



PROYECTO

DESARROLLO DE UN MODELO MATEMÁTICO Y SU ALGORITMO DE SOLUCIÓN DE PDM (PROGRAMACIÓN DINÁMICA MULTIDIMENSIONAL) Y EL AOP (ALGORITMO DE OPTIMIZACIÓN PROGRESIVA) PARA EL DISEÑO DE “POLÍTICAS ÓPTIMAS DE OPERACIÓN DE SISTEMAS HIDROELÉCTRICOS EN CASCADA”

Reporte final

En este reporte se presentan los diagramas de flujo del funcionamiento detallado de la implementación de la solución propuesta por Jiang en el artículo base de este proyecto. Se presenta una comparación detallada de los resultados obtenidos con los reportados por Jiang y una fase de experimentación para probar las conclusiones de este proyecto

Alberto Hernández Espinosa
(28 de diciembre de 2018)

Table of Contents

Balance de volumen	3
Cálculo del volumen disponible.	4
Cálculo del volumen que sale por las turbinas	5
Cálculo de la energía total generada.....	6
Discretización del sistema	7
MDP	9
POA.....	14
IMDP	17
Comentarios y conclusiones. (La equivalencia entre el modelo de Jiang y su implementación en MATLAB).	19
Sobre la exactitud en el cálculo de la trayectoria inicial	19
Escalamiento y adaptación	20
Adaptación: Aprovechamiento del posible volumen vertido	22
Comparación directa de los resultados de Jiang y la implementación de este proyecto y conclusiones.....	24
MANUAL DE USUARIO	26
Cálculo de trayectoria inicial usando MDP	26
APLICACIÓN DE POA A UNA TRAYECTORIA INICIAL.....	28

Aplicación de Programación Dinámica Multidimensional (MDP), Principio de Optimización (POA) y MDP mejorado (IMDP) al problema de Optimización de Operación de Sistemas de presas en cascada (CROO).

En el artículo de Jiang se exponen dos metodologías para poder encontrar las curvas de operación de un sistema de presas en cascada: POA e IMDP. Estos dos métodos de optimización requieren de una trayectoria inicial a optimizar. Para calcularla Jiang propone encontrar la aplicación de PD (Programación Dinámica) de muy gruesa aproximación, en otras palabras, aplicar MDP con una discretización del problema de pocos puntos.

El objetivo de aplicar métodos de optimización sobre una trayectoria inicial es para encontrar la trayectoria de operación que genere la mayor cantidad de energía posible dadas las características específicas del sistema, como el numero de presas, sus capacidades de almacenamiento, los datos de escurrimientos al sistema y el número y capacidades de generación de energía de las turbinas en cada una de las presas.

Para calcular en cada paso la cantidad de energía posible entre dos puntos en la discretización del sistema se requiere hacer lo que Jiang llama un “balance de volumen” que da como resultado para cierto periodo de tiempo t y entre dos puntos, uno inicial (h_0 o m_1) y final (h_1 o m_2) en la discretización del sistema la cantidad de energía generada. Del calculo iterativo del balance de volumen entre todas las posibles combinaciones de puntos en el tiempo de planeación se seleccionan, cada ciclo de cálculo, las combinaciones que generen los máximos posibles. Esta es, en esencia, la aplicación del principio de programación dinámica al problema CROO.

En resumen, el balance de volumen es un procedimiento clave en la aplicación de las metodologías propuestas por Jiang. En los siguiente secciones se detalla este proceso y cómo es usado en MDP, POA e IMDP para encontrar las trayectorias o curvas de operación de un sistema de presas en cascada.

Balance de volumen

El balance de volumen consta de 3 pasos que se muestran en la figura 4(A), después de tener los datos de todas las entradas (en términos de volúmenes) a una presa, 1) se calcula el volumen disponible, 2) el volumen que sale por las turbinas y 3) la energía generada con el volumen que sale por las turbinas.

Cada uno de estos pasos se explica con las figuras 4(B), 4(C) y 4(D). En los siguientes párrafos se explica cómo estos diagramas corresponden al código entregado en el desarrollo de este proyecto y la prueba de concepto (paso a paso) entregado como hoja de cálculo.

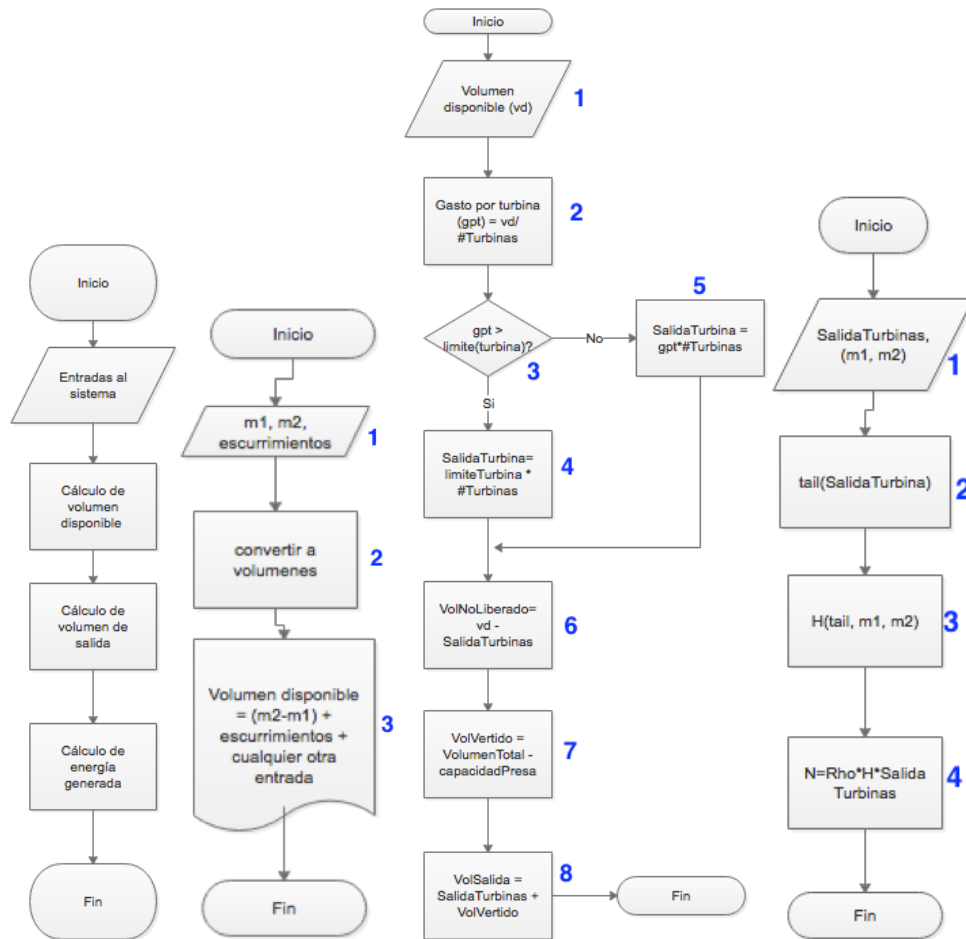


Fig. 4. Detalle de balance de volumen usado en MDP. (A) Proceso general del balance de volumen. (B) Cálculo de gasto disponible, (C) Cálculo del gasto que sale por las turbinas, (D) Cálculo de la energía generada por la salida de volumen por las turbinas

Cálculo del volumen disponible.

El volumen disponible corresponde al volumen de agua total con el que se cuenta para generar energía, y no es más que la suma de todas las entradas de volumen al sistema, lo que comprende, en esencia, los datos de escurrimiento mas el volumen que sale de la presa al ir de h_0 a h_1 en el periodo de tiempo t .

El cálculo de este volumen se presenta en la figura 4(B) que corresponde a la rutina **availableVolumeUpstream** y **availableVolumeDownstream** del código en MATLAB. La primer rutina se muestra en la figura 5. En ésta se muestra como los pasos 1-3 del diagrama de flujo 4(B) corresponden a los pasos numerados en el código. Un ciclo de cálculo numérico del gasto disponible, corresponden a las líneas 1-9 de la tabla 2 de la pagina 13, en el reporte 1 de este proyecto, como se muestra en la figura 6.

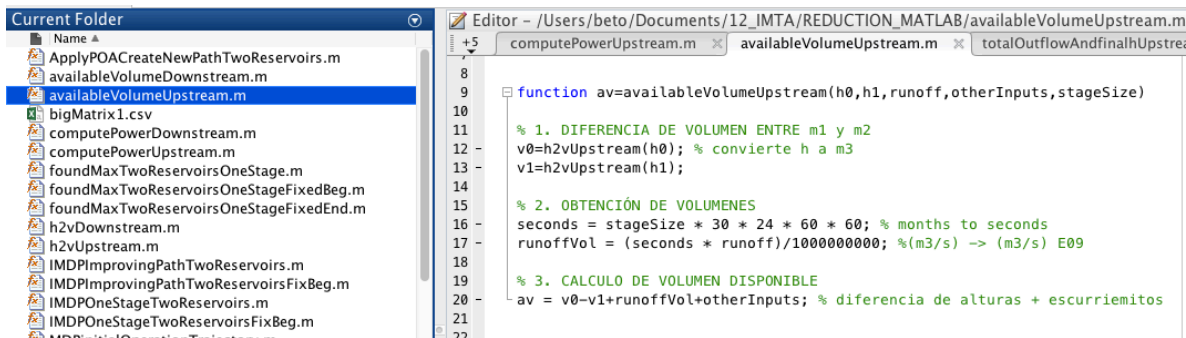


Fig. 5. Código para el proceso de cálculo de gasto disponible

Initial vol	h0	normal level	(msnm)	400	1
	v0	by table	(m3) E09	4.1158	2
Final volume	h1	discretized level	(msnm)	385	3
	v1	by table	(m3) E09	3.3512	4
Runoff volume calculation	runoff	runoff data (from paper data)	m3/s	178	5
	t	1 month in hours	(h)	720	6
	v3	vol of runoff data in one month	(m3)	461376000	7
	v3	runoff volume	(m3) E09	0.4613760	8
Water availability	available V	v_0-v_1+v_runoff	(m3) E09	1.2259210	9

Fig. 6. Paso a paso del calculo de gasto disponible

Cálculo del volumen que sale por las turbinas

Del gasto disponible, puede ser que por las capacidades de las turbinas, pueda o no salir por ellas, de modo que hay que calcular cuanta agua es posible usar para efectivamente generar energía, este es precisamente el cálculo del volumen que sale por las turbinas

Este cálculo se muestra en la figura 4(C), y corresponde al código de las rutinas **totalOutflowAndFinalhUpstream** y **totalOutflowAndFinalhDownstream**. La primera de estas rutinas se muestra en la figura 7, y se emparenta con el diagrama 4(C) mediante la numeración de pasos. Numéricamente este proceso corresponde a los renglones 10-21 de la tabla 2, en la página 13 del reporte 1 de este proyecto como muestra la figura 8.

Current Folder

- h2vDownstream.m
- h2vUpstream.m
- IMDPImprovingPathTwoReservoirs.m
- IMDPImprovingPathTwoReservoirsFixBeg.m
- IMDPOneStageTwoReservoirs.m
- IMDPOneStageTwoReservoirsFixBeg.m
- MDPInitialOperationTrajectory.m
- MDPInitialOperationTrajectoryFixedBeg.m
- MDPInitialOperationTrajectoryFixedEnd.m
- POAFixedEndAndBegOneStep.m
- runoffVolume.m
- setCorridorLimits.m
- tailDownstream.m
- tailUpstream.m
- totalOutflowAndfinalhDownstream.m
- totalOutflowAndfinalhUpstream.m**
- totalOutflowAndfinalhUpstream.m~
- v2hDownstream.m

totalOutflowAndfinalhUpstream.m (Function)

```

31 % 1. VOLUMEN DISPONIBLE
32 if availableVol > 0 % (m3) E09 (it must to be possitive)
33     potentialq=(availableVol/seconds)*1000000000;% (m3)
34 end
35
36 % 2. GASTO POR TURBINA
37 qPerTurbine = potentialq/4; % four turbines (m3/s)
38
39 % if outflow per turbine > limit, then outflow = limit * number of turbines
40 rateOutflowAllTurbines = 0;
41
42 % 3. ¿ES EL GASTO POR TURBINA MENOR AL LIMITE POR TURBINA?
43 if qPerTurbine < turbineLimit
44     % 4. SI SÍ, ENTONCES DIVIDE EL GASTO TOTAL ENTRE EL #TURBINAS
45     rateOutflowAllTurbines=(qPerTurbine*4)/10000; % 4 turbines in dam, (m3)
46 else
47     % 5. SI NO, ENTONCES, LIMITE * NUMERO DE TURBINAS
48     rateOutflowAllTurbines=(turbineLimit*4)/10000;
49 end
50
51 % fixing units
52 turbinesOutflow = (rateOutflowAllTurbines * seconds)/100000; % (m3) E09
53
54 % flow that cannot be leaved thought turbines is added to h1, this is vol
55 % remaining in reservoir
56 % 6. CALCULO DE VOLUMEN NO LIBERADO
57 nonDeliveredVol = availableVol-turbinesOutflow;
58 newV1=v1+nonDeliveredVol;
59
60 % if v1 (end vol) > storage reservoir capacity, then is spilled
61 % 7. CALCULO DE VOLUMEN VERTIDO
62 spilledVol = 0;
63 if newV1 > storageReservoirCapacity
64     spilledVol = newV1-storageReservoirCapacity;
65 end
66
67 % in any case, delivered vol is vol-delivered by turbines + spilled one
68 totalOutflow = turbinesOutflow + spilledVol;
69 % 8. SALIDA TOTAL DE LAS TURBINAS
70 % real and final h in reservoir
71 newh1 = v2hUpstream(newV1-spilledVol);

```

Fig. 7. Código para el cálculo del volumen que sale por las turbinas

Q through turbines	Potential q	Potential q to be delivered througuh turbines	(m3) E09/s	0.00000047296333	10 ¹
	Potential q		(m3/s)	472.96	11
	q/turb	4 turbinas	(m3/s)	118.24	12
	q/TurbLim	Q limit per turbine	(m3/s)	325.00	13
	q dif/turb	non delivered q	(m3/s)	-206.76	14
	Real qby4 Turb		(m3) E04/s	0.047296333	15 ^{2, 3}
Real volume delivered	delivered V	delivered Q by all turbines realOutflowAllTurbines	(m3) E09	1.2259210	16
	ND v	Non delivered q	(m3) E09	0.0000000	17 ^{5,6}
	real v1	remained	(m3) E09	3.3512	18
	spilled water	volume spilled	(m3) E09	0.0000	19 ⁷
	total outflow	q through turbines + spilled vol	(m3) E09	1.2259	20
	real h1	only considers q through turbines	(m3)	384	21 ⁸

Fig. 8. Paso a paso del calculo de volumen que sale por las turbinas

Cálculo de la energía total generada

El cálculo de la energía generada por el volumen de agua que sale por las turbinas se explica en el diagrama 4(D), que básicamente se reduce a multiplicar el gasto que sale por las

turbinas por un factor de eficiencia y por la altura de la columna de agua a la salida de una presa. El código que realiza este cálculo se muestra en la figura 9, que está emparentado con los números de pasos con el diagrama 4(D). Numéricamente este código está explicado en las líneas 22 a 27 de la tabla 2, en la pagina 13 del reporte 1 de este proyecto.

```

14 %
15 % computePowerUpstream(400,400,178,0,1,325,4.12)
16
17 function [powerUp, totalOutUp, realh1Up]=computePowerUpstream(h0Up, h1Up, runoffUp, inputsUp, stageSize);
18
19 %1. CALCULO DEL VOLUMEN DISPONIBLE
20 avUpstr = availableVolumeUpstream(h0Up, h1Up, runoffUp, inputsUp, stageSize);
21 v1=h2vUpstream(h1Up); % convert h to volumen using charts
22 %2. CALCULO DE LA SALIDA DE VOLUMEN EN LAS TURBINAS
23 [turbinesRateOutUp, totalOutUp, realh1Up] = totalOutflowAndfinalhUpstream(stageSize, h1Up, v1);
24 %3. CALCULO DE ENERGÍA GENERADA POR LAS TURBINAS
25 tail=tailUpstream(turbinesRateOutUp);
26 headUp = ((h0Up+h1Up)/2)-tail;
27 powerUp = headUp*turbinesRateOutUp*8.5; % 8.5 -> efficiency factor
28
29
30
31

```

Fig. 9. Cálculo de la energía generada por las turbinas

Head calculation	tail	tail for q (by table)	m	190.9794092	22
	Hreal			201.0893367	23
	H (hypotetic)	Avg of h's - tail	m	201.5205908	24
Power generation	K	data	no-dim	0.85	25
	N	result	MW	81.02	26
	N (real)	result	MW	80.84	27 3

Fig. 10. Paso a paso del cálculo de energía generada por las turbinas

Discretización del sistema

Las metodologías de Jiang trabajan sobre una trayectoria inicial. Para poder calcular esta y también para aplicar sus métodos de optimización se requiere de la discretización del sistema y su esquematización. Este proceso se explica en esta sección.

Para cada una de las presas del sistema, tomando como referencia el nivel de agua almacenada, se define: 1) un punto de inicio y final para la discretización del volumen como se muestra en la figura 11. Donde se define un fin e inicio de 350-400 metros en el nivel de almacenamiento. 2) Un numero M de discretizaciones, 3) Un punto fijo en donde el periodo de planeación se espera empiece y termine (puntos 1 y 2 en la figura 11)

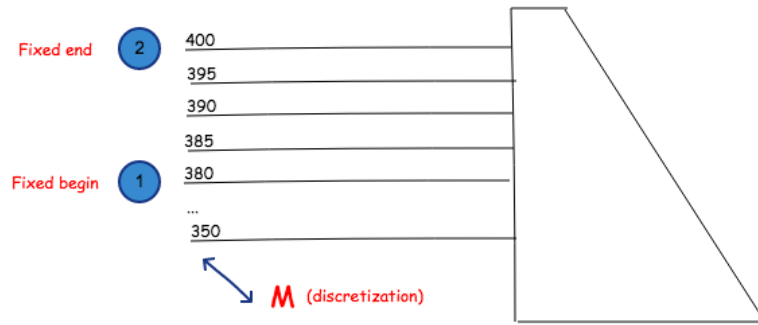


Fig. 11. Representación de la discretización de una presa de un sistema en cascada.

La discretización mostrada en la figura 1 corresponde al esquema de la figura 12, en donde el eje x define la duración de la planeación, en este caso, 12 periodos de un mes, con $\Delta t = 1$, y $T = 12$

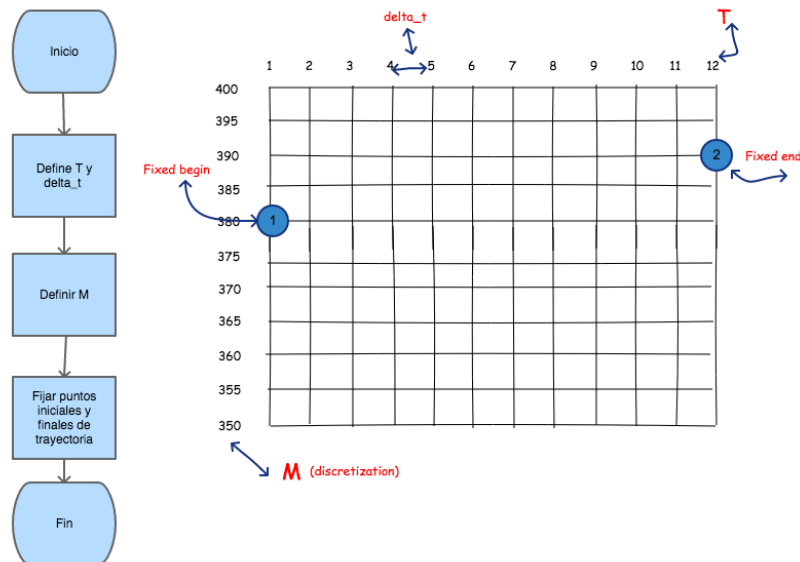


Fig. 12. Esquemmatización de la discretización representada en la figura 1.

La malla de la figura 12 se usa para trazar la trayectoria o curva guía de operación del sistema de presas en cascada. Una malla por cada presa del sistema en cuestión, por lo que cada presa se le asigna una curva guía final del proceso de cálculo.

Una vez que se tiene la discretización del sistema se aplica AMD para encontrar las trayectorias iniciales o curvas guías de operación iniciales sobre las cuales, en un paso posterior se les aplica POA e IMDP para optimizarlas y encontrar las curvas guías finales.

MDP

El diagrama de flujo de aplicación del proceso iterativo MDP se muestra en la figura 13

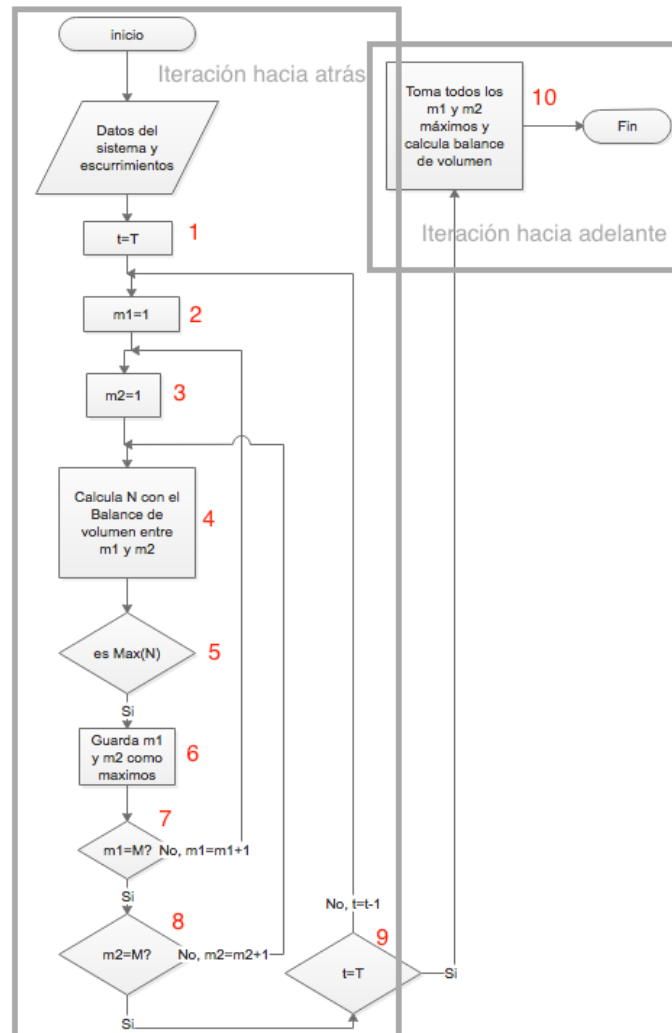


Fig. 13. Diagrama de flujo de MDP

La aplicación del diagrama de la figura 13 puede explicarse con la figura 14, que a su vez corresponde a la figura 8 del reporte 1 de este proyecto.

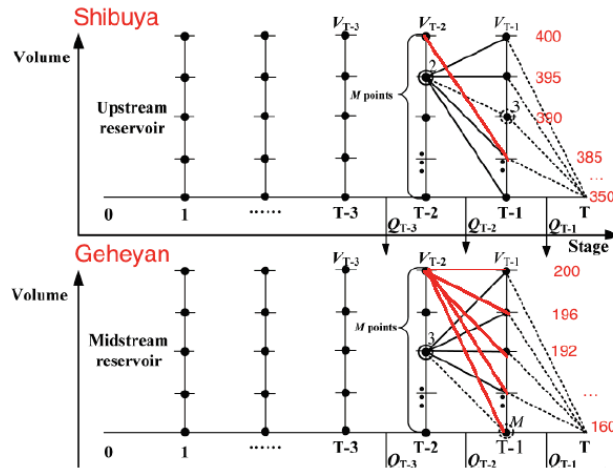


Fig. 14. Iteración y aplicación de MDP

En Breve, la relación entre las figuras 13 y 14 se puede resumir de la siguiente manera: Suponga un $m1$ (punto inicial/nivel inicial en el periodo de tiempo t) = 200 para la presa Geheyen estando en $T-2$ como el la figura. Se hace que $m2$ recorra todos los puntos, de 1 a M sobre $T-1$. Para cada combinación $m1$ - $m2$ se calcula el balance de volumen y se escoge el par $m1$, $m2$ que genere la N máxima (Fig. 13, pasos 3, 4 y 5). Cuando $m2$ llega a M , $m1$ da un paso mas (pasos 7 y 8) hasta que $m1$ también alcance M . Una vez que se recorrión toda la discretización de 1 hasta M tanto para $m1$ como para $m2$, se da un paso más en T . Así sucesivamente hasta que $t = 1$.

Al final de este proceso iterativo se tiene T pares $(m1, m2)$ que guardan los máximos encontrados en cada periodo t . El paso 10 de la figura 13 indica que estos máximos se usan para calcular la trayectoria inicial-definitiva cuando se termina MDP.

Este proceso es explicado con detalle junto con un ejemplo numérico en la sección MDP, en la página 14 del reporte 1 de este proyecto.

El código que ejecuta una iteración en un periodo de tiempo se presenta en la figura 15. Es decir, la iteración hacia delante para un periodo t donde $m1$ va de 1 hasta M , y $m2$ va de 1 hasta M , o en otras palabras, el proceso indicado en líneas rojas en figura 14.

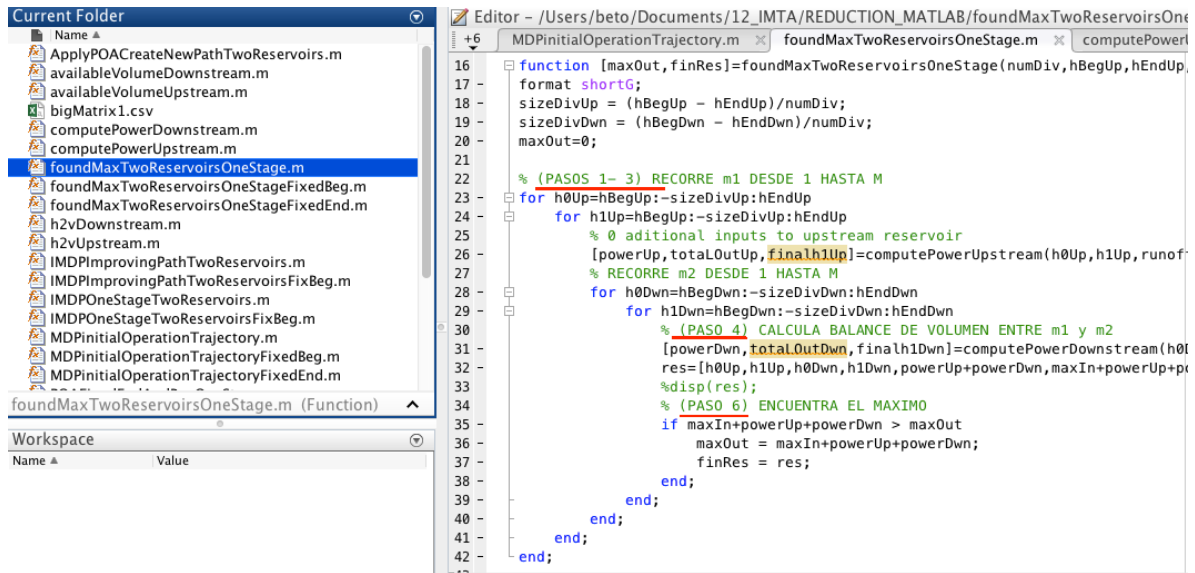


Fig. 15. Una iteración hacia delante del proceso MDP

El código del proceso completo de MDP para encontrar la trayectoria inicial se presenta en la figura 16 y está codificado en la turinta **MDPInitialOperationTrajectory**

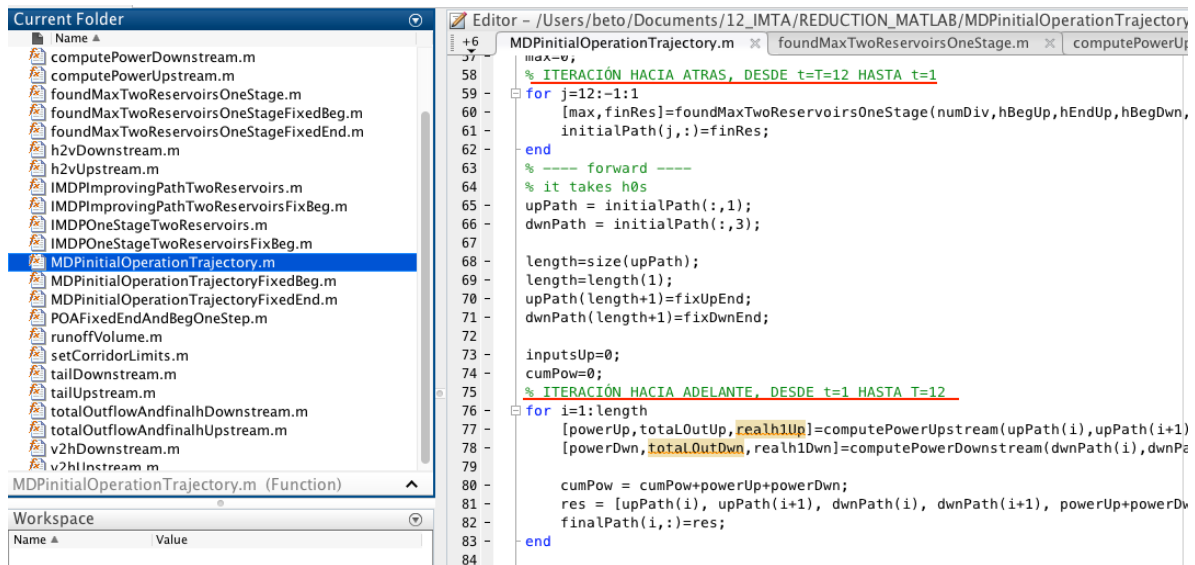


Fig. 16. Código de MDP para encontrar la trayectoria inicial

La iteración hacia atrás traza una primer trayectoria compuesta de puntos de inicio y final para cada periodo de tiempo t . Numéricamente, una trayectoria tiene la siguiente forma:

$$(m1_1^n, m2_1^n), (m1_2^n, m2_2^n), \dots, (m1_T^n, m2_T^n) \quad (\text{Eq. 1})$$

Donde

- n es el numero de presa del sistema

- $m1$ es el punto inicial de un fragmento de trayectoria de operación para un tiempo t de T
- $m2$ es el punto final de un fragmento de trayectoria de operación para un tiempo t de T
- T es el numero total de periodos de tiempo que dura la trayectoria de operación.

El resultado de aplicar MDP se muestran en la figura 17 del reporte 1 que se presenta enseguida

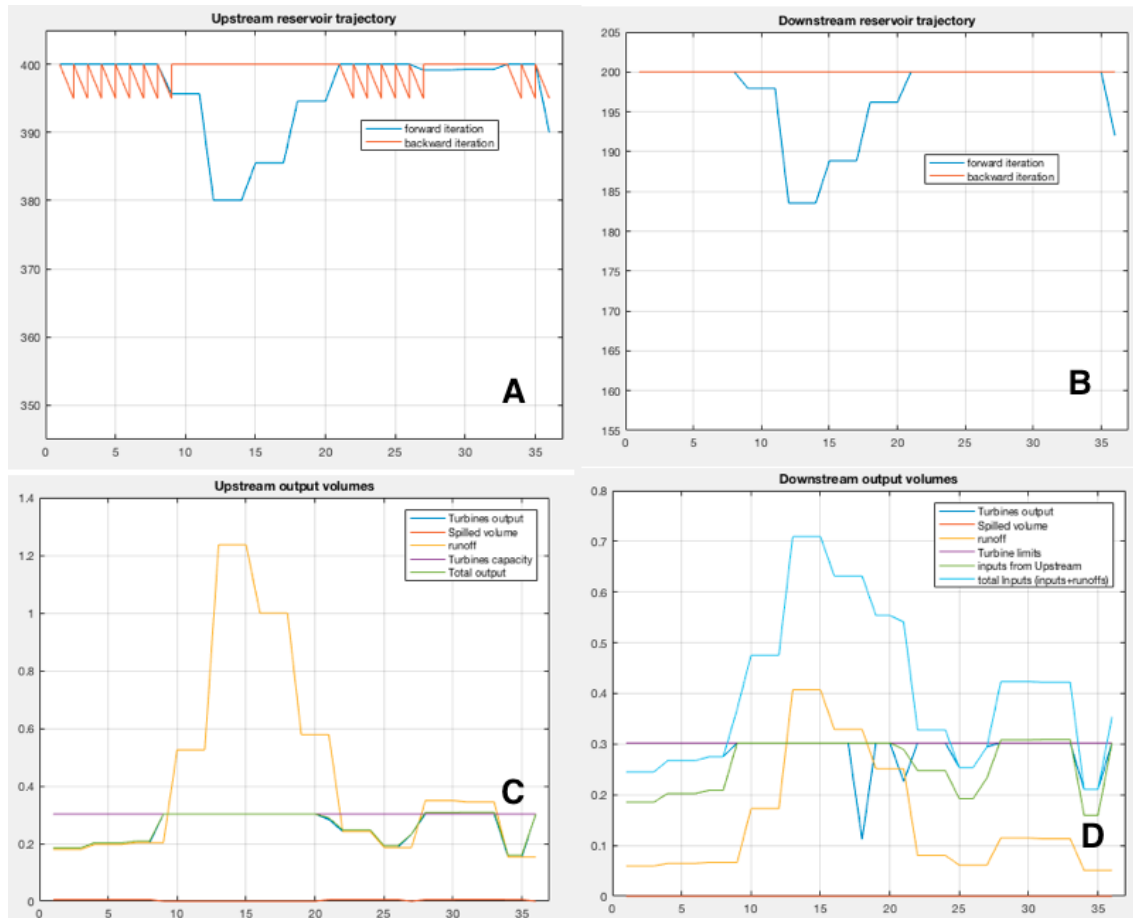


Fig. 17. Resultado de aplicar MDP. (A) Resultados de iteración hacia atrás y adelante para la presa aguas-arriba (Suibuya). (B) Resultados de iteraciones hacia atrás y adelante para la presa aguas-abajo (Geheyan). (C) Registro de los volúmenes para la presa Shuibuya. (D) Registro de los volúmenes para la presa Geheya.

Las trayectoria presentada en la figura 17, en su forma numérica, tiene la forma de Eq.1. Por ejemplo, para la presa aguas-arriba, para $t=1$ hasta $t=8$, la Eq.1 toma la forma: $(400, 400) \times 8$. Esta fracción de trayectoria corresponde a la línea naranja de la grafica izquierda de la figura 17.

La línea azul corresponde a la unión de los $m1s$ o puntos iniciales de la trayectoria naranja. Es sobre esta línea azul que se aplica la iteración hacia delante de acuerdo con Jiang.

La figura 18 muestra como ejecutar el programa en MATLAB para obtener una trayectoria inicial

Command Window												
<pre>>> dwnsDry=[69,75,77,200,471,381,291,93,71,133,131,59]; upsDry=[208,228,235,608,1431,1157,669,280,216,404,398,178]; [ini, fin]=MDPInitialOperationTrajectoryII(upsDry,dwnsDry,10,400,350,200,160,390,192,1/3,87.5,87.5,4.11,3.4)</pre>												
Upstream		Downstream		Totals			Upstream		Downstream		Totals	
h0	h1	h0	h1	Power	Cum Power	finalH	turbOut Up	Spilled Up	turbOu tDwn	spilled Dwn	totalOut Up	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	
400.0	400.0	200.0	200.0	0.678	0.678	200.1	0.1797	0.0	0.245	0	0.185	
400.0	400.0	200.0	200.0	0.678	1.356	200.1	0.1797	0.0	0.245	0	0.185	
400.0	400.0	200.0	200.0	0.678	2.034	200.1	0.1797	0.0	0.245	0	0.185	
400.0	400.0	200.0	200.0	0.741	2.776	200.1	0.1970	0.0	0.268	0	0.203	
400.0	400.0	200.0	200.0	0.741	3.517	200.1	0.1970	0.0	0.268	0	0.203	
400.0	400.0	200.0	200.0	0.741	4.259	200.1	0.1970	0.0	0.268	0	0.203	
400.0	400.0	200.0	200.0	0.764	5.022	200.1	0.2030	0.0	0.275	0	0.209	
400.0	400.0	200.0	200.0	0.764	5.786	200.1	0.2030	0.0	0.275	0	0.209	
400.0	395.7	200.0	197.9	0.994	8.540	200.9	0.3024	0.0	0.302	0	0.302	
395.7	395.7	197.9	197.9	0.984	11.462	200.1	0.3024	0.0	0.302	0	0.302	
395.7	395.7	197.9	197.9	0.984	14.384	200.1	0.3024	0.0	0.302	0	0.302	
395.7	380.1	197.9	183.6	0.939	17.238	200.1	0.3024	0.0	0.302	0	0.302	
380.1	380.1	183.6	183.6	0.894	19.208	189.7	0.3024	0.0	0.302	0	0.302	
380.1	380.1	183.6	183.6	0.894	21.178	189.7	0.3024	0.0	0.302	0	0.302	
380.1	385.5	183.6	188.8	0.910	23.172	189.7	0.3024	0.0	0.302	0	0.302	
385.5	385.5	188.8	188.8	0.926	25.214	193.3	0.3024	0.0	0.302	0	0.302	
385.5	385.5	188.8	188.8	0.926	27.255	193.3	0.3024	0.0	0.302	0	0.302	
385.5	394.6	188.8	196.2	0.733	29.117	196.0	0.3024	0.0	0.112	0	0.302	
394.6	394.6	196.2	196.2	0.976	31.859	199.3	0.3024	0.0	0.302	0	0.302	
394.6	394.6	196.2	196.2	0.976	34.601	199.3	0.3024	0.0	0.302	0	0.302	
394.6	400.0	196.2	200.0	0.860	35.461	200.1	0.2836	0.0	0.226	0	0.289	
400.0	400.0	200.0	200.0	0.878	36.339	200.4	0.2419	0.0	0.302	0	0.248	
400.0	400.0	200.0	200.0	0.878	37.216	200.4	0.2419	0.0	0.302	0	0.248	
400.0	400.0	200.0	200.0	0.878	38.094	200.4	0.2419	0.0	0.302	0	0.248	
400.0	400.0	200.0	200.0	0.703	38.797	200.1	0.1866	0.0	0.254	0	0.192	
400.0	400.0	200.0	200.0	0.703	39.500	200.1	0.1866	0.0	0.254	0	0.192	
400.0	399.2	200.0	200.0	0.849	41.052	200.1	0.2333	0.0	0.295	0	0.233	
399.2	399.2	200.0	200.0	1.001	43.056	201.6	0.3024	0.0	0.302	0	0.308	
399.2	399.2	200.0	200.0	1.001	45.059	201.6	0.3024	0.0	0.302	0	0.308	
399.2	399.3	200.0	200.0	1.001	47.063	201.6	0.3024	0.0	0.302	0	0.308	
399.3	399.3	200.0	200.0	1.001	49.067	201.6	0.3024	0.0	0.302	0	0.309	
399.3	399.3	200.0	200.0	1.001	51.071	201.6	0.3024	0.0	0.302	0	0.309	
399.3	400.0	200.0	200.0	1.003	52.073	202.1	0.3024	0.0	0.302	0	0.309	
400.0	400.0	200.0	200.0	0.582	52.655	200.1	0.1538	0.0	0.211	0	0.160	
400.0	400.0	200.0	200.0	0.582	53.237	200.1	0.1538	0.0	0.211	0	0.160	
400.0	390.0	200.0	192.0	0.976	54.213	200.7	0.3024	0.0	0.302	0	0.302	

Fig. 18. Obtención de la trayectoria inicial mediante el código en MATLAB. **Línea de comandos:** código para ejecutar MDP para obtener trayectoria inicial para datos de escurrimientos de un año seco según Jiang. 10 divisiones de discretización, con límites 400-350 para Shuibuya y 200-160 para Geheyan, con puntos fijos para final de trayectoria de 390m para Shuibuya y 192m para Geheyan, para un periodos de tiempo de (1/3) de mes. Con 87.5m³/s como capacidades límite de las turbinas de las presas cuyas capacidades son de 4.11 y 3.4 m³ E09. A continuación la tabla que describe las trayectorias mostradas en la figura 17: columnas **h0 y h1** puntos inicial y final para la trayectoria para un periodo de tiempo t. **Power:** Total de energía generada en cada periodo de tiempo. **CumPower:** Energía generada acumulada (suma de columna Power). **finalH:** nivel del agua en Geheyan and final de un periodo (aproximado). **turbOut:** volumen que sale por las turbinas para Shuibuya(Up) y Geheyan(Dwn). **Spilled:** Volumen vertido. **TotalOut:** Salida aproximada de volumen total de Shuibuya (volumen que sale de las turbinas + volumen vertido).

POA

POA es el método sugerido por Jiang para la optimización de la trayectoria inicial obtenida vía MDP. En resumen POA consta de un proceso iterativo de adelante hacia atrás en donde en cada paso se toman 3 puntos en la trayectoria correspondientes a $t-1$, t y $t+1$, siendo el punto en t el punto a optimizar y los otros quedan fijos. El punto t se hace variar de 1 a M calculando el balance de volumen para cada triada de puntos hasta encontrar la combinación de puntos que maximice de energía generada.

El proceso de POA está explicado en las figuras 9 a 12 del reporte 1 de este proyecto. La figura 19 muestra el diagrama de flujo para POA.

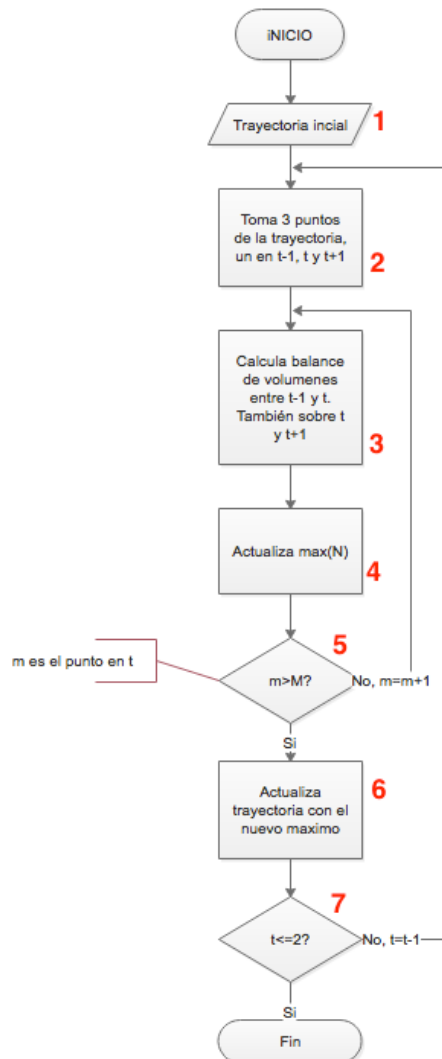


Fig. 19. Diagrama de flujo de POA

Este proceso se ejemplifica en la imagen 20

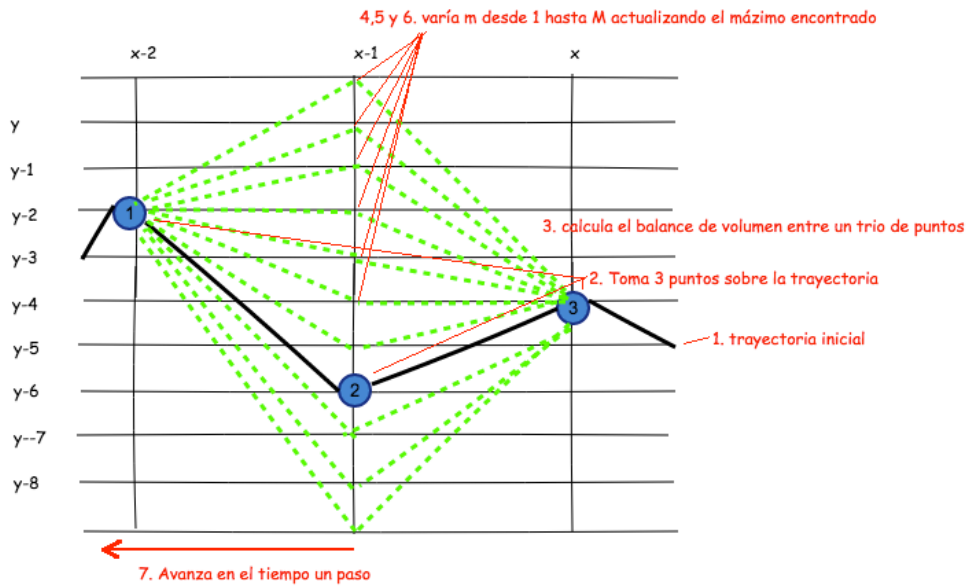


Fig. 20. Gráfico del proceso POA, donde los incisos corresponde a la numeración en el diagrama de flujo de la figura 19. 1) La línea negra es la trayectoria inicial obtenida via MDP. 2) Se toman 3 puntos de la trayectoria, en donde 1 y 3 ($t-1$ y $t+1$) quedan fijos, el punto 2 se varia desde 1 hasta M o el número de divisiones establecidas. 3) Para cada combinación de tres puntos obtenida cada vez que se varia el punto 2 se calcula el balance de volumen para obtener la energía generada por eta combinación de puntos. 4,5 y 6) al variar el punto 2 se actualiza el máximo encontrado. 7) Si el punto 2 se varió en todas sus posibilidades, se avanza en el tiempo hacia atrás, quedando el punto 2 con su nuevo valor máximo encontrado y establecido como punto fijo. El punto 1 ahora se varía y el toma un punto más en la trayectoria inicial.

El código que ejecuta el proceso POA entre tres puntos de la trayectoria original, es decir, el proceso mostrado en la figura 20, se muestra en la figura 21. Este proceso corresponde a los pasos 2 a 6 del diagrama de flujo de la figura 19 para una sola triada de puntos. La variación iterativa del tiempo hacia atrás y la actualización de la trayectoria se hace en el código mostrado en la figura 22.

Debe notarse que el código de la figura 22 usa iterativamente el código de la figura 21, en otras palabras, la rutina **ApplyPOACreateNewPathTwoReservoirs** usa iterativamente la rutina **POAFixedEndAndBegOneStep** en cada paso para actualizar la trayectoria inicial. El código de la figura 22 correponde a los pasos 1 y 7 del diagrama de flujo de la figura 19, donde se hace variar en el tiempo la triada de puntos a optimizar de la trayectoria original.

```

123 Editor - /Users/beto/Documents/12_IMTA/REDUCTION_MATLAB/POAFixedEndAndBegOneStep.m
124 IMDPOneStageTwoReservoirs.m x IMDPOneStageTwoReservoirsFixBeg.m x POAFixedEndAndBegOne
125
126 % 1. CALCULA DE ADELANTE HACIA ATRAS
127 for hUp=hBegRangeUp:-sizeDivUp:hEndRangeUp
128     h0Up=begUpFixed;
129     h1Up=endUpFixed;
130
131     % 0 additional inputs to upstream reservoir
132     % computes power from fixed beginning to h
133     % and from h to fixed end
134     % 2. BALANCE DE VOLUMEN ENTRE t-1, t (AGUAS ARRIBA)
135     [powerUpFixBeg,totalOutUpFixBeg,finalh1UpFixBeg]=computePowerUpstream(h0Up,hUp,
136     % 2.1 BALANCE DE COLUMEN ENTRE t y t+1 (AGUAS ARRIBA)
137     [powerUpFixEnd,totalOutUpFixEnd,finalh1UpFixEnd]=computePowerUpstream(hUp,h1Up,
138
139     % last data stay fixed to review all possible combinations in
140     % downstream
141     for hDwn=hBegRangeDwn:-sizeDivDwn:hEndRangeDwn
142         h0Dwn=begDwnFixed;
143         h1Dwn=endDwnFixed;
144
145         % computes power from fixed beginning to h
146         % and from h to fixed end.
147         % 3. BALANCE DE VOLUMEN ENTRE t-1, t y t+1 (AGUAS ABAJO)
148         [powerDwnFixBeg,totalOutDwnFixBeg,finalh1DwnFixBeg]=computePowerDownstream
149         [powerDwnFixEnd,totalOutDwnFixEnd,finalh1DwnFixEnd]=computePowerDownstream
150
151         % sum power in the beginning
152         powerSumFixEnd = powerUpFixEnd+powerDwnFixEnd;
153         % sum power in end
154         powerSumFixBeg = powerUpFixBeg+powerDwnFixBeg;
155         % total power in both stages
156         %currentPowerSum=powerSumFixEnd+powerSumFixBeg;
157         % maxIn corresponds only for the end stage, it is not for both
158         % stages here considered since in the next step, beginning stage
159         % turns on end stage for next computation
160         % 4. GUARDA EL MAXIMO ACTUAL ENCONTRADO
161         totalPowerSum=maxIn + powerSumFixBeg + powerSumFixEnd;
162
163         %resArray=[powerUpFixBeg,powerDwnFixBeg,powerUpFixEnd,powerDwnFixEnd,powerS
164         %disp(resArray);
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fig. 21. Código de un paso de POA

```

123 Editor - /Users/beto/Documents/12_IMTA/REDUCTION_MATLAB/ApplyPOACreateNewPathTwo
124 ApplyPOACreateNewPathTwoReservoirs.m x v2hUpstream.m x MDPInitialOperationTrajecto
125
126 % backward
127 for i=size(pathUp):-1:3
128     % 1. TOMA TRES PUNTOS EN LA TRAYECTORIA ORIGINAL
129     h1Up=pathUp(i);
130     h0Up=pathUp(i-2);
131     runoffUpBeg=runoffUpArray(i);
132     runoffUpEnd=runoffUpArray(i-2);
133     h1Dwn=pathDwn(i);
134     h0Dwn=pathDwn(i-2);
135     runoffDwnBeg=runoffDwnArray(i);
136     runoffDwnEnd=runoffDwnArray(i-2);
137     % fix begin and end, up and dwn and tries every options in the middle
138     % to find max
139     % finRes format: [hUp,h1Up,hDwn,h1Dwn,powerSumFixEnd,finalh1DwnFixEnd,h
140     % 2. CALCULA EL MAXIMO EN LA TRIADA DE PUNTOS ACTUAL
141     [maxOut,finRes]=POAFixedEndAndBegOneStep(numDiv,hBegRangeUp,hEndRangeUp,
142     pathUp(i-1)=finRes(1);
143     pathDwn(i-1)=finRes(3);
144     newPathLinei = [finRes(1),finRes(2),finRes(3),finRes(4),finRes(5)];
145     newPathLinei_1=[finRes(7),finRes(8),finRes(9),finRes(10),finRes(11)];
146     %newRes=[finRes(2),finRes(3),finRes(5),finRes(6),maxOut,maxIn+maxOut];
147     % 3. ACTUALIZA LA TRAYECTORIA
148     newPath(i,1:5)=newPathLinei;
149     newPath(i-1,1:5)=newPathLinei_1;
150     maxIn=maxOut;
151 end
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fig. 22. Iteración de POA sobre toda la trayectoria

Los resultados de aplicar POA se muestran en la figura 18 del reporte 1 de este proyecto. En este reporte se muestran en la figura 23, en donde la línea azul es la trayectoria inicial obtenida mediante la aplicación de MDP (compare la trayectoria inicial mostrada en la figura 17 de este reporte) y la línea naranja es la trayectoria optimizada tras una primer aplicación de POA.

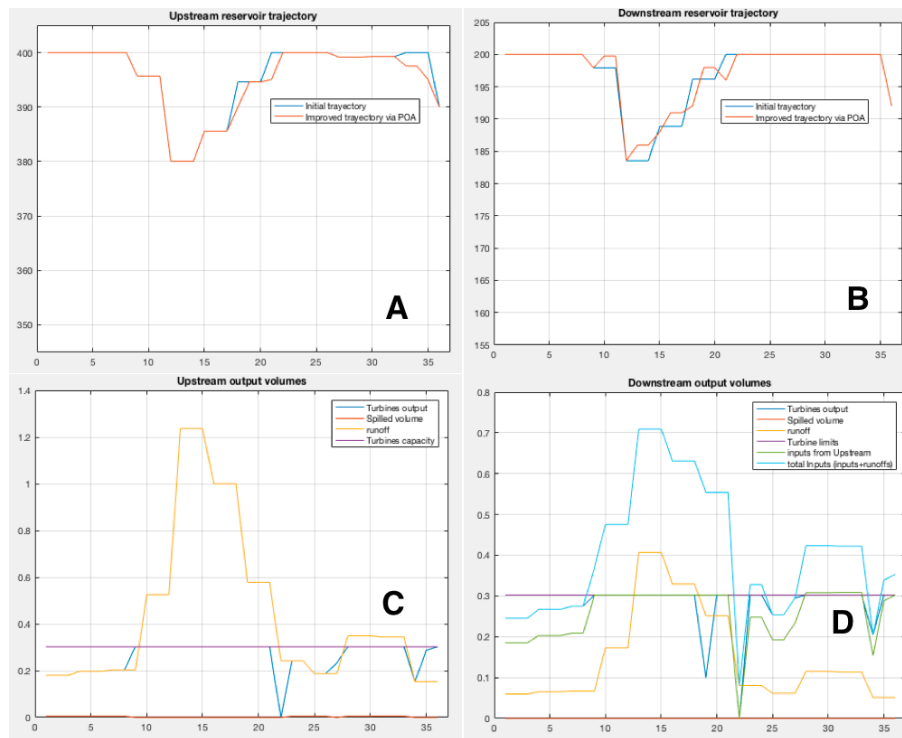


Fig. 23. Resultado de aplicar POA (línea naranja) a una trayectoria inicial (línea azul)

Como se puede notar en la figura 23(A), la línea azul corresponde a la trayectoria inicial calculada vía MDP mostrada en la figura 17(A). La línea naranja corresponde a la trayectoria optimizada mediante POA. La figura 23(A) muestra el comportamiento de los volúmenes de agua para la trayectoria mejorada. Notese que la línea azul que corresponde al volumen que sale por las turbinas para la presa Shuibuya, el cual no rebasa el límite de las turbinas (línea púrpura). La línea roja describe el volumen vertido y la línea amarilla el volumen que entra vía escurrimientos a Shuibuya. La misma descripción es válida para 27(B) y (D) que corresponden a la presa Geheyan con una excepción: la línea verde describe las entradas a Geheyan que vienen de Shuibuya (las salidas de Shuibuya se convierten en entradas para Geheyan) y la línea azul cielo que corresponde a la suma de entradas a Geheyan, las que provienen de Shuibuya y los escurrimientos.

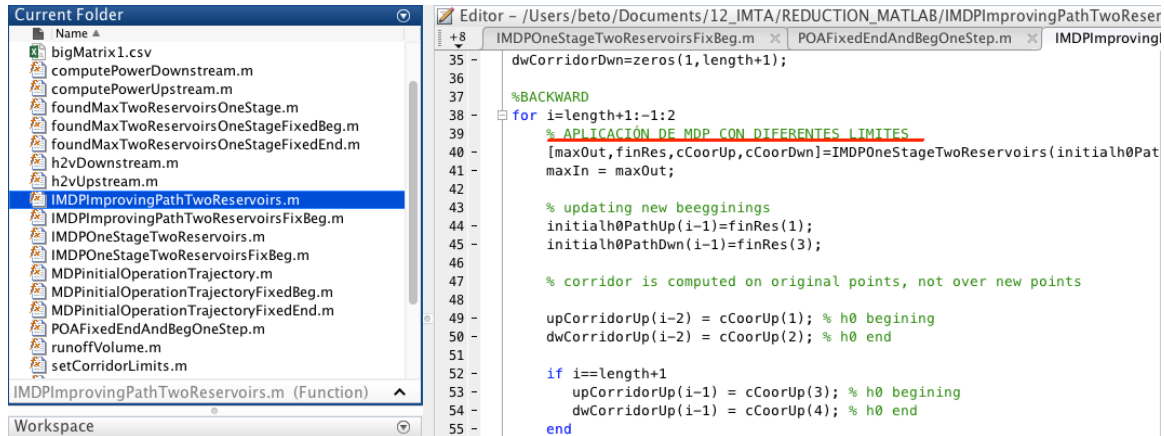
IMDP

IMDP, de acuerdo con Jiang, es un MDP mejorado (improved). La mejora de Jiang a MDP son 2 que se pueden explicar, con la figura 13 y 14 de este reporte:

- 1) IMDP reduce la búsqueda de solución a un “corredor”, esto es, en la figura 13, en vez de variar $m1$ y $m2$ desde 1 hasta M (pasos 7 y 8), lo hace solo desde cierta distancia en la vecindad de la trayectoria inicial. Jiang sugiere entre 1 y 4 puntos de discretización alrededor de la trayectoria inicial.

- 2) El corredor de búsqueda es subdividido de manera mucho mas fina (hasta 100 puntos) para hacer la búsqueda de la solución óptima, de manera que, si en MDP se usa una discretización –digamos- de 10 puntos para el tiempo t , IMDP puede usar hasta 40 puntos solo en el corredor determinado.

El código que aplica IMDP a una trayectoria inicial se muestra en la figura 24 y el resultado de aplicarlo se muestra en la figura 25.



```

35 - dwCorridorDwn=zeros(1,length+1);
36
37 %BACKWARD
38 - for i=length+1:-1:2
39     % APLICACIÓN DE MDP CON DIFERENTES LÍMITES
40     [maxOut, finRes, cCoorUp, cCoorDwn]=IMDPOneStageTwoReservoirs(initialh0Pat
41     maxIn = maxOut;
42
43     % updating new beegginings
44     initialh0PathUp(i-1)=finRes(1);
45     initialh0PathDwn(i-1)=finRes(3);
46
47     % corridor is computed on original points, not over new points
48
49     upCorridorUp(i-2) = cCoorUp(1); % h0 beginning
50     dwCorridorUp(i-2) = cCoorUp(2); % h0 end
51
52     if i==length+1
53         upCorridorUp(i-1) = cCoorUp(3); % h0 beginning
54         dwCorridorUp(i-1) = cCoorUp(4); % h0 end
55     end

```

Fig. 24. Código para IMDP

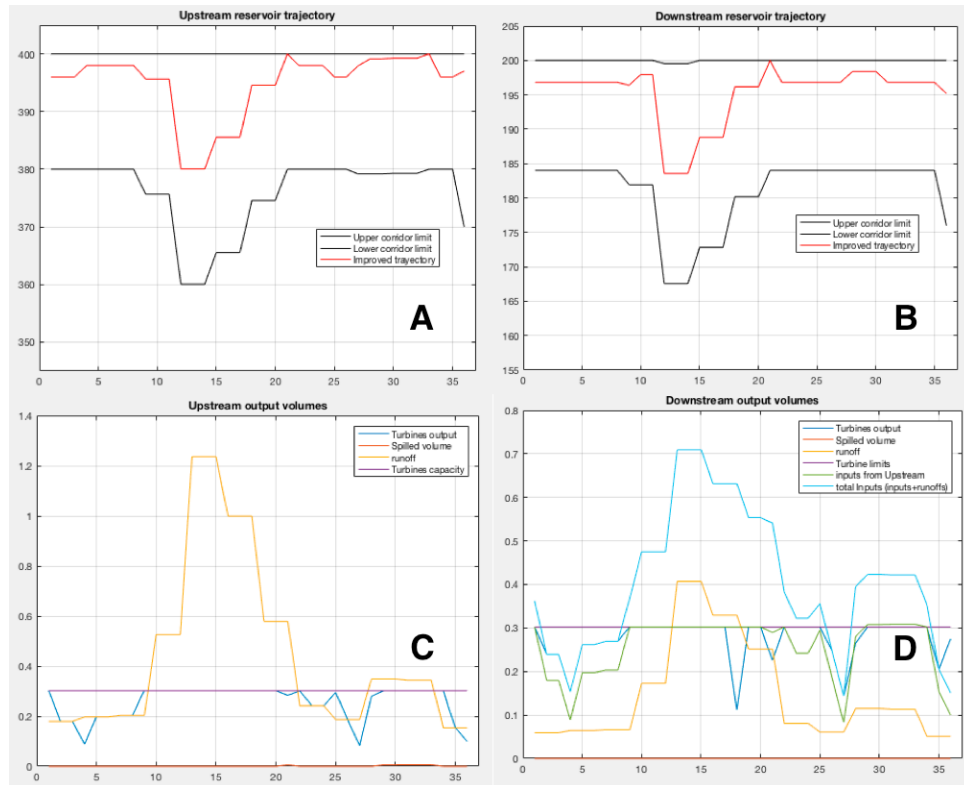


Fig. 25. Resultados de aplicar POA a la trayectoria inicial mostrada en la figura 17. (A) Trayectoria maximizada para la presa Shuibuya usando IMDP (línea roja) y el corredor de cálculo (líneas en negro). (B) Comportamiento de los volúmenes para la presa Shuibuya. (C) Trayectoria maximizada para la presa Geheyan. (D) Comportamiento de los volúmenes para la presa Geheyan.

Comentarios y conclusiones. (La equivalencia entre el modelo de Jiang y su implementación en MATLAB).

En esta sección se discute la exactitud del estudio e implementación del modelo de Jiang con relación a los resultados obtenidos por el autor en el artículo base de este proyecto.

En la comparación aquí presentada se deben de tomar en cuenta varios factores que inciden sobre la exactitud de los resultados en comparación con los resultados obtenidos por el autor del artículo base, a saber:

1. **Los datos disponibles del sistema de presas en cascada es limitado.** Particularmente no se cuentan con las curvas nivel-almacenamiento de la presa Shuibuya. Tampoco se cuenta con la curva gasto-altura. Para subsanar este dato se hizo una generalización y escalamiento (adaptación) de las curvas de la presa Geheyan para ser usadas en la presa Shuibuya. Para ello se requirió de fuertes supuestos, como el de que la presa Shuibuya es un modelo escalada de la presa Geheyan.
2. **No se cuentan con datos exactos de las capacidades de generación de energía del sistema de presas en cascada,** por lo que también se supuso que ambas presas tienen el mismo numero de compuertas, el mismo numero de turbinas con las mismas capacidades de generación de energía.
3. Como el autor señala, **se desprecian las perdidas en volumen** en el recorrido de aguas arriba a aguas abajo.
4. En el artículo base **no existe detalle sobre las restricciones usadas en el cálculo de balance de volumen,** que desde un punto de vista técnico, son restricciones importantes que afectan de manera decisiva la forma de las curvas guías de operación, tales como el gasto limite permitido para ser desalojado por las turbinas. En este proyecto se trabajó sobre el supuesto de las turbinas pueden trabajar a su máxima capacidad todo el periodo de planeación.

Sobre la exactitud en el cálculo de la trayectoria inicial

En el artículo base no se reportan resultados o ejemplos de trayectorias iniciales mediante el uso de MDP. Tampoco se explica el método MDP y su adaptación al problema CROO. Más bien referencia otros artículos donde, aunque se explica el método y se presentan diagramas de flujo del mismo, no se presentan ejemplos del cálculo de estas trayectorias.

El método MDP implementado en este proyecto sigue rigurosamente la información encontrada en los artículos referenciados.

A manera de conclusión parcial, se estima que existe la “ventaja” de que como dice el autor, la trayectoria inicial no es del todo relevante para calcular la curva de operación optimizada final, sino mas bien, es solo un punto inicial del cálculo de la trayectoria final.

Aunque bien es cierto que la aseveración anterior es correcta para el caso de la aplicación de POA como método de optimización de la trayectoria inicial, no lo es del todo para la aplicación de IMDP. La razón es que POA deja puntos fijos para reducir el espacio de búsqueda del óptimo, y su espacio de búsqueda abarca la discretización completa del punto a optimizar, es decir, sigue barriendo el total de las opciones optimizando al reducir las opciones por su uso de puntos fijos. El caso de IMDP es diferente, pues la búsqueda del óptimo está reducido a un corredor de búsqueda. Entonces, si a la trayectoria inicial no se le asignan un corredor de búsqueda amplio, entonces la forma de la curva inicial incide sobre la exactitud de la curva de operación final. Más aún, si la trayectoria inicial no tuviera una forma original semejante a la óptima, con un corredor no muy amplio, la solución óptima puede estar fuera de su alcance.

Para lidiar con las lista de 4 limitaciones presentadas al inicio de esta sección se adoptaron dos estrategias: 1) el escalamiento y adaptación de datos y especificar corredores de búsqueda amplios para encontrar asegurar lo más posible el barrido del espacio de soluciones. 2) Una adaptación a los métodos de optimización que busca maximizar el uso de posibles volúmenes vertidos. Esta última adaptación viene de un estudio detallado del comportamiento de los volúmenes de una trayectoria, de donde se deduce que el autor lo hace pero nunca menciona nada al respecto. Esta adaptación se explica en las secciones seguidas.

Escalamiento y adaptación

Esta estrategia consistió en reproducir una trayectoria reportada en el artículo base y usar los puntos que definen dicha trayectoria para hacer el cálculo por la implementación hecha en este proyecto.

La intención de hacer esta comparación es ver el desfase que pudiese existir entre los datos reportados por el artículo base y los calculados por la implementación aquí hecha. La hipótesis es que el desfase encontrado sería serio debido a las fuertes limitaciones de acceso a datos exactos.

Primero se calcularon los puntos que definen las trayectorias mostradas en la figura 6 del artículo base para el año seco, las cuales se muestran aquí como la figura 26

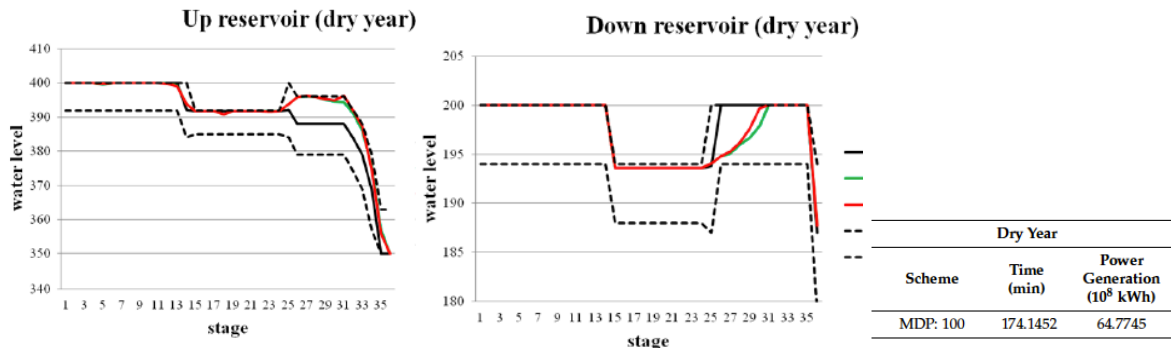


Fig. 26. Trayectorias tomadas del artículo base (fig. 6) para año seco, y datos de la tabla 7 cuya generación de energía final para estas trayectorias es de más o menos 64.77 kWh E08

Las trayectorias que se tomaron como referencia fueron las trayectorias en líneas rojas de la figura 26. Al introducir dichas trayectorias a la implementación en MATLAB, se obtuvieron los siguientes resultados mostrados en la figura 26.

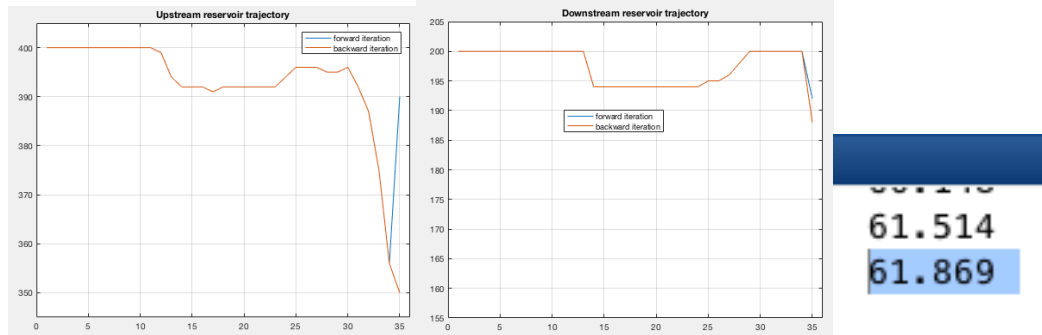


Fig. 27. Cálculo de la energía generada por la trayectoria mostrada en la figura 25.

Como se puede apreciar en la figura 27, después del ajuste a la implementación en MATLAB la discrepancia entre resultados es de $64.77 - 61.87 = 2.9$.

Para calcular estos resultados que buscan aproximar los obtenidos por el autor se hicieron 2 adaptaciones: 1) dividir el año de programación de operación en 36 lapsos de tiempo en vez de 12 (tal y como se hace en el artículo base), y 2) cambiar la eficiencia con la que se calcula la generación de energía a 1 en vez de 85% como se muestra en la figura 28. (ver cambio en subrutina **computePowerUpstream** y **computePowerDownstream**).

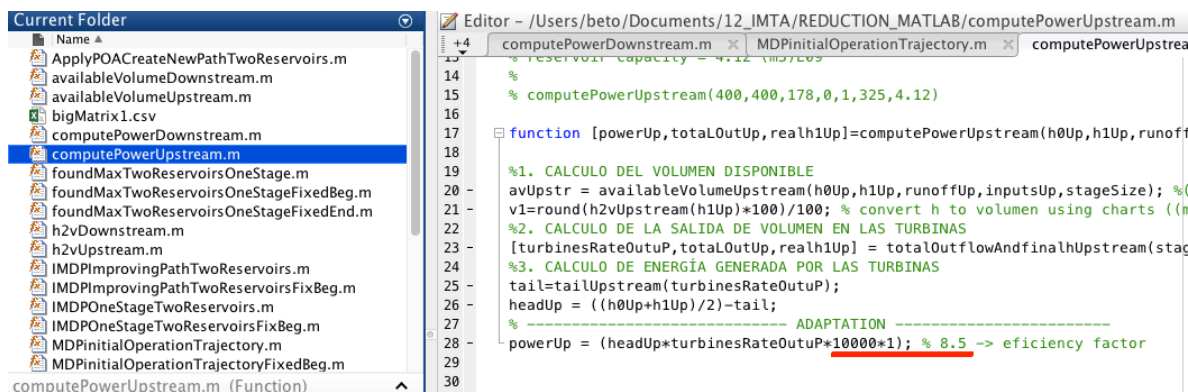


Fig. 28. Cambio de factor de eficiencia como adaptación a la implementación MATLAB.

Adaptación: Aprovechamiento del posible volumen vertido

Básicamente, esta adaptación consta en evitar que haya volumen vertido en ninguna de las presas. Aunque Jiang no hace mención de esta estrategia, al estudiar el comportamiento de los volúmenes y trayectorias presentadas por el autor del artículo base, parece que así se hace. Además, desde el punto de vista técnico, parece lógico aprovechar un volumen para crear energía en vez de dejarlo salir de la presa sin crear ningún beneficio en términos de generación de energía.

En pocas palabras esta adaptación consta en que, en el proceso de optimización de una trayectoria se encuentra que existe volumen vertido, entonces se explora en periodos de tiempo anteriores si por las condiciones existentes puede abrirse espacio en la presa dejando salir volumen que genere energía y que sea suficiente para contener el posible volumen vertido en momentos posteriores.

Este método puede explicarse con la figura 29 en donde se presenta un ejemplo de optimización usando IMDP.

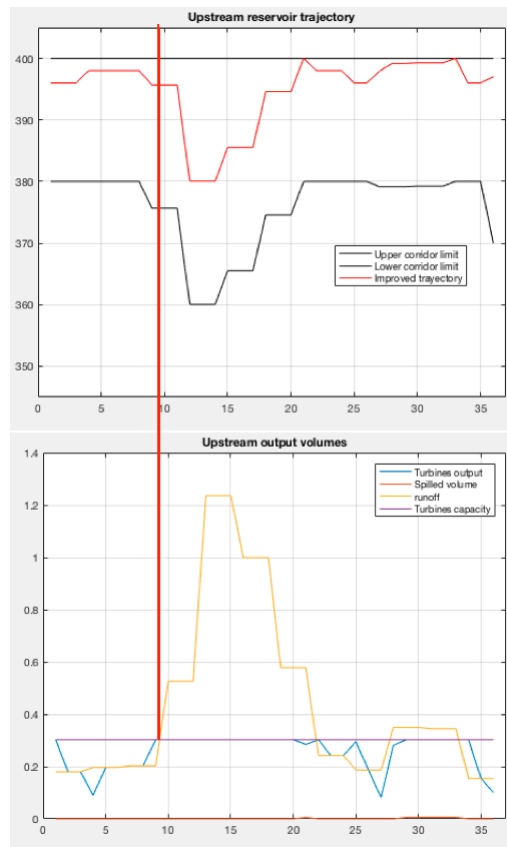


Fig. 29. Diagrama de aprovechamiento de volúmenes vertidos

Como se puede notar en la figura 29, arriba se presenta en color rojo la optimización de una trayectoria inicial (que no se grafica) usando IMDP, rodeada de un corredor de cálculo. Abajo, el comportamiento de los volúmenes de acuerdo a la trayectoria final (la línea roja). Nótese que de acuerdo al comportamiento de volúmenes, la línea amarilla indica que casi llegado el periodo de tiempo 10 (indicado con la línea roja que atraviesa ambos gráficos de la figura 29) las entradas en forma de escurrimientos rebasa las capacidades de las turbinas (línea purpura). En este punto, debe haber escurrimientos si el nivel del agua alcanza el nivel máximo y las turbinas no pueden desalojar lo suficiente. Como se puede ver en el gráfico de la trayectoria de operación (grafico superior, línea roja), la tendencia antes y después del punto donde empieza a haber posibilidades de volumen vertido, es hacia abajo, esperando abrir espacio anticipadamente para contener el volumen que entra y pudiera ser vertido.

La línea que indica el volumen vertido (línea en $y=0$ del gráfico inferior de la figura 29) indica que no hay volumen vertido, y la línea azul indica que el volumen desalojado por las turbinas y que genera energía nunca rebasa las capacidades limite de las turbinas.

Lo anterior muestra que la implementación del aprovechamiento del posible volumen vertido es eficiente y útil. Por supuesto, esto habrá que compararlo con los resultados reportados por Jiang. Esto se hace en la siguiente sección.

Comparación directa de los resultados de Jiang y la implementación de este proyecto y conclusiones

En la figura 30, 31 Y 32 se ponen frente a frente los resultados de Jiang y los obtenidos en este proyecto para los datos del año seco, normal y húmedo respectivamente.

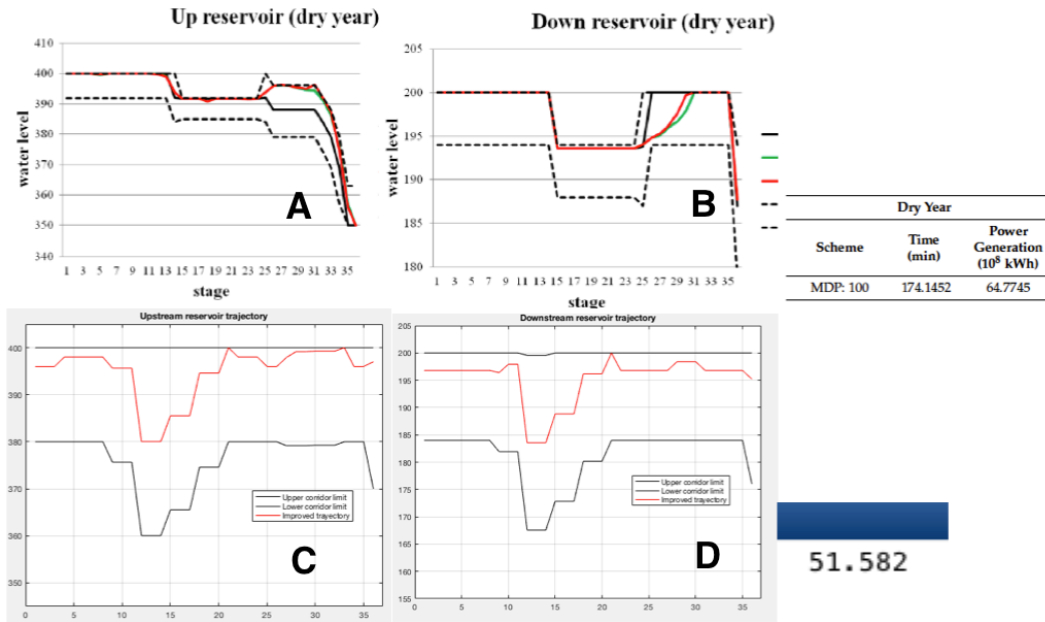


Fig. 30. Comparación de resultados para año seco. (A) Jiang: Shuibuya, año seco. (B) Jiang: Geheyan, año seco. (C) MATLAB: Shuibuya, año seco. (D) MATLAB: Geheyan, año seco

Como se puede notar en las figuras 30 a 32 las trayectorias reportadas por Jiang y las calculadas por la implementación en MATLAB de este proyecto tienen semejanzas y diferencias las cuales se explican en seguida:

- Para año seco, las trayectorias (líneas en rojo) son semejantes en la primera mitad de la curva de operación y no así para la segunda mitad. Esto se debe tal vez a que no se cuentan con todas las restricciones que se le deben imputar a la planeación. De la forma de las trayectorias de las figuras 30 (A y B) puede deducirse que pudiera existir la restricción de que al final del periodo de planeación se espera que se vacíe la presa. Esa restricción no se implementa en MATLAB dado que se trata de una mera especulación y no se sabe si en la vida real exista tal restricción. Esta ultima hipótesis es la razón de la diferencia entre la cantidad de energía generada al final del periodo de planeación (64.77 para Jiang, y 55.58 para nuestro planteamiento). En cualquiera de los casos se puede ver una tendencia semejante en ambos casos, tanto para Jiang como para nosotros se ve una tendencia hacia abajo y luego hacia arriba cuando empieza y termina el periodo de lluvias (primera mitad del año). La razón por la cual se le atribuye la diferencia en la forma de las trayectorias a restricciones que no se conocen se explican con la experimentación presentada en seguida que corresponden al año normal y húmedo.

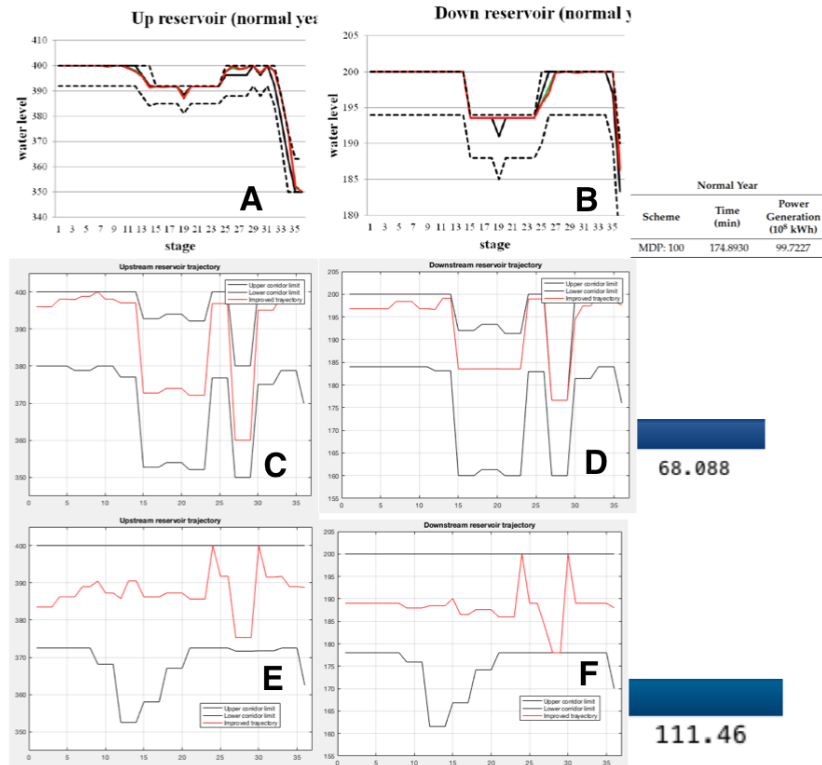


Fig. 31. Comparación de resultados para año normal. (A y B) Jiang. (C y D) MATLAB con una salida límite de turbinas restringida a 87.5m³/s. (E y F) MATLAB con salida límite de turbinas restringido 250 m³/s

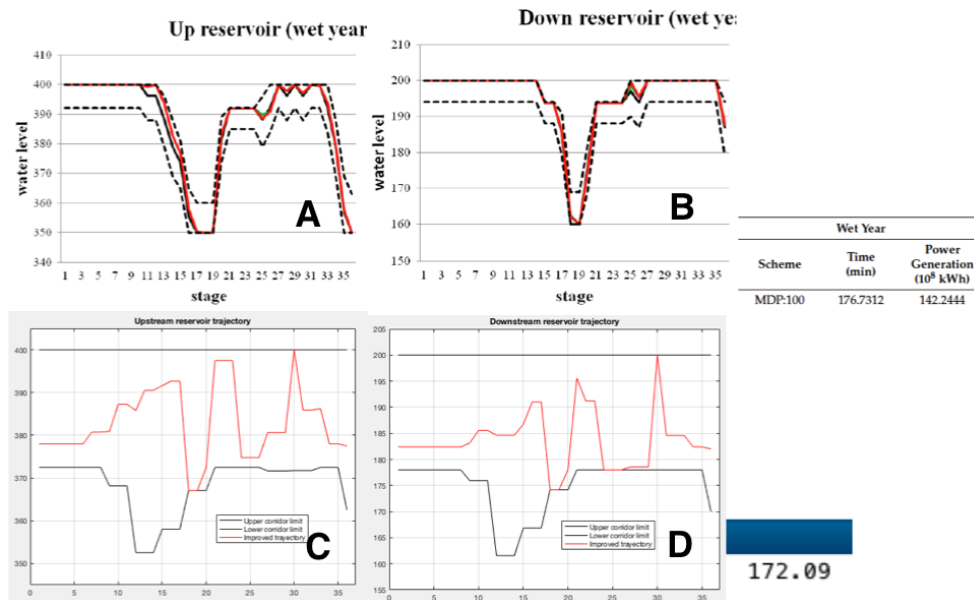


Fig. 32. Comparación de resultados para año húmedo. (A y B) Jiang, año húmedo. (C y D) MATLAB, año húmedo.

Como se puede notar en las figuras 31 y 32, cuando se restringe la capacidad de las turbinas de volumen de agua que pueden usar para generar energía, las trayectorias se parecen más o menos a las reportadas por Jiang con resultados en generación de energía aproximados o

mayores a los reportados por Jiang. Por ejemplo, las figuras 31 (C y D) que, aunque sus trayectorias se asemejan en forma a las reportadas por Jiang (Fig 31(A y B)) los resultados en energía son inferiores a los originales, mientras que si se “abren” las capacidades de las turbinas, los volúmenes aprovechables de agua se distribuyen a lo largo del periodo de planeación y se encuentran cantidades de energía superiores a las encontradas por Jiang. En cualquier caso, se estima que si se conocen con exactitud las capacidades de las turbinas y engeneral, las especificaciones exactas del sistema de presas, además de las restricciones, como la (hipotética) de que se vacié la presa al final del periodo de planeación, con seguridad se pueden igualar los resultados de Jiang. Esta última conclusión se deriva de la observación de que en general, las tendencias presentadas en esta sección son similares con sus consiguientes diferencias por lo ya antes explicado.

MANUAL DE USUARIO

Cálculo de trayectoria inicial usando MDP

Suponga que queremos calcular la trayectoria inicial para los datos de un año seco para el sistema de presas en cascada considerado en el artículo base de este proyecto. Estos son los datos de escurrimientos:

month	Dry year	
	Shuibuya	Geheyan
J	208	69
F	228	75
M	235	77
A	608	200
M	1431	471
J	1157	381
J	669	291
A	280	93
S	216	71
O	404	133
N	398	131
D	178	59

Ahora, suponga las siguientes condiciones

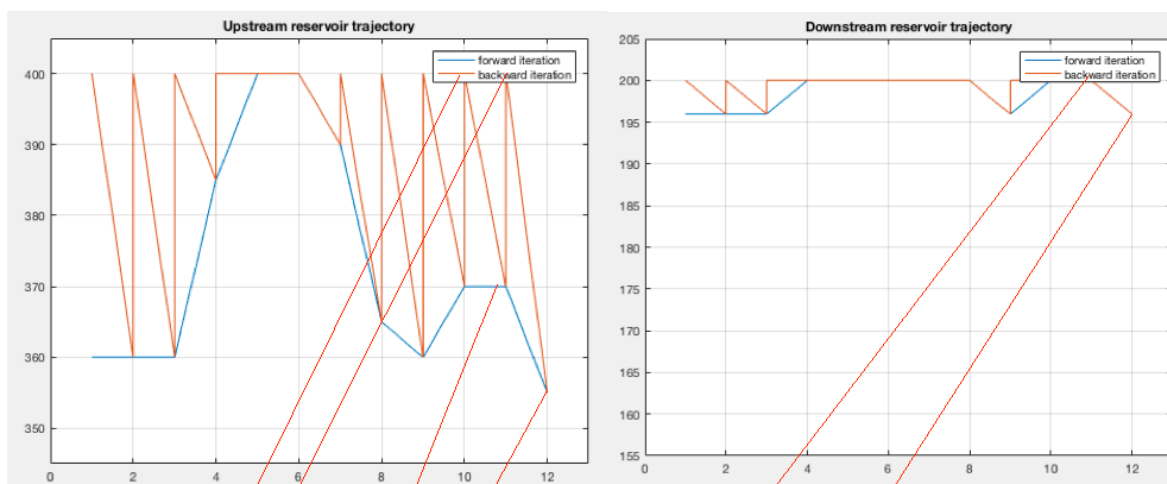
- **10** puntos para discretización del sistema
- **[400m,350m]**: límites de la altura del agua almacenada a ser usados para encontrar la trayectoria inicial para el caso de la presa aguas arriba
- **[200m,160m]**: límites de la altura del agua almacenada a ser usados para encontrar la trayectoria inicial para el caso de la presa aguas abajo.
- **390**: punto inicial estimado de la trayectoria inicial resultante (punto fijo al inicio)aguas arriba

- **192 m**: punto inicial estimado de la trayectoria inicial resultante (punto fijo al inicio)aguas abajo
- periodo de tiempo de planeación: **1 mes**
- capacidad máxima de turbinas aguas arriba: **200 m³/s** (Dato inventado)
- capacidad máxima de turbinas aguas abajo: **325 m³/s**
- capacidad de almacenamiento de la presa aguas arriba: **4.11 E09 m³**
- capacidad de almacenamiento de la presa aguas abajo: **3.4 E09 m³**

En MATLAB:

```
>> upsDry=[208,228,235,608,1431,1157,669,280,216,404,398,178];
dwnsDry=[69,75,77,200,471,381,291,93,71,133,131,59];
[ini, fin]=MDPinitialOperationTrajectory(upsDry,dwnsDry,10,400,350,200,160,390,192,1,200,350,4.11,3.4)
```

Resultados:



Aguas arriba		Aguas abajo		N	N acum.	h final
m1	m2	m1	m2			
400	360	200	196	238.84	2937.3	199
400	360	200	196	239.16	2698.4	199.27
400	360	200	196	239.16	2459.3	199.34
400	385	200	200	250.18	2220.1	202.96
400	400	200	200	255.79	1969.9	202.96
400	400	200	200	255.79	1714.1	202.96
400	390	200	200	250.74	1458.3	202.96
400	365	200	200	242	1207.6	200.09
400	360	200	196	239.16	965.6	199.14
400	370	200	200	244.57	726.44	201.16
400	370	200	200	244.57	481.86	201.1
400	355	200	196	237.29	237.29	198.74

Resultados de la iteración hacia atrás en el cálculo de la trayectoria inicial (línea naranja)

360	360	196	196	58.003	58.003	195.77
360	360	196	196	63.476	121.48	195.77
360	385	196	200	0	121.48	198.44
385	400	200	200	107.15	228.63	200.09
400	400	200	200	255.79	484.42	202.96
400	390	200	200	252.05	736.47	202.96
390	365	200	200	238.96	975.43	202.96
365	360	200	196	107.92	1083.4	195.77
360	370	196	200	14.744	1098.1	200.09
370	370	200	200	117.16	1215.3	200.09
370	355	200	196	172.34	1387.6	195.77
355	390	196	192	17.185	1404.8	191.55

Resultados de la iteración hacia delante de la trayectoria inicial (línea azul)

Note como los gráficos corresponden a los datos numéricos de la siguiente forma: Las columnas de los datos para la presa aguas arriba –por ejemplo- especifican el punto inicial y final (m1 y m2) de un segmento t de la trayectoria. Para la iteración hacia atrás, la línea naranja, para t=1, m1=400, m2=365. La línea azul tiene también su representación numérica correspondiente.

La línea naranja indica los valores encontrados en la iteración hacia atrás, mientras que la línea azul indica los valores tomados en la iteración hacia delante (todos los m1, de acuerdo con Jiang).

APLICACIÓN DE POA A UNA TRAYECTORIA INICIAL

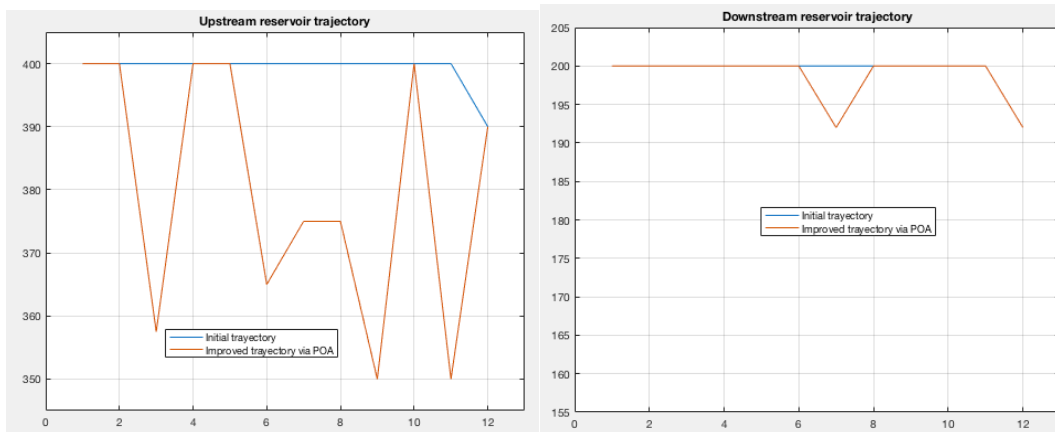
Suponga que se aplicará también para el año seco de la sección anterior.

- fin: trayectoria inicial almacenada en la variable fin y resultado del cálculo de la sección anterior.
- 20 Divisiones en la discretización de un periodo de tiempo t
- [400,350]: limites en las alturas de almacenamiento para la planeación de la presa aguas arriba
- [200,160]:]: limites en las alturas de almacenamiento para la planeación de la presa aguas abajo
- periodo de tiempo de planeación: 1 mes
- capacidad máxima de turbinas aguas arriba: 200 m3/s
- capacidad máxima de turbinas aguas abajo: 325 m3/s
- capacidad de almacenamiento de la presa aguas arriba: 4.11 E09 m3
- capacidad de almacenamiento de la presa aguas abajo: 3.4 E09 m3

En MATLAB:

```
>> [maxOut,newPath]=ApplyPOACreateNewPathTwoReservoirs(fin,upsDry,dwnsDry,20,400,350,200,160,1,325,325,4.11,3.4)
```

Resultados



La línea azul indica la trayectoria inicial, y la línea naranja es la nueva trayectoria o trayectoria optimizada via POA. La representación numérica de las trayectorias se muestra en la siguiente figura.

```
maxOut =
    1869.7

newPath =
    400      400      200      200      66.247      2010.6
    400      400      200      200      74.661      1944.4
    400      357.5      200      200      244.19      1869.7
    357.5      400      200      200      7.8731      1625.6
    400      400      200      200      74.661      1617.7
    400      365      200      200      327.56      1543
    365      375      200      192      321.16      1215.5
    375      375      192      200      306.67      894.3
    375      350      200      200      274.87      587.63
    350      400      200      200      9.7569      312.76
    400      350      200      200      265.43      303
    350      390      200      192      37.571      37.571
```

- fin: trayectoria inicial almacenada en la variable fin y resultado del cálculo de la sección anterior.
- 8: Tamaño del corredor (4 unidades por abajo y arriba de la trayectoria inicial)
- 5: Tamaño de las divisiones en la discretización en la presa aguas arriba
- 4: Tamaño de las divisiones en la discretización en la presa aguas abajo
- 10: Numero de divisiones en el corredor para m1
- 10: Numero de divisiones en el corredor para m2
- [400m,350m]: limites en las alturas de almacenamiento para la planeación de la presa aguas arriba
- [200m,160m]: limites en las alturas de almacenamiento para la planeación de la presa aguas abajo.
- 290m: punto fijo al final de la trayectoria aguas arriba
- 192m: punto fijo al final de la trayectoria aguas abajo
- upsDry: datos de escurrimiento para la presa aguas arriba

- dwnsDry: datos de escurriamiento para la presa aguas abajo
- periodo de tiempo de planeación: 1 mes
- capacidad máxima de turbinas aguas arriba: 200 m3/s
- capacidad máxima de turbinas aguas abajo: 325 m3/s
- capacidad de almacenamiento de la presa aguas arriba: 4.11 E09 m3
- capacidad de almacenamiento de la presa aguas abajo: 3.4 E09 m3

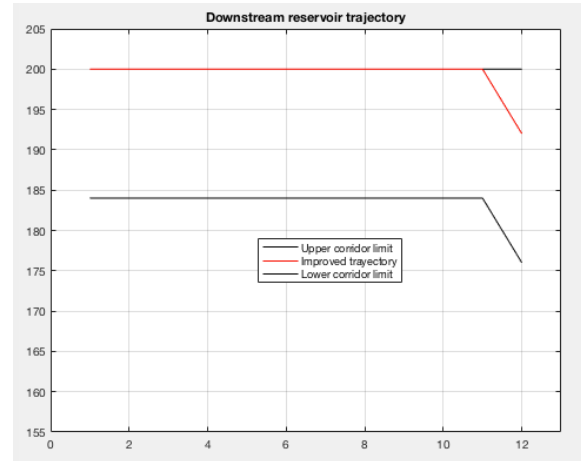
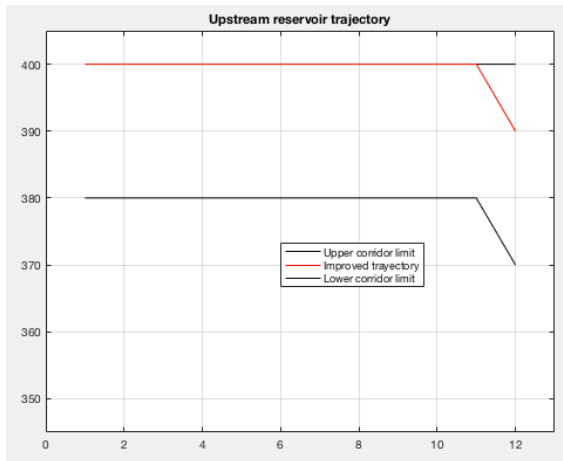
EN MATLAB

```
Command Window
>> [backNewPath,forNewPath]=IMDPImpovingPathTwoReservoirs(fin,8,5,4,10,10,400,350,200,160,390,192,upsDry,dwnsDry,1,325,325,4.11,3.4)
```

Resultados

```
Command Window
backNewPath =
    400    380    200    184    203.45    3124.2    183.36
    400    380    200    184    209.21    2920.7    183.36
    400    380    200    184    211.22    2711.5    183.36
    400    380    200    196.8    295.58    2500.3    196.63
    400    400    200    200    360.7    2204.7    202.96
    400    392    200    200    356.28    1844    202.96
    400    380    200    200    308.32    1487.7    201.4
    400    380    200    184    224.39    1179.4    183.36
    400    380    200    184    205.71    955.01    183.36
    400    380    200    185.6    256.4    749.3    185.67
    400    380    200    185.6    255.4    492.9    185.32
    400    370    200    178.4    237.49    237.49    178.76

forNewPath =
    400    400    200    200    66.247    66.247    200.09
    400    400    200    200    72.488    138.74    200.09
    400    400    200    200    74.661    213.4    200.09
    400    400    200    200    191.1    404.5    200.09
    400    400    200    200    360.7    765.2    202.96
    400    400    200    200    335.91    1101.1    202.96
    400    400    200    200    217.1    1318.2    200.09
    400    400    200    200    88.935    1407.1    200.09
    400    400    200    200    68.703    1475.8    200.09
    400    400    200    200    127.69    1603.5    200.09
    400    400    200    200    125.81    1729.3    200.09
    400    390    200    192    134.13    1863.5    191.55
```



Las líneas negras indican el corredor de búsqueda definido para la trayectoria inicial. La línea roja indica la trayectoria optimizada via IMDP.