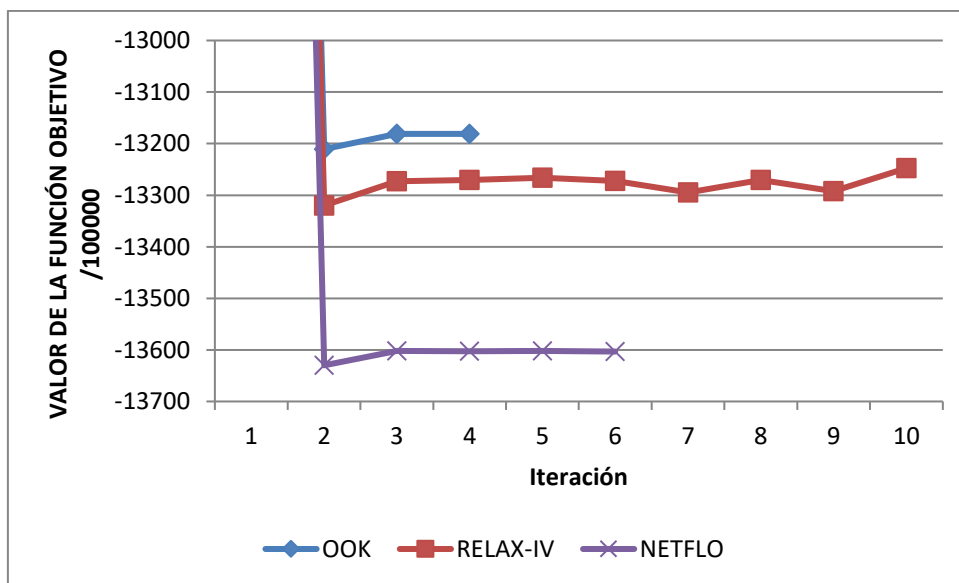


Figura 2.12. Evolución del valor de la función objetivo con cada iteración para un horizonte de optimización de 720 meses.

	EVAPORACIÓN TOTAL		
	OOK	RELAX-IV	NETFLO
ARENOS	176.87	173.82	156.74
SICHAR	164.81	173.34	213.37
TOTAL	341.68	347.16	370.11

Tabla 2.11. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.

	RETORNO TOTAL		
	OOK	RELAX-IV	NETFLO
RETORNO 1	138.56	138.46	138.38
RETORNO 2	1171.28	1171.36	1170.83
TOTAL	1309.84	1309.82	1309.21

Tabla 2.12. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.

	DÉFICITS TOTALES		
	OOK	RELAX-IV	NETFLO
COTA 220	41.66	42.63	43.55
COTA 100	57.19	57.85	58.19
R. TRAD.	108.92	108.7	110.72
Total	207.77	209.18	212.46

Tabla 2.13. Déficit total en cada demanda para el horizonte de optimización de 720 meses.

	TOTAL AL MAR		
	OOK	RELAX-IV	NETFLO
TOTAL	5944.11	5941.62	5922.52

Tabla 2.14. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.

En cuanto al tiempo de ejecución, RELAX-IV sigue siendo el algoritmo el más rápido.

2.4. Análisis de resultados.

Aunque a lo largo de la presentación de resultados ya se ha analizado parte de los mismos, en este apartado se comentarán/resumirán los más importantes para el desarrollo de este estudio.

Para comenzar, se ha podido observar, claramente, que se cumplen las consideraciones realizadas en el capítulo anterior acerca de la eficacia de los algoritmos empleados en la comparación realizada en el presente apartado. Así, hemos comprobado que el algoritmo RELAX-IV es, de los tres estudiados, el más eficiente en la resolución de una red de flujo. Al respecto de los tiempos de ejecución también se puede observar,

apoyándose en la figura 2 y las tablas 2 y 3, que el incremento del tiempo de ejecución con el tamaño de la red no es tan rápido en el algoritmo RELAX-IV como en los otros dos, y que éste puede llegar a ser casi 80 veces más rápido que el algoritmo Out-of-Kilter, utilizado hasta ahora.

Se ha podido ver que, dependiendo del algoritmo empleado para la resolución de una misma red, se obtenían resultados que no tenían demasiado que ver unos con otros. Este aspecto es mucho más notable en la operación de los embalses, especialmente debido a que la red resuelta sólo tiene en cuenta la prioridad de uno respecto del otro en el último mes del periodo de optimización. Por esta razón, cada algoritmo, por sus diferentes metodologías de búsqueda del óptimo asigna los flujos en los arcos como mejor le conviene aunque, al final, el valor de la función objetivo sea el mismo. Se comprobó que si la prioridad de un embalse respecto a otro se hacía extensiva a todo el periodo de optimización, todos los algoritmos ofrecían un mismo valor de todos los resultados.

Una vez comprobado qué algoritmo de resolución resulta más rápido, es necesario centrarse en su implementación en OPTIGES y comprobar su comportamiento cuando se hace uso de las posibilidades que ofrece el programa de introducir algunos elementos de comportamiento no lineal mediante iteraciones. Dado que las iteraciones en OPTIGES se realizan resolviendo cada vez una red de flujo distinta, ya podía discernirse de los resultados de las comparaciones en la resolución de una red de flujo pura que RELAX-IV sería también el algoritmo más eficiente aunque, mediante la resolución de los diferentes casos, se ha podido comprobar la existencia de problemas en cuanto a la convergencia de los resultados.

Aunque no ocurre en el caso de los retornos de las demandas, cuando se introducen evaporaciones en los embalses, se comprueba cómo el modelo funcionando con el algoritmo RELAX-IV no alcanza la convergencia de las soluciones y se llega al máximo de iteraciones. Que esto se dé sólo en uno de los casos en que se realizan iteraciones (dos si se tiene en cuenta el ejemplo en que se consideran las dos posibilidades conjuntamente) permite pensar que el problema no es debido al algoritmo en sí, sino al criterio de convergencia escogido.

Cuando se realizan las comparaciones entre el valor calculado en la iteración anterior y el calculado en la presente iteración, se exige que la diferencia sea menor que un valor definido internamente en el programa (NEPSI) y que actualmente es igual a 4, lo que viene a representar $0.04Hm^3$, sin que el usuario pueda modificarlo. Cabe la posibilidad de que este valor sea muy reducido según el sistema considerado y que, sumado al hecho de que el óptimo no se corresponde con una única solución, el algoritmo arroje soluciones próximas entre sí cada vez, pero no lo suficiente como para cumplir el criterio de convergencia, mientras que el valor de la función objetivo se mueve alrededor de un valor central.

Resulta interesante, por lo tanto, estudiar el efecto que tendría modificar el criterio de convergencia en los resultados arrojados por el modelo. Para ello se ha empleado el último de los casos estudiados anteriormente, en el que se tenía evaporación y retornos al mismo tiempo. Se incrementó el valor de NEPSI gradualmente para comprobar en qué momentos se reducía el número de iteraciones efectuadas por RELAX-IV. De esta forma se obtiene una gráfica como la que se presenta en la figura 2.13. En ella se puede ver como para valores de NEPSI menores de 9, se realizan más de 200 iteraciones, entre 9 y 10 se realizan 157 iteraciones y, a partir de 11, sólo tres iteraciones. Con esto, se compararon los resultados pormenorizados de emplear cada uno de los valores clave de la variable NEPSI. Así se obtuvieron los valores mostrados en las tablas 2.15 a 2.18.

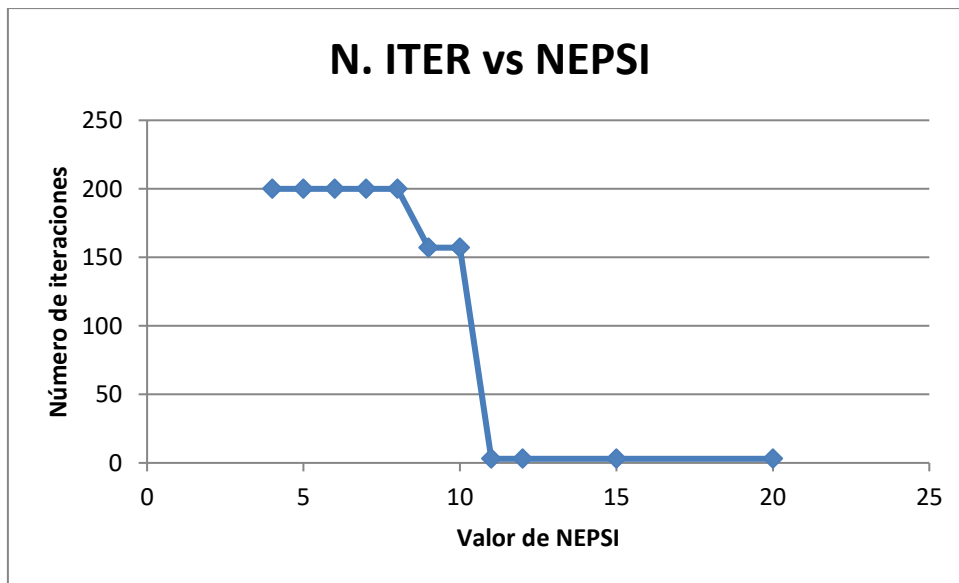


Figura 2.13. Número de iteraciones según el valor de la variable NEPSI.

EVAPORACIÓN TOTAL			
	NEPSI=4	NEPSI=9	NEPSI=11
ARENOS	173.82	172.94	173.3
SICCHAR	173.34	175.97	174.6
TOTAL	347.16	348.91	347.9

Tabla 2.15. Volumen total evaporado en los embalses para un horizonte de optimización de 720 meses.

RETORNO TOTAL			
	NEPSI=4	NEPSI=9	NEPSI=11
RETORNO 1	138.46	138.47	138.56
RETORNO 2	1171.36	1171.44	1171.31
TOTAL	1309.82	1309.91	1309.87

Tabla 2.16. Volumen total retornado por las demandas a través de cada elemento de retorno para el horizonte de optimización de 720 meses.

DÉFICITS TOTALES			
------------------	--	--	--

	NEPSI=4	NEPSI=9	NEPSI=11
COTA 220	42.63	42.29	42.21
COTA 100	57.85	58.2	58.48
R. TRAD.	108.7	108.36	108.19
Total	209.18	208.85	208.88

Tabla 2.17. Déficit total en cada demanda para el horizonte de optimización de 720 meses.

TOTAL AL MAR			
	NEPSI=4	NEPSI=9	NEPSI=11
TOTAL	5941.62	5940.12	5940.95

Tabla 2.18. Volumen total vertido al mar a través de la última conducción del sistema para el horizonte de optimización de 720 meses.

Como se puede ver en las tablas de resultados mostradas, éstos no difieren mucho unos de otros independientemente del valor de la variable NEPSI. Este hecho nos hace pensar que el criterio de convergencia está influido directamente por las características del sistema estudiado y, por lo tanto, deberá definirse un criterio distinto según éstas cada vez. Una posibilidad que se plantea es sustituir el proceso de comprobación de la convergencia que se está empleando actualmente, que consiste en comprobar la convergencia de todos y cada uno de los resultados relacionados con la variable correspondiente a cada proceso no lineal, por uno más sencillo, pero no por ello de menor exigencia, como podría ser realizar la comprobación de convergencia únicamente con el valor que más se diferenciase de la iteración anterior y confirmar que la desviación no es mayor que el orden de magnitud que se está manejando con dicha variable.

Pese a lo comentado en el párrafo anterior sobre el criterio de convergencia, se ha podido observar anteriormente en los resultados que el valor de la función objetivo oscilaba alrededor de un valor prácticamente fijo a lo largo de todas las iteraciones. En la figura 2.14 se muestra la evolución de la función objetivo a lo largo de las 200 iteraciones que realiza el modelo con RELAX-IV. En ella se puede observar como ésta oscila alrededor de un valor medio que se ha estimado en $-13273 \cdot 10^5$, marcado en la gráfica. Esto, unido al hecho observado anteriormente de la escasa variación de los resultados independientemente del número de iteraciones, permite considerar también la posibilidad de aceptar el resultado obtenido al final de un cierto número de iteraciones sin necesidad de variar el criterio de convergencia.

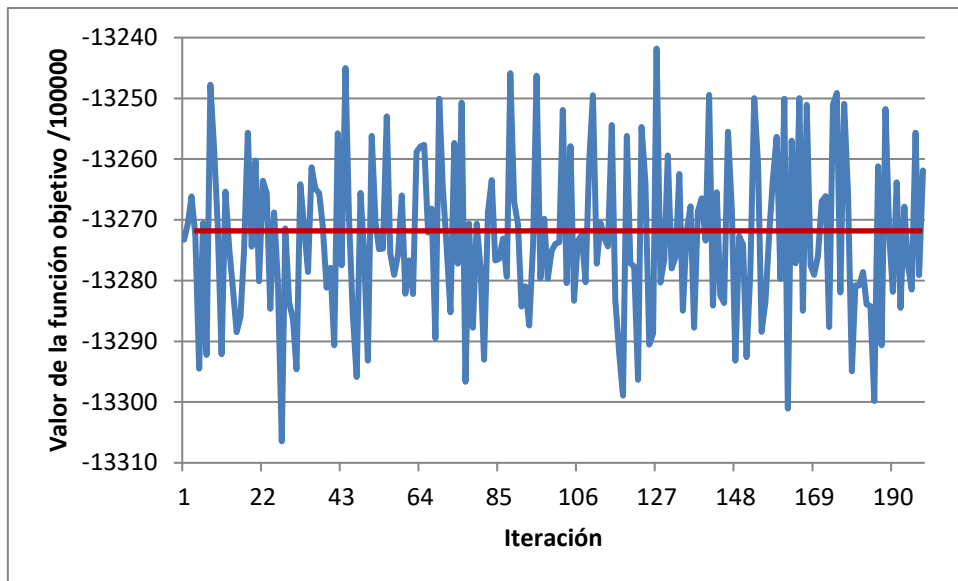


Figura 2.14. Evolución del valor de la función objetivo a lo largo de 200 iteraciones con OPTIGES con RELAX-IV.

Los dos aspectos comentados acerca de la convergencia del modelo trabajando con RELAX-IV han de comprobarse más en profundidad mediante la aplicación de este algoritmo a otros esquemas de sistemas de recursos hídricos para averiguar si este comportamiento es generalizado y, entonces, poder definir las posibles metodologías para darle una solución, y que serán germen de futuros estudios al respecto de lo tratado en el presente informe.

Una última consideración que se puede hacer al respecto de la convergencia del algoritmo, y que diverge en cierta manera del camino definido en los párrafos anteriores, es la posibilidad de alcanzar una convergencia más rápida si se emplease como solución factible inicial de cada iteración de RELAX-IV la solución de la iteración anterior, como así se hace con Out-of-Kilter y que, además, acelera el proceso de búsqueda del óptimo al partir de una solución mucho más cercana a este que si se parte cada vez de la solución de compromiso en que todos los flujos iniciales son nulos.

2.5. Consideraciones finales del capítulo

De la comparación de los algoritmos de resolución de redes de flujo aplicados a la optimización de un sistema de recursos hídricos, se puede concluir que RELAX-IV va a permitir mejorar, notablemente, el funcionamiento de OPTIGES respecto a su funcionamiento actual con el algoritmo Out-of-Kilter.

La eficiencia del nuevo algoritmo, más de 80 veces más rápido que el actual, debería permitir plantear la optimización de sistemas de tamaños y complejidad mucho mayores en tiempos lo suficientemente razonables como para plantear estudios de mayor extensión, pudiéndose plantear más alternativas, aspecto poco atractivo si cada ejecución del algoritmo lleva aparejada una gran cantidad de tiempo. Si unimos esto a los últimos avances en técnicas de computación, mediante paralelización de los

procesos en un mayor número de núcleos de cálculo, es razonable pensar que las mejoras pueden ser muy beneficiosas.

Aparte de la posibilidad de llevar a cabo estudios con periodos de optimización mucho más largos, la velocidad del algoritmo RELAX-IV permitirá también la realización de muchas más iteraciones en menos tiempo, lo que propicia dar un paso más en el desarrollo del modelo OPTIGES y pasar a la incorporación de nuevos elementos al modelo. Estos nuevos elementos permitirán crear esquemas mucho más detallados cuyos resultados serán más representativos del funcionamiento real de los sistemas de recursos hídricos estudiados.

Pese a lo prometedor del algoritmo, se han planteado problemas de convergencia de las soluciones cuando éste se emplea para resolver sucesivamente redes de flujo con pequeñas diferencias entre sí. Se han localizado diversas causas y posibles soluciones a las mismas que tendrán que estudiarse, y compararse, en subsiguientes investigaciones de forma que podamos plantear la mejor metodología a seguir en el futuro.

A la lectura de este documento, ya existirá una versión del modelo OPTIGES que permitirá la resolución del esquema de recursos hídricos con el algoritmo RELAX-IV, y en que se puedan incluir aspectos tales como los volúmenes mínimos de embalse y caudales mínimos en conducciones que, para el desarrollo de este apartado, se habían obviado.