Murdoch
UNIVERSITY

MURDOCH RESEARCH REPOSITORY

http://dx.doi.org/10.1049/cp:20000422

Fung, C.C., Chow, S.Y. and Wong, K.P. (2000) A low-cost parallel computing platform for power engineering applications. In: 5th International Conference on Advances in Power System Control, Operation and Management, 30 October - 1 November, Hong Kong, pp 354-358.

http://researchrepository.murdoch.edu.au/14405/

Copyright © 2000 IEEE

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Proceedings of the 5th International Conference on Advances in Power System Control, Operation and Management, APSCOM 2000, Hong Kong, October 2000.

A LOW-COST PARALLEL COMPUTING PLATFORM FOR POWER ENGINEERING APPLICATIONS

C.C. Fung **S.Y. Chow**

School of Electrical and Computer Engineering
Curtin University of Technology
GPO Box U1987 Perth
Western Australia 6845

K.P. Wong

Dept. of Electrical and Electronic Engineering
The University of Western Australia
Nedlands
Western Australia 6907

ABSTRACT

This paper develops and evaluates a low-cost parallel computing platform for the implementation of parallel algorithms in Power Engineering applications. The proposed approach utilises an existing local area network without incurring any additional hardware costs. Application of computational intelligence techniques based on the developed computing platform to the economic dispatch problem is outlined. The performance of genetic algorithms in parallel and cluster structures and their abilities in coping time constraint applications are also demonstrated. It is found that when the workload is large, a parallel computing structure should be exploited for cost-effectiveness purpose.

1. INTRODUCTION

This paper reports the development and evaluation of a low-cost computing platform for the implementation of parallel algorithms [1,2,3,5,8] in power engineering applications. In particular, the implementation of genetic algorithm techniques [4, 7] to the problem of economic dispatch [9,10,11,12] is used to demonstrate the proposed approach. Unlike dedicated multi-processors or high performance computing platforms, the proposed method is based on an existing Local Area Network (LAN) or Intranet. Examples are typical office or laboratory environments with multiple interconnected computers. Apart from exceptional circumstances when all the computers are fully utilised, most of them are either idling or not loaded to their full capacities. Hence, it is the intention of this study to explore the possibility of exploiting the combined processing power without the need of any extra specialised hardware installation.

Over the past years, Computational Intelligence (CI) techniques have been recognised as flexible yet powerful approaches for solving challenging engineering problems. Characteristics of computational intelligence techniques include their inherent massive parallelism, and their capability for distributed representation and computation. Theoretically, these CI algorithms could be incorporated with large number of interconnected simple processors to solve a variety of difficulty computational problems. CI techniques mainly comprise of Artificial Neural Network (ANN), Fuzzy logic and Evolutionary Computation, which bases primarily on Genetic Algorithms (GA).

Although investigation on the implementation of ANN on parallel structures has been carried out, this paper only reports the development and testing of parallel algorithms on GA's, incorporated with simulated annealing (SA) and Tabu Search (TS) methods. They are employed to seek global optimal solutions for the economic dispatch problem. As GA is highly suitable for parallel computation, its performance in a distributed computer network environment is evaluated. Two classes of parallel GA's have been assessed: a cluster structure and a parallel structure. In this paper, a test study on the determination of the optimal loading of 13 generators is solved using ten Pentium II-350 computers. They are configured as one master and nine slaves interconnected with TCP/IP as the backbone network protocol. Java programming is used for synthesising the above configuration due to its portability [2].

2. A PARALLEL COMPUTING PLATFORM

A JAVA-based command line driven program designed to facilitate the implementation of parallel algorithms over the defined structure has been developed. The program is called Para Worker. It can be operated both as a slave node and a master node with no reduction in functionality. Commands are issued to the program via the standard input and output streams. In the event that a node becomes a slave to another, the standard input and output streams are redirected to this other node, via the master-slave socket connection.

There are four distinct mechanisms that facilitate the implementation of parallel algorithms. These are the *Master Mechanism, Peer Mechanism, Class Transfer Mechanism,* and the *Message Passing Mechanism*, as shown in Figure 1. Each mechanism performs a fundamental function that facilities the development of parallel and distributed algorithms. They can be loaded into and unloaded from the Para Worker at will, and their presence is entirely dependent on the requirements of the parallel algorithm.

Three of these Mechanisms use TCP/IP Socket communication facilities. The other uses UDP/IP. This means that any number of computers can be connected to each other via the Internet setup. The distance may range from half a metre or even half a world away. Because the use of Internet and Intranet is growing prolifically around the world, the protocols used have become more standardized and are more appropriate to be used as the communications medium for the Para Worker. The Para Worker effectively turns an arbitrary number of networked desktop machines into a cluster of machines. One machine can be designated the master which is able to issue commands to all of its slave nodes. The computing structure is presented as a single computing entity with the capability to transparently grow in size as resources become available.

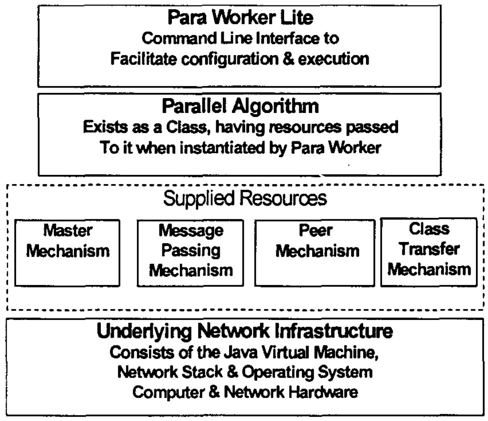

Figure 1: The Para Worker Hierarchy

Utilizing the multi-tasking capabilities of modern operating systems. The structure can operate as a cluster even while users are working at the slave machines. In order to reduce the effect of the extra loading due to this software, the Para Worker slaves can be given a relatively low priority during execution.

There are two versions of the Para Worker program that were developed in this project. Both are essentially the same, as the Para Worker Lite was based on the original Para Worker. The distinction between the two programs is that the Para Worker Lite can be considered as a lightweight component. All functionality, to as great an extent as possible, has been removed from the main file and placed into other classes.

3. GENETIC ALGORITHMS, SIMULATED ANNEALING AND TABU SEARCH

GAs are search algorithms that incorporate natural evolution mechanisms, including crossover, mutation and survival of the fittest. GA paradigms work on populations of individuals, rather than on single data point or vectors. They are often used for optimisation problems. GAs have been used in approaches for the determination of the global optimum solution in power engineering including economic dispatch, unit commitment, distributed planning & operation, reactive power planning and long-range generation expansion planning problems. Other methods such as SA and TS have also been combined with GA's to improve the solution process.

SA replicates the annealing process of molten metals. The metal is first heated to a very high temperature and then cooled slowly until it is solidified at a low temperature. In this study, the SA method is used as a probabilistic replacement test to evaluate the replacement of crossover and mutated genes in a set of chromosomes. Performance of GA's can be improved by introducing more varieties among the chromosomes in the early stage so that premature convergence of the solution can be eliminated.

TS is based on the premise that intelligent problem solving requires incorporation of adaptive memory [5]. TS is a global search technique in that it provides means to escape from local minima. In integrating TS with GAs, a finite list of forbidden chromosomes called the tabu list is maintained. As a new solution of chromosome is generated via crossover or mutation process, it is added to the list and the oldest member of the list is removed. Thus the tabu list prevents cycling by disallowing repetition of chromosomes in the population. Certain solutions are accepted even if they are in the tabu list, provided they satisfy an acceptance/aspiration criterion.

3.1 Parallel implementation of Genetic Algorithms, Simulated Annealing and Tabu Search

Application of GA to find an optimum solution involves the establishment of a large population size of chromosomes, such that the chances of obtaining the optimum solution by mixing the data (via crossover and mutation) is high. To facilitate the computation of a large populated data, cluster and parallel structures with the algorithms are investigated.

For cluster computing, a cube consisting of 9 processors (plus one master node) is used. After each specified generation, each node will communicate with its neighbouring node and sends its best chromosome to each neighbour. It also receives the best chromosome from the neighbour nodes. The received chromosome will be included in the regeneration process. This model of the cluster computation is illustrated in Figure 2.

For parallel computing, the overall population is divided among 10 processors to perform crossover and mutation. The data is then recombined to perform crossover and mutation. After that, it is sent back to the 10 processors for repeated manipulation of data until the maximum number of generations is reached. The model implemented for parallel structure is illustrated in Figure 3. Techniques of enhancement like SA and TS are incorporated in both the parallel and clustering structure.

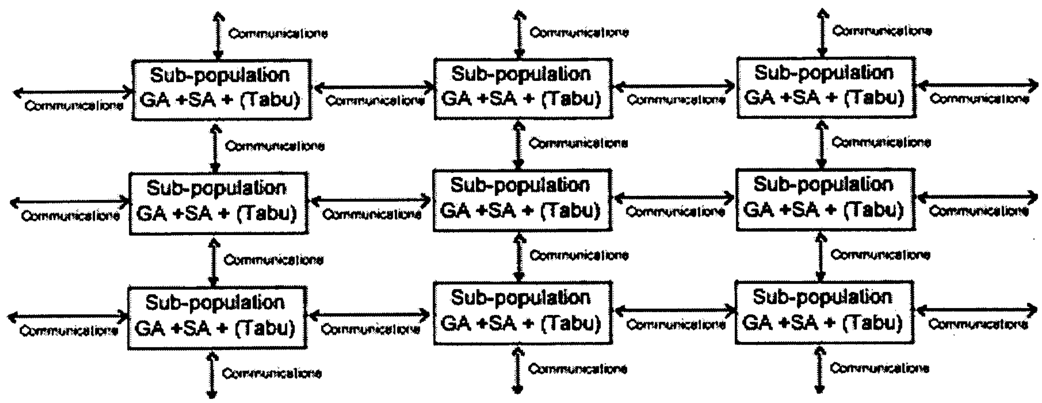

Figure 2: Cluster Structure (Coarse-grain) of parallel platform with a computer on each node

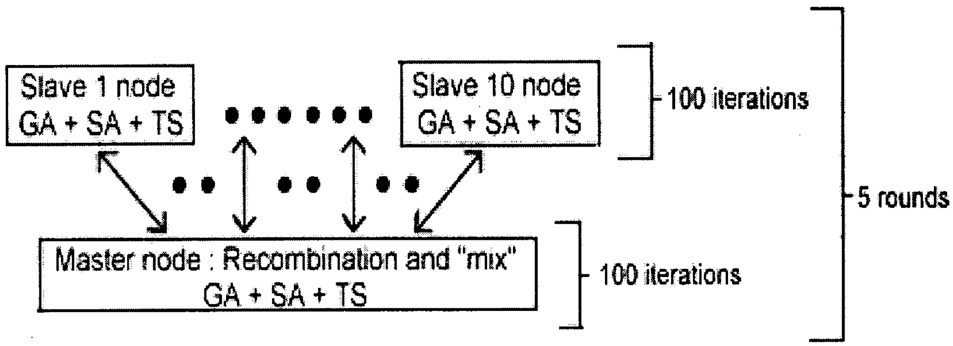

Figure 3: Model of Parallel Structure

3.2 The Economic Dispatch Problem

The principal objective in economic dispatch of thermal generators in a power system is to resolve the optimal loadings of the generators so that the load demand can be met and the loadings are within the feasible operating regions of the generators. The problem can be expressed as follows:

$$Minimise F_T = \sum_{i=1}^{n} f_i(P_i) \qquad (1)$$

where F_T = total fuel cost
n is number of online thermal generators,
$f_i(P_i)$ is the operation fuel cost of generator

The minimisation in Expression (1) is subject to the power balance constraint:

$$\sum_{i=1}^{n} P_i = D + P_L \qquad (2)$$

where D is the total load demand,
P_L is the total transmission loss

In addition, each generator also has to conform with the inequality constraints. This is due to the operation limits of the generators as shown below:

$$P_{i,\min} \le P_i \le P_{i,\max} \text{ for } 1,2,\ldots,n \qquad (3)$$

To implement in genetic algorithm, the loadings of the generator loadings are expressed as floating-point numbers. The crossover and mutation operations are outlined in equations (4) and (5) respectively.

$$P_d = D + P_L - \sum_{i=1, i \ne d}^{n} P_i \qquad (4)$$

$$P_i^{"} = P_{i,\max} - P_i + P_{i,\min} \qquad (5)$$

where $P_i^{"}$ is the mutated loading,
P_d is the dependent generator loading [11] and
$P_{i,\max}$ & $P_{i,\min}$ are the maximum operation limits and the minimum operation limits of the i[th] generators respectively.

In this study, the cost function is expressed as follows:

$$f(P) = c + b{*}P + a{*}P^2 + [e \sin[f(P_{min} - P)]$$

where a to f are coefficients of the heat-rate functions of the generators. In the test study, the coefficients used are shown in the following tables and they are adapted from Reference [11]. The chromosomes implemented for this GA consist of 13 genes which are the generator loadings. Each chromosome associates with a fitness value that is the total operation cost in equation (1). The probability of acceptance incorporated in the SA process is calculated from

$$\Pr(\Delta) = [1/\{1+\exp(\Delta/T)\}] \qquad (6)$$

where Δ is the amount of deterioration between old and new solutions and T is the temperature at which the new solution is generated.

At each iteration, the initial temperature is reduced by $r^{(k-1)}$ where k is the iteration number and r is the temperature reduction factor. In other words, the new solution (chromosome) is formed by perturbing the old solution according to the Gaussian probabilistic distribution function (GPDF). The mean of GPDF is the old solution and the standard deviation is the product of temperature and a scaling factor γ.

TABLE 1 - Coefficients of heat-rate functions of generators in Test system.

Ge	a	b	c	e	f
1	0.00028	8.1	550	330	0.035
2-3	0.00056	8.1	309	200	0.042
4-9	0.00324	7.74	240	150	0.063
10-13	0.00284	8.6	126	100	0.084

TABLE 2 - Generator Max and Min Loadings (in MW)

Generators	Min. loading	Max. loading
1	0	680
2-3	0	360
4-9	60	180
10-13	40	120

The acceptance criteria for the TS is a comparison between the current solution with the best "fitness value" which in this case is the total operation fuel cost. The solution process continues until the maximum number of generations is reached.

3.3 Description of the Combined algorithm

The implementation of GA with SA and TS is illustrated in Figure 4. The Tabu Search algorithm is also implemented as shown in flowchart as shown in Figure 5.

3.4 Test Study

In this case study, the following parameters have been used:

D = 2520MW
Population size =90 (for 9 computers)
Probability of crossover = 0.6
Probability of mutation =0.01
Total generations performed =1000
Scaling factor γ =0.01
Temperature reducing factor =0.98
Initial temperature =50000
Size of Tabu List = 100
Tolerance for acceptance criteria (since it is comparing floating point numbers) = within 5% of each value stored in the tabu list.

For clustering structure processing, at each 100 generations the best chromosomes is passed to its neighboring node.

For parallel structure processing, each slave node will process for 100 generations and the master will process for 100 generations and this is repeated for 5 rounds.

For all the structures, various methods of TS are evaluated:- No Tabu List (where only GA and SA is performed) , Local Tabu List (where each slave and master have its own Tabu List) and common Tabu List (each slave and master have a common Tabu List). All of the above methods utilize GA and SA.

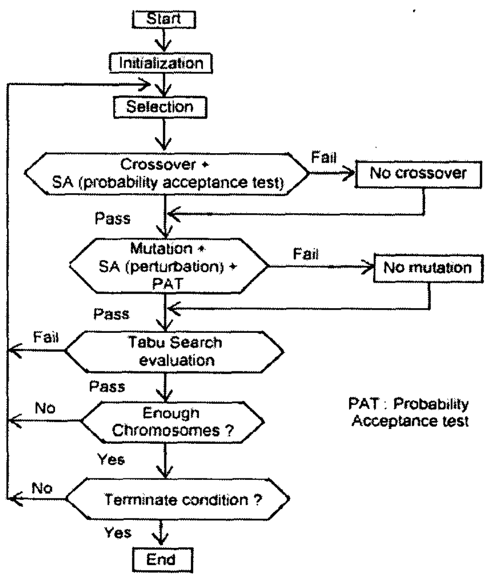

Figure 4: Integrated GA, SA and TS Flowchart

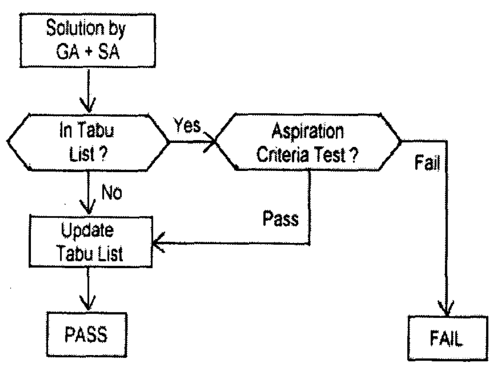

Figure 5: Tabu Search Flowchart

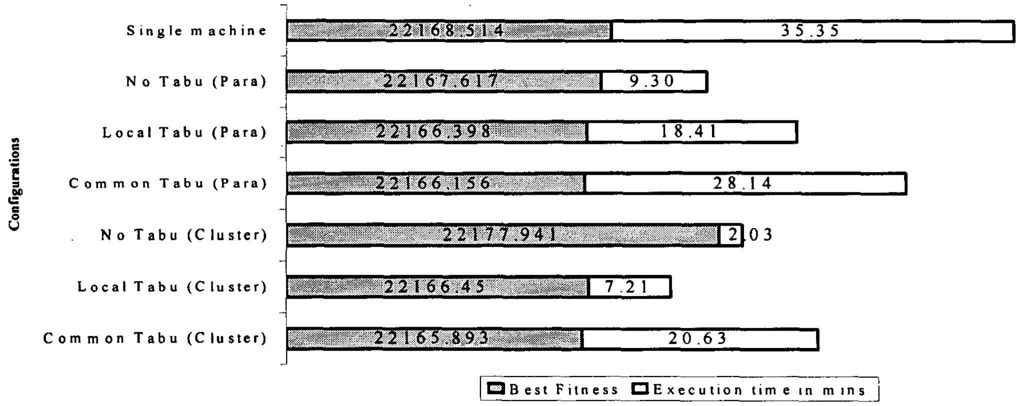

Figure 6: Performance comparison between Cluster, Parallel and Single machine structures

3.5 Results

The results of the best operation fuel cost obtained and the total execution time required are illustrated in Figure 6. Single machine has also been implemented with conventional GA, SA and TS. As observed from the above results, the algorithm using a common Tabu List produced the lowest best fitness value in both cluster and parallel structures. However, this advantage is traded off with the execution time required. This is anticipated as both communication time and synchronization to access the tabu list are substantial. Comparing with the single machine, the parallel processing structures have provided better performance in execution time and with the ability to find a better optimisation solution.

4. CONCLUSION

This paper has demonstrated the use of a low-cost Java-based computing platform for the implementation of parallel algorithms. In this paper, the problem of economic dispatch of 13 generators was solved. It is shown that performances in parallel structures can be heavily affected by the communication time and synchronization between the processors. If the problem space is huge, parallel structures indeed yield faster execution time and better solutions than sequential processing in a single machine. In view of the significant number of computers are idling after business hours, they can be used in parallel structures for solving practical applications that require huge amount of computing resources. By integrating the parallel structures and computational intelligent techniques, this proposal will provide a powerful alternative to an expensive platform.

Reference

1. Akl, S.G., 1989, "The Design and Analysis of Parallel Algorithms", Prentice-Hall International, New Jersey
2. Finkel, D., Will, C.E., Brennan, B. and Brennan C., 1999, "Distriblets: Java-based distributed Computing on the Web" in Internet Research: Electronic Networking Applications and Policy, Vol. 9, No. 1, 35-40
3. Fountatin, T.J., 1994, "Parallel Computing; principles and practice", Cambridge University Press, London, England
4. Fukuyama, Y., Chiang; H.C., 1996, "A parallel genetic algorithm for generation expansion planning"; IEEE Transactions on Power Systems, Vol.11, No.2, 955-961
5. Glover, F., Laguna, M., 1997 "Tabu Search"; Kluwer Academic Publishers
6. Hennessy, J.L. and Patterson, DA., 1998, "Computer Organization and Design: The Hardware/Software Interface", Morgan Kaufmann Publishers, Inc., San Francisco, USA
7. Nara; K. 1997, "Genetic Algorithm for Power Systems Planning"; Proceedings of the 4th International Conference on Advances in Power System Control, Operation and Management, APSCOM-97, Hong Kong.
8. Pfister, G., 1995, "In Search of Clusters: The Coming Battle in Lowly Parallel Computing", Prentice-Hall International, New Jersey, USA
9. Sheble, G.B. and Brittig, K., 1995, "Refined Genetic Algorithm - Economic Dispatch example"; IEEE Transactions on Power Systems, Vol.10, No.1, 117-124
10. Walters, D.C. and Sheble; G.B. 1993, "Genetic Algorithm solution of economic dispatch with valve point loading"; IEEE Transactions on Power Systems, Vol. 8, No.3, 1325-1331
11. Wong, K..P., Wong, Y.W.; 1994, "Genetic and genetic/simulated-annealing approaches to economic dispatch"; IEE Proceedings for Power Generation. Transmission and Distribution. Vol. 141, No.5, 507-513
12. Wong, K.P, Wong, Y.W.; 1996 "Combined genetic algorithm/simulated annealing approach to short-term generation scheduling with take-or-pay fuel contract"; IEEE Transactions on Power Systems, Vol.11, No.1, 128-136