Abstract—The Aggregate Production Plan (APP) is a schedule of the organization’s overall operations over a planning horizon to satisfy demand while minimizing costs. It is the baseline for any further planning and formulating the master production scheduling, resources, capacity and raw material planning. This paper presents a methodology to model the Aggregate Production Planning problem, which is combinatorial in nature, when optimized with Genetic Algorithms. This is done considering a multitude of constraints of contradictory nature and the optimization criterion – overall cost, made up of costs with production, work force, inventory, and subcontracting. A case study of substantial size, used to develop the model, is presented, along with the genetic operators.

Keywords—Aggregate Production Planning, Costs, and Optimization.

I. INTRODUCTION

AGGREGATE Production Plans (APP) concern about the allocation of resources of the company to meet the demand forecast. Optimizing the APP problem implies minimizing the cost over a finite planning horizon. This can be done by adjusting production load as well as inventory and employment levels over a certain period of time to achieve the lowest cost while satisfying demand and considering the specific constraints for each particular case (company dependent). A good APP has the capacity to positively influence the bottom line and also permit a long-term view of the organization performance. This avoids having to make short-term decisions and fire-fight problems, adversely affecting the organization’s long-term perspective [1].

Managers have access to the break-down monthly or weekly demand forecast for the next planning horizon, normally 1 year. In practice, managers capitalize on the forecasted demand to achieve long-run profitability. They face major constraints in the number of workers, facilities and plant capacity to fulfill the demand. Therefore, not only all the demand must be met in each planning period (month/week), but costs have to be minimized. Managers may decide if meeting market demand results in lower long-term profit, to backorder and/or ask the subcontractors to do a part of the products. The APP problem deals with how to employ the available workforce, resources and facilities, including external contractors, to best satisfy the demand which is defined through APP [1].

Although a number of production planning approaches have been developed in order to improve planning automation and increase efficiency of production planning [2], but a lot of problems in the area of production planning are subject to highly complex constraints which make them very difficult to solve using traditional optimization methods and approaches. Despite the importance of APP which forms the basis for the formulation for all other schedules and materials management, the results of the APP optimization are far from perfect, leaving way to major improvements.

This paper uses Genetic Algorithm (GA), and presents an optimization approach to APP modeling, which permits the search for an optimum, while considering, simultaneously, a large number of constraints of contradictory nature. A realistic case study illustrates the model and the development of the GA to an APP problem with the conditions found in an industrial context is presented.

II. LITERATURE REVIEW

The APP problem considering minimum changes in workforce level as well as inventory and backorders minimization simultaneously was solved for an 8-period planning horizon [3]. In 1998, the APP problem was solved using Mixed Integer Programming and considering different optimization criteria, including revenue maximization as well as inventory, backorder and set-up cost minimization [4]. Baykasoglu added further constraints to the previous models such as subcontractor selection and set-up decisions [5].

Later on, a number of artificial intelligence approaches, alone or combined with mathematical programming models have been used to solve the production planning problems.
considering more constraints. GA, fuzzy logic and stochastic programming have been among the most popular ones. Among all, Wang and Fang proposed a fuzzy programming model to imitate the human decision procedure for production planning ended with a family of inexact solutions within an acceptable level [6]. A fuzzy multi-objective linear programming model for solving the multi-product APP decision problem in a fuzzy environment considering inventory level, labor levels, capacity, warehouse space and the time value of money is presented in [7]. A model to optimize the multi-site APP problem by considering a wider range of constrains describing a two-stage stochastic programming model [8].

However, little attention has been given to develop a strategy taking into account the many constraints and their combination, as they appear in practice. The combination of factors simultaneously affecting the quality of the APP is a characteristic of real-life problems and their consideration can make the difference between a purely academic treatment of the subject and a result that can be applied or transferred immediately in practice.

In this paper, a complex and realistic mathematical model is built and a GA is developed for its optimization. It goes beyond developing heuristics to solve simple strategies to optimize the APP. Instead, the approach is general, all optimization constraints are implemented into the Fitness Function and a penalty is incurred for any suboptimal solution. The model contain a large number of practical constraints including production cost, labor cost, hiring and laying off costs, holding costs (carrying inventory during plan period) and subcontracting costs.

III. RESEARCH METHODOLOGY

In developing the methodology for modeling and optimizing the APP a number of strategies can be and were considered:

Strategy 1: fill the requirements using overtime – workers (all or them or only veteran personnel – workers with at least one week stage in the company) are used to work for an integer number of hours. In this case the inventory and contracting out the units to be delivered are underutilized/disregarded;

Strategy 2: fill the un-met requirements using external contractors – is a lean, outsourcing strategy in regards to keeping inventory, reducing, at the same time the workforce available to a minimum;

Strategy 3: fill requirements using up to the equivalent of a given number of weeks output in inventory, by minimizing at the same time the variation of workforce. This strategy can also minimize the use of contractors, taking advantage properties of keeping inventory to increase or keep a service level [9].

Other strategies or any linear combinations of strategies can be developed and the results of their application assessed. These strategies can be implemented as heuristics in algorithms to optimize the planning process. However, any of these strategies is likely to produce desired results – i.e. minimum costs - only for a narrow combination of conditions and input values, which might appear briefly, as windows - during the planning horizon. The use of any set strategy would, in this case, be suboptimal in the rest of the planning horizon.

Also, it became obvious that, by using a set strategy, there would be a set relation between a number of variables (see next section) e.g. production plan in a period, number of veteran and new workers, the production, hours worked, inventory each day and cumulative inventory and the respective costs.

After examining the results of implementing the strategies presented above, it emerged that a better approach would be to avoid constraining the planning to just one of these strategies. The independent variables in this case are chosen as the number of workers each planning period and the number of hours worked, with all production and inventory levels derived from this. The only constraint imposed is the maximum level of inventory, which is a sensible condition in practice.

It was decided to use the evolutionary character of GA to determine an optimum result by exploring the whole search space. This is the equivalent of finding the best strategy or combinations of strategies at any point in time, and varying it, as necessary, to produce an optimum result. When choosing GA for the optimization process, an important element was their capacity to implement any cost function [10].

IV. MODELING OF THE APP PROBLEM

As a realistic model is sought for the APP problem, a complex combination of conditions is applied. The list of variables is by no means exhaustive, but it incorporates many decisions variables, economies of scale, hard constraints and costs, etc.

A. Variables:

Planning data:

T - Planning horizon;
DY - Total forecasted demand in year (units/year);
Pt - Production of current week (units/week);
DY (units/t);
DPtmax - The maximum forecasted demand for each period in the planning horizon (units/t);
DPMax - The minimum forecasted demand for each period in DY (units/t);
P - Production of current week (units/week);
Pm - Forecasted demand for each period in the planning horizon (units/t);

Labor costs:

CRL - Regular wage – including overheads ($/h)
COL - Overtime wage – including overheads ($/h)

PNL - productivity of a new worker in first week (units/h)
\[ L_t \] - number of full-time permanent labors in period t – variable;  

Personnel policy: 
- \[ C_H \] - The cost of hiring one labor ($ / Labor) 
- \[ C_L \] - The cost of laying off one labor ($ / Labor) 

Plant running costs: 
\[ N \] – Actual hours company works per week - variable; 
- \[ C_P \] - Plant running cost per hour – normal hours ($/h) 
- \[ C_{PO} \] - Plant running cost per hour – overtime ($/h) 

Inventory policy: 
- \[ I_t \] – inventory level; 
- \[ C_I \] - inventory cost to hold a single unit of product at the end of each period ($ / unit - period); 
- \[ C_S \] - shortage cost per unit associated with subcontracting ($/unit); 

\[ Lt \] - number of full-time permanent labors in period t – variable; 
\[ N \] – Actual hours company works per week - variable; 
\[ C_P \] - Plant running cost per hour – normal hours ($/h) 
\[ C_{PO} \] - Plant running cost per hour – overtime ($/h) 

B. Constraints 
The assumptions listed below are implemented as a list of feasibility constraints. Violating any of these constraints would produce an infeasible solution: 

1. The company works at least \( N_t \) hours each week; 
2. The number of hours worked in a week is integer; 
3. A worker will only produce an integer number of units per week. If the worker cannot produce a whole unit, he/she will be reassigned during that time for maintenance work (paid – equivalent cost for the time worked - but no direct output is obtained); 
4. The company has to deliver all products corresponding to the demand each week (service level 100%); 
5. The company uses the products made/kept/contracted out to satisfy demand in the following order: 
   - A. units made that week by the workforce; 
   - B. the shortage will be covered, if possible, from inventory; 
   - C. if the company is still short of units, they will be outsourced to contractors; 
6. All excess products will be stored in inventory; 
7. In the first week of the planning horizon, the company has a number of workers and a number of items in storage equal with the average number of workers per week to fulfill average demand and the equivalent of an average week of production, respectively; 
8. The capacity of the warehouse storing the inventory is maximum three times the average weekly output; 

The assumptions above, very realistic in any manufacturing context, have also the potential to significantly simplify the modeling of the problem and the implementation of the algorithm. 

It is important to point out that assumption 5 in combination with assumption 8, in fact, guide the decisions regarding the make-or-buy of products or, on the other hand, rely on your work force or adopt a very flexible hire-or-fire policy for employed personnel. As it is set, it tends to favor the existing workforce, with contracting out used only as a last resort. However, this set of assumptions can be modified to be aligned with the management's general strategies and the company's external context. 

V. CHROMOSOME ENCODING 
The chromosome encoding is presented with relation to the case study below. The chromosomes encode the solutions of the problem, in this case assembling the independent variables of the problem – namely the number of workers employed and the number of hours worked each period. The planning horizon was chosen 1 year with a granularity of the model of 1 week. This implies the chromosome is an array of 52 x 2 variables (104 independent variables). 

The chromosome is illustrated in Fig. 1 part A, in a vertical format for space-saving purposes. For reference, the number of the week is displayed at the left of the chromosome. The number of hours worked is minimum 80, as explained in the previous section and set in the case study. 

VI. THE GENETIC ALGORITHM 
The structure of the GA is classic [10], with genetic operators adapted to the particularities of the problem. Instead of working with strings, they are tailored to work with arrays. They are illustrated in the following sections. 

1. Handling Constraints 
The probability to obtain an infeasible chromosome by random genetic operators (GO) is reduced as long as the operators are implemented correctly, taking into account the set of constraints detailed in Section 4. The major source of infeasibility is constraint 8. A chromosome has to be tested for feasibility after generation or application of a GO (crossover or mutation). The repair strategy proposed and tested successfully checks the level of inventory and, if constraint is not satisfied, to reduce the number of workers at the point of infeasibility (for the week when the inventory level exceeds three times the average weekly output) until the gene becomes feasible. Even if rare, it is possible to have more than one infeasible gene in a chromosome. In this case, the repair is to be done successively, from the first to the last week. 

2. Crossover 
The crossover is, in principle, a simple cut and swap operation. Figure 1 part B presents an example of crossover. 

In this example, parents P1 and P2, randomly selected from the initial operation, undergo the crossover. The cut point is randomly selected after week 13, and the two bottom-parts of the parents’ genetic information are exchanged. After the operation, a feasibility check/repair is necessary. 

3. Mutation 
The mutation operator is, again, classic in its principle. An example of mutation operator is presented in Figure 2 part B (M1 and M2). In this example, a randomly selected chromosome of the population undergoes mutation. The genetic information from weeks 15 and 39, randomly selected, is swapped. After mutation, the chromosome has to undergo a feasibility check/repair operation.
4. Evaluation

The Fitness Function (FF) of each chromosome is dependent upon the costs associated with the application of the strategy associated with the corresponding particular solution. GA has a remarkable ability to incorporate and use almost any conceivable type of cost structure [10], [11]. The total cost (TC) for a solution/chromosome is the sum of all costs attached to operating the company for the next forecasting horizon:

\[
TC = \sum_{i=1}^{n} (PC + WC + IC + SC)
\]

- **PC** - Production cost – takes into account the normal and overtime rate;
- **PC** = \(C_P\) if \(N < N_W\);
- **PC** = \(C_{PO}\) if \(N \geq N_W\)

**WC** - Costs associated with workforce

\[
WC = WC_1 + WC_2
\]

---

**Fig. 1** The chromosome and the crossover operators

**Fig. 2** The chromosome and the mutation operator
WC - made of wages (WC1) + hiring and firing costs (WC2);

\[ WC1 = N \times L_t \times C_{RL} \] if \( N \leq N_W \); - normal working time
\[ WC1 = L_t \times N \times C_{RL} + L_t \times (N_w \times N \times C_{OL}) \] if \( N \geq N_W \); - if overtime is needed

\[ WC2 = C_H \times (L_t - L_{t-1}) \] if \( L_t \geq L_{t-1} \); - if workers hired
\[ WC2 = C_L \times (L_{t-1} - L_t) \] if \( L_t < L_{t-1} \); - if workers fired

IC – Inventory keeping costs –only if inventory is positive;
\[ IC = (I_{t-1} + P_t + D_{Pt}) \times C_I \] if IC \( \geq 0 \)
\[ IC = (I_{t-1} + P_t + D_{Pt}) \times C_I \] if IC \( < 0 \); - given for first week, calculated subsequently;

Where
\[ I_{t-1} \] – previous week's inventory – given for first week, calculated subsequently;
\[ P = N \times L_t \times P_L \] if \( L_t \leq L_{t-1} \) – production by veteran workers
\[ P = (N \times L_{t-1} \times P_L) + N \times (L_t - L_{t-1}) \times P_{NL} \] if \( L_t > L_{t-1} \) – production by veteran workers and newly hired workers.
\[ D_{Pt} \] – forecasted demand for each period in the planning horizon (units/t) - given;

SC – Costs associated with subcontracting a part of production;
\[ SC = IC \times C_S \]

5. Selection

The stochastic sampling mechanism is used to select the next generation of chromosomes, associated with the Holland’s proportionate selection or roulette wheel selection (Holland, 1975). Because the weighed roulette works for maximization of the fitness values and the GA in this case is designed to minimize the cost, a simple double transformation is applied: the inverse solutions’ cost is multiplied with \( 10^{10} \). After the GA has been applied, the true costs are restored, using the inverse operation – i.e. multiplying the inverse of the FF with \( 10^{10} \) [12].

VII. A CASE STUDY

A case study has been developed in conjunction with the model presented in last sections. The forecast for the next year is broken down in Table I. The case study is based on the following data and has co-evolved with the model of the APP problem:

<table>
<thead>
<tr>
<th>Week</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>D</td>
</tr>
<tr>
<td>1</td>
<td>12000</td>
</tr>
<tr>
<td>2</td>
<td>10500</td>
</tr>
<tr>
<td>3</td>
<td>8000</td>
</tr>
<tr>
<td>4</td>
<td>11500</td>
</tr>
<tr>
<td>5</td>
<td>8000</td>
</tr>
<tr>
<td>6</td>
<td>10000</td>
</tr>
<tr>
<td>7</td>
<td>9000</td>
</tr>
<tr>
<td>8</td>
<td>10500</td>
</tr>
<tr>
<td>9</td>
<td>11500</td>
</tr>
<tr>
<td>10</td>
<td>12000</td>
</tr>
<tr>
<td>11</td>
<td>8000</td>
</tr>
<tr>
<td>12</td>
<td>11000</td>
</tr>
<tr>
<td>13</td>
<td>8500</td>
</tr>
</tbody>
</table>

The values presented above were used to develop and test the genetic operators. The cost function is implemented as a subroutine composed of the relevant cost modules. The cost structure is flexible and can be easily modified to suit any similar problem if necessary, since constraints can be varied in magnitude and other constraints can be added as required.

VIII. CONCLUSION

A complex and realistic model for the optimization of the APP has been developed. It incorporates the most important constraints and costs currently encountered in a manufacturing company.

The GA for the optimization of the APP is in an advanced implementation state. All operators have been developed and tested and will be integrated in the full algorithm shortly. Preliminary results are promising.

Further work will address the following:
- Finalization of the full GA and its testing;
- Implementation of a yet more complex cost structure, ideally by developing a framework to incorporate all realistic costs that can appear in practice;
- Optimality of results and how they are influenced by the relative level of different classes of costs on the strategy to employ, patterns of strategies as the level of costs vary;
- The possibility to address stochastic events and their influence on the optimality of the APP.

ACKNOWLEDGMENT

As the first author of this paper, I am indebted to Dr. Romeo Marian and Prof. Lee Luong for their positive influence in conducting this research. I am also very much grateful to my dear brother, Mr. Behdad Fahimnia, and whole family members for their all-time supports and I dedicate this research to them all. I hope this study can add to the body of knowledge in Manufacturing.

REFERENCES


Mr. Behnam Fahimnia was born in Tehran, the capital city of Iran, in 1978. He graduated with bachelor degree in Mechanical Engineering, Solids Designing from Tehran Azad University in 2001 ranked 2nd and granted the university fellowship in 1999. At present, he is in the completion of a master program in Advanced Manufacturing Technology at the University of South Australia. Mr. Fahimnia ranked the top student with High Distinction in all subjects and is now in the process of joining the PhD program late 2006. He has published several papers in national and international conferences and the most recent ones are listed below:


He is also a research assistant at the University of South Australia, School of Advanced Manufacturing and Mechanical Engineering, under supervision of Dr. Romeo Marian. His research interests are Production Planning, Project Planning and Control, Manufacturing Automation, Genetic Algorithms. Mr. Fahimnia is a member of Institute of Engineers Australia (IEAus), Australia, and Institute of Mechanical Engineers (IMechE), UK. He is now registered as a professional engineer in Australia and is a registered employee at the University of South Australia, with the School of Advanced Mechanical and Manufacturing Engineering.