A-TEAMS EN LA OPTIMIZACIÓN DEL CAUDAL TURBINADO DE UNA REPRESA HIDROELÉCTRICA

Benjamín Barán[1] *bbaran@cnc.una.py*

Enrique Chaparro V. *eviveros@cnc.una.py*

Néstor Cáceres *gustafson@sce.cnc.una.py*

Centro Nacional de Computación
Universidad Nacional de Asunción
Casilla de Correos 1439
Campus Universitario de San Lorenzo – Paraguay
Tel./Fax**:** (595) (21) 585.619 y 585.550
http://www.cnc.una.py

Resumen

Una represa hidroeléctrica de gran envergadura tiene generalmente un número considerable de unidades generadoras, cada una con sus propias características físicas y restricciones operativas. El presente trabajo propone aprovechar la infraestructura computacional de las redes de computadoras existentes para optimizar la generación de la potencia demandada, minimizando el recurso hídrico (caudal) a ser turbinado.

La complejidad del problema considerado se debe a diversos factores como el número y características de las unidades generadoras, la altura variable del agua y la necesidad de parar determinadas unidades generadoras para mantenimiento o reparaciones. Debido a esta complejidad, los métodos numéricos exhaustivos tienen un alto costo computacional, razón por la cual, se propone la utilización de *A-Teams* (*Asynchronous Teams*) que combinan los Algoritmos Genéticos (AG), posiblemente paralelos, con métodos numéricos tradicionalmente utilizados en la resolución de este tipo de problemas.

El trabajo describe la implementación de un *A-Team* distribuido que aprovecha diversos procesadores de una red de computadoras personales, levantando resultados experimentales que demuestran las ventajas de utilizar *A-Teams*, en la medida que aumenta la complejidad y el tamaño del problema.

Palabras claves

Algoritmos Genéticos Paralelos, *A-Teams***,** *Team Algorithm*, usina hidroeléctrica.

[1] Este trabajo ha sido desarrollado con el apoyo de la Dirección de Investigaciones, Postgrado y Relaciones Internacionales de la Universidad Nacional de Asunción.

1. Introducción

El Paraguay es un país exportador de energía eléctrica gracias a sus represas hidroeléctricas de gran envergadura, como Itaipú y Yacyretá, entre otras. La generación de energía eléctrica en estos sistemas reales de gran porte, está caracterizada por complejas ecuaciones matemáticas [1], para las cuales no siempre se dispone de un método único de resolución. En consecuencia, se estudian diversos métodos de optimización, cada uno con sus propias características que lo hacen apropiado para un subconjunto de problemas. En algunos casos, un método tiene un buen desempeño bajo ciertas características operativas, pero falla para otros escenarios del problema para los cuales existen alternativas diferentes de solución, que a su vez sólo son aplicables a una determinada clase de problemas. Surge así la necesidad de utilizar un método de optimización adecuado para cada clase de problema. Ocasionalmente, puede suceder que ninguno de los métodos disponibles consigue resolver adecuadamente el problema en cuestión, cuando son utilizados aisladamente [2].

Ante esta dificultad, surge la posibilidad de descomponer un problema complejo en subproblemas menores, de forma a utilizar métodos diferentes en cada uno de estos subproblemas, según sean sus características específicas. Esta combinación de algoritmos diferentes interactuando en la solución de un mismo problema global, se conoce como *Team Algorithm* [2]. La solución de problemas utilizando *Team Algorithm* puede ser naturalmente implementada en paralelo, asignando los diferentes subproblemas a los diversos procesadores de un sistema distribuido asíncrono, como una red de computadoras, en cuyo caso se lo conoce como *A-Team* (Asynchronous Team) [3-5].

En un trabajo anterior realizado por Talukdar et al. [3], se implementó un *A-Team* combinando Algoritmos Genéticos (AG) con reconocidos métodos numéricos, como el Método de Newton, aplicados a la resolución de sistemas algebraicos de ecuaciones no lineales. De esta forma, se obtuvo una combinación asíncrona de algoritmos que interactuaban sinergéticamente reduciendo significativamente los recursos computacionales que demandaría la sola utilización de métodos numéricos tradicionales, reportándose muy buenas aceleraciones (*speedup*). A la fecha, se puede observar un creciente interés en la utilización de *A-Teams* para diversas aplicaciones [2-5].

En el problema aquí estudiado, se tiene una represa hidroeléctrica con diversas turbinas (por ejemplo, 18 en Itaipú), cada una de las cuales debe turbinar una determinada cantidad de agua para que, en conjunto, se genere la potencia demandada. La potencia entregada por cada unidad generadora depende de sus características de funcionamiento, dadas por *Curvas de Eficiencia* propias de cada turbina (*Potencia* [Mw] vs. *Caudal* [m^3/s]), levantadas experimentalmente mediante un "*index-test*". Estas *Curvas de Eficiencia* determinan las características de funcionamiento de cada unidad

generadora. Por lo tanto, el objetivo del presente trabajo es generar la potencia total demandada (P_D), minimizando el caudal a ser turbinado, optimizándose así la utilización de los recursos hídricos, atendiendo siempre a las restricciones de la usina hidroeléctrica.

En resumen, el problema planteado puede ser formalizado como:

$$\textit{Minimizar} \qquad Q_T = \sum_{i=1}^{N_T} f_i(P_i, h) \quad . \quad \textbf{(1)}$$

$$\textit{sujeto a} \qquad \sum_{i=1}^{N_T} P_i = P_D \quad \textbf{(2)}$$

donde las curvas $f_i(P_i, h)$ se obtienen experimentalmente, para cada generador i, a partir de un "*index-test*".

Los métodos numéricos tradicionalmente utilizados en este problema típicamente están formados por una combinación de métodos exhaustivos de búsqueda global con métodos numéricos de optimización local (como el método del Gradiente), resultando en un algoritmo cuyo tiempo de ejecución crece exponencialmente con la complejidad del problema (dado por ejemplo, por el número de unidades generadoras). Teniendo en cuenta la posibilidad de utilizar *A-Teams* para reducir la complejidad de los cálculos y los recursos computacionales requeridos, el presente trabajo propone la combinación de Algoritmos Genéticos (AG), reconocidos por su capacidad de búsqueda global [6-7], con los métodos numéricos tradicionalmente utilizados en la actualidad, en forma conceptualmente similar a lo propuesto en [3].

Si bien la utilización de Algoritmos Genéticos en problemas eléctricos no es nueva [8], la presente propuesta facilita una implementación en paralelo, aprovechando el asincronismo intrínseco a las redes de computadoras existentes, de forma a mejorar el desempeño de los Algoritmos Genéticos, al ser combinados con métodos numéricos utilizados en la actualidad [9-10].

La sección 2 describe al método numérico y al Algoritmo Genético, cuya combinación se describe en la sección 3. Resultados experimentales son presentados en la sección 4 y las conclusiones finales en la sección 5.

2. Métodos de Optimización Utilizados

El presente trabajo propone la combinación de dos técnicas de optimización, cada una con sus características bien diferenciadas. Por un lado están los métodos numéricos, de probada eficacia en optimizaciones locales, pero de complejidad típicamente exponencial cuando se requiere una búsqueda exhaustiva. Por el otro, los Algoritmos Genéticos, intrínsecamente paralelos en su búsqueda, han demostrado una excelente robustez en su capacidad de búsqueda global [6-7]. De hecho, la idea de combinar estas técnicas de tal forma a aprovechar las mejores propiedades de cada técnica ya ha sido propuesta para diversas aplicaciones [11].

El método numérico utilizado combina las ventajas de un método exhaustivo de búsqueda global con otro de optimización local dentro de la región de búsqueda. El método consiste en descomponer el dominio global de búsqueda en pequeños subdominios monótonos dentro de los cuales se pueda realizar una optimización local, utilizando métodos numéricos tradicionales como el algoritmo del Gradiente para encontrar los óptimos correspondientes a los subdominios más prometedores.

El método numérico inicia la búsqueda a partir de un punto inicial que se va comparando con muestras de cada subdominio. Si la muestra es *prometedora*, se la optimiza con una variante del Método del Gradiente [12] que calcula el óptimo, posiblemente local, del subdominio en cuestión, que de resultar mejor al candidato inicial, lo reemplaza. Recorridos todos los subdominios, el método presenta la mejor solución encontrada. Lógicamente, la *calidad de la solución* encontrada puede mejorar con el número de subdominios, lo que a su vez requiere de mayores recursos computacionales.

La curva de eficiencia para una turbina *i* está dada por una lista de puntos experimentales de potencia generada P_i en función del caudal turbinado Q_i, para valores parametrizados de la altura de agua h, disponible en la usina.

$$P_i = f_i (Q_i, h) \qquad \textbf{(3)}$$

Los parámetros de entrada del método numérico son: la potencia demandada (P_D), la altura disponible (h), la lista de turbinas que están en servicio (S) y el número de turbinas en servicio (N_T), trabajándose con curvas de eficiencia interpoladas con respecto al parámetro h.

El dominio en el cual se realiza el análisis global, se divide en subespacios, formando un CUADRÍCULA. La dimensión del espacio de búsqueda es igual al número N_T de turbinas en servicio. En la CUADRÍCULA, los puntos son analizados sistemáticamente, y se consideran sólo aquellos puntos que atienden a las restricciones del problema (2).

```
Entrada de Parámetros (P_D, h, S, N_T);
Obtención de Curvas de Eficiencia respecto a h;
Crear CUADRÍCULA (Espacio de Búsqueda);

DO WHILE ( No se analice toda la CUADRÍCULA )
   IF ( Punto Prometedor que atiende restricciones ) THEN
            Método del Gradiente( );
   END IF
END DO
```

Pseudocódigo 1: Método Numérico.

En resumen, el método exhaustivo se utiliza para definir los puntos del espacio de búsqueda, recorriendo cada subespacio, optimizando localmente los puntos prometedores.

La segunda técnica computacional considerada es el Algoritmo Genético, basado en los mecanismos naturales de Selección y Genética [6]. En él, una población de posibles soluciones *evoluciona* mediante la aplicación de operadores típicamente probabilísticos (*Selección*, *Cruzamiento* y *Mutación)*, de modo que la población contenga soluciones cada vez mejores. El AG ha sido utilizado satisfactoriamente como herramienta de búsqueda de óptimos globales [6-8], debido a su relativa sencillez de implementación y la robustez de su aplicación. Adicionalmente, su implementación paralela resulta sumamente sencilla y eficiente [9], especialmente en el contexto asíncrono de las redes de computadoras existentes en la actualidad.

```
t ← 0;
Iniciar Población ( t );
Evaluar Población ( t );
DO WHILE ( No se cumpla Criterio de Parada )
      t ← t + 1;
      Seleccionar Población ( t ) a partir de Población ( t – 1 );
      Cruzar y mutar Población ( t );
      Evaluar Población ( t );
END DO
```

Pseudocódigo 2: Algoritmo Genético**.**

El método más natural para mejorar el tiempo de respuesta del AG (Pseudocódigo 2), es la paralelización del mismo [13-14]. En efecto, la población de *individuos* puede descomponerse en subpoblaciones, asignadas a diversos procesadores posiblemente heterogéneos, que periódicamente intercambian *individuos* entre sí de acuerdo a una política migratoria definida, en nuestra implementación, por los siguientes parámetros:

- El *intervalo de migración*: establece cada cuantas generaciones se realizará la migración de una cierta cantidad de individuos desde una subpoblación a otra.
- La *tasa de migración*: que indica cuantos individuos han de comunicarse a la otra subpoblación.
- El *criterio de selección de migrantes*: determina la política que se aplicará para la selección de los individuos que han de migrar. Por ejemplo, se los puede elegir al azar o entre los mejor adaptados de la subpoblación considerada.
- La *topología de comunicación*: define el sentido de las migraciones, es decir, el o los procesadores destino de la migración.

Otro concepto inherente a una eficiente paralelización en un ambiente distribuido heterogéneo, es el *balanceamiento de cargas* [2]. En efecto, como los diversos procesadores de un sistema distribuido pueden tener capacidades diferentes de procesamiento, es sumamente importante que las

subpoblaciones tengan los tamaños relativos adecuados, proporcionales al desempeño de los procesadores que los albergarán. Desafortunadamente, este balanceamiento de carga es muy difícil de lograr.

Un alivio a este problema lo constituye la *implementación asíncrona* [2] que permite a cada subpoblación realizar tantas *generaciones* como sean necesarias, comunicando eventualmente sus mejores *individuos* a otros procesadores e incorporando en su subpoblación los migrantes que van llegando, formándose de esta manera una población extendida. En este contexto, se pierde el concepto global de generación, pues algunos procesadores podrán realizar más iteraciones que otros, en el mismo periodo de tiempo.

3. Implementación del *A-Team*

Debido al ingrediente de aleatoriedad del AG y sus dificultades en atender la restricción (2), el presente trabajo propone aplicar dicho algoritmo sólo en algunas turbinas, aplicando el método numérico en las unidades restantes, asegurando así el cumplimiento de las restricciones del problema. De esta forma, el espacio de búsqueda exhaustiva para el método numérico es reducido considerablemente, quedando el AG como responsable por encontrar las *regiones prometedoras* ha ser exploradas en detalle por el método numérico. De esta forma, se espera obtener las ventajas combinadas de ambos métodos, reduciendo considerablemente el tiempo de ejecución, de forma similar al *A-Team* propuesto en [3].

La función objetivo que se busca optimizar en una usina, está dada por la eficiencia η en la generación de energía [1], conforme**:**

$$\eta = \frac{P_D}{\delta \times h \times \sum Qi} \qquad \textbf{(4)}$$

donde**:**

Q_i caudal turbinado en la unidad generadora *i*,
$\Sigma\, Q_i$ Caudal Total turbinado por todas las turbinas en servicio,
δ peso específico del fluido turbinado.

Los valores extremos de la función objetivo están muy próximos entre sí, lo que limita el desempeño del operador de Selección. Para corregir esto, se *escala* la función objetivo [4]. A esta función escalada se le llama función de adecuabilidad, o *fitness f*. El presente trabajo utiliza un escalamiento lineal dado por**:**

$$f = a \times \eta + b \qquad \textbf{(5)}$$

El AG implementado, representa un *individuo* como una estructura de datos, formada por las siguientes variables**:**

- la codificación binaria de las potencias correspondientes a las turbinas asignadas al Algoritmo Genético (que denominamos: *cromosoma*);
- los valores reales de dichas potencias: P_i ($i \leq N_T$). Por lo tanto, las

potencias P_i son las variables independientes de la función objetivo;

- los valores de caudal para cada turbina: Q_i ($i \leq N_T$), obtenidos a partir de los valores de potencia P_i en función de la ecuación (3);
- la eficiencia η, en la generación de la energía.

Para cada *individuo*, el AG calculará la potencia de k unidades ($k < N_T$) y el método numérico de las ($N_T - k$) unidades restantes, sirviendo de complemento para el cálculo de la eficiencia η. De esta forma, se asegura el cumplimiento de las restricciones del problema, quedando conformado un *A-Team* que trabaja de la siguiente manera:

a) Para cada *individuo j* de la población inicial, se generan aleatoriamente todas sus potencias P_i ($0 < i \leq k < N_T$); siendo P_i la generación de la unidad i (asignada al AG).

b) Para cada *individuo j* de la población inicial, se aplica el método numérico, de forma a calcular las potencias P_i ($k < i \leq N_T$). Dicho cálculo se realiza asignando a las ($N_T - k$) turbinas restantes, una potencia deseada de generación de ($P_D - \Sigma P_l$) donde ($0 < l \leq k$).

c) Conocidas las N_T potencias P_i, se calcula la eficiencia η usando las ecuaciones (3) y (4), y luego el *fitness f* conforme (5).

d) Si no se cumple el criterio de parada del algoritmo, se transmiten los migrantes seleccionados, se reciben migrantes de otros procesadores y se aplican los operadores genéticos, siendo el operador de selección el encargado de mantener constante el tamaño de la población.

e) El proceso continua hasta que se cumple el criterio de parada. Entonces, el *A-Team* proporciona el mejor *individuo*. Eventualmente, la solución obtenida puede ser mejorada aplicando el método del Gradiente a la solución proporcionada por el *A-Team*.

```
Leer Datos;
Levantar Procesos Esclavos;
Enviar Parámetros a cada Esclavo;
Fin ← FALSE;
DO WHILE ( Fin )
  Recibir Mensaje Terminación de Esclavos;
  IF ( Todas las máquinas terminaron ) THEN   Fin ← TRUE
END DO
Enviar Mensaje de Fin a Esclavos;
Eliminar Procesos Esclavos;
```

Pseudocódigo 3: Proceso *Master*.

La implementación asíncrona realizada está compuesta por un proceso *Master* (Pseudocódigo 3) que se encarga de administrar los recursos, incluyendo el lanzamiento de diversos procesos *Esclavos* en cada uno de los procesadores disponibles. Estos procesos *Esclavos* realizan los cálculos propiamente dichos, conforme se ilustra en el Pseudocódigo 4.

```
Recibir Parámetros;
Iniciar Población;
Estadística de la Población;
Selección de Individuos;
DO WHILE ( TRUE )
    Reproducción;                    /*Cruzamiento y Mutación*/
    Cálculo fitness de individuos;   /*Usando Método Numérico del*/
                                     /*Pseudocódigo 1 para cumplir ( 2 )*/
    Escoger y Enviar Migrantes;      /*a otros procesos Esclavos*/
    Recibir Migrantes;               /*de otros procesos Esclavos*/
    Seleccionar Individuos;          /*Manteniendo tamaño de Población*/
    Estadística de la Población;
    IF ( Criterio de Fin ) THEN
        Aplicar Gradiente ( Individuo solución );
        Mensaje al Master;
    END IF
END DO
```

Pseudocódigo 4: Proceso *Esclavo*.

En la implementación asíncrona, cuando una subpoblación recibe migrantes se forma una población extendida sobre la que se aplica el Operador de Selección de tal forma a seleccionar en cada *generación* la misma cantidad de *individuos*, manteniendo constante el tamaño de la subpoblación correspondiente [13].

4. Resultados experimentales

El problema ejemplo considera una represa hidroeléctrica con 9 unidades generadoras que pueden generar entre 300 MW y 710 MW cada una, y curvas de eficiencia obtenidas experimentalmente a partir de un "*index-test*". Los resultados que siguen fueron calculados para h = 116,5 m. El lenguaje de programación utilizado en los algoritmos secuenciales fue el ANSI C, utilizándose PVM (*Parallel Virtual Machine*), en la versión extendida del ANSI C, para la codificación paralela del *A-Team*.

Los resultados experimentales fueron levantados en una plataforma de 5 computadoras personales en una red *Ethernet* (10 Mbps), cada una con procesador Pentium de 75 MHz y 8 MB de memoria RAM. Se realizaron 20 corridas de la combinación secuencial de AG y método numérico (que llamaremos AG combinado), así como del *A-Team*, promediándose los resultados.

Para el AG combinado se consideró una población de 200 *individuos*. El modelo del Operador de Selección que se aplicó sobre dicha población es el de la *Ruleta* [6]. La probabilidad de cruzamiento utilizada fue de 0.6, siendo la probabilidad de mutación igual a 0.003. Esta población global fue

dividida en 4 subpoblaciones de 50 *individuos* cada una, para la implementación del *A-Team*, quedando la quinta máquina como *Master*.

Los resultados experimentales fueron obtenidos para diferentes valores de potencia demandada (P_D), utilizándose la siguiente nomenclatura para la descripción de los resultados obtenidos en las tablas que siguen:

- MN ..... Método Numérico utilizado tradicionalmente (sin AG).
- AG+MN .... implementación secuencial que combina el AG y el MN.
- η_{MN} .....eficiencia obtenida por el MN.
- t_{MN} ..... tiempo total de ejecución del MN para obtener η_{MN}.
- η_{prom} ..... promedio de la eficiencia en las 20 corridas.
- t_{prom} ..... promedio de tiempo en las 20 corridas.
- σ_{t_AG+MN} ..... desviación típica de tiempos de cómputo con AG+MN.
- $\eta_{max\text{-}AG}$ ..... eficiencia máxima obtenida por el AG+MN.
- σ_{AG+MN} ..... desviación típica de $\eta_{max\text{-}AG}$ con AG+MN.
- $\eta_{max\text{-}A\text{-}Team}$ .. eficiencia máxima obtenida por el *A-Team*.
- $\sigma_{A\text{-}Team}$ ..... desviación típica de $\eta_{max\text{-}A\text{-}Team}$ con *A-Team*.
- $\sigma_{t\_A\text{-}Team}$ ..... desviación típica de tiempos de cómputo con *A-Team*.

Algoritmo	**Variable**	**$N_T = 3$**	**$N_T = 4$**	**$N_T = 5$**	**$N_T = 6$**	**$N_T = 7$**	**$N_T = 8$**	**$N_T = 9$**
MN	η_{MN}	85.878	85.878	85.878	85.878	85.878	85.878	–
Intensidad 9	t_{MN} (seg.)	0.1	0.56	4.17	43.63	1040.92	56211.02	Muy Grande
MN	η_{MN}	85.964	85.964	85.964	85.964	85.964	85.964	–
Intensidad 11	t_{MN} (seg.)	0.14	2.04	15.05	136.59	3688.51	136576.3	Muy Grande

Algoritmo	**Variable**	**$N_T = 3$**	**$N_T = 4$**	**$N_T = 5$**	**$N_T = 6$**	**$N_T = 7$**	**$N_T = 8$**	**$N_T = 9$**
AG+MN	$\eta_{max\text{-}AG}$	85.718	85.991	85.991	85.991	85.991	85.991	85.991
	η_{prom}	85.718	85.991	85.991	85.991	85.908	85.987	85.983
	σ_{AG}	2.6×10^{-5}	2.6×10^{-5}	0.00243	0.00243	0.00243	0.00377	0.00292
	t_{prom} (seg.)	13.61	38.256	128.256	566.71	2984.1	9018.02	9058.38
	σ_{t_AG+MN}	0.1805	0.5553	0.4347				
A-Team	$\eta_{max\text{-}A\text{-}Team}$	85.991	85.991	85.991	85.991	85.991	85.991	85.991
	η_{prom}	85.989	85.989	85.988	85.989	85.988	85.990	85.984
	$\sigma_{A\text{-}Team}$	0.00286	0.00351	0.00630	0.00453	0.00364	0.00317	0.00708
	t_{prom} (seg.)	8.663	16.502	75.982	171.46	223.488	1313.89	3500.18
	$\sigma_{t\_A\text{-}Team}$	1.687	7.430	19.143	38.659	52.911	470.856	3135.408

Tabla 1: Resultados experimentales para $P_D = 1.500$ MW.

En las tablas que siguen, el término *Intensidad* indica el número de subdominios por dimensión, introducida como parámetro para la formación de la CUADRÍCULA (espacio de búsqueda) del MN. La dimensión del espacio de búsqueda está dada por el número N_T de turbinas en servicio. Lógicamente, a medida que aumenta la *Intensidad*, la solución (eficiencia) obtenida mejora y en contrapartida aumenta el tiempo de computación.

Al considerar el tiempo de computación para obtener soluciones similares, resulta evidente que el método numérico presenta mejor desempeño cuando la dimensión del problema es suficientemente pequeña.

Algoritmo	Variable	$N_T = 3$	$N_T = 4$	$N_T = 5$	$N_T = 6$	$N_T = 7$	$N_T = 8$	$N_T = 9$
MN *Intensidad 9*	η_{max}	–	87.710	87.710	87.710	87.710	87.710	–
	t_{max} (seg.)	–	0.32	14.04	176.01	1186.54	22137.4	Muy Grande
MN *Intensidad 11*	η_{max}	–	87.906	87.992	87.992	87.992	87.992	–
	t_{max} (seg.)	–	0.52	21.05	676.88	3886.14	61608.8	Muy Grande

Algoritmo	Variable	$N_T = 3$	$N_T = 4$	$N_T = 5$	$N_T = 6$	$N_T = 7$	$N_T = 8$	$N_T = 9$
AG+MN	$\eta_{max\text{-}AG}$	–	87.882	87.910	87.910	87.910	87.910	87.910
	η_{prom}	–	87.881	87.893	87.893	87.906	87.904	87.909
	σ_{AG}	–	5.9×10^{-5}	0.00265	0.0378	0.0165	0.0124	0.0028
	t_{prom} (seg.)	–	18.231	33.654	63.138	149.628	854.17	3037.52
	σ_{t_AG+MN}	–	0.0702	0.1111				
A-Team	$\eta_{max\text{-}A\text{-}Team}$	–	87.964	87.992	87.992	87.992	87.992	87.994
	η_{prom}	–	87.964	87.991	87.970	87.987	87.974	87.991
	$\sigma_{A\text{-}Team}$	–	0.0001	0.00290	0.0565	0.0170	0.0350	0.0030
	t_{prom} (seg.)	–	12.443	19.421	32.199	106.465	370.921	1759.30
	$\sigma_{t\_A\text{-}Team}$	–	5.401	13.480	13.577	57.940	136.044	498.618

Tabla 2: Resultados experimentales para $P_D = 2.550$ MW.

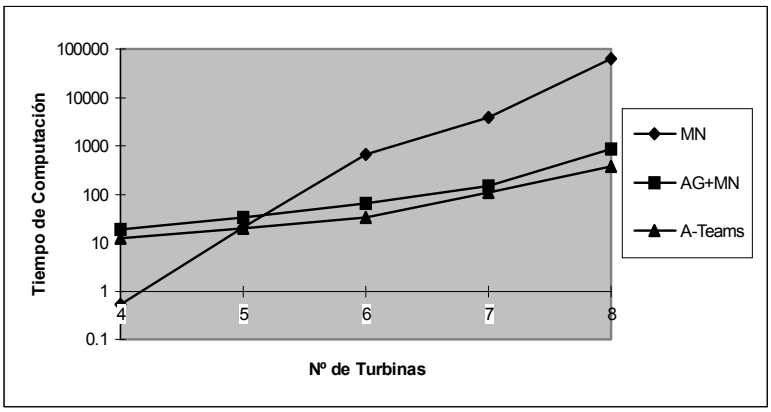

Figura 1: Tiempo vs. N° de Turbinas para $P_D = 2550$ MW.

Si analizamos la *calidad* de la solución (eficiencia obtenida), podemos notar que la combinación de métodos permite obtener muchas veces soluciones superiores a las obtenidas con el MN para la misma *Intensidad* de búsqueda, aunque en promedio, las soluciones obtenidas resultan bastante similares. De entre las implementaciones combinadas, el *A-Team* supera generalmente al AG combinado puramente secuencial, reiterándose el efecto positivo que tiene el asincronismo sobre la diversidad de la población [10].

Sin embargo, cuando la dimensión N_T del espacio de búsqueda aumenta, el problema se vuelve más complejo y con esto, el desempeño del *A-Team* supera ampliamente a los demás métodos considerados (Tablas 1 y 2). En efecto, la Figura 1 muestra que claramente, para $N_T > 5$ el *A-Team* es más rápido que el método numérico, para soluciones equivalentes, y esta ventaja tiende a crecer rápidamente con el tamaño del problema.

Para el presente trabajo, se define la Aceleración (*speedup)* como el cociente entre el tiempo de ejecución secuencial del MN (tradicionalmente utilizado) y el tiempo de ejecución del *A-Team*, para soluciones equivalentes [2], esto es:

$$Sp = \frac{T_{MN}}{T_{A\ -\ Team}} \qquad (6)$$

Comparando la aceleración para diferentes valores de Potencia Demandada P_D (Figura 2), se puede observar una vez más la superioridad del *A-Team* en la medida que crece la complejidad del problema. Así, la aceleración aumenta no solo con el tamaño del problema, sino también con el valor de P_D. Esto porque en la medida que se demande mayor generación de energía, más turbinas deben entrar a operar efectivamente, e inclusive algunas pueden necesitar operar en condiciones extremas de generación en alta carga, con alta alinealidad. Por consiguiente crece la complejidad del problema, lo que crea un escenario apropiado para la implementación de un *A-Team*.

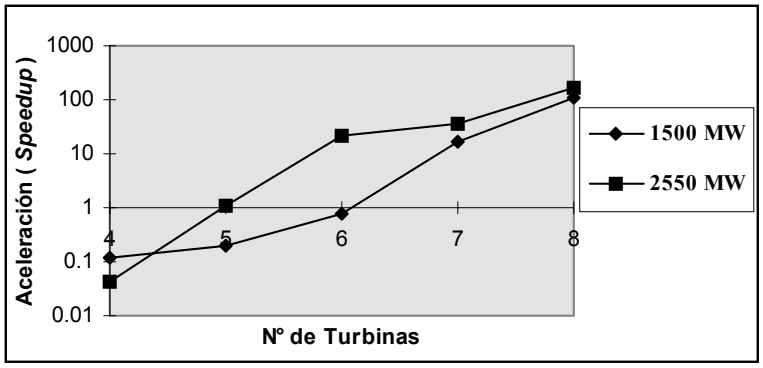

Figura 2: Aceleración (*Speedup*) vs. N° de Turbinas.

5. Conclusiones

De acuerdo a los resultados obtenidos, el *A-Team* que combina Algoritmos Genéticos Paralelos y tradicionales Métodos Numéricos, mejora considerablemente el tiempo de cálculo requerido para optimizar la eficiencia en la generación de energía hidroeléctrica, cuando es comparado con el método numérico trabajando secuencialmente. Esta mejora se acrecienta en la

medida que la dimensión del espacio de búsqueda (equivalente al número de turbinas en servicio) crece, siendo perceptible a partir de unas 5 turbinas, número muy inferior al utilizado en represas hidroeléctricas como la Itaipú, que posee 18 unidades en operación con capacidad de expansión futura. Sin embargo, el método numérico presenta mejor desempeño que el *A-Team* cuando el problema es simple, como se puede observar en las Tablas de la sección anterior. Con todo, teniendo en cuenta el tamaño de las represas hidroeléctricas consideradas, el *A-Team* propuesto promete ser una herramienta sumamente útil en la optimización de los recursos hídricos utilizados para la generación de energía.

Una ventaja adicional de la propuesta es su facilidad de paralelización, utilizando implementaciones asíncronas que enriquecen la diversidad de la población del AG y permiten así aprovechar al máximo las redes locales de computadoras personales y *workstations* existentes en las organizaciones modernas, en lugar de utilizar computadoras de alto desempeño más costosos y menos accesibles. En efecto, la Figura 1 muestra los excelentes resultados alcanzados con una implementación en una red de computadoras personales, lo que resulta sumamente prometedor, alentando otras aplicaciones combinadas de algoritmos diversos en ambientes distribuidos, cada vez más populares y accesibles.