

## **Optimización de Generación de un Sistema Hidroeléctrico, en un Clúster de Computadoras, paralelizando los métodos de Relajación Lagrangiana y Programación Dinámica**

**Eustaquio Martínez<sup>1,a</sup>, Esteban Vargas<sup>1,2,b</sup>, Sebastián Arce<sup>1,2,c</sup>**

**<sup>1</sup>Facultad Politécnica - Universidad Nacional del Este, <sup>2</sup>Itaipú Binacional**

**Campus Universitario - Km 8 Ciudad del Este, Hernandarias, Paraguay**

**amartinez@fpune.edu.py<sup>a</sup>, evargas@fpune.edu.py<sup>b</sup>, arce@itaipu.gov.py<sup>c</sup>**

**Paraguay**

### **RESUMEN**

Este trabajo presenta un estudio de la paralelización de la Relajación Lagrangiana (RL) y la Programación Dinámica (PD), en la optimización de generación de un sistema hidroeléctrico. Expone la utilización de ambas técnicas, basada en una heurística de optimización, paralelizando cada uno de los métodos por separado y en conjunto para analizar su desempeño en la resolución del problema. Presenta además los resultados experimentales, para los modelos de paralelización mencionados, de la optimización del sistema hidroeléctrico brasileño, compuesto por 78 usinas, para los casos mencionados, en un ambiente distribuido constituido por un clúster de computadoras personales conectadas en red. Los resultados obtenidos plantean que la paralelización tanto de la RL, la PD y ambos contribuyen a disminuir el tiempo de resolución del problema tratado, cuando resuelto con más de un procesador.

### **PALABRAS CLAVES**

Optimización de generación, Relajación Lagrangiana paralela, Programación Dinámica paralela, Clúster de Computadoras Personales.



## 1. INTRODUCCIÓN

La optimización de generación o de despacho óptimo de unidades generadoras de un sistema hidroeléctrico [1, 2], de gran interés, en el contexto económico, consiste en la adecuada programación de las unidades generadoras, de forma a minimizar los costos asociados a las pérdidas de generación, y el arranque y parada de las mismas [1, 2]. Este problema se puede resolver utilizando una heurística de Relajación Lagrangiana (RL) y Programación Dinámica (PD), según lo propuesto en [1], y la dimensión del mismo requiere de una gran capacidad de procesamiento, lo que lo convierte en un problema dimensional extenso, lo que justifica, en principio el estudio de la paralelización del método [3, 11]. En [13], se propone la paralelización de la RL para el problema hasta aquí mencionado, por lo tanto como continuación de ese trabajo, aquí se propone paralelizar la PD con el objeto de analizar su impacto en el tiempo de procesamiento necesario para la optimización.

Para exponer lo tratado en este trabajo, el artículo se organiza de la siguiente manera: en la sección II una breve revisión del método de RL, en la sección III lo propio con PD, dejando para la sección IV, un rápida descripción del Problema de Despacho

## 2. RELAJACIÓN LAGRANGIANA

La RL [7, 10, 12, 15, 16], se vale de los conceptos de la teoría de la dualidad [17] para tratar el conjunto de restricciones del problema de manera diferenciada. Las restricciones consideradas con cierta complicación son transferidas a la función objetivo, ponderadas por Multiplicadores de Lagrange (ML). Así el problema se transforma en uno relajado, en la que la estructura de las restricciones resultantes, expone subproblemas que pueden ser resueltos con menos dificultad si comparado con la versión original.

## 3. PROGRAMACIÓN DINÁMICA

La PD fue creada para el tratamiento matemático de situaciones en que el problema se reduce a procesos de decisión multietapas, esta decisión es denominada política y esa política presentará mayores ventajas en función a algún criterio pre asignado, denominado política óptima. En resumen la idea básica de la PD es, en función a la política óptima, determinar la decisión requerida a cada estado del sistema, basada en el principio de optimalidad [6], que establece que una óptima política tiene la propiedad de que cualquiera sean el estado y la decisión iniciales, las decisiones remanentes deben constituir una óptima política, respecto al estado resultante de la primera decisión.

Un problema de optimización puede ser planteado como [8]: encontrar la secuencia de decisiones  $(u(1); \dots; u(n))$ , que puede describirse como política y su correspondiente secuencia de estados  $(x(1); \dots; x(n))$ , denominada generalmente trayectoria, que minimiza la función costo

$$J = \sum_{k=1}^n L(x(k); u(k); k) \quad (1)$$

donde

$$x(k+1) = g(x(k); u(k); k) \quad (2)$$

y sujeto al conjunto de restricciones de los estados y las variables de decisión, formuladas como:

$$x \in X(k); u \in U(x(k); k) \quad (3)$$

En este caso,  $x$  es la variable de estado,  $X$  es el conjunto de estados posibles,  $u$  es la variable de decisión (política),  $U$  es el conjunto de decisiones admisibles,  $k$  es la etapa, y  $J$  es el costo de la función objetivo;  $L$  representa el costo de una etapa en particular.

Entonces, la función costo mínimo en la etapa  $k$ , en el final del problema de decisión, se define como:

$$I(x; k) = \min_{u(k); u(k+1); \dots; u(n)} \sum_{j=k}^n L(x(j); u(j); j) \quad (4)$$

Según el principio de optimalidad de Bellman [6], todas las porciones de una trayectoria óptima, son trayectorias óptimas en si mismas, es decir:

$$\begin{aligned} I(x; k) &= \min_u \{ L(x(k); u(k); k) + I(g(k); u(k); k; k+1) \\ &= \min_u \{ L(x(k); u(k); k) + I(x(k+1); k+1) \end{aligned} \quad (5)$$

con  $I(x; n) = \min_{u(n)} \{ L(x(n); u(n); n) \}$  para la última etapa.

#### 4. DESPACHO ÓPTIMO DE UNIDADES GENERADORAS DE UN SISTEMA HIDROELÉCTRICO

El despacho óptimo de unidades generadoras de un sistema hidroeléctrico [1], generalmente utiliza como criterio de desempeño un modelo que evalúa las pérdidas en la generación y el costo asociado a arranques y paradas de las máquinas. Este problema se caracteriza por ser entero-mixto y no convexo, no lineal y de gran porte, lo que dificulta de sobremanera la obtención de la solución óptima.

Para resolver este problema se utiliza una metodología heurística propuesta en [1] y cuya paralelización de la RL se propone en [13]. La idea se basa en descomponer el problema en dos subproblemas: el de Despacho de Unidades Generadoras (DU), que determina la configuración de unidades generadoras en operación en cada una de las usinas y en cada intervalo de tiempo [2]; y el que corresponde al Despacho de Generación (DG), que determina la asignación óptima de generación para una configuración de unidades generadoras en una operación dada. Estos dos subproblemas pueden ser resueltos en forma iterativa hasta obtenerse la solución global del sistema. La Figura 1 expone este razonamiento.



**Figura 1: Esquema de la Metodología Heurística de solución del problema de Despacho Óptimo [1]**

El subproblema DG puede ser formulado como un caso particular del problema del despacho hidroeléctrico, para el cual el número de unidades generadoras en operación ya está definido y su función objetivo puede expresarse como:

$$\min \sum_{t=1}^T \sum_{i=1}^N c^p \cdot f_i(n_i^{pt}; p_i^t) \quad (6)$$

s. a.:

$$\sum_{i=1}^N p_i^t = d^t \quad \forall t \quad (7)$$

$$\sum_{i=1}^N p_i^t = T \cdot m_i \quad \forall i \quad (8)$$

$$p_i^{\min}(n_i^{pt}) \leq p_i^t \leq p_i^{\max}(n_i^{pt}) \quad (9)$$

donde:

$T$  es el número de intervalos de tiempo;

$N$  es el número de usinas hidroeléctricas;

$f_i(p_i^t; n_i^{pt})$  es la función de pérdidas en la usina  $i$  [MW], que depende del número de unidades generadoras  $n$  y de la potencia generada  $p$  durante el intervalo de tiempo  $t$ ;

$m_i$  es la meta energética de la usina hidroeléctrica  $i$  [MWmedia];

$d^t$  es la carga propia del sistema durante el intervalo de tiempo  $t$  [MW];

$n_i^{pt}$  es el número de unidades generadoras en operación en la usina  $i$  en intervalo de tiempo  $t$ ;

$c^p$  costo de las pérdidas asociadas al sistema de generación;

$p_i^t$  potencia generada en la usina  $i$  durante el tiempo  $t$ ;

$p_i^{\min}(\cdot); p_i^{\max}(\cdot)$  potencia mínima y máxima, respectivamente, asociada a  $n$  unidades generadoras despachadas en la usina  $i$ , en un tiempo  $t$ ;

Este problema se puede resolver utilizando RL [20-23]. Para detalles de la formulación matemática referirse a [13].

La función objetivo a optimizar es:

X SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRÉ  
19, 20 y 21 de setiembre de 2012

$$f(X) = \sum_{t=1}^T \sum_{i=1}^N c^p : f_i(n_i^{pt}; p_i^t) \quad (10)$$

que siguiendo los lineamientos de [13], puede ser escrita como:

$$\sum_{t=1}^T \sum_{i=1}^N c^p : f_i(n_i^{pt}; p_i^t) + \sum_{t=1}^T \sum_{i=1}^N d_i^t \cdot p_i^t + \sum_{t=1}^T \sum_{i=1}^N m_i : T_i \cdot p_i^t \quad (11)$$

Lo que permite escribir la función Dual como:

$$\min_X L(X; u) = \min_{p_i^t} L(p_i^t; \lambda, \mu) \quad (12)$$

$$s.a : p_i^t \cdot \lambda_j ; j = 1; 2; \dots; p_i^t \cdot \mu_i^{min}(n_i^{pt}) \cdot p_i^t \cdot \mu_i^{max}(n_i^{pt}); \delta_i ; t$$

finalmente la solución del problema (6) al (9) puede ser obtenida resolviendo el siguiente problema Dual D:

$$\max \mathcal{E}(u) = \max h(\lambda, \mu) \quad (13)$$

$$s.a : u \in R^m, \lambda \in R^T; \mu \in R^N$$

La solución propuesta en [1]. Plantea la estructura jerárquica, que en el nivel superior (coordinador o programa maestro) determina los valores de los ML  $(\lambda, \mu)$  y en el nivel inferior resuelve los subproblemas primales, uno para cada usina y para cada intervalo de tiempo, en función a los multiplicadores de Lagrange fijados por el coordinador.

Los bloques  $p(i;t)$  en la Figura 2, representan a la solución de los subproblemas primales asociados a la usina  $i$  en el intervalo de tiempo  $t$ . En este nivel resuelve  $N \times T$  subproblemas y en el nivel superior el coordinador determinan los valores de  $\lambda$  y  $\mu$  para resolver el problema dual.

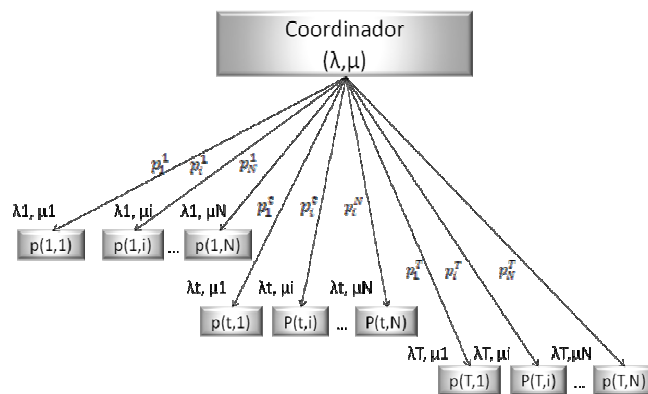


Figura 2: Esquema general de la solución del subproblema DG [1].

#### 4.1 Solución del problema y la Relajación Lagrangiana Paralela

La solución del problema se logra a través del procedimiento heurístico mencionado en el apartado anterior, alternando PD para el subproblema DU y RL para el subproblema DG, hasta obtener la mejor solución. Este procedimiento, considerando la estructura de la figura 2, observando la posibilidad de la ejecución de cálculos concurrentes, fue aprovechado para paralelizar la RL, a través del paradigma Maestro-Eslavo, en un ambiente computacional distribuido, construido a partir de Computadoras Personales en

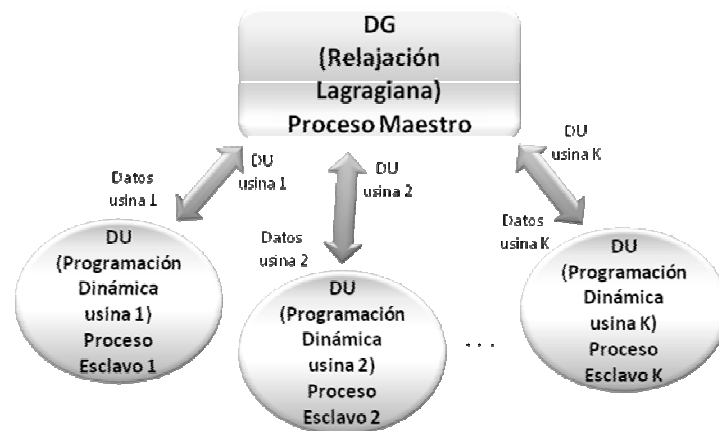
[13], detalles de la implementación de una solución secuencial, se pueden obtener en [1].

#### 4.2 Solución del problema y la Programación Dinámica

En la solución propuesta en [13], se observó que el subproblema DU, se resuelve a través de PD, en más de una oportunidad, por lo tanto la realización de los cálculos asociados, en forma concurrente, a primera vista, parece interesante, especialmente teniendo en cuenta las dimensiones del sistema ejemplo resuelto a través de la RL paralela [13]. La mencionada idea se puede concretar, resolviendo el subproblema DU en más de un procesador, con miras a analizar su impacto en la reducción de los tiempos de procesamiento en la resolución del problema en cuestión.

### 5. PROGRAMACIÓN DINÁMICA PARALELA EN EL DESPACHO OPTIMO DE UNIDADES GENERADORAS

La paralelización de la PD no es una idea nueva [8], así como en el contexto de la ingeniería eléctrica [11], lo que es interesante, puesto que forma parte de la solución heurística [1] del problema de optimización de generación de un sistema hidroeléctrico. Analizando el esquema de la figura 1, parece interesante proponer una modificación, en que la RL se realice en un proceso Maestro y la PD en varios procesos esclavos, siguiendo los lineamientos de la figura 3.



**Figura 3: Modelo Maestro Esclavo para el Problema DG y DU, paralelizando la Programación Dinámica**

La idea es que cada procesador o proceso esclavo resuelva el DU para un horizonte de corto plazo (24 horas), por cada usina, a cada iteración del proceso de optimización en base a la RL, lo que implica ejecutar, en el procesador Maestro, el Algoritmo de la Figura 4 y en cada procesador esclavo el algoritmo de la Figura 5.

**X SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRÉ**  
**19, 20 y 21 de setiembre de 2012**

Proceso MAESTRO:  
Inicio.  
Iniciar los procesos esclavos (*MPI\_Init()*)  
**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)  
Enviar (*switch open*) a todos los procesos esclavos (*MPI\_Send()*)  
Enviar *Datos* a todos los procesos esclavos (*MPI\_Send()*)  
Fin Para ( $k$ )  
**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)  
Recibir Despacho de pos proceso esclavos  
Fin Para ( $k$ )  
Mientras existe alteración en la última configuración de unidades generadoras.  
**Calcular** subproblema DG (Se utiliza Relajación Lagrangeana)  
Calcular multiplicadores de Lagrange ( $\lambda, \mu$ )  
Para  $t=1$  hasta  $T$   
    Resolución de la ecuación (21)  
Fin Para.  
**Calcular errores** (Errores de restricciones de demanda y metas energéticas)  
**Prueba de Optimalidad.** (Errores de generación energética y atención a la demanda)  
Si no encuentra solución volver a *Calcular subproblema DG*.  
    **Fin** subproblema DG.  
**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU, nuevas configuraciones )  
Enviar (*switch open*) a todos los procesos esclavos (*MPI\_Send()*)  
Enviar *Datos* a todos los procesos esclavos (*MPI\_Send()*)  
Fin Para ( $k$ )  
**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)  
Recibir Despacho de los proceso esclavos (*MPI\_Recev()*)  
Fin Para ( $k$ )  
Fin Mientras  
**Para**  $k=1$  **hasta**  $(K - 1)$   
Enviar (*switch close*) a todos los procesos esclavos (*MPI\_Send()*)  
Fin Para ( $k$ )  
**Fin.** (*MPI\_Finalize*)

**Figura 4: Algoritmo del Proceso Maestro para el modelo de la Figura 5.**

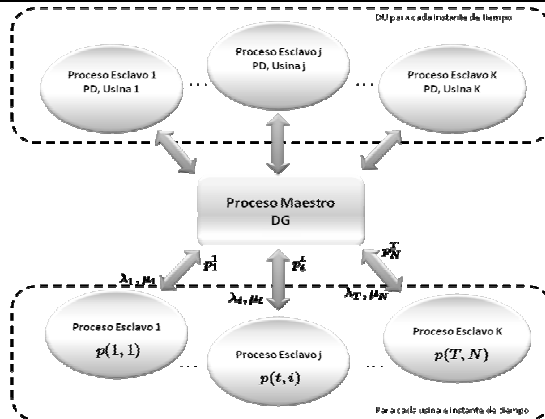
Proceso ESCLAVO:  
Recibir (*switch open*) del Maestro  
**Mientras** (*switch open*)  
Recibir los datos fijos del proceso MAESTRO (*MPI\_Recev()*)  
**Calcular** Despacho de Unidades **por** PD  
**Enviar** Despacho de Unidades **al** MAESTRO (*MPI\_Send()*)  
Recibir (*switch close*) del MAESTRO  
Fin Mientras

**Figura 5: Algoritmo del Proceso Esclavo para el modelo de la Figura 5.**

Los algoritmos de las figuras 4 y 5 fueron implementados en la misma plataforma de [13], y los resultados y comparaciones se presentan en la sección VI.

Con el mismo objetivo de analizar el impacto en los tiempos de ejecución, se plantea la paralelización de la RL, y de la PD, al mismo tiempo, es decir, el DG y el DU en el mismo modelo Maestro-Esclavo. La idea sigue el esquema propuesto por la Figura 6.

X SEMINARIO DEL SECTOR ELÉCTRICO PARAGUAYO - CIGRÉ  
19, 20 y 21 de setiembre de 2012



**Figura 6: Modelo Maestro Esclavo para el Problema DG y DU, paralelizando la Relajación Lagrangiana y la Programación Dinámica**

Para codificar lo planteado por el esquema de la figura 6, se han utilizado los algoritmos de las figuras 7 y 8.

**Proceso MAESTRO:**

Inicio.

Iniciar los procesos esclavos (*MPI\_Init()*)

**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)

Enviar (*switch open*) y *Datos* a todos los procesos esclavos (*MPI\_Send()*)

Fin Para ( $k$ )

**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)

Recibir Despacho inicial de los proceso esclavos

Fin Para ( $k$ )

Mientras existe alteración en la última configuración de unidades generadoras.

**Calcular** subproblema DG (Se utiliza Relajación Lagrangeana)

Calcular multiplicadores de Lagrange ( $\lambda, \mu$ )

**Para**  $k=1$  **hasta**  $(K - 1)$ .

Enviar (*switch open*) a todos los procesos esclavos (*MPI\_Send()*)

Enviar ( $\lambda, \mu$ ) a todos los procesos esclavos (*MPI\_Send()*)

Fin Para ( $k$ )

**Para**  $k=1$  **hasta**  $(K - 1)$

Recibir  $p_i^t$  de todos los esclavos (*MPI\_Recv()*)

Fin Para ( $k$ ).

**Calcular errores** (Errores de restricciones de demanda y metas energéticas)

**Prueba de Optimalidad** (Errores de generación y atención a la demanda)

Si no encuentra solución volver a *Calcular subproblema DG*.

**Fin** subproblema DG.

**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU, nuevas configuraciones)

Enviar (*switch open*) y *Datos* a todos los procesos esclavos (*MPI\_Send()*)

Fin Para ( $k$ )

**Para**  $k=1$  **hasta**  $(K - 1)$  (Resolución del subproblema DU)

Recibir Despacho de los proceso esclavos

Fin Para ( $k$ )

Fin Mientras

**Para**  $k=1$  **hasta**  $(K - 1)$

Enviar (*switch close*) a todos los procesos esclavos (*MPI\_Send()*)

Fin Para ( $k$ )

**Fin.** (*MPI\_Finalize*)

**Figura 7: Algoritmo del Proceso Maestro para el modelo de la Figura 8.**



X SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRÉ  
19, 20 y 21 de setiembre de 2012

```
Proceso ESCLAVO:
Recibir (switch open) del Maestro
Inicializa valores necesarios para cálculos.
Procesa división de trabajo acorde a cantidad de procesadores disonibles (Tiempo inicial y final)
Mientras (switch open)
Recibir los datos fijos del proceso MAESTRO (MPI_Recev())
Calcular Despacho de Unidades por PD
Enviar Despacho de Unidades al MAESTRO (MPI_Send())
Recibir (Switch close) del MAESTRO (MPI_Recev())
Fin Mientras
Recibe switch del maestro. (MPI_Recv( ))
Mientras (switch open)
Recibir ( $\lambda, \mu$ ) del MAESTRO (MPI_Recv())
Para inicio tiempo hasta final tiempo
Calcular ecuación (21).
Fin Para.
Enviar  $p_i^t$  al proceso MAESTRO (MPI_Send())
Recibir switch del MAESTRO (MPI_Recv( ))
Fin Mientras
```

**Figura 8. Algoritmo del Proceso Esclavo para el modelo de la Figura 8.**

En el modelo de la Figura 6, cada proceso esclavo recibe  $\lambda$  y  $\mu$  del Maestro y calcula la potencia generada ( $p_i^t$ ) en la usina  $i$  en un instante de tiempo  $t$ , devolviendo el valor al proceso Maestro. Del mismo modo en cada proceso esclavo se calcula el DU para la configuración inicial de las máquinas por PD, hecho que se repite a cada intervalo de tiempo, para determinar las nuevas configuraciones, para cada usina del sistema hidroeléctrico, mientras no termine el proceso iterativo de la optimización.

## 6. RESULTADOS EXPERIMENTALES

### 6.1 Plataforma computacional

Los algoritmos de las Figuras 3, 4, 6, 7, 9 y 10 fueron codificados utilizando el lenguaje de programación Octave [22], y las directivas de la biblioteca MPITB (MPI ToolBox) [20], sobre una arquitectura PelicanHPC [23], es decir un clúster de estaciones de trabajo, consistente en computadoras personales. El mencionado clúster, consta de 7 computadoras idénticas con procesador AMD Opteron de 2.2Ghz, 1GBytes de memoria RAM, interconectadas por una red Ethernet de 100Mbps y totalmente aislada de otras redes. Una de las máquinas del grupo se estableció como Maestro y las demás como Esclavos.

### 6.2 Resultados

El desempeño de la paralelización se pudo verificar efectuando una serie de pruebas de ejecución, utilizando como problema ejemplo el despacho de máquinas para el horizonte de programación diaria, de 78 usinas del sistema hidroeléctrico brasileño. La discretización utilizada fue horaria. Como métrica, en el contexto paralelo, se utilizó el SpeedUp ( $Sp$ ), definido como:

$$Sp = \frac{t_s}{t_p} \quad (14)$$

X SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRÉ  
19, 20 y 21 de setiembre de 2012

donde  $t_s$  corresponde al tiempo (en segundos) que se requiere para que el programa se ejecute en forma secuencial y  $t_p$  al tiempo que se requiere para ejecutar el programa en  $p$  procesadores.

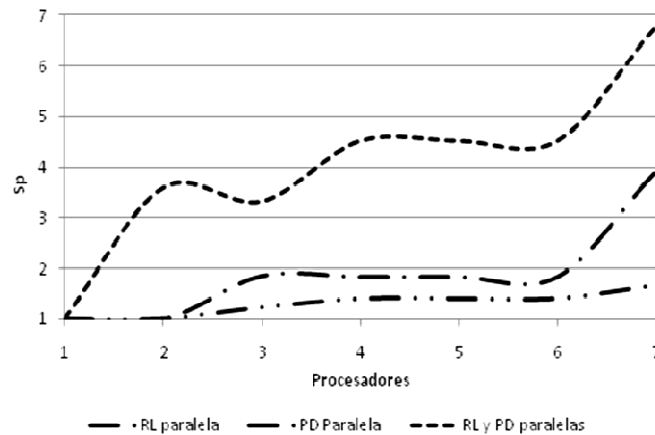


Figura 11: Gráfico del  $S_p$  para cada modelo de paralelización presentado

El gráfico de la figura 11, muestra que paralelización de la RL en comparación con la paralelización de la PD, presenta mejores resultados, puesto que el  $S_p$  llega a aproximadamente a 4 con 7 procesadores, lo que implica una disminución del tiempo de procesamiento a prácticamente un 25%, del tiempo de ejecución secuencial, mientras que para la PD paralela es menor que 2, lo que implica que en este caso la disminución del tiempo no es significativa, con la misma cantidad de procesadores; sin embargo la implementación de la RL y la PD paralelas es la que presenta mejor  $S_p$ , puesto que llega a prácticamente 7 con 7 procesadores, es decir es prácticamente 7 veces más rápido que la solución secuencial. Esto implica que paralelizar ambos métodos (RL y PD) reduce el tiempo de procesamiento, en la plataforma mencionada, de la optimización de la generación de un sistema hidroeléctrico con las características del problema ejemplo utilizado.

## CONCLUSIONES

Con este estudio se pudo constatar, que la PD en la resolución del problema planteado en este trabajo, no afecta sustantivamente al tiempo de ejecución en el proceso de optimización, puesto que su paralelización, dejando el despacho de generación para el procesador Maestro, no ha significado un aumento del  $S_p$ , sino al contrario, se ha reducido, si comparado con la paralelización de la RL solamente, en el que se deja para el Proceso Maestro la PD, sin embargo la Paralelización de ambos modelo la RL y la PD, ha significado un interesante aumento en el  $S_p$ . Lo que finalmente implica que la aplicación de técnica de paralelización en el problema tratado trae interesantes ventajas en lo relativo al tiempo de ejecución en un ambiente distribuido.

Como consecuencia se plantea como trabajos futuros el estudio de otros modelos de paralelización y ambientes computacionales distribuidos y heterogéneos.

## REFERENCIAS

- [1] A. Arce, Despacho Ótimo de Unidades Geradoras em Sistemas Hidroelétricos via Heurística Baseada em Relaxação Lagrangeana e Programação Dinâmica, P.hD. Thesis, Universidade Estadual de Campinas, Brasil, enero de 2006.
- [2] A. Arce, T. Ohishi and S. Soares, "Optimal dispatch of Generating Units of the Itaipu Hydroelectric Plant", IEEE Trans, on Power Systems, Vol. 17, No 1, Febrero 2002.
- [3] F. Almeida, "Introducción a la programación paralela", Ed. Paraninfo Cengage Learning, 2008.
- [4] S. Bazaraa, C. Shetye; *Nonlinear Programming*, Jhon Wiley & Sons, Atlanta, 1979.
- [5] D. P. Bertzakas, *Nonlinear Programming*, Athena Scientific, Massachusetts, 1999.
- [6] R. Bellman, "The Theory of Dinamic Programing", Bull. Amer. Math. Soc. Volume 60, Number 6 (1954), 503-515.
- [7] H. Everett III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", Oper. Res. 11, 399-417, 1963.
- [8] S. Dormido, et al, "Parallel Dynamng Programming on Cluster of Workstations", IEEE Trans. on Parallel and Distributed System, Vol. 16, No. 9, setiembre de 2005
- [9] L. Feng, "A Parametric Iteration Method for Sthochastic Dynamng Programming for Optimal Dispatch of Hydroelectric Plants", IEE 2nd International Conference in Power System Control, Operation and Management, Hong Kong, Diciembre de 1993
- [10] M. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems", Management Science 27, 1-18, 1981.
- [11] C. Fung, S. Chow, K. Wong, "A low Cost Parallel Computing Platform for Power Engineering Applications", Proceedings of the 5th. International Conference on Advances in Power System Control and Management, APSCOM 20, Hong Kong, octubre de 2000.
- [12] A. Geoffrion, "Lagrangean relaxation for integer programming", Mathematical Programming Study 2, 82-114, 1974.
- [13] F. González, L. Fariña, E. Martínez, E. Vargas, A. S. Arce, "Relajación Lagrangiana Paralela en la Optimización de Generación de un Sistema Hidroeléctrico", XXXVII Conferencia Latinoamericana de Informática, CLEI, 2011, Quito, Ecuador, 10 al 15 de octubre de 2011.
- [14] M. Held, R. Karp, "The traveling salesman problem and minimum spanning trees. Operations Research", 18:1138-1162, 1970.
- [15] M. Held, R. Karp, "The Traveling salesman problem and minimum spanning trees: Part II", Mathematical Programming I, 6-25, 1971.
- [16] C. Rodríguez, et al., "Paradigms for Parallel Dynamic Programming", Proceedings of Euromicro -22, 1996
- [17] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming", in Proc. of the 2nd Berkeley Symposium on Mathematical Statistic and Probability, J. Neyman (Ed.), University of California Press. Berkeley, CA, 1951.



Comité Nacional Paraguayo



Unión de Ingenieros de ANDE

X SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRÉ  
19, 20 y 21 de setiembre de 2012

---

- [18] L. S. Lasdon, "Optimization Theory of Large Systems", Macmillan Company, New York, NY, 1970.
- [19] D. G. Luenberger, "Introduction to Linear and Non-linear Programming", Second Edition, Addison-Wesley, Reading, Mass., 1984.
- [20] L. Morris L., MATLAB Laboratory for MPI toolbox (MPITB) URL: [http://www.tech.plym.ac.uk/spmc/links/matlab/matlab\\_mpi.html](http://www.tech.plym.ac.uk/spmc/links/matlab/matlab_mpi.html)
- [21] L. Moreno, I. Acosta, J. Sánchez, "Design of algorithms for Spatial-time Reduction Complexity of Dynaming Programming", IEE Proceedings-D, Vol 139, No.2, marzo de 1992.
- [22] Octave, URL: <http://www.gnu.org/software/octave/>
- [23] PelicanHPC GNU Linux, URL: <http://idea.uab.es/mcreel/ParallelKnoppix/>
- [24] R. Singiresu, Engineering Optimization, Theory and Practice, Fourth Edition, John Wiley and Sons, Inc., 2009.