HYDROELECTRIC OPTIMIZATION USING *A-TEAMS*

Benjamin Baran, Enrique Chaparro Viveros and *Nestor Caceres*
National Computing Center, National University of Asuncion
P.O. Box 1439, San Lorenzo – Paraguay
http://www.cnc.una.py/invest/
E.mail: bbaran@cnc.una.py

Abstract: Paraguay exports electricity thanks to the generation of several large scale hydroelectric plants as Itaipu, Yacyreta and others. The optimization of the generation process is very complex, due to the large number of generation units (up to 18 in Itaipu) with different physical characteristics, operational and maintenance restrictions and so forth. The purpose of this work is to propose a technique that optimizes the use of water resources, using a distributed computer network existing in most organizations.

As a consequence of the complexity of the problem, several numerical algorithms were developed, each one for a different operating scenario. In a new approach to solve complex problems, *Team Algorithms* were proposed as combinations of different methods working together to solve the same problem. In particular, a parallel combination in a distributed asynchronous computing environment, such as a computer network, is receiving a great deal of attention with the name of *A-Team*. Using this concept, the present work proposes a synergetic combination of traditional numerical methods (NM) with Parallel Genetic Algorithms (PGA), getting an *A-Team* capable of finding very good solutions, outperforming methods in use nowadays. Experimental results are presented showing the advantages of using *A-Teams* when both the complexity and the size of the problem increase.

Keywords: Parallel Genetic Algorithms, *A-Teams*, *Team Algorithm*, Hydroelectric Plant.

1. Introduction

Today Paraguay has two large scale hydroelectric plants: Itaipu (with 18 turbines and 12600 MW of power generation) and Yacyreta (with 20 turbines and 3500 MW). The optimization process required demands a high computational load, because the problem's complexity grows up exponentially with the number of turbines involved.

Electrical power generation using hydroelectric plants is a well studied field and today several mathematical models exist allowing to study the generation and its associated transmission [1]. The present problem consists in minimizing the turbined flow (Q) in a hydroelectric power station, given the demanded power (P_D) and the number of generators (N_T).

The solution is not trivial, because each turbine has its own operative restrictions characterized by the way in which the turbined flow (Q) turns into useful electrical power (P_i) and the height (h) of the water in the reservoir. The powers (P_i) generated by each one of the turbines and the flows (Q_i) required in each one of them are related by the functions (f_i). These are experimentally obtained by "*index-test*".

$$P_i = f_i (Q_i, h) \qquad \textbf{(1)}$$

To solve the presented problem there are several optimization methods [1,2], each one of them showing advantages and disadvantages depending on the peculiar characteristics of each particular problem. However, it could happen that none of the chosen methods is able to solve the considered problem.

A strategy to overcome the presented inconveniences is to split a big complex problem in minor subproblems, and to solve each one of them using an advantageous method according to the characteristics of each subproblem. This combination of several algorithms working together towards the solution of a same problem is known as *Team Algorithm* [3].

The introduction of both the concept of parallelism, i.e., the simultaneous utilization of several processors in a problem's resolution, and, in this context, the asynchronous-like data transfer between processors, constitute a valid alternative in order to reduce the computing time. The resulting combination of algorithms in an asynchronous environment is known as *A-Team* (*Asynchronous Team*), which has a proved performance in several disciplines [4–6].

The present work explains the integration in an *A-Team* of: a well-known sequencial numeric optimization method, like the Gradient Method, and a distributed version of Genetic Algorithms, similar to the one proposed in [6]. This *A-Team* is used to minimize the turbined flow in a large scale hydroelectrical plant.

Section 2 describes the combined algorithms. The strategies used in order to combine the algorithms are explained in section 3. Experimental results and conclusions are presented in sections 4 and 5, respectively.

2. Optimization methods

This section describes the numerical methods commonly used to solve the considered optimization problem, as well as the methodology followed in order to apply Genetic Algorithms to the same problem.

2.1 Traditional Numerical Method

The numerical methods traditionally used in order to solve the problem concerning us are typically made up of an exhaustive search algorithm and a local optimization method. The denomination 'local', when talking about numerical optimization methods, refers to the fact that these methods usually have a quick convergence to a solution, although many times this solution does not correspond to a global optimum, but to an optimum which is in the neighbourhood of the initial point. Such is the case of the Gradient Method, the Newton Method and others [7].

In the numerical methods used, the global domain of search is divided, using a Grid, into small monotonous subdomains. The search of a local optimum is carried out in each one of these subdomains. Then, when the analyzed points satisfy a certain condition, a variant of the Gradient Method is applied to these points. The best solution between all the subdomains' solutions is the problem's solution. This method requires a high overhead, since the dimension of the search space is equal to N_T and, in order to increase the quality of the solution, it is also necessary to increase the amount of subdomains. This is the reason why the Gradient Method is only applied on those "promising" points which fulfill the problem's restriction:

$$\sum_{i=1}^{N_T} P_i = P_D \qquad (2)$$

2.2 Genetic Algorithm

The Genetic Algorithms have been inspired by the natural mechanisms of selection and genetics [8]. In these algorithms, an initial population of possible solutions (called *individuals* in analogy with biology [8]) evolves by applying probabilistic operators (*Selection, Crossover and Mutation*) in such a way that the population contains every time better solutions.

The *Selection* operator uses a function called objective function in order to evaluate the goodness of the solution represented by each *individual*. The efficiency η in the generation of energy in a Hydroelectric Plant is the objective function to be optimized:

$$\eta = \frac{P_D}{\delta \times h \times \sum Q_i} \tag{3}$$

where**:**

Q_i turbined flow in the generation unit i,
ΣQ_i total turbined flow with all the turbines in service,
δ specific gravity of the turbined fluid.

The *Selection*, *Crossover* and *Mutation* operators provide the mechanisms to generate offspring beginning with preexisting *individuals*, therefore allowing new regions of the search space to be explored, obtaining in this way the global optimum.

Besides being easy to implement, an outstanding characteristic of the GA is that it is easily parallelized. Indeed, the population of *individuals* can be divided in subpopulations, and these are assigned to different processors that periodically exchange certain amount of *individuals* (called *migrants* [9]).

The fitting of load is something very important to keep in mind, i.e., the size of the subpopulations must be proportional to the computational power of the processors to which those subpopulations have been assigned.

3. *A-Team* (*Asynchronous Team*)

Due to the randomness of the GA, it turns out difficult to satisfy restriction (2). Consequently, the present work proposes to solve the problem in the following way:

- The GA will calculate the P_i power to be generated by the first k generating units, where $\forall\ i \in \{1, ..., k\}: k < N_T$.
- The NM will receive as parameter the remaining required power: $P_D - \Sigma P_i$, to determine the P_i powers, where $\forall\ i \in \{k + 1, ..., N_T\}$
- Known all the P_i Powers, the efficiency η is calculated using equations (1) and (3).

In this way it is achieved, on the one hand, that all the analyzed points satisfy the problem's restriction, and, on the other hand, with adequate values of k a good performance of the NM is maintained upon reducing the search space to the one in which the search is accomplished. However, the performance of the selection operator is limited due to the fact that, according to the definition in [3], the extreme values of η are too close each other. For this reason the objective function is scaled [8] using a linear scaling function.

For the paralelization of the method, the utilization of the *Master-Slave* model [9] is proposed. The *Master* process is entrusted with the administration of the resources, including launching and ending *Slave* processes in each one of the available processors. The *Slave* processes are entrusted with the accomplishment of the fittingly said calculations, and with the transmission and reception of information to and from other *Slaves*. Pseudocodes 1 and 2 describe the operations accomplished by processes *Master* and *Slave*, respectively.

```
Read Data;
Spawn slave processes;
Send parameters to each slave;
End ← FALSE;
DO WHILE ( End )
    Receive end messages from slaves;
    IF ( All the machines ended ) THEN    End ← TRUE
END DO
Send end messages to slaves;
Eliminate slave processes;
```

Pseudocode 1: *Master* Process

```
Receive parameters;
Begin population;
Calculate Statistics of the Population;
Select individual;
DO WHILE ( TRUE )
  Reproduction;                    /*Crossover and Mutation*/
  Individual´s fitness evaluation; /*Applying Numerical Method in A-Team */
                                   /* to fulfill the (2) constraint */
  Choose and send Migrant;         /*to other Slave processes*/
  Receive Migrant;                 /*from other Slave processes*/
  Select Individual;               /*Maintaining the size of the Population*/
  Calculate Statistics of the Population
  IF (End criteria) THEN
          Apply Gradient ( Individual solution );
          Send End Menssage to the Master;
  END IF
END DO
```

Pseudocode 2: *Slave* Process executed in each processor of the Distributed System

4. Experimental Results

The experimental results, for the *A-Team*, were obtained with a platform of 5 personal computers (Pentium), having each one of them 8 MB of RAM and 75 MHz, on an Ethernet network (10 Mbps). The global population, with 200 individuals, was split into 4 sub populations of 50 individuals. Each subpopulation was assigned to each one of 4 PC's, remaining the fifth computer as the manager (*Master*). For the combination of the sequential GA with the numerical method (that we will call combined GA) as well as for the *A-Team*, 20 bullfights were accomplished, with the results being averaged.

As an example we considered a hydroelectric dam with 9 turbines (each one of them generating from 300 MW up to 710 MW of electrical power). The following results were calculated using h = 116,5 m.

Despite the fact that numerous experiments with different values of P_D have been accomplished, due to the limited space only results corresponding to P_D = 1500 MW are shown.

For the description of the results the following nomenclature is used:

NM	numerical method (without GA);	η_{prom}	efficiency average
GA+NM ...	sequential combination of GA and NM;	t_{prom}	time average
η_{max}	maximum efficiency obtained by the NM;	$\eta_{max\text{-GA}}$	max.efficiency obtained by the GA+NM
t_{max}	total time of the NM to obtain η_{max};	$\eta_{max\text{-}A\text{-}Team}$	max. efficiency obtained by the A-TEAM

Analyzing the quality of the solution (obtained efficiency), we observe in Table 1 that the average results of both the GA+NM and the *A-Team* are better than those obtained by the NM. Furthermore, the *A-Team* presents a speedup considerably greater than the rest of the sequential implementations, as it is appreciated in Table 1, showing in this way the advantages of using a computer network to solve problems of great freightage, such as the one presented here.

Algorithm	Variable	$N_T = 3$	$N_T = 4$	$N_T = 5$	$N_T = 6$	$N_T = 7$	$N_T = 8$	$N_T = 9$	AVERAGE
NM	η_{opt}	85.964	85.964	85.964	85.964	85.964	85.964	-	85.964
	t_{total} (seg.)	0.14	2.04	15.05	136.59	3688.51	136576.3	Too Big	23403.097

Algorithm	Variable	$N_T = 3$	$N_T = 4$	$N_T = 5$	$N_T = 6$	$N_T = 7$	$N_T = 8$	$N_T = 9$	AVERAGE
GA+NM	η_{max}	85.818	85.991	85.991	85.991	85.991	85.991	85.991	85.962
	η_{prom}	85.718	85.991	85.991	85.991	85.908	85.987	85.983	85.931
	t_{prom} (seg.)	13.61	38.256	128.256	566.71	2984.1	9018.02	9058.38	2124.825
A-Team	η_{max}	85.991	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	85.989
	t_{prom} (seg.)	8.663	16.502	75.982	171.46	223.488	1313.89	3500.18	301.664

Table 1: Experimental results for PD = 1500 MW.

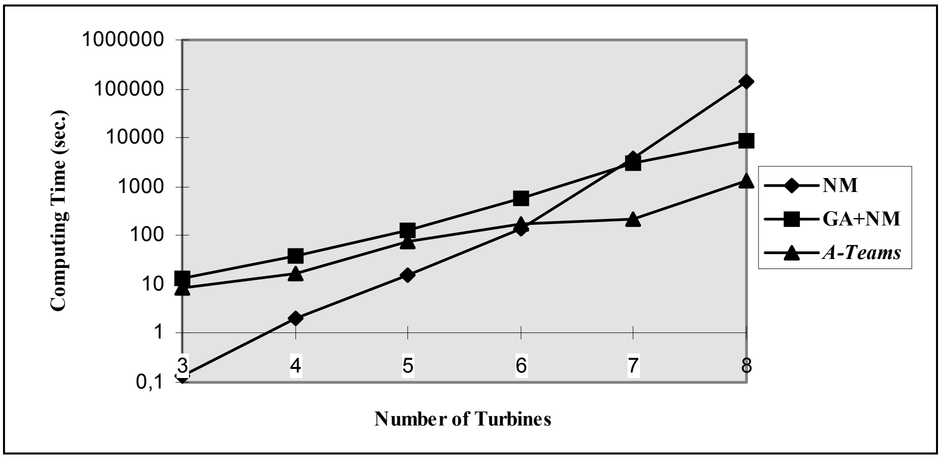

Figure N° 1: Time vs. Number of Turbines for P_D = 1500 MW.

Upon considering the computation time to obtain similar solutions, the NM presents better performance when the dimension of the problem (number of turbines in service) is small. However, when the dimension N_T of the search space increases, the problem is more complex and the *A-Team* outperforms widely the other considered methods (Table 1). Indeed, Figure N° 1 shows that for $N_T \geq 6$ the *A-Team* is faster than the NM for equivalent solutions, and this advantage increases quickly with the size of the problem.

5. Conclusions

According to the results obtained, the *A-Team* reduces considerably the calculation time required to optimize the efficiency in hydroelectrical energy generation, when it is compared with the NM. This improvement grows up as the search space (number of turbines in service) increases, being noticeable when $N_T = 6$, number a lot smaller than the ones used in Hydroelectric Dams like Itaipú, with 18 turbines, and Yacyretá, with 20 generating units. However, the NM has better performance than the *A-Team* when the problem is simple (see Table 1). Thus, we can observe also in Table 1 that the proposed *A-Team* presents average results that are better than the ones obtained by the sequential implementations, and, taking into account the size of the considered Hydroelectric Dams, the *A-Team* promises to be a useful tool in the optimization of the resources used for electrical energy generation.

6. Acknowledgment

This work has been partially supported by a grant of the Research, Postgraduate and International Relationships Department (DIPRI) of the National University of Asuncion (UNA). At the same time, the authors want to thank the support received from the UNA Engineering Faculty.

7. References

[1] Wood A. J. & Wollenberg B. F., "*Power Generation, Operation and Control*". John & Sons, 1983.

[2] Falcão D. M., Bomfim A. L. B., Dornellas C. R. R. & Taranto G. N. "Genetics Algorithms in Power Systems Optimization", *V SEPOE*, Recife – Brasil, 1996.

[3] Barán B., Kaszkurewicz E. & Bhaya A., "Paralell Asyncrhonous Team Algorithms: Convergence and Perfomance Analysis". *IEEE Transactions on Paralell and Distribuited Systems*, Vol. 7, N° 7, 1996.

[4] Souza P. S. & Favilla J. R., "Asynchronous Teams for Steel Industry: Mini Mills". *ISAS´96*. Orlando – Florida, Estados Unidos, 1996.

[5] Barán B. & Chaparro E., "Algoritmos Asíncronos Combinados para una Red Heterogénea de Computadoras". *XXIII Conferencia Latinoamericana de Informática* (*CLEI-PANEL 97*). Valparaíso – Chile, 1997.

[6] Souza P. S. & Talukdar S., "Genetics Algorithms in Asynchronous Teams", *ICGA – 91*, pp. 392 – 397, San Diego – California, 1991.

[7] Press W. H., Flannery B. P., Teukolsky S. A. & Vetterling W. T., "*Numerical Recipes in C: The Art of Scientific Computing*", Cambridge University Press, 1988.

[8] Goldberg D. E., "*Genetic Algorithm in Search, Optimization & Machine Learning*". Addison Wesley, 1989.

[9] Mejía M. & Cantú E., "DGENESIS: Software para la Ejecución de Algoritmos Genéticos Distribuidos". *XX Conferencia Latinoamericana de Informática* (*CLEI-PANEL 94*). México, 1994.