International Journal of Emerging Engineering Research and Technology
Volume 4, Issue 11, November 2016, PP 1-11
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Job Shop Scheduling Application Using Genetic and Branch and Bound Algorithms

Briggs, Tobinson, A.

[1]*Department of Mechatronic Engineering, University of Port Harcourt, Port Harcourt, Nigeria*

ABSTRACT

The paper tackled two methodologies in solving job shop scheduling problems, the "genetic and branch and bound" algorithm. The aim is to advance the existing schedule of jobs assigned to machines in a local glass-container factory as a mean of reducing delay in the current schedule so that the mean processing time is minimized. In the study, data used was supplied by the manufacturer along side of the development of model representing each manufacturing line processing time of each job. Their performances are compared for the jobs constructed and appraise each algorithm applied Technological constraints and other constraints such as precedence relations and resource availability mainly the furnace capacity were considered.

Keywords: Genetic Algorithm, Branch and Bound Algorithm, Scheduling, Processing Time, Job Shop.

INTRODUCTION

Pinedo in his work defined "scheduling as the allocation of distributed assets to tasks over a given period of time" (Pinedo, 2001). Job shop scheduling problem can be illustrated generally by comprises of n jobs in a set $\{J_i\}_{1\leq i \leq n}$ and which will be processed on m machines of a set $\{M_j\}_{1\leq j \leq n}$. The problem can be characterized as follows:

1. On each machine each job is processed in an order given by a predefined sequence of operations
2. One job is processed on each machine at a time
3. Job i J is process on machine j M which is defined by the operation O ij
4. Each operation ij O requires an uninterrupted processing on machine j M and preemption is not allowed
5. The processing times for each operation are known in advance

Products or parts are produced in mechanized facility where all necessary operation are performed is known as a shop. Emblematic shops may be divided into single or multiple machines in its operations. Multi-machine shops have machines that may be the same, similar, or different based on the required doling out technique. In a plant all machines are alienated into shops.

Plants are decomposed into smaller shops for more specialized and improved solution measures is developed for planning, scheduling and quality control purposes. On the other hand, it gives room for increased need of coordination among shops.

THE STRUCTURE OF GENETIC ALGORITHM

Inspired by the principles of natural evolution, GA's are one of the well known types of evolutionary algorithms (Holland, 1985). Akin to biological reproduction, sexual GA's produce offspring from two parents, while asexual GA's produce offspring from one parent. At the start, a set of possible solutions termed initial population (NPOP) of chromosomes is populated. Each generation of offspring is called iteration. Reproductive phenomena that enable diversity like crossover and mutation are achieved in GA's by using case-specific operators. In the next few paragraphs, the working of sexual and asexual reproduction processes in GA is presented.

In sexual reproduction the GA's work similar to that of human genetics. Although mutation which involves the refinement of single chromosomes is important, the crossover operation defines sexual reproduction. Individual chromosomes from the NPOP are assessed and given fitness values which are stored. Using the fitness values as the selection criterion, the crossover operator forms two new off springs from two chromosomes in the first iteration. This iterative process is repeated producing a new generation each time until the loop closes according to set criteria. Finally, the last generation of chromosomes which form the solution to the problem are evaluated.

Asexual reproduction in GA is similar to sexual reproduction in that fitness values are assessed, calculated and stored from the NPOP, but in this case, rearrangement of component genes in each chromosome is performed. Since there is only one parent, mutation is the basis of asexual reproduction like in single-celled species. The last process involves sorting to find the best chromosomes. Various strategies of gene rearrangement have been presented in the literature.

In this study, asexual reproduction with two mutation operations was applied to find best solutions.

The following pseudo code below (Holland, 1975), describing the general working of a GA with asexual reproduction, (crossover is not performed) was used.

```
t = 0;
Initialize (K (t=0));
Evaluate (k (t =0));
While not Terminated () do
Kp(t) = k(t).select parents();
Kc(t) = Mating(Kp);
Mutate1 (Kc(t) );
Mutate2 (Kc(t) );
Evaluate (Kc(t));
k (t+1) = build next generation from (Kc(t), k(t));
t = t+1;
End
```

The Structure of Branch and Bound Algorithm

According to **Chinneck,** (2010) the terminology used in structuring the branch and bound algorithm are given below:

i *Node*: it represents incomplete or complete solution. Each task that could be scheduled based upon precedence and resource constraint

ii Leaf (leaf node): it stands for the search consists of traversing the tree until the best root-to- leaf path is found; that is the solution is complete with all know variable values

iii Bud (bud node): a partial solution, either feasible or infeasible. Think of it as a node that might yet grow further, just as on a real tree.

- *Bounding function*: by growing a bud node further, it gives the method of calculating approximately the best value of the objective function. Bounding function values are only connected from the bud nodes. Objective function values are contained in the Leaf nodes, these gives actual values but not estimates. It is imperative that an optimistic estimator should be from the bounding function. Likewise, if you are minimizing, it must underestimate the actual achievable objective function at its best value; then, maximizing it must overestimate the achievable objective function at its best value. You want it to be as accurate an estimator as possible so that the resulting branch and bound tree is as small as possible, but it must err in the optimistic direction. The bounding function is the real magic in branch and bound. It takes ingenuity sometimes to find a good bounding function, however the payoff in amplified efficiency is marvelous. Every problem has its diverse conditions.

- *Branching, growing, or expanding a node*: the process of creating the child nodes for a bud node. One child node is created for each possible value of the next variable. For example, if the next variable is binary, there will be one child node associated with the value zero and one child node associated with the value one.
- *Incumbent*: "is the best complete feasible solution found so far". When the solution process begins, there may not be an incumbent. In that instance, the first incumbent is taken as the first complete feasible solution found during the solution process.

The below BnB algorithm according to Pinedo, (1995) was used

Step 1: (Initial condition)

Ω : = {Each job first operation}

r_{ij} : = 0 for all $(i, j) \in \Omega$

Step 2: (Machine selection)

Compute $t(\Omega)$ for current partial schedule.

$t(\Omega)$: = min{$r$ij + Pij}

$i\_$: = machine on which the minimum is achieved.

Step 3: (Branching)

Ω' : = { $(i\_, j)$|ri_j < $t(\Omega)$ }

- For all $(i\_, j) \in \Omega'$, extend partial schedule by scheduling (i, j) next on machine i.
- For each such choice, delete $(i\_, j)$ from.
- Add job successor of $(i\_, j)$ to.
- Return to Step 2

PROBLEM FORMULATION AND ANALYSIS

This section provides a detailed description of the job shop scheduling problem dealt in this study. Considering n jobs $i = 1, \ldots, n$ and m machines $M_1, \ldots, M_m$. Each job i consists of a set of operations O_{ij} ($j = 1, \ldots, n_i$) with processing times p_{ij} . Each operation O_{ij} must be processed on a machine $\mu_{ij} \in \{M_1, \ldots, M_m\}$. All jobs operations may have precedence relationships. Each job can be processed by more than one machine at a time while each machine can only process one job at a time. The aim is to minimizes some objective function especially the mean flow time of the finishing times C_i of a feasible schedule of the jobs $i = 1, \ldots, n$ of a local glass container factory. Obeying the constraints of precedence and resource availability such as furnace capacity, we got an optimal processing time for five production lines with the feasible tasks schedule.

The problem is denoted by

$$J_m \left| prec \right| \left(\sum_{j=i}^{n} \left(C_j / n \right) \right) \tag{1}$$

Where;

J denotes a job shop with m machines,

prec denotes precedence constrains on jobs

C j denotes the completion time of job *j*

The following assumptions are considered while formulating the solution approach for the job shop scheduling problem:

- There exists a feasible schedule.
- All jobs are ready at the time of processing
- Preemption of jobs is not allowed
- Setup times are sequence-independent
- Setup times are added to processing times
- Shift Break times are not considered in processing times

For the purposes of this paper, the manufacturer supplied data were obtained from manufacturer for five actual jobs. These data were well defined. The job build variables in table below have a direct impact on the time required to complete a job.

Table1. *Line Performance Parameters*

Symbol	Line Performance Parameter
W	Average Operating Weight (g)
V_s	Machine Sectional Speed (BPM)
T_p	Actual Production Time (min)
G_m	Melted Gross Pull (Tons)
G_p	Packed Gross Pull (Tons)
B_T	Theoretical Gross Production
B_A	Actual Gross Production
E	Efficiency (%)

Decision Variables

p_{ik} processing time of job *j* on machine *i* (tonnes per day)

C_j Total Completion Time

m Number of Machines

n Number of jobs

O*ij* set of operations O_{ij}

C*g* Capacity of furnace in period t (tones)

I_{it} Stock of product *k* at the end of period t

I_{jt} Backlog of product *k* at the end of period t

The optimal processing time after a pre-specified number of generation(s) was determined using Matlab Release 2007b (R2007b) running on an AMD Turion (tm) X2 Dual Core Mobile RM – 72 210GHz processor with 32 – bit operating system Windows 7 Home Premium.

EXPERIMENTAL RESULTS AND DISCUSSION

The results of the computational methods are given below for ten, twenty, thirty and forty operational runs. A run is a completed number of generation(s) or branches for genetic algorithm and branch and bound algorithm respectively. The results obtained here are not reproducible, that is, two runs of identical characteristics may not necessarily produce the same results. This is due to the random generators and the nodes pruning used in the techniques employed. Also, the nature of asexual genetic algorithm also increases the irreproducibility of the results.

Table2. *Computational Results for five operational runs*

No. of Run	GA	BnB	No. of Run	GA	BnB	No. of Run	GA	BnB	No. of Run	GA	BnB	No. of Run	GA	BnB
1	550	375	1	600	380	1	550	370	1	788	380	1	351	370
2	350	375	2	350	375	2	350	390	2	350	380	2	600	370
3	600	380	3	350	370	3	350	370	3	900	375	3	350	400
4	350	390	4	350	390	4	350	370	4	355	380	4	350	375
5	600	375	5	351	375	5	351	370	5	351	370	5	550	400
6	600	375	6	350	370	6	355	370	6	600	375	6	788	390
7	350	370	7	350	390	7	800	375	7	351	370	7	350	390
8	350	370	8	350	370	8	350	370	8	350	400	8	350	375
9	350	370	9	355	380	9	350	390	9	350	380	9	355	380
10	355	390	10	350	370	10	600	380	10	350	375	10	350	380
			11	350	380	11	600	380	11	351	370	11	550	370
			12	350	380	12	900	400	12	600	370	12	351	370
			13	350	370	13	350	375	13	350	375	13	600	370
			14	350	375	14	350	370	14	350	370	14	350	380
			15	351	375	15	350	375	15	355	375	15	350	375
			16	350	370	16	351	375	16	351	370	16	600	370
			17	600	380	17	550	375	17	350	375	17	350	375
			18	350	375	18	351	375	18	351	375	18	550	375
			19	350	370	19	550	375	19	350	380	19	355	370
			20	350	375	20	350	375	20	600	375	20	788	370
						21	350	370	21	350	370	21	351	390
						22	350	380	22	788	375	22	350	375
						23	550	375	23	350	370	23	350	370
						24	788	375	24	351	375	24	350	375
						25	350	370	25	350	370	25	600	380
						26	550	375	26	788	375	26	350	375
						27	355	380	27	350	370	27	355	370
						28	351	375	28	350	380	28	788	390
						29	350	370	29	350	375	29	600	375
						30	355	375	30	350	370	30	350	400
									31	350	375	31	550	370
									32	550	400	32	350	370
									33	350	375	33	600	375
									34	350	380	34	788	375
									35	351	375	35	788	375
									36	350	380	36	350	375
									37	350	370	37	350	375
									38	350	380	38	600	380
									39	350	390	39	800	375
									40	355	390	40	351	375
												41	800	380
												42	600	380

												43	**355**	**400**
												44	**350**	**390**
												45	**788**	**380**
												46	**350**	**370**
												47	**550**	**370**
												48	**350**	**375**
												49	**600**	**375**
												50	**350**	**380**

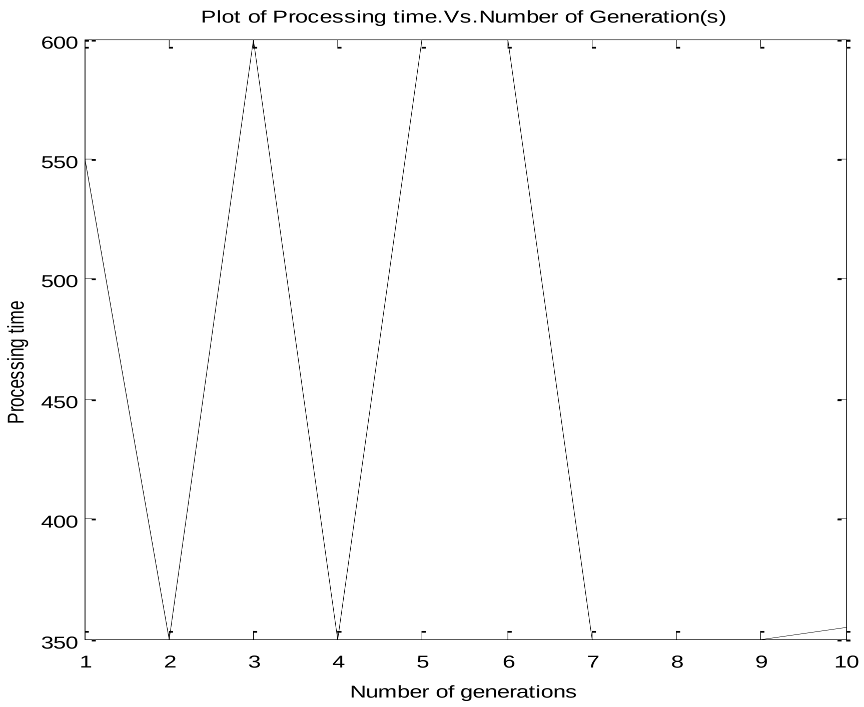

Figure1. *Plot of Processing Time Vs ten generations for GA*

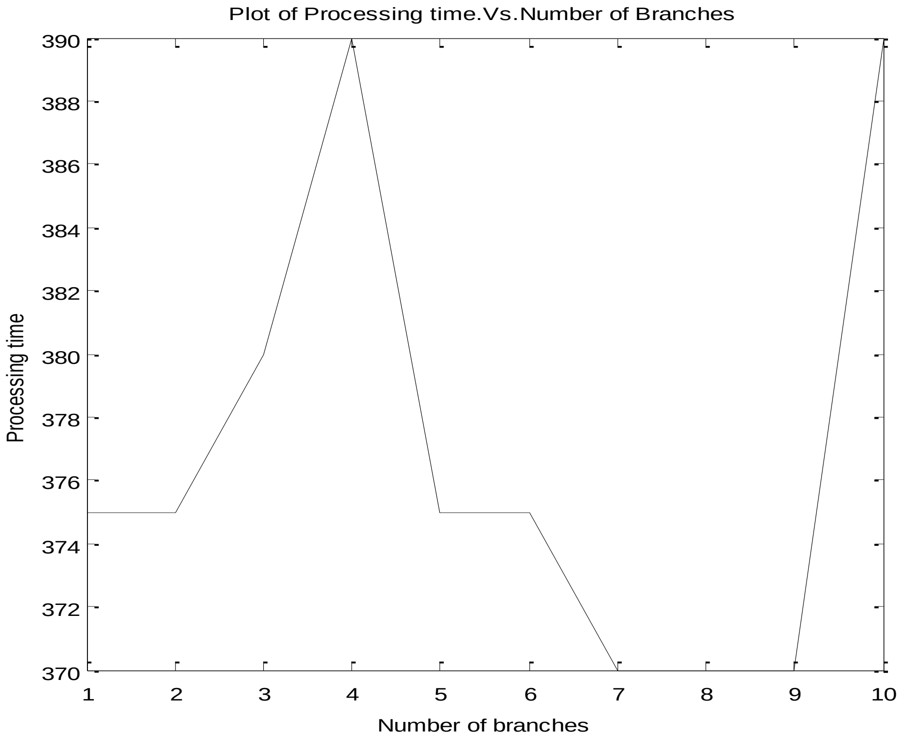

Figure2. *Plot of Processing Time Vs ten branches for BB*

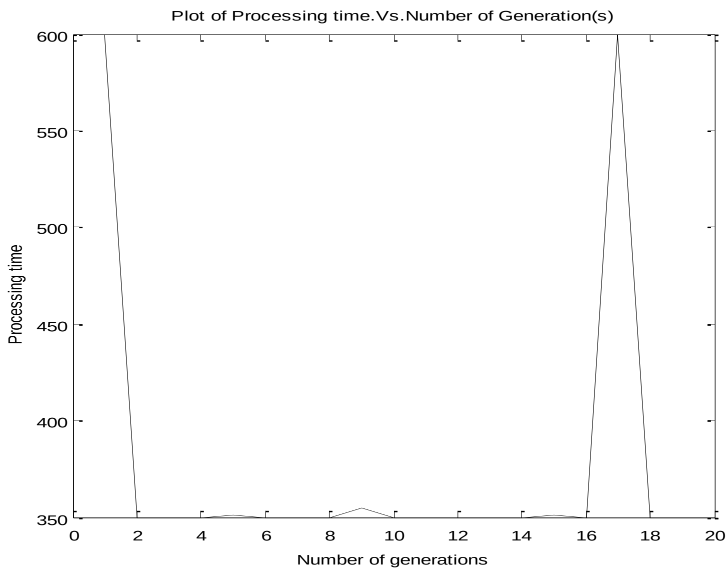

Figure3. *Plot of Processing Time Vs twenty generations for GA*

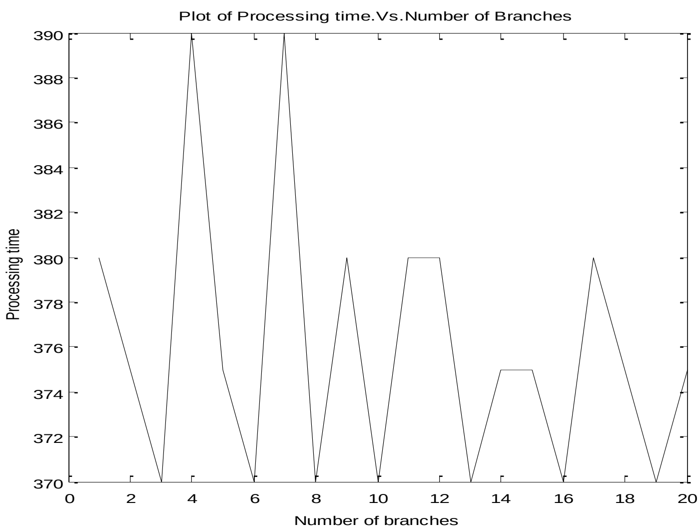

Figure4. *Plot of Processing Time Vs twenty branches*

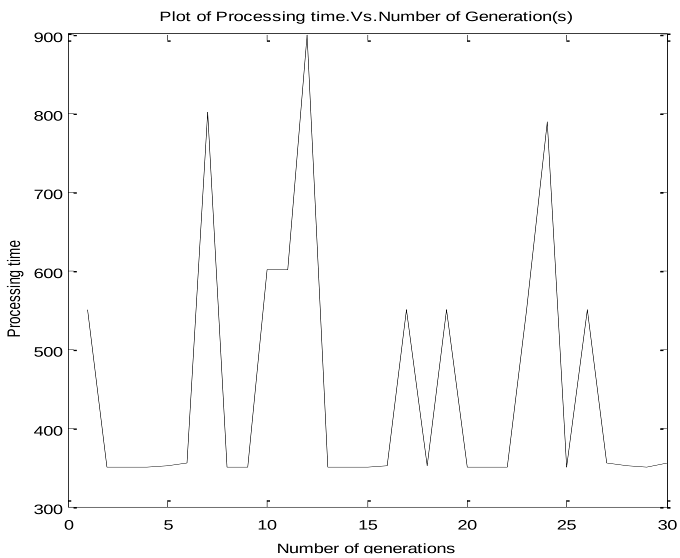

Figure5. *Plot of Processing Time Vs thirty generations*

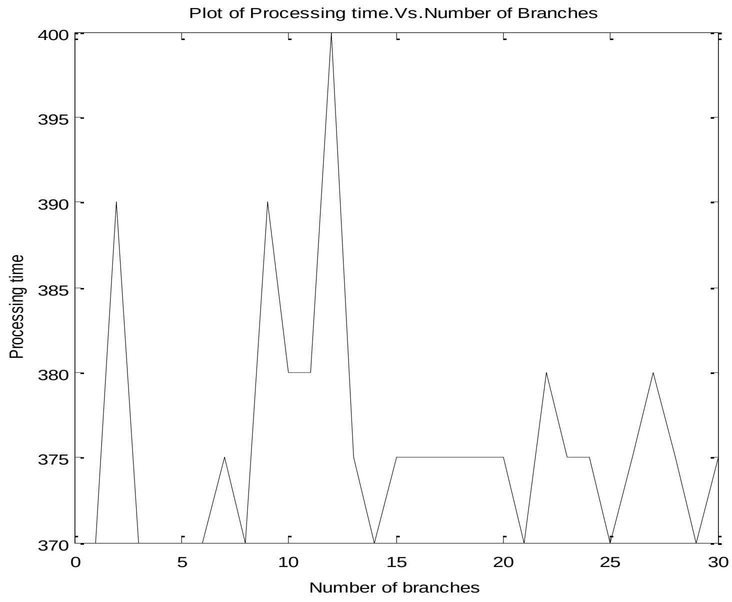

Figure6. *Plot of Processing Time Vs thirty branches for BB*

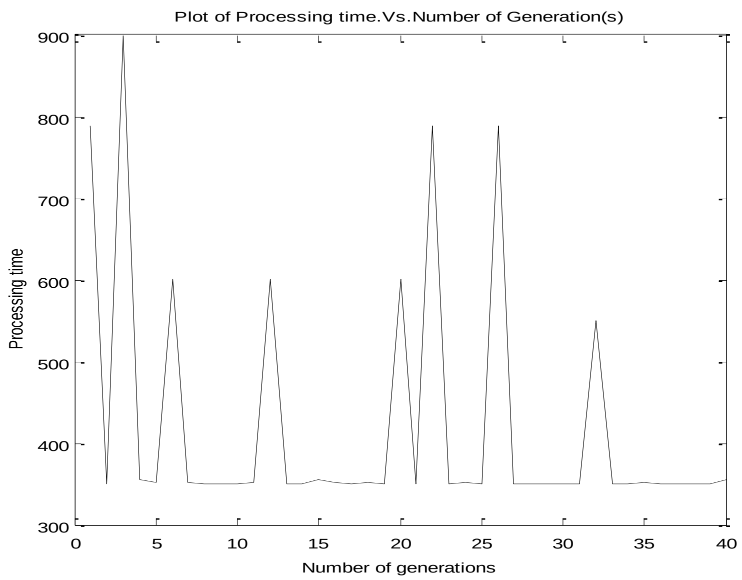

Figure7. *Plot of Processing Time Vs forty generations for GA*

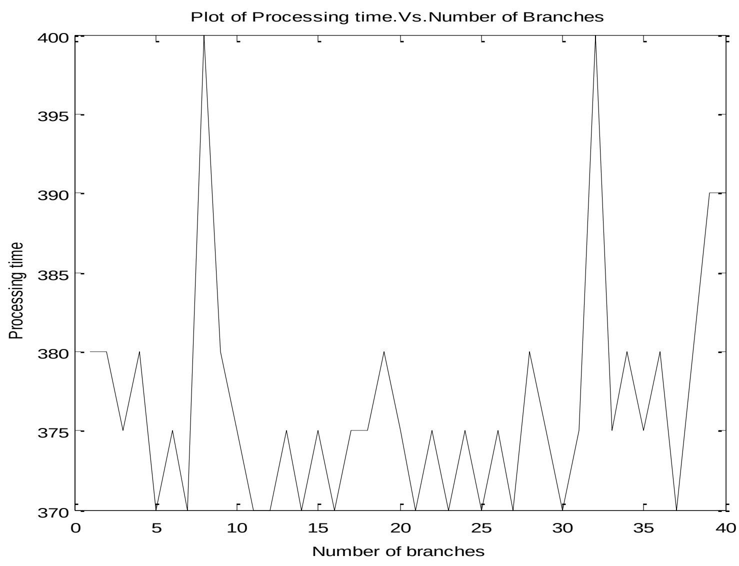

Figure8. *Plot of Processing Time Vs forty branches for BB*

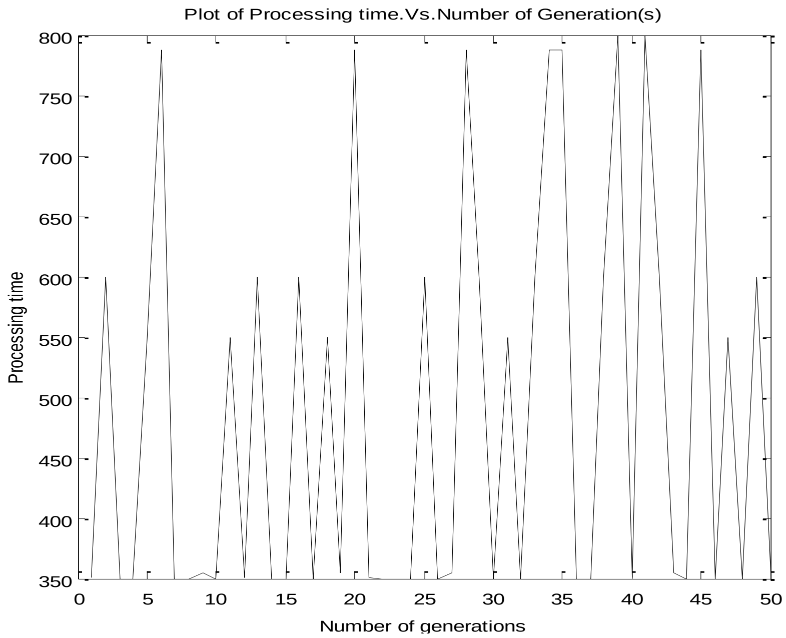

Figure9. *Plot of Processing Time Vs fifty generations for GA*

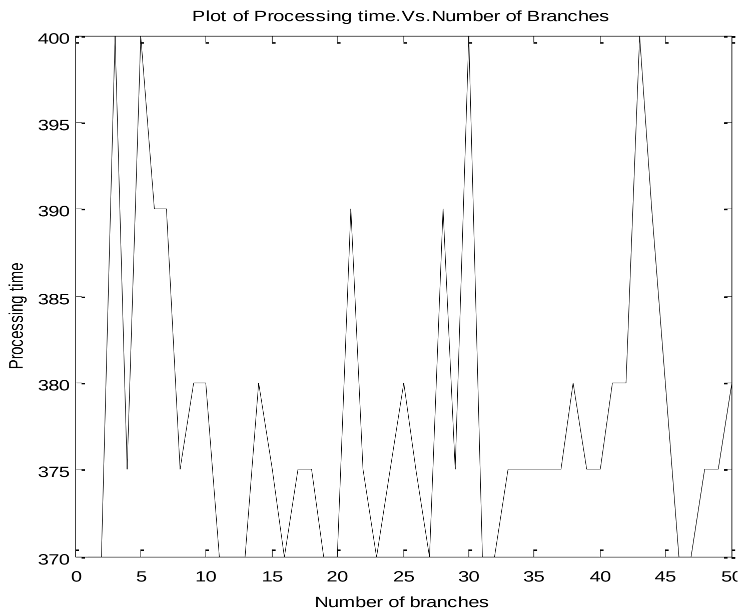

Figure10. *Plot of Processing Time Vs fifty branches for BB*

From the computational analysis, genetic algorithm, GA has proved to obtain near optimal solutions with reasonable computational time than the branch and bound algorithm, BnB. With the high relative speed of computational GA possesses over the BnB algorithm, results were easily obtain.

BnB algorithm was found not able to solve large size problems unlike genetic algorithm which operates within large range of possible solutions. The operational runs for both techniques show that GA often gives the best minimum processing times for the various parallel machines. Also, it was observed that the minimum processing times given by GA for the five parallel machines were always same while the minimum processing times given by BnB were always slightly different with variations ranging from 0.5 to 5.0

Therefore the research work has indicated the meta-heuristics nature of GA for solving hard optimization problems than BnB which is an exact method.

CONCLUSION AND RECOMMENDATION

This paper presents a model for optimizing production schedule with the objective of minimizing the "mean processing time" of various jobs in a current schedule for five uniform parallel machines based on the development and analysis of "genetic algorithm" and "branch and bound algorithm".

Analyses were coded in genetic algorithm and branch and Bound algorithm and programmed with the aid of matlab software. Both computational methods improve on the current schedule by optimizing the mean processing time.

The two resource constraints were precedence and furnace capacity. The techniques calculated the optimal duration of the glass manufacturing process to be about three hundred and fifty (350) minutes for GA and three hundred and seventy (370) minutes for BnB for specific quantity of bottle products. One of the major advantages of using computational techniques is their relative speed of computation. With relatively low processing time, computational techniques can solve problems with a very large number of jobs in seconds. Also, other methods of solving scheduling problems (Critical Path and Project Evaluation Review Technique) cannot account for resource constraints.

Several comparisons were made for the minimum processing time and duration generated for ten, twenty, thirty and twenty operational runs. We observed that the optimal minimum processing time with the existing schedule was always obtained for ten and higher generational runs. The reproducibility of the G.A was calculated to be about 60%.

Many recommendations have been suggested for most works on genetic algorithm, including local searches in the components of the genetic algorithms to optimizing the operators of the algorithm to using an effective replacement strategy; several strategies exist for improving the results of a genetic algorithm. For this particular work, the following recommendations are been made for further studies:

- Using a complex task schedule with large number of jobs as this will emphasize the effectiveness of the algorithms.
- Including cross over operation in the analysis of genetic algorithm
- Including tardiness and earliness of activities in the analysis
- Compelling job shop scheduling with various computational techniques.

ACKNOWLEDGEMENT

I will like to appreciate students who spend time and help to carry out the field work. Also thank the Indigenous Glass Company who gives assistance in carrying out the study.

REFERENCES

[1]. Aldakhilallah, K. A., and Ramesh, R., (2001), "'Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment", International Journal of Production Research, Vol. 39, Issue 12, pp. 2635 – 2657.

[2]. Brooks, G. H., and White, C. R., (1965), "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem", Journal Of Industrial Engineering, Vol. 16, Issue 1, pp. 34-40.

[3]. Brucker, P., Jurisch, B., and Sievers, B., (1994), "A Branch and Bound Algorithm for the Job-Shop Scheduling Problem", Discrete Applied Mathematics, Vol. 49, pp. 109-127.

[4]. Carlier, J., and Pinson, E., (1989), "An Algorithm for Solving the Job Shop Problem" Management Science, Vol. 35, Issue 2, pp. 164-176.

[5]. Chu, C., Portmann, M. C., and Proth, J. M., (1992),"A Splitting-Up Approach to Simplify Job-Shop Scheduling Problems", International Journal of Production Research, Vol. 30, Issue 4, pp. 859-870.

[6]. French, S., (1982), "Sequencing and Scheduling - An Introduction to the Mathematics of the Job-Shop", John-Wiley and Sons, New York.

[7]. Glover, F., (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence", Journal of Computer and Operations Research, Vol. 13, No. 5, pp. 533-549.

[8]. Glover, F., and Greenberg, H. J., (1989), "New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", European Journal of Operations Research, Vol. 39, Issue 2, pp. 119-130.

[9]. Graham R.L., Lawler E.L., Lenstra J.K. and Rinnooy Kan A.H.G.,(1979), "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey", Annals of Discrete Mathematics 5, pp. 287–326.

[10]. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., (1983), "Optimization by Simulated Annealing", Journal of Science, Vol. 220, No. 4598, pp. 671 - 680.

[11]. Kubiak, W., Lou, S.X.C., and Wang, Y., (1996), "Mean Flow Time Minimization in Reentrant Job Shops with a Hub", Journal of Operations Research, Vol. 44, pp. 764-776.

[12]. Manne, A. S., (1960),"On the Job-Shop Scheduling Problem", Journal of Operations Research, Vol. 8, pp. 219-223.

[13]. McMahon, G. B.,and Florian, M., (1975), "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness", Operations Research, Vol. 23, Issue 3, pp. 475-482.

[14]. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., (1953), "Equation of State Calculations by Fast Computing Machines", The Journal of Chemical Physics, Vol. 21, Issue 6, pp. 1087-1092.

[15]. Mitchell, M., Holland, J. H., Forrest, S., (1994), Advances in Neural Information Processing Systems, Morgan Kaufmann Publishers, San Fransisco, CA, pp. 51-58.

[16]. Moon, I., and Lee, J., (2000), "Genetic Algorithm Application to the Job Shop Scheduling Problem with Alternative Routings", Technical report, Brain Korea 21 Logistics Team, Pusan National University, South Korea.

[17]. Nasr, N., and. Elsayed, E. A.,(1990), "Job Shop Scheduling with Alternative Machines", International Journal of Production Research, Vol. 28, No. 9, pp. 1595-1609.

[18]. Pinedo, M., (2001), "Scheduling: Theory, Algorithms, and Systems", Prentice Hall, New York.

[19]. Yamada, T., and Nakano, R., (1997), "Genetic Algorithms for Job-Shop Scheduling Problems", Proceedings for the Modern Heuristic for Decision Support, London, March 18-19, 1997, pp. 67-81.

AUTHOR'S BIOGRAPHY

Dr. T. A. Briggs, obtained his Doctoral degree (Ph.D.) in Mechanical Engineering and a Postgraduate certificate in Advanced Studies in Academic Practice from University of Newcastle. Dr. Briggs has published some research papers in learned reputable journals both here and abroad. He has a high sense of research curiosities with a great intellectual drive. Currently the Head of Department and a lecturer of Mechatronic Engineering and Acting Director of Intellectual Property Technology Transfer Office (IPTTO).