Parallel computation for production and distribution planning

A F Suahati\textsuperscript{1,*}, K J Wang\textsuperscript{2} and A A Nurrahman\textsuperscript{1}

\textsuperscript{1} Department of Industrial Engineering, Universitas Islam Bandung, Bandung, Indonesia
\textsuperscript{2} Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan

\*luthfi69ti@gmail.com

Abstract. Due to the TFT LCD (Thin-Film-Transistor Liquid-Crystal-Display) manifestation into many related products, this industry has been growing rapidly along with increasing demand. To satisfy the demand, most companies have increased their production capacity and capability by increasing their number of factories in different places and causing complexity in this industry. This research develops a production and distribution planning model for the multi-stage and multi-site supply chain in the TFT LCD industry. Genetic algorithm proposed in this research to solve the problem. Maximizing capacity utilization and total profit in the supply chain are become the major performance indicator in this model. Regarding the high computational effort of genetic algorithm, then parallel computation performed. Genetic algorithm conducted in multi-processor computation using OpenMP for time efficiency. To compare the computational time, the genetic algorithm conducted in five different number of processors; 1, 2, 4, 8, and 16, to know how many processors are needed to get the optimal computational time. The result is the genetic algorithm using 4 processors has the optimal expected net profit compare to the others. The result shows that a larger number of processors doesn’t mean the computation time will become automatically faster.

1. Introduction

Thin-Film-Transistor Liquid-Crystal-Display (TFT LCD) panel manufacturing industry is a highly complex multi-stage and multi-site production network. The manufacturing process divided into three general stages: array, cell, and module processes defined as “multi-stage”. Each production stage involves more than one factory, forming a “multi-site” production network. Cost of glass substrate and color filter accounted more than 30% of the total material cost of TV. Therefore, TFT LCD industries have to concerns to determine the most economical way to cut the glass substrate. As a product which involves many manufacturing plants those are located in different places, the company will be faced with the problems of production and distribution in multi-stage and multi-site supply chain.

Production and distribution planning is one of big issue in TFT LCD industry, Lin and Chen presents a multi-site supply network planning problem considering variable time buckets [1]. They proposed a monolithic model of multi-stage and multi-site production planning problem with combining two different time scales, i.e. monthly and daily time buckets and considering a real-world practical example of a TFT LCD manufacturer in Taiwan. Chen et al. explains about capacity allocation and expansion model for TFT LCD multi-site manufacturing [2]. Chen explains single-stage and multi-site capacity planning under demand uncertainty for TFT LCD industry [3]. He developed
resource allocation and capacity extension scheme to reduce cost and wastage incurred by cutting of glass substrates and color filters by considers the economic cutting problem of glass substrates in factories of different generations. Zegordi et al. proposed a novel genetic algorithm for solving production and transportation scheduling in two-stage supply chain [4].

Owing to the complexity of TFT LCD panel manufacturing problem, proposed algorithm for solving this problem are usually time consuming. Some researchers have developed computation method to make their algorithm are more efficiency toward the computational time. One of the computational method to make the time consumption is more efficient is to use either a computer or a processor in parallel. Related to the parallel computation, Rocha proposed a hybrid distributed memory parallel genetic algorithm for optimization [5]. This works presents a genetic algorithm combining two types of computational parallelization methods, resulting in hybrid shared, distributed memory algorithm based on the island model using both OpenMP (Open Multi-Processing) and MPI (Message Passing Interface) libraries. The resulting gains in execution time due to parallel implementation allow the use of high fidelity analysis procedures based on the finite element method in the optimization of composite laminate plates and shells.

In other studies, Wang and Lee present about parallel genetic annealing algorithm with OpenMP [6]. In his paper a parallel genetic annealing algorithm (GAA) combining with simulated annealing algorithm and genetic algorithm was proposed, which was implemented using distributed computing model that employed OpenMP on four core computer. In this algorithm, several each sub-population evolved independently, and the current best individual was distributed into all the sub-population. The algorithm overcame premature convergence, and found global optima efficiently in less time. And the experiment results shown that the performance had been significantly improved. Umbarkar and Joshi proposed a dual population genetic algorithm versus OpenMP genetic algorithm for multimodal function optimization [7]. The experimental result shows that the performance of the parallel genetic algorithm using OpenMP is remarkably superior to that of the sequential genetic algorithm in terms of execution time and speed up. This research shows that OpenMP GA gives optimum solution in comparison with the other methods. In summary, most of the previous researches only focused on production and distribution within a single production a stage two production stages. High computational using parallel computation to solve the production and distribution problem is also not considered.

2. Methods
Parallel computing aims to reduce the time needed to solve computational problem. In this research, OpenMP is used to conduct the parallel computation. OpenMP is an application programming interface (API) for parallel programming on multiprocessors in C, C++, and Fortan, on most processor architecture and operating systems, includes Windows, Mac OS X, Linux, and Solaris platforms. OpenMP using shared-memory model to do the parallel computation, processor will synchronize and communicate with each other through shared variables. The shared-memory model is characterized by fork/join parallelism (Figure 1), in which parallelism comes and goes. At the beginning of execution, only a single thread (called master thread) is active. The master thread executes the serial portions of the program. It forks additional threads to help it execute parallel portions of the program. These threads are deactivated when serial execution resumes.
OpenMP directives used in loops through the individuals, such as the objective function calculation and the genetic algorithm operators (crossover and mutation). The section of code which is need to be run in parallel is marked accordingly, a #pragma is added before the section of code and the task is automatically assigned and balanced between processors. OpenMP also provides a function to set how many processor is needed to do the parallel computation using #omp_set_num_threads(). Figure 2 describes how the parallel computation applied to genetic algorithm in this research.

**Figure 1.** Multithreading for execute the parallel codes.

**Figure 2.** Schematic diagram of parallel-genetic algorithm.
In this research, the first step is that algorithm randomly generates an initial chromosome population. Then the fitness value of each chromosome calculated by the expected total profit. Chromosomes with higher fitness value have higher chance to survive until the next generation. The proposed model applied the roulette wheel method as the selection method on the reproduction process.

After the fitness value calculation and reproduction process finished, then the parallel phase begins. Parallel phase begins with crossover procedure. Uniform crossover applied to the algorithm to execute the crossover procedure. The algorithm selects a pair of chromosomes and exchange randomly gene values of the selected chromosomes. After the crossover procedure completed, mutation procedure will continue the algorithm. New values replace randomly the original values following the mutation rate. To ensure the entire chromosome is a feasible solution during the evolution, evaluation process performed in this process. The procedure of the proposed parallel – genetic algorithm presented in Figure 3.

![Figure 3. Procedure of parallel – genetic algorithm.](image)

### 3. Results and discussion

The text of your paper should be formatted as follows: To compare the computational time, the proposed algorithm conducted in five different number of processor as a parameter; 1, 2, 4, 8, and 16. It aims to know how many processors are needed to get the optimal computational time. To compare computational time of the proposed algorithm under different number of processor, each parameter is run with time limit as the termination condition. The proposed algorithm stops as exceeding its computational time. Then, 180 seconds is determined as the CPU time limit to run the proposed algorithm with large problem size. This time is to approve that genetic algorithm still has a better solution than PSO to solve the problem in the same computational time. Figure 4 shows fitness evolution of genetic algorithm in different number of processor.

![Figure 4. Fitness evolution of genetic algorithm in different number of processor.](image)
Figure 4 shows the fitness evolution of genetic algorithm using 4 processors has the optimal expected net profit compare to others. The result shows that bigger number of processor doesn’t mean the computation time will become automatically faster since to send the task to different processors is takes a time. It means to send the task into many processors is takes more time too. Therefore, the optimum number of processors is determined by the section of code which is needs to be run in parallel. In this proposed algorithm, the parallel section would be optimum if done with 4 processors.

Table 1 shows the comparison under different number of processor details include the net expected profit and CPU usage. CPU usage is based on task manager when the program is running. Since all the experiment is conducted in 16 cores computer, the proposed algorithm using 16 processors has 100% CPU usage.

| Number of Processor | Net Expected Profit | CPU Usage (%) |
|---------------------|--------------------|---------------|
| 1                   | 17,649,636,617.94  | 6             |
| 2                   | 17,597,977,559.16  | 12            |
| 4                   | 17,720,182,427.51  | 27            |
| 8                   | 17,583,755,564.97  | 50            |
| 16                  | 17,543,760,663.12  | 100           |

4. Conclusion

This study has proposed the genetic algorithm to solve production and distribution planning problem in multi-stage and multi-site TFT LCD panel manufacturing supply chain. In order to time efficiency, genetic algorithm conducted in parallel computation. The parallel – genetic algorithm tested under different number of processor. The result shows that bigger number of processor doesn’t mean the computation time will be faster, because it takes time to send the task to different processor. It means that to send the task into many processors is takes more time too

References

[1] Lin J T and Chen YY 2007 A multi-site supply network planning problem considering variable time buckets – A TFT LCD industry case The International Journal of Advanced Manufacturing Technology 33(9-10) 1031-1044
[2] Chen T L, Chen Y Y and Lu H C 2013 A capacity allocation and expansion model fit TFT LCD multi-site manufacturing Journal of Intelligent Manufacturing 24(4) 847-872
[3] Chen C C 2007 Capacity and transportation planning for a TFT-LCD supply chain (National Taiwan University of Science and Technology)
[4] Zegordi S H, Kamal I N A and Beheshti M A N 2010 A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain Computers & Industrial Engineering 58(3) 373-381
[5] Rocha I B C M, Parente J E and Melo A M C 2014 A hybrid shared distributed memory parallel genetic algorithm for optimization of laminate composites Composites Structures 107 288-297
[6] Wang K J and Lee C H 2015 A revised ant algorithm for solving location-allocation problem with risky demand in a multi-echelon supply network Applied Soft Computing 32 311-321
[7] Umbarkar A J and Joshi M S 2013 Dual population genetic algorithm (GA) versus OpenMP GA for multimodal function optimization International Journal of Computer Applications 64(19) 29-36