Evolutionary Production Planning and Finite Capacity Scheduling

Zablon. A. Mbero[1], H. O. Nyongesa*[2]

[1]Department of Computer Science, University of Botswana
[2]*University of the Western Cape, South Africa

ABSTRACT

In this paper, we discuss an optimization approach to real-world production planning and finite capacity scheduling (FCS). Due to production flexibility, it's usually possible to generate many feasible process plans for the jobs in a production environment. Production planning and finite capacity scheduling are tightly linked. The optimality of capacity scheduling depends on optimal production planning, and vice versa. The integration of production planning and capacity scheduling is therefore important for an efficient utilization of manufacturing resources. In this paper, based on data from real instances, we have developed a system for production planning and scheduling for manufacturing lines in small and medium enterprises (SME). The optimization approach is based on an evolutionary approach, genetic algorithms (GA). The schedules are constructed using rules in which the priorities are determined by the GA, using a procedure that generates parameterized activities. After a schedule is obtained local search heuristics are applied to improve the search solution.

Keywords: Production Planning, Finite Capacity Scheduling, Evolutionary Algorithms, Genetic Algorithms, Process Optimization, Priority Ordering.

1. INTRODUCTION

We consider, production planning and production scheduling as an application, which involves the allocation of resources over time to perform a collection of tasks. In a generalized scheduling problem, there are $\boldsymbol{w}$ work orders and $\boldsymbol{r}$ resources. Each $\boldsymbol{w}_i$ comprises a set of operations, $\boldsymbol{o}_i$ that are carried out by $\boldsymbol{r}_i$ resources over a finite processing time, $\boldsymbol{t}_i$ in an operation dependent sequence. A production operation is comprised of processes or tasks defined over a route used to identify which resources are to be used, and the time taken on each resource. Route information contains a sequenced list of operations needed to complete a work order, including the details of amount and type of resources needed at each operation stage. The sequence is identified by having operation name predecessors and successors against each operation on the route.

Production planning and production scheduling problems are known to be NP-hard [6], and as such in many industrial production environments human expert knowledge is relied upon, which in most cases is not optimal and can lead to significant variations in production times and costs [6]. On one hand, jobs, resources, constraints and priorities constitute a huge and complex search space that cannot be timely solved in practice by heuristic or operations research traditional techniques. On the other hand, schedules often need to be frequently updated in response to changes in job priorities or the availability, or non-availability of resources. The use of GA is, thus, well suited to such problems because of their effectiveness at searching large spaces [11]. GA is a search, optimization and machine learning in which a population of abstract representations of candidate solutions lead to near optimal solution through pseudo-Darwinian operations [9]. Thus, GA are a useful tool in analyzing competing solutions in order to find an optimized plan in a drastically reduced time period. In the proposed approach, the objective is to minimize deviations from prescribed due-dates for the different work orders under given constraints, in particular improvement in delivery performance and reduction in inventory costs.

The rest of the paper is organized as follows: Section 2 provides overview of GA, and production planning and finite capacity scheduling, Section 3 presents an approach for optimization production planning, section 4 details the application of GA to production scheduling. Section 5 outlines an experimental case study. Conclusions are contained in section 6.

2. OVERVIEW

2.1 GENETIC ALGORITHMS

Genetic algorithms have been proven to be a powerful tool for solving complex optimization and search problems. In nature, given a population of individuals, environmental pressures cause natural selection (survival of the fittest) and thus a rise in the fitness of future populations. In GA, given a quality function to be maximized we initially create random sets of candidate solutions and apply the quality function as an abstract fitness measure (the higher the better). Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to them. Recombination or crossover is an operator applied to two or more selected candidates (parents) and results in two new candidate solutions (offspring). Mutation is applied to a single parent and results in one new candidate. The execution of the above leads to a set of new candidates that compete based on their fitness. This process can be iterated until a candidate with sufficient quality is found.

Evolutionary algorithms have components, procedures or operators that must be specified in order to define a particular EA. Some of these components are; representation (definition of individuals), fitness function, population, parent selection mechanism, and evolution operators, such as, recombination and mutation. Related techniques include; Evolutionary Strategies (ES) [7] and Genetic Programming (GP) [8].

GA is a bio-mimetic evolutionary process with emphasis on genotype based operators (genotype/phenotype dualism) [7,11]. Objects forming possible solutions within the original problem context are referred to as phenotypes, while their encoding, the individuals within the GA are called genotypes. GA is attractive in engineering design because they are easy to apply and in many cases, they are able to find the global near-optimal solution.

GA works on a population of artificial chromosomes, referred to as individuals. Each individual is represented by a string of L bits. Each segment of this string corresponds to a variable of the optimizing problem in an encoded form. The population is evolved in the optimization process by iterative application of crossover and mutations operations. The following is a typical pseudo-code for GA [9]:

```
initialize(P(time, t=0));
evaluate(P(0));
while isNotTerminated() do
    $P_p(t)$=P(t).selectParents();
    $P_c(t)$= crossover($P_p(t)$);
    mutate($P_c(t)$);
    evaluate($P_c(t)$);
    P(t+1)= buildNextGeneration(($P_c(t)$,P(t));
    t=t+1;
end
```

Figure 1: GA Pseudo-code

2.2 PRODUCTION SCHEDULING

Production scheduling is the process of generating “to-do” lists or dispatch lists for the shop floor. As part of a larger planning and scheduling process, production scheduling is essential for the proper functioning of a manufacturing enterprise. A good production schedule helps manufacturing firm lower cost, reduce inventory, and improve customer service. FCS involves defining a sequence of activities under constraints, example, labour, machines, materials, work order delivery times. FCS often groups similar activities to minimize changeover times. So instead of holding one set-up time for a work order, a typical FCS holds a table of set-up times between work orders. The scheduler assigns a starting and ending time to each activity and sequences the activities across the resources so that no resource conflict occurs. In this regard, a scheduler will consists of three components, namely, priority scheduling, due-date and start-date scheduling and efficiency scheduling according to machine capability. These are then integrated to achieve stated production goals, over all work orders. FCS can be modelled as follows :

- Task representation: a given task or work order is represented by start time, end time , duration or due date and the resources required to complete the task. Furthermore a task priority can be appended to discriminate between the tasks, especially where tardiness may occasion severe penalties.

- Operations list: each work order is decomposed into an operations list, taking into account any operation dependencies, such as precedence, on a schedule route. The task of a scheduler is to arrange an operations list, such that it is possible to complete all tasks within the constraints of available resources.

- Attribute matching: a scheduler attempts to identify all possible resources that can be used to carry out a task, by matching the attributes of each resource against the requirements of each operation. Generally, a resource may be matched to several operation attributes, and similarly an attribute may be matched by several resources.

- Time calculation: for each operation on a route, the scheduling system calculates the time required based on the operation times. There are four time types: Fixed; where the time of operation is fixed irrespective of the size or number operations of the batch; Fixed to Batch; where the time for an operation is an integer multiple of the number of batches, irrespective of the batch sizes; Variable to Batch; where time for the operation will be determined by the batch size; and finally variable to Unit; where the operation time is determined by each specific operation.

- Resource calendar: a calendar function enables incorporation of any holidays, shift changeovers including preparation of a resource for the next operation, and resource outages in the scheduling. The calendar by nature is organized hierarchically with information inherited from a top, supervisory level down to the operation level.

3. OPTIMIZATION OF PRODCUTION PLANNING

In general, it's advantageous that a scheduling and planning system starts with an initial rule-of-thumb work order schedule, and then using forward or backward chaining attempt to balance the given constraints. Subsequent improvements to the schedule are usually necessary to achieve a good plan. There are many techniques that can be employed to improve initially derived schedules, including heuristics and local search. In this paper, we have proposed a schedule optimization approach based on evolutionary learning, using GA. The role of GA is to evolve a near-optimal schedule from an initial population of random candidate solutions.

Consider a schedule of N work orders. Each order, o_i is defined by five parameters: start_time, s_t, duration, d_r, end_time, e_t, due_date, d_d, and priority, p_r. Furthermore we define, shift change over time, sh, resources outage time, ro and any embedded holiday times in the calendar, ho. However, for simplicity and clarity, the latter parameters are ignored. The optimization objective is to meet specified due dates, by minimizing the number of late, or for that matter early orders. Thus, $\forall\ i \in (1 .. N)$

$$s\_t_i = e\_t_{i-1} \qquad (1)$$

$$e\_t_i = s\_t_i + d\_r_i \qquad (2)$$

$$d\_r_i = \sum_{j=1}^{o_{in}} t_j \qquad (3)$$

where n is the number of processes in order o_i, and t is the time taken on each process, j.

Next we define tardiness, d as,

$$d = \begin{cases} +1, & if \quad e\_t_i \geq d\_d + \varepsilon \\ 0, & \text{if} \quad \left| e\_t_i - d\_d \right| \leq \varepsilon \\ -1, & if \quad e\_t_i \leq d\_d - \varepsilon \end{cases} \qquad (4)$$

where, ε is a tolerance on order completion times.

We also define work order priority, p as,
+

$$P = \begin{cases} +1, & high \\ 0, & normal \\ -1, & low \end{cases} \qquad (5)$$

The optimization objective is to maximize the function;

$$\mathrm{M}_o = \Theta\left(\frac{1}{1+\left|\sum_{i=1}^{N} d_i * p_i\right|}\right) \qquad (6)$$

4. EVOLUTIONARY OPTIMIZATION of PRODCUTION PLANNING using GA

GA function by working with an encoding of the parameters of the problem domain, rather the parameters themselves. This is unlike conventional optimization techniques, such as gradient descent methods. The encoding equips GA on different types of multi-modal problems, with getting stuck on local minima, and therefore reaches near-optimal solutions. Many different types of encoding can be implemented, including binary, numeric, alphabetic or alphanumeric.

Considering the scheduling of N work orders. As stated previously, the GA approach will create several initial random orderings of all work orders,

$$\mathrm{P} = \bigcup_{i=1}^{p} o_{i1}\; o_{i2}\; o_{i3} \cdots o_{iN} \qquad (7)$$

where, P is the population of candidate solutions, and p the size of the population. The population is then evolved according to the procedure given in Fig. 1, in relation to the fitness function in equation. (6).

To illustrate the evolution operators, crossover and mutation, consider the scheduling of N=10, work orders for simplicity of experssion. Typical candidate schedules would then take the form:

$$P_k = 3\,9\,8\,0\,7\,1\,5\,6\,2\,4$$
$$P_l = 9\,5\,8\,1\,6\,2\,0\,4\,7\,3$$

where, indexing and subscripting have been omitted for the sake clarity. Crossover between two candidate solutions is carried out as follows: two crossover points are selected at random; say the first is in position 3, between 8 and 0 on the P_k and 8 and 1 on P_l. In forming the offspring of these two parents, the genes between the crossover points are preserved. Subsequently, starting from the right hand crossover point corresponding offspring inherit genes from the opposite parent, in an ordered manner, if that gene is not already present. This continues until all the gene positions on either offspring filled. In type of crossover, amongst other types of crossover operators, is unsurprisingly called *ordered crossover*. Thus, the offspring that will emerge from this illustration are:

$$O_k = 8\,6\,2\,0\,7\,1\,5\,4\,3\,9$$
$$O_l = 8\,7\,5\,1\,6\,2\,0\,4\,3\,9$$

An unintended observation from this illustration, is the emergence of particular ordering of genes, for example 4 3 9 in the last three positions of both offspring. This consequence, proven by the schema theorem [1], indeed demonstrates the power of GA.

Mutation, similarly, is carried out by randomly selecting and exchanging two gene positions. However, as in nature mutation occurs, with a very low frequency. But it plays a crucial role in re-ordering the genes in a stagnated sub-optimal population.

5. CASE STUDY

We simulated the application of GA to production planning and scheduling on a simple production process in a SME. The process involves production of different coloured taps. The basic types are produced in one batch, and the task of optimization is to deliver orders of packs of different coloured taps. In this scenario, there are only three types of operations: hot spray, cold spray and packing. The two sprays can be carried out by two different machines with significantly differing efficiencies.

In general, at commencement of a production cycle a schedule of work orders is prepared. We compare a heuristic scheduling approach against GA-derived schedules. In the heuristic approach scheduling software in use applies rule-of-thumb, combining order dates, due-dates, and order sizes. The parameters of the GA parameters were optimized and fixed; however, the population size and number of generations were factors of the number of orders to be scheduled.

Figures 2 and 3, show the comparison of the performance of the two scheduling approaches. Figure 2 shows result of product mean output of different sized works orders, assuming no priority of the orders. In Figure 3, however, randomised priorities were assigned to the work orders.

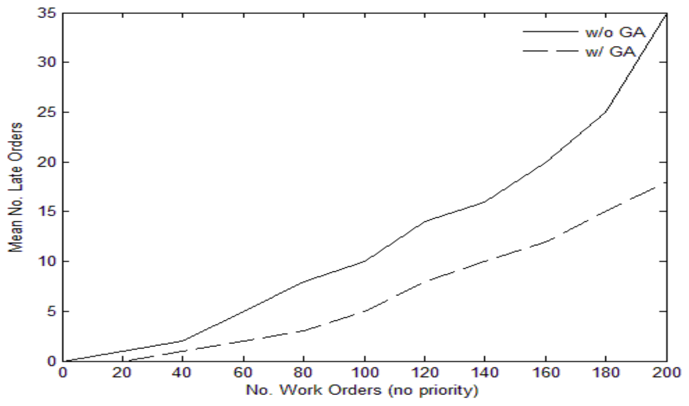

Figure 2: Production Output with no priorities

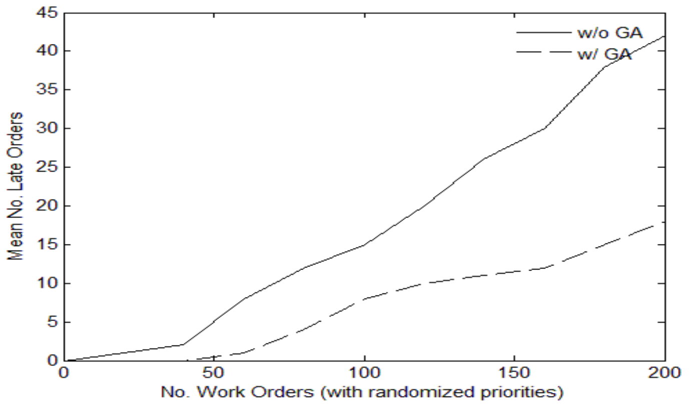

Figure 3: Production Output with randomized priorities

The results show that for low order quantities, there was less significant difference between the heuristic and the GA scheduling schemes. This, however, becomes markedly more significant as the number of orders increases. This is even more so, when priorities are introduced to the work orders. The results demonstrate the suitability and applicability of GA to production and finite capacity scheduling.

CONCLUSION

GA have been studied and applied to many types of problems, including NP-hard problems, such as the travelling salesman and transportation problems. Their ease of application lends suitable for small scale optimization problems. In this paper, we have shown that GA could be used to optimize operations in a real-world environment of SME. Significant production and inventory costs are achieved using a GA-optimized production schedule.

REFERENCES

[1]. Goldberg, David E (2002), “The Design of Innovation: Lessons from and for competent Genetic Algorithms, Addison-Wesley, Reading MA.
[2]. Candido, M.A.B., Khator, S.K. and Barcia, R.M. 1998: A Genetic Algorithm based procedure for more realistic job shop scheduling problem. International Journal of production Research 36, 3437-3457
[3]. Srikath K. Iyer, BarkhaSaxena: Improved Genetic Algorithm for the permutation flowshop scheduling problem, Computers and Operations Research, v.31 n.4, p.593-606, April 2004.
[4]. Colin R. Reeves: A Genetic Algorithm for flowshop sequencing, Computers and operations Research, v.22 n.1, p.5-13, Jan, 1995.

[5]. Biegel, J E. and Davern, J.J., 1990: "Genetic Algorithms and Job Shop scheduling" Computers and industrial Engineering, Vol. 19, Nos. 1-4, pp 81-91.

[6]. Saeedeh, Maleki-Dizaji, Henry Nyongesa and Bobak Khazaei, UPlanIT: An Evolutionary Based Production planning and Scheduling System, pg 1.

[7]. Koza, J.R (1990). Genetic programming: A paradigm for Genetically Breeding populations for computer programs to solve problems

[8]. Evolutionary Strategy: http://en.wikipedia.org/wiki/Evolution_strategy, last access 21-8-13.

[9]. Prof. Dr. Zell, Evolutionary Algorithms, Chapter 3. Algorithms: http://www.ra.cs.uni-tuebingen.de/software/JCell/tutorial/ch03s05.html last access 21-8-13

[10]. Genotype-phenotype distinction: http://en.wikipedia.org/wiki/Genotype-phenotype_distinction, last access 21-8-13.

[11]. Peter Bentley, An Introduction to Evolutionary design by Computers: chapter 1 p 8-10