Intelligent Design of Manufacturing Systems

Liam Quinn

A thesis submitted in partial fulfilment
of the requirements of De Montfort University
for the degree of Doctor of Philosophy

June 1996

De Montfort University

Abstract

The design of a manufacturing system is normally performed in two distinct stages, i.e. steady state design and dynamic state design. Within each system design stage a variety of decisions need to be made of which essential ones are the determination of the product range to be manufactured, the layout of equipment on the shopfloor, allocation of work tasks to workstations, planning of aggregate capacity requirements and determining the lot sizes to be processed.

This research work has examined the individual problem areas listed above in order to identify the efficiency of current solution techniques and to determine the problems experienced with their use. It has been identified that for each design problem, although there are an assortment of solution techniques available, the majority of these techniques are unable to generate optimal or near optimal solutions to problems of a practical size. In addition, a variety of limitations have been identified that restrict the use of existing techniques. For example, existing methods are limited with respect to the external conditions over which they are applicable and/or cannot enable qualitative or subjective judgements of experienced personnel to influence solution outcomes.

An investigation of optimization techniques has been carried out which indicated that genetic algorithms offer great potential in solving the variety of problem areas involved in manufacturing systems design. This research has, therefore, concentrated on testing the use of genetic algorithms to make individual manufacturing design decisions. In particular, the ability of genetic algorithms to generate better solutions than existing techniques has been examined and their ability to overcome the range of limitations that exist with current solution techniques.

For each problem area, a typical solution has been coded in terms of a genetic algorithm structure, a suitable objective function constructed and experiments performed to identify the most suitable operators and operator parameter values to use. The best solution generated using these parameters has then been compared with the solution derived using a traditional solution technique. In addition, from the range of experiments undertaken the underlying relationships have been identified between problem characteristics and optimality of operator types and parameter values.

The results of the research have identified that genetic algorithms could provide an improved solution technique for all manufacturing design decision areas investigated. In most areas genetic algorithms identified lower cost solutions and overcame many of the limitations of existing techniques.

Acknowledgements

A special thanks to Dr. David Stockton, my supervisor, for his constant help and advice that was necessary for the completion of this work. Also thanks to Professor Jeffrey Knight, Head of Department of Mechanical and Manufacturing Engineering for securing funding to enable me to undertake this work. This research was funded by the Science and Engineering Research Council (SERC).

Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A	Prime Costs
β	Penalty Function Weighting Value
BL	Balancing Loss
BOM	Bill Of Materials
C	Cycle time in minutes
CAN-Q	Computer Analysis of Networks of Queues
COMSOAL	Computer Method of Sequencing Operations for Assembly Lines
CRAFT	Computerised Relative Allocation of Facilities Technique
Ca	Additional costs incurred
C_e	Prime Costs
Ce_p	Prime Costs for p standard sizes (£'s)
Ce_{p+1}	Prime Costs for p+1 standard sizes (£'s)
C_h	Holding Costs Per Item
C_p	Cost of Placing an Order
C_{sv}	Cost saving obtained from adding one extra standard size (£'s)
c_{ij}	Cost of Carrying One Unit between machines i and j
D	Annual Demand
DP	Dynamic Programming
d_{hij}	Minimum Clearance Distance between Machines i and j in the x axis
d_{vij}	Minimum Clearance Distance between Machines i and j in the y axis
EOQ	Economic Order Quantity
F	Baseline Function Value
FCFS	First Come First Served
FIFO	First In First Out
FMS	Flexible Manufacturing System
f_{ij}	Frequency of Movements between Machines i and j
GA	Genetic Algorithm
GRAI	Groupe de Recherche en Automatisation Integrere
GT	Group Technology
IDEF	Integrated Computer-aided manufacturing Definition
JIT	Just In Time
LCFS	Last Come First Served
LFL	Lot For Lot rule
LP	Linear Programming
LTC	Least Total Cost
LUC	Least Unit Cost
l_i	Length of Machine i
M	Lead Time for a new order to arrive
MOM	McLaren's Order Moment
MRP	Materials Requirements Planning
N	Number of parts to be produced
n	Number of Machines/Workstations
Np	Number of Precedence Constraints Broken
Nt	Number of Workstations exceeding Cycle time
η_{min}	Theoretical Minimum Number of Workstations

OR	Operational Research
PFA	Product Flow Analysis
POQ	Periodic Order Quantity
POR	Planned Order Release
Q^*	Economic Order Quantity
QAP	Quadratic Assignment Problem
R	ReOrder point
RPW	Rank Positional Weighting
SA	Simulated Annealing
SADT	Structured Analysis and Design Technique
Smax	The upper bounding Model
Smin	The lower bounding Model
SSADM	Structured Systems Analysis and Design Method
SWL	Safe Working Load
T	Time available to produce N parts in minutes
TS	Tabu Search
u(x)	Performance Variable
VLSI	Very Large Scale Integration
W	Window Size
w	The number of standard frame sizes in the interval
WIP	Work In Progress
w_i	Width of Machine i
x_i	x co-ordinate of machine i
y_i	y co-ordinate of machine i

1. Introduction

Manufacturing systems are essentially input-output systems (Lupton 1986, Parnaby 1979 and Hitomi 1979) that produce outputs, in the form of saleable products, from the use and transformation of their inputs. Inputs to the manufacturing system include people, capital, materials, machines, information and a variety of external social and economic factors. Input-output diagrams represent the manufacturing system itself as a "black box" and it is the design of this "black box" that determines how efficiently inputs are converted to outputs and hence the competitive advantage of the organisation.

In this respect a systems approach has been found to be essential to the successful design of manufacturing systems. This approach considers the manufacturing system as a collection of sub-systems that form an integrated whole. Although each sub-system will possess its own function and characteristics, the complete system according to Williams (1994) is synergistic in that "it has more properties than the sum of the properties of its parts". The systems approach has, therefore, been widely adopted to enable the complexity of a manufacturing system to be broken down into smaller units each of which can then be individually designed. Particular attention can then be paid to the characteristics, relationships, boundaries, environment, functions, strengths and weaknesses of each sub-system and their relationships with the system whole. Individual system elements need to be designed such that the overall system satisfies a company's specific business, market and operational requirements (Iwata et al 1984). These individual system elements cannot be considered in isolation because of the inter-relationships that exist between them, i.e. a change in one element may have a significant effect on other elements leading to a reduction in the overall performance of the system.

To aid the manufacturing systems developer, a number of design and analysis methodologies are available, the most notable being IDEF, Groupe de Recherche en Automatisation Integrere, GRAI (Doumeingts et al., 1987), Structured Systems Analysis and Design Method, SSADM (Ashworth 1988) and Structured Analysis and Design Technique, SADT (Marca and McGowan, 1988). These methodologies assist the analyst in breaking down the higher level systems into smaller sub-systems, by identifying the inputs, functions and outputs of each of the sub-systems. This "top down" approach (Williams 1994) assists in identifying the interactions between sub-systems and their effects on the overall system. "Bottom up" approaches have also been adopted which use the basic elements that make up a system to construct the higher level system. Normally, this approach is feasible only if all such system elements and their inter-relationships are known prior to the manufacturing system design process.

1.1 Manufacturing Systems Design

Mathematical models and algorithms are available to help optimise manufacturing systems design. The use of these aids has been limited due to the number of variables involved and the complexity of the inter-relationships that exist between them. Often such effects are either unknown or too complex to accurately model. To model a manufacturing system mathematically, the system is normally simplified, leaving the model unable to evaluate the effect of individual parameters on the overall system performance. To overcome this, simulation modelling is increasingly being used to model systems as it allows complex interactions to be modelled in sufficient detail. However, it is difficult to use simulation modelling to optimise the design of manufacturing systems since such techniques only evaluate system efficiencies and often do not provide insights into how system design can be improved.

The design of a manufacturing system is normally undertaken in the following basic stages;

1. Input - Output analysis of sub systems.
2. Steady state design.
3. Dynamic state design.
4. Specification of data collection and information flow functions.
5. Definition of control functions and control systems design.

The stages of the design of a manufacturing system most relevant to the work in this thesis are steady state design and dynamic state design.

1.1.1 Steady State Design

During the steady state design stage the system is designed with the assumption that the following factors remain constant, i.e.

a. process cycle times,

b. change-over times,

c. demand volumes,

d. product mix, and

e. operator performance levels,

It is at this stage of the design process that estimates of the resources required by the system are calculated, i.e. number of machines, approximate manning levels, materials and tools required and approximate inventory levels. It is also assumed that processing operation details and times are correct, breakdowns and other stoppages can be ignored and scrap quantities are negligible. Although these parameters have a significant effect on the overall operation of the system, they are only considered at a later stage during the dynamic design process.

The assumption of steady state conditions when calculating the resources required simplifies the models needed since only average values are determined. Steady state design also assumes that there are no fluctuations in parameter values. The result of steady state design provides only a starting point for further modifications to the design and allows only major problem areas, such as bottlenecks, to be highlighted.

The amount of inventory, particularly work-in-progress (WIP), is affected by a number of factors, for example number of machines, process batch sizes, transfer batch sizes and the methods adopted for sequencing and scheduling parts through the system.

The Steady State calculations allow approximate values to be calculated for the system such as resources required and cell layout. Following this, the design is normally refined at the dynamic state design stage, where the aims are then to reduce WIP and the number of change-overs.

Reductions in WIP allow reduced lead times and buffer space on the shopfloor. Small batches allow reductions in WIP to be achieved but increase material handling requirements and, therefore, handling costs. The sequencing and routing of the parts through the system is also considered during steady state design since these planning decisions effect the number of changeovers that are required, which in turn affect the process capacity of the system.

In steady state design, the identification of bottlenecks is essential since they affect throughput rates, WIP quantity, and manufacturing lead times.

1.1.2 Dynamic State Design

Dynamic design is concerned with the variables and parameters that change over time. During Steady State design, these variables were either assumed to be constant or were simply not examined. The use of constant values, at the steady state design stage can often mask the significance of an individual variable's influence on overall systems performance. It is in a dynamic state that interdependencies between the subsystems lead to major changes in the operational behaviour of the overall system. The effect of these interactions must be identified in the design process in order to avoid performance problems becoming evident after the system is in operation. Assessment is normally undertaken with the use of more sophisticated models than those used in the steady state design stage with the use of computer simulation being prominent.

The first stage of the dynamic design process is to identify the parameters that are to be tested and the range of values over which these parameters will be examined. A model of the system is then constructed using as a basis the basic model identified from the steady state design stage. The model is then used to perform sensitivity analysis of the system variables. An interactive process then takes place which involves modification and testing of the model parameters until no further improvements can be obtained or time constraints prevent further modifications being tested.

Models used at the steady-state design stage primarily compare capacity requirements with available demand. In dynamic design, such models would yield inaccurate results due to their inability to assess the dynamic changes and interactions of sub-systems on the entire system performance. Therefore, more accurate models are required, which take a more detailed, less aggregate, viewpoint. As the models become more detailed, the amount of calculation required increases, necessitating the use of a computer.

Sensitivity analysis is essential in determining the effects on system performance of key dynamic parameters such as production volume, production mix and changeover times. Sensitivity analysis is, therefore, undertaken by individually testing the model against average and extreme values of each dynamic variable. If little change in the overall system performance results from a change then it can be concluded that the variable is not significant, otherwise, experiments are continued to determine the degree of sensitivity exhibited by the system to changes to the parameter concerned. Although sensitivity analysis allows the effects of changes in system variables to be identified it is difficult to determine the effects of two or more variables changing at the same time. In this respect, two or more variables that change simultaneously may have a completely different effect on the system performance than if they change sequentially. Sensitivity analysis, therefore, aids the designer to identify the parameter types that have the greatest effect on system performance.

Once the detrimental effects of changes to the parameters have been identified then solutions to these problems are designed either by correcting the source of the variation or modifying the design of the system to suit the dynamics of the environment. After modifications have been completed, the design is retested and this iterative process repeated until no further improvement can be made or no further time is available.

Previously an efficient manufacturing system incorporated conventional plant in a functional layout with slow changeover times between part types. Such systems although aiming to achieve high utilisation of machines and processes resulted in the whole system being inflexible. In such systems, queues build-up at machines and components are only processed a small percentage of the time they are on the shopfloor. In addition, buffer stocks are used to act as safety stocks to offset the effects of equipment breakdowns and enable the achievement of high resource utilisation. However, long manufacturing lead times result and components are made to a forecast of customer demand rather than from an actual customers order.

An effective system allows customer's order requirements to be achieved such that inventory is minimised and production control is simplified. The system is, therefore, characterised by responsive product availability and good delivery performance. The objectives of such a system are simplicity, effectiveness and ease of control and accountability.

As discussed, the elements which form the manufacturing system are highly interrelated. This makes the process of designing such systems difficult due to the large number of possible variables and combinations of elements that have to be considered.

The various decisions to be taken in a manufacturing system's design can be categorised as design and operational planning based which currently take place during the steady state and dynamic state design stages.

Chapter 2, therefore, examines the main problems involved in steady state design including assortment planning and the design of facility layouts. The chapter analyses current methods for solving these problems, namely the Minaddition Approach for the assortment problem. Chapter 3 then deals with operational planning problems involved during dynamic state design, aggregate planning, Materials Requirements Planning and Kanban Material Control, each decision area is described and current solution methods critically examined.

Chapter 4 identifies the main techniques available for modelling, evaluating and improving the design of manufacturing systems and discusses techniques such as Dynamic

Programming, Simulated Annealing, Tabu Search and Genetic Algorithms. Genetic algorithms (GA) are identified as one such method which have been successfully applied to many areas of engineering. A detailed description is provided of the GA technique.

In Chapter 5 each of the problems involved with the manufacturing design process is examined and methods proposed for their solution using genetic algorithms. The ability of the Genetic Algorithm to determine optimal solutions is investigated using traditional methods, where possible, as an indication of how well the algorithm performs. The sensitivity of the algorithm to changes in its parameters is also investigated.

Chapter 6 examines the results obtained from the GA experiments performed in each decision area. A discussion is provided on how the various elements, (i.e. coding, objective function, cross-over methods, selection methods and mutation), of the genetic algorithm technique has been applied to each decision area.

Chapter 6 also provides guidelines for the selection of operator parameter values. Here the operator parameter values used to obtained the best solutions within each decision area are presented in a manner that enables general relationships to be established between the size of a problem and the most suitable parameter value to adopt.

2. Steady State Design Decisions

2.1 Introduction

Steady state design is normally concerned with the following decisions, i.e.:

a. determining the type and number of each item of processing equipment,

b. determining the type of facilities layout required, and

c. designing the layout in terms of the positions of equipment on the shopfloor.

Identifying relevant processing equipment needs to take into consideration the individual operations required and the processing capacity. In this respect an important decision to be made is that of the number and type of standard models that are included in a product range (section 2.2), since these decisions affect the range of process variables that need to be addressed.

Determining the shopfloor layout of the processing equipment (section 2.3) is a fundamental problem in the design of manufacturing systems which has attracted much research interest. In terms of layout design three main types are recognised, i.e.:

1. Functional or process based layouts, in which all operations of a similar nature are grouped together in the same part of the factory.
2. Product or flow line based layouts, in which the facilities are arranged according to the needs of the product and in the same sequence as the operations necessary for manufacture of the product.

3. Group technology or cellular layouts in which similar components are initially grouped to form families and then all the operations required to manufacture these families are located in the same area of the shopfloor.

2.2 Assortment Planning

In order to optimise the design of manufacturing systems, a fundamental decision needs to be made, at the concept design stage, concerning the number of standard models to produce and the basic design specifications for each of these models. Decisions of this type fall into an important class of problems defined as "the assortment problem" which according to Swanson (1970) are composed of two basic sub-classes, i.e. the "set-up and inventory" assortment problem and the "job shop" assortment problem.

The general form of the assortment problem looks at the determination of the number of standard model types which could be manufactured such that the reduction in the economies of scale obtained through manufacturing a larger number of standards more than offset the lost extra capacity, i.e. material and manufacturing time, in supplying products whose specification is greater than the customer requires, Bongers (1980) and Tryfos (1985). Silver and Kelle (1989) examined a restricted form of the assortment problem in which the set of standards to be stocked was considered fixed and only the relative quantities to stock needed to be determined.

With the "set-up and inventory" assortment problem the method of manufacture is assumed to be associated with a certain set-up cost and optimal production run length. Therefore, when different product models are combined to form a single standard, the inventory, set-up and manufacturing costs must all be considered in deciding whether such a combination should take place. During the "job-shop" assortment problem, manufacturing is assumed to be performed exactly to the specification of the customer. Hence, there is relatively little inventory and the main consideration is the differing costs for various methods of manufacture. The assortment problem has associated with it the subjective criteria listed in Table 2.1, all of which may influence the optimality of solutions and hence makes this type of problem difficult to formulate objectively, as Table 2.1 indicates any solution developed must ideally reconcile the competing objectives of all functional departments involved in product development, marketing and manufacture.

2.2.1 Solution Techniques

Since many of the factors listed in Table 2.1 are difficult to quantitatively define the assortment problem often centres around the estimation of the quantifiable benefits in order to weight these against the subjective criteria. The problem definition then becomes either:
(i) determine the minimum number of standard models such that the loss is acceptable from manufacturing standards, for example with size specifications, greater than customer requirements, or
(ii) determine the number of standard models such that the disadvantages of manufacturing an additional model type are greater than the benefits achieved.

Marketing Factors

* Customer acceptance of sizes other than those historically available.
* Determining whether customers would accept sizes larger than their requirements.
* Competitive product sizes which may prove to provide poor price comparisons if standards were chosen that were larger
* The actual demand pattern, i.e. the preferential sizes favoured by the market.
* The range of sizes provided for the market.
* The profitability pattern, i.e. product size versus profit.
* Manufacturing lead times required.
* The need to market a full range of product sizes, e.g. product sales maybe strongly dependent on customers standardizing product specifications in order to simplify stock holding and maintenance procedures. Therefore, customers would tend to buy from manufacturers who can supply all their specific requirements.

Manufacturing Factors

* Production flexibility, i.e. can the manufacturing facilities handle the level of product and component design variation associated with increasing the number of standard sizes.
* Stock policy, i.e. stock costs can be expected to rise as the number of standard sizes manufactured increases.
* Production control methodology, e.g. batch scheduling, material requirements planning.
* Machine utilisation.
* Machine set-up and operation costs.
* Machine type profile, i.e. the number and type of automated machines.
* Type of manufacturing system, i.e. job shop, batch, high volume production, group technology layout or automated production lines.
* Cost of standardization.

Design Factors

* The development resources available influence the number of standard sizes that can be designed.
* "Bought out" to "made in" component levels in the product design.
* The degree of standardization between product sizes.
* The suitability of the product to modularisation.
* Level of material, labour and overhead costs.

Corporate Factors

* New product range development time scale.
* Corporate policy.
* Long term objectives for the product range.
* Labour policy, i.e. degree of flexibility in labour allocation.
* Working capital constraints tend to limit stock and work-in-progress levels therefore reducing the number of standard sizes that could be manufactured.

Table 2.1 Subjective Criteria for the Assortment Problem

Each of the constraints listed in Table 2.1 must be examined for its relative importance to the assortment problem. For example, development time scale may be an important constraint during the development of a new product range if a market launch is required before a serious loss occurs in the sales levels of existing products.

There are several techniques for solving the assortment problem, i.e. dynamic programming and the Minaddition technique which attempt to determine the number of standard models and their respective design specifications that minimise manufacturing costs. In addition, the use of complete enumeration or random enumeration could be possible.

Complete enumeration for example involves the analysis of all possible combinations of standard models to determine the relative savings obtained from each, with the solution yielding the maximum benefits being adopted. However, the number of possible combinations to consider increases exponentially with a rise in the number of possible models and the number of models adopted as standard. Random enumeration of a limited number of combinations is possible but difficulties arise in determining how close to optimal any trial solution is. Selecting combinations based on intuition and experience is also liable to the same uncertainty.

2.2.1.1 Dynamic Programming

Swanson (1970), Jackson and Zerbe (1968) and Martin and Piff (1971) have used Dynamic Programming (DP) algorithms to solve the assortment problem. The underlying philosophic argument of the DP technique is known as the "principle of optimality", which states that an optimal policy has the property that whatever the initial state and initial decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. The assortment problem is, therefore, represented as a sequential decision which is solvable in stages and where the first stage is to specify the initial standard model, e.g. the largest size or largest volume. This decision is normally a marketing or manufacturing constraint, e.g. the largest size or volume required to cover the intended market segment or the maximum size/volume that can be manufactured due to component dimension limitations on production equipment. The subsequent stages involve selecting each individual standard in order of decreasing size or volume.

Jackson and Zerbe (1968) initially solved the assortment problem using the DP technique but according to Swanson (1970) they specified the problem illogically since the solution required the pre-determination of the number of standard models, i.e. this is a decision variable and cannot, therefore, be pre-specified. The solution also determined which particular models should become standards whilst leaving the other models as custom designed. However, the overall objective of determining the optimum number of standards is basically to reduce the number of models produced by manufacturing.

Taking these factors into account Swanson (1970) found it necessary at the initial decision stage to examine all feasible single standards and at the second decision stage examining all feasible double standards. This methodology requires extensive amounts of computation when considering even small numbers of standards.

A disadvantage of using the DP technique to solve assortment problems is the difficulty of including in the decision making process non-quantifiable criteria such as those listed in Table 2.1. This problem has been highlighted by Martin and Piff (1971) when attempting to determine the optimum number of fork lift trucks to manufacture. The DP solution to this problem considered factors such as manufacturing quantities, range of sizes possible, manufacturing costs and sales prices and recommended that 22 standard fork lift truck sizes should be manufactured. The subjective estimate of management indicated that only 5 standard sizes should be produced. The large variation in results indicated the effect that the subjective criteria, listed in Table 2.1, have on the solution to the problem.

2.2.1.2 The Minaddition Approach

The Minaddition technique has been used, Wolfson (1965), Stockton (1983) and Pentico (1988) to solve the assortment problem assuming that any required model not a standard can be provided from the next standard model up the product range. This constraint results in excess material and labour costs being required and the Minaddition technique is used therefore, to select the particular set of standard models that minimises these excess costs. When profit margins vary with product model then the problem can be respecified to minimise total profit loss. Using the Minaddition technique the assortment problem, therefore, resolves into two distinct stages, i.e.:

1. Use the Minaddition technique to determine the cost or profit loss from manufacturing various numbers of standard models to include in the product range.

2. Determine the extra costs incurred from increasing the number of standards in the product range.

The optimum number of standards to manufacture, therefore, is that number where the savings obtained from adding one extra standard is considered unjustified when compared with the extra costs and manufacturing problems involved, e.g. reduction in manufacturing efficiency, higher work-in-progress levels.

Initially the data to be obtained includes the range of standard models that are being considered, forecast sales quantities for each standard and the "cost" of each standard. The type of "cost" determined depends on the specific circumstances of each problem and may be profit per unit, manufacturing costs per unit, direct costs per unit, labour cost per unit or material cost per unit.

The Minaddition technique considers the range of possible standards in sections and temporarily regards the models that bound each section (termed "interval") as standards. The total prime costs for each interval are then calculated in stages, i.e. the cost that results when all other models in the interval are provided from the larger of the two bounding models.

The prime costs in any interval would, therefore, remain the same no matter which models greater or smaller than the interval bounding models were varied. Costs are denoted by:

$C_e = Aw\ (S_{min},\ S_{max})$ Equation 2.1

Where:

C_e, A = Prime costs (£'s),

S_{min} = The lower bounding model,

S_{max} = The upper bounding model, and

w = The number of standard models in the interval.

The objective of the Minaddition technique is, therefore, to minimise C_e for successive values of w.

Since the Minaddition technique determines the prime costs that arise from manufacturing a fixed number of standard models the cost saving resulting from adding one extra model can be determined using Equation 2.2.

$C_{sv} = Ce_p - Ce_{p+1}$ Equation 2.2

Where:

C_{sv} = Cost saving obtained from adding one extra standard model (£'s),

Ce_p = Prime costs for p standard models (£'s), and

Ce_{p+1} = Prime costs for p+1 standard models (£'s).

Having determined the cost savings it is necessary to compare them with the increases in manufacturing costs that are incurred through manufacturing additional standards. These cost increases arise primarily since:

a. increasing the number of standard models within a product range effectively lowers the average manufacturing and purchase batch sizes, hence incurring additional purchase and setting-up costs,

b. the increased number of batches that require processing effectively lowers the machine utilisation level,

c. increased costs are incurred for special tooling, e.g. moulding tools, and jigs and fixtures,

d. the efficiency of production planning and control systems could be lowered hence labour costs could be increased, e.g. extra idle time is incurred,

e. work-in-progress and stock levels increase, and

f. labour costs and scrap levels cannot easily be reduced by taking advantage of the "learning curve' effect, Sims (1975).

In order to determine the optimum number of standard models to manufacture it is necessary to compare the cost savings with the added costs involved in increasing the number of standards. The optimum number occurs when $C_{SV} < C_a$ where C_a represents the additional costs incurred.

2.3 Facilities Layout

2.3.1 Functional Layout

Tompkins and White (1984) estimated that material handling costs in a functional layout constitute up to 80 % of total manufacturing costs. In order to reduce such costs, therefore, effective facilities layout should attempt to minimise the distances that materials travel between successive operations. Poor facilities layout may also reduce the ability of parts to arrive at workstations on time, thus reducing throughput and increasing WIP.

2.3.1.1 Layout Planning Methods

There are a variety of manual methods available for determining plant layout designs of which travel charting (Wild 1989) is perhaps the most widely known. However, these methods are non optimal and time consuming to use for problems of a practical nature.

The quadratic assignment problem (QAP) has been used to model the machine layout problem, with Sahni and Gonzalez (1976) illustrating that the QAP is NP complete. However, Finke and Kuisak (1985) identified that the largest quadratic assignment problem that can be solved optimally is 15 facilities x 15 sites. Hence problems of a practical size cannot be solved using this procedure.

For larger problems, therefore, heuristic algorithms have been developed that are capable of yielding good but non-optimal solutions. Kusiak and Heragu (1987) have classified these heuristic methods as construction methods, improvement methods, hybrid methods and graph theoretic methods.

Construction methods produce solutions by assigning facilities to a shopfloor area until the layout is complete. For example, CORELAP (Lee and Moore 1967) develops an acceptable layout for departments or individual items of processing equipment from preferable workcentre relationships supplied by the user. This method uses a "closeness" ratings scale of A, E, I, O, U or X (Muther 1955), in which a numerical value is assigned to each. A total "closeness" rating is calculated for each workcentre for its relationships with other workcentres. The workcentre with the highest "total closeness rating" is then selected and placed at the centre of the layout. The remaining workcentres are examined in order to identify workcentres which have an 'A' relationship with the original workcentre selected. From those workcentres identified, the one with the highest total closeness relationship is selected and placed next to the machine with the largest total closeness rating. When all machines with 'A' relationships have been placed, the process is repeated using the machine with the next highest "total closeness rating" until all workcentres with 'A' relationships have been placed. The process is then completed using E, I, O, U and X relationships until all workcentres are placed. Other construction methods include ALDEP, Seehof and Evans (1967), HC66, Hillier and Connors (1966) , RMA CompI, Muther and McPherson (1970), MAT, Edwards, Gillet and Hale (1970) , FATE, Block (1978), INLAYT, O'Brien and Abdel Barr (1980) and PLANET, Deisenroth and Apple (1972).

Improvement methods, such as CRAFT (Computerised Relative Allocation of Facilities Technique) Armour and Buffa (1963), require an initial solution, which may itself be generated by a construction procedure. This initial solution is then subjected to modifications in which systematic exchanges of pairs of workstations take place that enable costs to be reduced until no further cost reduction can be attained. CRAFT relies on the principle of pairwise exchanges in order to improve the initial solution provided by the users and requires an interdepartmental flow matrix, interdepartmental movement cost matrix, initial layout and information about any practical restrictions that apply. After the program interchanges pairs of workcentres, the total cost of the new layout is calculated. If this new layout achieves a lower cost then this layout is used as the next solution to be modified.

2.3.2 Product Based Layout Design

Here the basic design task is to allocate work tasks to work stations such that:

a. precedence constraints are not broken,

b. the line is balanced such that idle time throughout the line is minimised,

c. equal amounts of work are allocated to work stations, and

d. the total work task time allocated to workstations is not greater than the cycle time.

When designing flow lines, therefore, constraints exist in the areas of production rates required, sequencing of tasks and grouping of tasks. For example, precedence constraints would influence where work elements can be allocated, e.g. these may be limited by zoning constraints. In addition, constraints may require that work elements are placed near each other or exclude work elements from being placed together. Although many constraints do exist, the design of product flow lines is a combinatorial problem since there are a wide variety of feasible alternative ways in which work tasks can be allocated to work stations when the duration of operation times, precedent relationships between tasks, resource allocation, work balancing and cycle times need to be considered.

Production forecasts are used to determine the rate at which parts must be produced, i.e. Equation 2.3. For example, if 2000 components need to be produced daily and assuming 8 working hours/day, the cycle time for completion of a single component is 4.16 minutes.

$$C = \frac{T}{N} \qquad \text{Equation 2.3}$$

Where:

N = the number of parts to be produced,

T = time available to produce N parts (minutes), and

C = the cycle time (minutes).

The minimum number of workstations can be calculated using Equation 2.4 and the balancing loss, i.e. the amount of idle time, calculated using Equation 2.5. In any real assembly line there will be a balancing loss because of the constraints discussed previously.

$$\eta_{\min} = \frac{n \cdot \sum t}{T} \qquad \text{Equation 2.4}$$

$$\text{Balancing Loss} = \frac{n(c) - \sum t}{n(c)} \cdot 100 \qquad \text{Equation 2.5}$$

Where:

η_{min} = the minimum number of work stations,

n = actual number of work stations,

t = total work content,

c = cycle time (mins), and

T = time available to produce N parts (Minutes).

2.3.2.1 Solution Techniques

Many approaches to the line balancing problem have been proposed. For example, Salveson (1955) and Bowman(1963) proposed solutions that made use of linear programming (LP). However LP has been found to be impractical for designing large flow lines which contain large numbers of workstations. To overcome these size limitations, heuristics methods have been developed by Kilbridge and Wester (1961), Arcus (1966) and Helgerson and Birnie (1961) who developed the Ranked Positional Weight technique (RPW). All heuristic methods, including the RPW method are approximate methods which although yielding good solutions are, however, non optimum (Wild 1989). Using the RPW method, tasks are initially prioritised according to their position in the overall processing sequence. The tasks are then assigned to lowest numbered feasible workstations in accordance with their priority.

A computer based heuristic method, COMSOAL (COmputer Method of Sequencing Operations for Assembly Lines), Arcus (1966), randomly generates solutions and hence can be used for a wide variety of layout problems. COMSOAL generates sequences by randomly selecting an individual task from the set of available tasks and placing the selected task next in sequence. Since measures such as accumulated idle time, available time remaining in the current workstation and, unassigned tasks are continuously monitored, only tasks that satisfy all constraints can be considered at each step. The generation of a sequence is discarded as soon as the number of workstations exceeds the number in the current best solution. The technique is easy to program and feasible solutions are found quickly. This method, however, generates random sequences which may lead to the re-examination of previous sequences hence wasting resources. By generating random sequences, the method may move around the set of possible sequences and is, therefore, likely to find a good solution.

2.3.3 Mixed and Multi Model Assembly Lines

When a variety of products are produced using multi-model or mixed model lines, three further planning decisions need to be answered i.e.:

1. How will the line be balanced ?

2. What are the batch sizes of the models ? This applies to mixed model lines only since multi-model lines normally produce in batch sizes of one.

3. What is the sequence in which the models will be introduced onto the line ?

Both mixed-model and multi-model assembly lines allow more than one model or model variations to be produced simultaneously. The major disadvantage of such a system is the differing work contents of each model, which can result in uneven flow of work through the line resulting in station idle time and work-in-progress.

The problem of sequencing is concerned with the ordering of models onto the line and also the time interval between placing the models onto the line. The objective of sequencing work through the line is to reduce both station idle times and the amount of semi-finished work-in-progress.

When launching product models into the line two basic methods are available, i.e., variable and fixed rate. In variable rate systems, the time interval between launching successive models onto the line is equal to the station cycle time of the model type that is produced in the largest quantities. In fixed rate systems, models are launched onto the line in an interval equal to the maximum time available to produce the models divided by the number of products being produced.

Both variable and fixed rate methods assume that the flow lines are 100 % reliable and that processing times are deterministic. However, in a practical situation this would not be the case since workstation loads may be slightly unbalanced because of the variability in processing times between machine types. In such cases, WIP buffers are often provided to cushion the effects of such processing time variability. Using inventory buffers, ensures that delays at workstations do not immediately affect the operation of subsequent stations. The determination of optimum buffer stocks, Young (1967), is hence essential to avoid excess inventory costs.

2.3.4 Cellular and Group Technology Layouts

2.3.4.1 Cellular Manufacturing

New technologies and manufacturing philosophies have been developed to help achieve the goal of greater effectiveness and efficiency. In this respect there has been a major emphasis on the introduction of cellular manufacturing layout techniques to replace traditional functional layouts. According to Gallagher and Knight (1986) cellular manufacturing involves identifying and grouping together components and processes that are related to take advantage of similarities which exist during all stages of design and manufacture. These manufacturing layouts have enabled the introduction of the Just-In-Time (JIT) philosophy to be established in batch manufacturing environments. Since its initial conception, (Flanders 1925), cellular manufacturing has been widely adopted throughout a wide variety of manufacturing industries.

Cellular manufacturing systems incorporate characteristics of traditional systems, i.e. general purpose machine types from job-shops, product flow layout from flow shops, small inventories from job shops and small in-process inventories from continuous flow process systems, Huang and Houck (1985). In essence, cellular manufacturing systems incorporate the best of other traditional systems to form an improved manufacturing system.

It has been observed that the important factor in designing and implementing manufacturing cells is to create the group in which the most expensive machines have a reasonably high utilisation at the expense of the less important machines which can have a relatively low utilisation, McManus (1980).

The benefits of cellular manufacturing have been identified as follows:

1. Reduction or elimination of set-up time, i.e. since parts are grouped into families which require identical or very similar production operations, the time required for re-tooling machines can be reduced or eliminated, Black (1983).

2. Reduction of material handling costs.

3. Potential for automating materials handling, i.e. Teresko (1980) states that if parts can be classified into families and machines into groups than the handling of parts during manufacture will become robot work. That is, with this approach, parts never become batches of anonymous WIP and their orientation for processing purposes can be maintained. Both these requirements are critical factors influencing the feasibility of robot machine loading and unloading.

4. In low volume/high variety batch manufacturing environments, each part is traditionally treated as unique from the initial design stage through to manufacture, Burbidge (1975), Edwards (1971), Ham (1975) and Gallagher and Knight (1986). However, by grouping similar parts into part families based on either their design characteristics or processes, it is possible to increase the productivity through more effective design rationalisation, data retrieval and manufacturing standardisation, Ham et al (1985).

5. Small lots can be produced economically and the system allows greater manufacturing flexibility, i.e. fixed costs of set-up are reduced and changes in production can be implemented with less change-over and set-up time.

6. Reduced lead times giving rise to:

    a. a faster response to customer demand,

    b. greater delivery reliability,

    c. work in progress and inventory are reduced,

    d. improved utilisation of resources increases output,

    e. less material handling is required,

    f. better resource utilisation,

    g. reduced inventory requires less space,

    h. better production planning and control,

    i. less variety of tools, jigs and fixtures and

    j. improved quality and less scrap.

7. Increased capability of economically justifying high capital investment machines

2.3.4.2 Classification Systems

Part family grouping is an important step for successful cellular manufacturing applications. In grouping part families, it is important to consider, production data such as lot-size, frequency, time, annual production plans, scheduling for optimum sequencing and machine loading. Four basic methods are used to form part families, i.e. Manual/Visual search, Nomenclatures/Functions, Production Flow Analysis and the use of Classification and Coding systems. Of these methods the two most commonly used methods are Production Flow Analysis and Classification and Coding systems since the other procedures are essentially manual/visual procedures which limits the number of parts that can be considered.

Component classification and coding systems (such as CINCLASS, MICLASS, CODE, BRISCH, OPTIZ and SALFORD) have been widely used and have been shown to be effective tools for the successful implementation of the Group Technology concept. The purpose of classification and coding systems are to enable the characteristics of a component to be numerically coded. Using these numeric codes, groups of components can be identified that possess similar design features (Ham et al 1985). When choosing a classification system, aspects that need to be considered are:

a. to predict the future growth and demands that will be placed upon the system, and

b. what will be the purpose of the classification system, e.g. primarily for group technology manufacture or for reducing the variety of parts processed.

2.3.4.3 Production Flow Analysis

When developing group technology cells, Production Flow Analysis (PFA), Burbridge (1975), uses actual processing data to group parts. PFA is, therefore, a direct route to the GT design solution, i.e. using other classification systems two parts may have identical shapes, but different dimensions and tolerances and therefore would not necessarily be manufactured using similar machine tools.

Production flow analysis has three stages; factory flow analysis, group analysis and line analysis. Factory flow analysis is concerned with the division of components or products into large groups that are to be made within individual manufacturing departments. Within each department Group Analysis is used to assign parts to GT cells. Line analysis then positions equipment within individual GT cells in order to achieve flow process production.

2.3.4.4 Clustering and Ranking algorithms

Clustering and ranking algorithms, McAuley (1972), create component groups using similar component/equipment matrices as used in Production Flow Analysis. Previous research (Wu et al (1986), Han and Ham (1986) and Kusiak, Chow and Wing(1987)) has concentrated on the development of heuristic methods which can take into account additional factors such as part quantity.

3. Dynamic Design Decisions

3.1 Introduction

Decisions that need to be made at this stage are essentially planning in nature and can be grouped by the length of the planning horizon as such:

a. medium to long term planning - aggregate planning,

b. short to medium term planning - master production scheduling, material requirements planning, capacity requirements planning and inventory planning and

c. short term planning - production scheduling, daily line sequencing.

3.2 Aggregate Planning

Aggregate planning takes expected sales demand and production capacity and translates this into future manufacturing plans for a family of products. The aggregate plan looks only at the production of families of products and is not concerned with individual products. A domestic goods manufacturer, for example, may produce both freezers and refrigerators in many different styles, colours and internal volumes. However, for aggregate planning purposes these may be placed into two groups, i.e. freezers and refrigerators. A typical aggregate plan is illustrated in Table 3.1 and normally consists of a number of monthly planning periods. The function of such a plan is to seek the best combination of manufacturing resources, (i.e. personnel, materials and processing equipment), and finished goods inventory that meets management and sales objectives and, in addition, reduces the overall costs of manufacturing.

Period	Opening Stock	Forecast Demand	Normal Production	Overtime	Sub Contract	Late Deliveries	Finish Stock
January	0	1500	1200	200		100	0
February	-100	1500	1000	200	200	200	0
March	-200	1000	1200	200			200
April	200	1500	1200		200		100
May	100	1500	1200		200		0
June	0	500	500				0
July	0	500	500				0
August	0	1500	1000	200	300		0
September	0	1500	1200	200	300		200
October	200	2000	1200	200	300	100	0
November	-100	1000	1200	200	300		600
December	600	2000	1000	100	300		0

Table 3.1 Typical Aggregate Plan

For each planning period it is necessary to forecast the sales demand for the aggregated product group and determine the normal production capacity that is available for achieving this demand. Planning periods vary between organisations but normally range between three and twelve months. The aggregate planning horizon must be sufficiently long to allow decisions, such as the hiring and layoff of personnel, to be optimal in the long term and not merely on a short term basis. The aggregate plan then forms the constraints under which detailed scheduling of facilities and personnel then proceeds. When variances exist between forecast and available manufacturing capacity, (i.e. too much capacity is available or too little is available), then decisions must be made concerning how to correct these differences.

The aggregate planning process is carried out at an aggregated level without the need to provide detailed material and capacity resource requirements for individual products and detailed schedules for facilities and personnel. This greatly reduces the amount of data used during the planning process and hence enables plans to be updated more frequently. Hence, changes occurring in factors such as forecast sales demand, labour cost rates, production capacity and raw material supply can be readily compensated for. In addition, when setting aggregate plans it is possible to focus on those resources that limit production capacity, i.e. bottlenecks.

3.2.1 Aggregate Planning Procedure

The basic steps involved in the development of an aggregate plan are shown in Table 3.2. Development of the plan initially begins with the identification of the long term objectives of the manufacturing organisation and the constraints under which the functional departments within the organisation must operate. These have a major affect on the feasibility of an aggregate plan. For example, an aggregate plan may require organisations to ensure that customers orders are always delivered on time, (i.e. strategic aim), but that inventory levels remain low, (i.e. operational constraint). Often such constraints conflict, for example in many organisations an overriding constraint when developing aggregate plans is the provision of stable and secure employment for the work force. However, this can often conflict with the inventory policy of maintaining low finished goods stock levels. A stable workforce size would enable stock to be built-up during periods of low sales demand and this stock then used during periods of high sales demand, i.e. when insufficient production capacity is available. The need to maintain low stock levels can be seen, therefore, to conflict with a stable workforce policy.

1. Determine relevant company and planning policies
2. Determine demand for each period
3. Determine production capacities for each period
4. Determine unit costs for labour, overtime, subcontracting and holding inventory
5. Develop alternative plans and compute the cost for each
6. Select the plan that best satisfies the objectives

Table 3.2 Aggregate Planning Steps

In order to perform the aggregate planning process, individual products are aggregated into groups according to similarities in such criteria as the types of processing operations required, production times required, types of labour skills required or the fact that each product in the group makes use of specific limited resources. The objective is to allow the capacity requirements of a product group to be measured in common units, such as "man hours of production time" on a specific item of processing equipment.

For each period within the planning horizon, forecasts of the sales demand for each product group are determined. Since accurate forecasting is essential at all planning levels, including aggregate planning, the accuracy of forecasts must be constantly checked. Aggregation of products may also be influenced by the accuracy and ease with which the sales demand for a group can be forecast. For each planning period and each product group it is then necessary to estimate the level of manufacturing resources required, (using the common resource units identified earlier), to produce the forecast volume of finished goods.

Aggregate planning is essentially a stochastic process since we are attempting to plan for events that will take place sometime into the future. Hence, the accuracy of sales forecasts will deteriorate the further into the future attempts to plan are carried out. In addition, estimates of the available resources will also be dependent on factors that are difficult to accurately predict, e.g. worker morale and motivation, level of operator experience and the processing reliability of equipment. The majority of these factors are themselves affected by the aggregate plan, e.g. operator morale is greatly influenced by hiring and layoff policies. The unpredictable nature of these variables results in a manager having to consider the trade-offs between costs of holding excess inventories, (i.e. when demand is lower then forecast), and the cost of lost sales, (i.e. when demand is greater then expected). The manager must, therefore, consider the relative costs of carrying stock from one period to the next, (e.g. obsolescence costs, interest charges, warehousing costs), and the effects of lost sales, (e.g. lack of revenue, lost customers), and choose the least cost alternative.

When variances exist between required and available capacity then one or more of the capacity management techniques listed in Table 3.3 must be used to achieve a balance. Choosing which technique to use, when to begin using the technique, the number of planning periods over which it will be used and the amount of production capacity to add or remove are the essential problems involved in generating an optimal and feasible aggregate plan.

Recruit/Layoff Staff/Redundancy
Work overtime/Shorter hours
Vary levels of inventory held
Vary lot sizes processed
Subcontract work
Transfer multi-skilled labour between work areas

Table 3.3 Capacity Management Techniques

The costs must be determined of the alternative methods that could be used to vary capacity. Examples of typical cost elements involved are listed in Table 3.4. In practice the options available for varying capacity will be limited by the need to meet such management policies as never running out of stock, allowing only an occasional stock-out or fulfilling customers orders within a specific time period. The planning process is also made more troublesome since many of the costs affected by the aggregate plan are difficult to quantify, e.g.:

- how will employee motivation be affected if additional staff are taken on without providing existing staff with the opportunity of working overtime,
- what will be the effects on manufacturing efficiency of machine breakdowns due to prolonged use of processing equipment, and
- how must the "learning effect" be taken into consideration when hiring new personnel.

Quantifiable Costs	**Non-quantifiable costs, that is costs arising due to**
Inventory	Stock-outs
Purchasing	Loss of orders
Overtime	Handling complaints
Subcontracting	Low morale
Shift premiums	Inexperienced personnel
Recruitment	Loss of efficiency
Redundancy	
Training	

Table 3.4 Cost Elements Involved in the EOQ Decision

3.2.2 Aggregate Planning Techniques

Nam and Logendran (1992) surveyed the range of techniques currently available for setting aggregate plans and classified each method in terms of their ability to produce either an exact optimal or a near-optimal solution. Here the optimality of a plan was measured primarily on the level of costs required to implement the plan.

Graphical techniques are described in Stevenson (1993), as the simplest to use and consist of developing graphs that enable planners to visually compare forecast demand requirements with future estimates of production capacity. Such comparisons are then used as a basis for developing alternative plans which can then be costed to determine their relative merits. The main limitation with this technique is that it does not normally result in optimal plans being identified.

The aggregate planning problem has been formulated and solved in terms of linear programming models, e.g. Bowman (1956) used a transportation-type linear programme. In order to use this approach it is necessary to identify for each planning period the amount and cost of overtime, regular time, subcontracting and inventory. Constraints involving limitations on the numbers of personnel, inventory levels and subcontracting costs can also be introduced into the linear programming technique which enables a more feasible solution to be obtained. In order to take into consideration the stochastic nature of the process data, one approach described in Feiring and Sastri (1990) has been to quantify the uncertainty in demand by a probability distribution and then formulate as a linear programming problem in which the expected demand is replaced by an appropriate fractile of the probability distribution for demand. This approach has the effect of converting the probabilistic nature of the problem into an equivalent deterministic situation. Other approaches have involved using sensitivity analysis to study the effects of inaccuracies in forecast demand, (which according to Taha (1971) represents only a partial answer to the problem), and stochastic optimal control, Love and Turner (1993), which is described as suitable when a significant amount of uncertainty is involved in the demand forecasts. A further approach that has shown promising results is the use of fuzzy logic to generate a crisp multiple objective linear programming model, Gen at al (1992). However, although the stochastic nature of the data can be taken into consideration, linear programming by its very nature must assume that costs are linearly related, e.g. as the number of personnel hired increases then the costs of hiring increase linearly. This assumption is frequently wrong and results in non-optimal plans being developed.

The well-known Linear Decision Rule developed by Holt et al (1960) uses linear equations as decision rules for specifying optimal production levels and work force size. Individual cost curves initially need to be developed that represent the cost relationships for hiring and laying off employees, the use of overtime, holding inventory and the costs arising through stock shortages. By differentiating these cost functions and combining them with labour costs, two basic rules can be obtained that can be used to define the size of the workforce for a planning period and the production rate for that same period. Although Kamien and Li (1990) introduced a third decision rule that enabled the effects of varying levels of subcontracting to be considered, the three rules are tedious to develop and do not perform well in situations where the assumed cost relationships are not representative of those that actually exist within a company.

The Management Coefficients approach, Bowman (1963), is a procedure that involves applying regression analysis to past aggregate planning decisions to generate formulae that can be used to calculate the costs of future decisions. The use of such formulae, however, are said to result in undesirable bias being set-up that weaken decisions made in previous periods, Riggs (1987).

The Search Decision Rule, Taubert (1968) procedure is a computer program that can perform a guided search through alternative aggregate plans until it finds the plan of minimum cost. The program must be capable of calculating the cost of implementing each plan found in its search. Unlike linear programming and the linear decision rule approaches, the Search Decision Rule is not restrictive in the type of cost relationships that can be used.

Other approaches to the aggregate planning problem have involved:

a. the use of a spreadsheet based simulation, Armacost (1990),

b. the use of heuristics, Gilbert and Madan (1991),

c. a model based approach incorporated within a decision support system, Vercellis (1991) and using multi-objective decision criteria, and

d. the use of fuzzy logic combined with multi-criteria decision rules that enable approximate reasoning to take place, hence enabling planning decisions to be made with imprecise and incomplete information, Satyadas and Chen (1992).

3.2.3 Limitations of Existing Methods

The disappointing use in the application within industry of existing aggregate planning techniques has been commented on by a number of researchers, Buxey (1993), Barman et al (1990), Vollman et al (1992) and Gilgeous (1989) and the primary reasons have been identified as:

a. Each method has been observed to be "situation dependent", i.e. it is only suitable for a limited range of planning situations. Hence identifying the most appropriate method to use can, therefore, present problems to production planners. Models are also not directly transferable to other planning situations, a limitation which can present difficulties when the conditions within which an organisation operates changes due to such factors as increases in material and labour costs and increased competitive actions in the market place.

b. None of the existing aggregate planning techniques can identify optimal or near optimal plans for real world problems that involve a range of planning variables. Also those techniques that can identify optimal plans do so by achieving only cost related objectives whereas many other non-cost objectives are often sought by managers. In addition, within many organisations the cost relationships used by these methods do not adequately represent those that actually exist. The mathematical procedures used by existing methods are also complex and difficult for manufacturing management to understand, hence management are reluctant to use such techniques.

c. Models require specific types of data items that are difficult to collect and quantify. Often, therefore, this information does not exist in a readily used format. Examples of such information include the costs of recruitment and training of new staff, effects of redundancy on employee morale, productivity reductions resulting from working longer hours due to overtime and length of time required for a new employee to become fully productive in terms of both output rate and quality.

Disaggregation is a particular problem that has been commented on by several researchers, Saad (1990), Miller (1991) and Azoza and Bonney(1990), and is the process of using the aggregate plan to generate detailed purchasing and manufacturing schedules for each product within the aggregate group. Existing aggregate planning techniques generate plans that may not be realistic when the problem of disaggregation is considered, i.e. implementation of aggregate plans may depend on the expertise of managers.

3.3 Material Requirements Planning

3.3.1 Existing MRP Lot Sizing Techniques

Material Requirements Planning is now an accepted method of planning the material requirements for items whose demand is dependant on higher level items within the Bill of Material, (BOM), structure.

The planning process starts with a production schedule and then applies MRP logic through BOM's and inventory records. The planned order releases that are output from the MRP process are those required to provide the quantity of items necessary to satisfy planned production requirements within each time bucket. Ideally MRP must be capable of producing a planned order receipt schedule such that this schedule minimises the total costs of holding and purchasing inventory over the entire planning horizon. In addition, the inventory demands for each period within that planning horizon must be met such that there are no stock outs in any period.

The aim of the MRP process is to determine the orders that need to be released in order that planned production can take place. Two items of information constitute a planned order release for a particular item i.e., an order size, and an order release date.

Although the logic underlying the MRP process is simple the results of the process can have a profound effect on the profitability of an organisation. In this respect the lot sizes chosen for the highest level components determine the quantities of lower level items required. Complex demand relationships can often exist between such items.

There are a wide variety of MRP lot sizing techniques that are currently available. Wagner and Whitin (1958) for example developed a procedure using dynamic programming as early as 1958 which is still recognised as one of the only methods available that can consistently generate least cost lot sizing conditions. Techniques vary from simple decision rules, (in which little effort is made to optimise the decisions made), to complex procedures that attempt to develop optimum planned order release schedules. Lot sizing techniques in current use are of three basic types (Nydick and Weiss, 1989), i.e.:

1. Based on the Economic Order Quantity, (EOQ), model. Such models include the use of the EOQ model to determine either a specific lot size to purchase or a frequency with which batches should be purchased, i.e. the Periodic Order Quantity, (POQ). The POQ model determines the time when an order should be placed and the actual quantity ordered is that quantity required between the placement of two successive orders. The use of the EOQ method results in a constant batch size being ordered whereas the POQ method allows the ordered batch size to vary.

2. Single Pass methods, of which there are a wide variety, are essentially ruled based. Each individual method applies a unique set of rules to determine schedules for order releases. Such methods include the Lot-For-Lot rule, (LFL), Least-Total-Cost, (LTC), Least-Unit-Cost, (LUC), and many others such as those developed by Silver and Meal (1973), Groff (1979), Freeland and Colley (1982) and Bahl and Zionts (1986).

3. Adjusted Single Pass methods initially make use of a specific single pass technique and then attempt to improve on the solution produced by this technique. This is achieved by examining the total costs involved in either increasing batch sizes such that the materials for one or more additional periods may be purchased, (i.e. termed "look ahead"), or decreasing purchase batch sizes to remove periods from the decision making process. If the total costs can be reduced by these actions then the new batch sizes are adopted.

Choosing which technique to use from the wide variety available is a major problem for materials managers since each technique will only yield acceptable results under a limited range of demand and inventory cost conditions. Recognising the need for materials managers to choose between the many techniques available Berry (1972) developed a framework for comparing such methods in terms of their ability to minimise inventory related costs over a range of cost and demand parameter values. In addition Berry proposed that this framework should also include comparisons of the amount of computing time required to make lot sizing decisions and estimates of the simplicity of the procedures used. Since this framework became available a number of researchers Axsater (1986) and Callarman and Hamrin (1984) have provided comparisons of existing techniques over a wide range of conditions whilst others Coleman and McKnew (1991), Aucamp (1985) and Benton (1985) have concentrated on developing improved lot sizing algorithms. However, the techniques developed have again been limited in their application areas. Gaither (1981) for example presented a technique that identified near-optimal MRP lot sizes and possessed procedural simplicity. However, this model was found to exhibit a built-in bias towards larger batch sizes when high "ordering cost to carrying cost" ratios existed. The procedures used, therefore, had to be modified, Gaither (1983) to compensate for this defect. This indicates that the limitations involved in using existing lot sizing techniques are not always immediately obvious.

3.3.2 Limitations of Existing Lot Sizing Methods

Existing methods have many fundamental problems that limit their usefulness in practical situations. For example, those methods that seek optimal schedules do so by considering all available options. As the MRP planning horizon grows larger the number of alternative schedules that need to be compared dramatically increases. This results in excessive computations being required. In order to obtain optimal schedules, therefore, MRP planning horizons must be limited. Hence, optimal short term schedules are obtained but these individual schedules do not necessarily result in optimisation of inventory over the long term. Generating optimal or near optimal schedules requires the use of complex procedures that are often difficult for operating personnel to understand. Their use within manufacturing industry is often limited for this reason.

A critical appraisal was carried out by St John (1984) who highlighted major misconceptions in the evaluation of existing lot sizing techniques. He argued correctly that the majority of the current methods in use are not applicable to MRP since:

1. They treat the lot sizing problem as a single stage process, i.e. determine lot sizes for single items. MRP is, however, a multi-stage process and any lot sizing techniques must take into consideration the relationships between items. Such methods must consider all items whose demand is related, both horizontally and vertically, to each other via the BOM structures. MRP items, therefore, exhibit dependent demand patterns whereas current lot sizing methods apply only to independent demand patterns.

2. Each individual technique is valid, (i.e. produces acceptable order schedules), under a particular set of conditions. The effectiveness of a technique for example is strongly dependent on such factors as the variability in the sizes of individual material requirements, the variability in the frequency of requirements, and the relative values of carrying and purchasing costs. In order to achieve good results, therefore, it is necessary to select with care the most appropriate method from amongst the many available. Since the demand and cost parameters for individual items within a BOM may vary then ideally the most appropriate method for each item should be selected. This is obviously impossible due to lack of resources and in practice a specific lot sizing technique is often applied to a group of items irrespective of how suitable it is for individual items within that group. Demand over time is not necessarily constant, i.e. lumpy demand patterns can exist. This is particularly true when a company's products exhibit seasonal bias in which case there can be months when there is no demand, months when there is little demand and months when demand reaches its peak. Choosing a specific lot sizing method for each item/demand period is again clearly impractical since regular checks would, therefore, need to be carried out to ensure that changes had not occurred that adversely effected the suitability of the technique being used.

3. All existing methods use costs to measure how effective a specific lot sizing policy is. Many methods place restrictions on the types of costs that are considered. In general the emphasis is on minimising the combined order/set-up and carrying costs. It is uncommon, when determining lot sizes, for non-cost variables such as the availabilities of working capital for purchasing stock or warehouse space to be considered. Hence, situations can

occur in which there is insufficient working capital or storage space to support the purchasing decisions made by a MRP system.

4. MRP systems often provide a range of lot sizing techniques. Choosing which technique to select often looks a routine decision since the likely results of such a change are not adequately identified. Changing lot sizing techniques in this way can often lead to disastrous consequences in particular excessive stockholding costs being incurred.

5. The use of existing dynamic lot sizing techniques can lead to system nervousness which occurs when relatively minor changes in the order schedule of a higher level component causes significant changes to the order schedules of lower level items. MRP nervousness is, therefore, the amplification of minor changes which create such conditions as late orders. Vollman et al (1992) suggested that the use of different lot sizing techniques at different levels in the BOM structure would assist in reducing the effects of nervousness. However this would result in the choice of a suitable lot sizing technique becoming even more difficult.

Despite St John's condemnation of existing lot sizing techniques, research has still focused on the comparison of lot sizing techniques and the development of more efficient single item, independent demand methods. Nydick and Weiss (1989) for example compared ten lot sizing techniques and evaluated them using a variety of demand and order cycles. They concluded that "the greater the variability in the pattern of demand and the more frequent the order cycle, the more erratic is the behaviour of all lot sizing techniques in determining the optimum conditions".

3.4 Kanban Material Control

Kanban control systems have been identified as one of the main characteristics of Just-In Time (JIT) (Finch and Cox (1986) and O'Grady (1988)). When Kanban systems are being designed then the decisions that need to be made at the dynamic design stage are:

a. the re-order level, i.e. the quantity of items remaining in the kanban areas/containers that signals the need for processing a further batch,

b. number of kanbans, and

c. the quantity of items that constitute a processing batch.

O'Grady et al (1989) identifies the major problem areas in implementing and operating a kanban system as,

1. the identification of flow lines,
2. the flowline loading problem, and
3. the operational control problem.

Philipoom, Rees, Taylor and Huang (1987) identified three factors in the production area which influenced the number of kanbans required i.e. container demand cycle, cycle processing time and cycle throughput velocity.

3.4.1 Number of Kanbans

The main problems in operational control are determining the number of kanbans and the batch sizes to be produced. Monden (1983) developed the formula shown in equation 3.1 to determine the number of kanbans to be used in a system.

$$\text{Number of Kanbans} \geq \frac{D \cdot T_1 \cdot S_f}{C_c} \qquad \text{Equation 3.1}$$

Where:

D = demand per unit time,

T_1 = processing time + waiting time + conveyance time + kanban collecting time,

S_f = safety factor, and

C_c = container capacity.

This equation does not take into consideration the variation or uncertainty involved in manufacturing, such as the processing reliability of machines and the uncertainty of customer demand. Philipoom et al (1987) identified that throughput velocity, the variation in processing times, the machine utilisation and correlation of processing times, all had significant effects on the optimum number of kanban systems. Current methods of calculating kanbans do not allow such uncertainties to be included in the determination.

3.4.2 Processing Batch Size

Processing batch sizes are a fundamental factor that determines the efficiency of a manufacturing cell, i.e. they influence:

a. throughput time of the system, and

b. work-in-progress.

In conventional manufacturing systems, the processing batch size is normally calculated using Economic Order Quantity (EOQ) models.

Economic Order Quantity models are in widespread use throughout industry for determining the quantity of an individual item to purchase from suppliers or to process through a production facility. There are a variety of models available and all originate from the classical Economic Order Quantity model developed in the early 1900's, Evans et al (1990). This procedure requires the costs associated with the procurement and holding of stock to be identified and then quantified. A model must then be developed that relates the total costs of a specific lot sizing policy to the costs incurred in operating that policy. Differential calculus is then applied to derive the model shown below which enables a lot size to be calculated that minimises total costs. This model, therefore, identifies the minimum point on the "total cost" versus "lot size" curve.

$$Q^* = \sqrt{\frac{2CpD}{Ch}} \qquad \text{Equation 3.2}$$

Where:

Q* = Economic Order Quantity,

D = Annual demand,

Cp = Cost of placing an order, and

Ch = Holding costs per item per annum.

Hence the EOQ (Equation 3.2) decision involves identifying a minimum cost compromise between:

1. Maintaining low inventory levels, (low inventory holding costs), and placing frequent replenishment orders, (high ordering costs).
2. Maintaining high inventory levels, (high inventory holding costs), and placing replenishment orders infrequently, (low ordering costs).

The cycle with which stock is assumed to be used and replaced is illustrated in Figure 3.1.

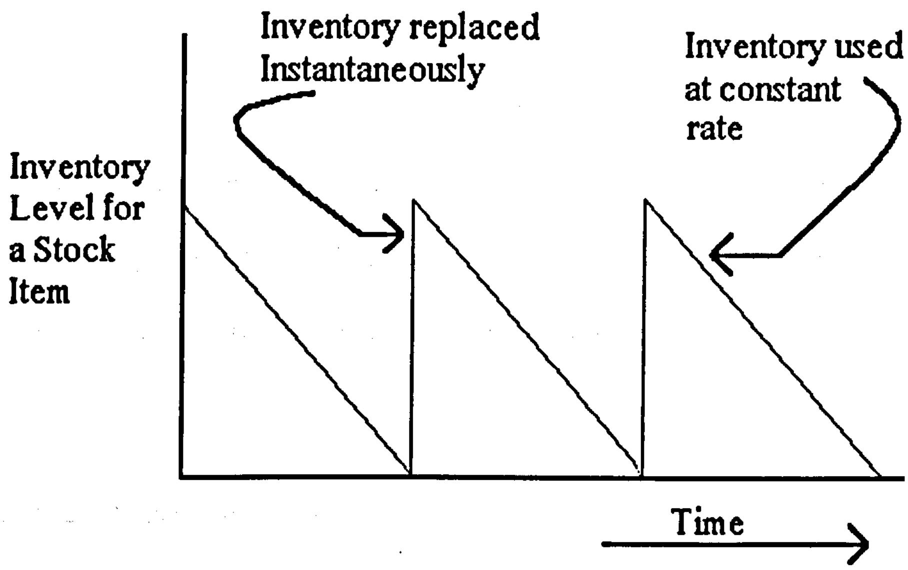


Figure 3.1 Inventory Replenishment/Demand Cycle

This pattern of demand and supply is used to determine both the quantity of items that need to be replenished and the point in time when new procurement orders need to be placed with suppliers.

3.4.3 Reorder Quantity

From the EOQ model the reorder point, i.e., the level to which stock may fall before a further order needs to be placed, is determined as such:

$$R = D \cdot M \qquad \text{Equation 3.3}$$

Where: $R =$ reorder point, and

$M =$ lead time for a new order to arrive

This equation (Equation 3.3) assumes that the reorder point should be equal to the amount of items that the company would expect to use during the time taken from placing an order to the delivery of the items. The calculation of optimum reorder levels is essential within a JIT system to ensure that material shortages do not occur or excess WIP is not produced.

3.4.3.1 EOQ Assumptions

As stated, to derive the EOQ model, Figure 3.1 is used to represent the cycle with which stock is replenished and used from stock points. The basic assumptions made in order to develop and make use of the EOQ model, therefore, are:

a. demand rates for finished products have little variability, hence the rate at which stock is used is constant or nearly constant,

b. the lot size ordered is supplied to the stock point at one point in time, i.e., instantaneously,

c. the replenishment/usage cycle is repetitive, i.e., conditions remain constant over the period the economic order quantity is in use,

d. no stock-outs are allowed, i.e., the order quantity and reorder point are set to ensure that there is always sufficient items in stock to supply customers, and

e. since the EOQ is used to determine reorder points then an assumption must be made that the lead time for this replenishment is constant.

An essential assumption that must be made concerns the holding and order costs that are used in the total cost model. It must be assumed that all the relevant cost elements, (Table 3.5), that make up these two basic cost types are known and can be quantified. In addition, the relative effect of each cost element must be assumed to remain the same across the batch size range covered by the total cost model.

Holding Cost Items	Ordering Cost Items
Cost of Capital	Purchasing departmental salaries
Insurance, lost goodwill, lost orders	Clerical support staff wages
Taxes	Transportation Costs
Breakage, obsolescence	Miscellaneous costs of paper, postage
Pilferage	Telephone Charges
Warehouse overhead costs	Lost production cost, e.g. equipment set-ups
Bulk factor	

Table 3.5 Main Costs Items Involved in the EOQ Decision

Normally holding cost rates are calculated as a percentage of the value of inventory because of convenience when a range of items are involved. Many of the items that make up the ordering cost rate would be estimated using some form of work measurement technique.

It is widely accepted that if reasonable estimates of the costs involved are provided then good approximations of the minimum cost order quantity can be determined. However, Woolsey (1988), analysed the total annual cost model used to generate the EOQ formula and highlighted the problems in estimating accurately each variable within the equation. He highlighted such problems as:

a. inflation causing instability in prices,

b. holding cost rates being dependant of both the market demand for a product and the product's profit margin, and

c. the problems in forecasting demand.

3.4.3.2 Model Variety

The need for the assumptions made, and therefore, the model used, to accurately reflect the real life situation the EOQ model is being used under has resulted in individual EOQ formulas being developed for specific situations. Hence there are a variety of EOQ models many of which are listed in Table 3.6. Each of these models is only applicable under a narrow set of conditions and models must be chosen that take into consideration the variables that have the greatest effect on the total inventory costs.

Because of the wide diversity of conditions that can exist in the states of the variables influencing the EOQ, (Table 3.7), it is difficult to correctly identify the most appropriate model for the inventory conditions that prevail. Indeed it is often difficult to identify the conditions that prevail for individual stock items. The fact that conditions certainly change over a period of time adds to the complexity and difficulty in using traditional EOQ models.

In addition, new criteria are being introduced into the inventory decision. For example, Fagan (1991), points out the need to consider the problem of exchange rate fluctuations when companies are involved in global sourcing.

Deterministic Models	Probabilistic Models
Single item, static model	Continuous review model
Single item, static model with quantity discounts	Simple period models
Multi-item, static model with storage limitations	Instantaneous demand with no set-up cost model
Single item, N-Period Dynamic model	Uniform demand with no set-up cost model
Single item, N-Period Dynamic model with constant or decreasing marginal costs	Instantaneous demand with set-up cost model
N-Period Dynamic Production Scheduling Model with no shortages	
N-Period Dynamic Production Scheduling model with shortages allowed	Miscellaneous multi-period models with combinations of backlog, no backlog, zero delivery time, positive delivery lag

Table 3.6 EOQ Models

The situation is increasingly confusing to the user particularly so since models are still being developed. For example, Chyr et al (1990), have developed an extension to the conventional EOQ model to incorporate the costs arising through the damage of items held in stock. In this situation conventional stockholding costs are understated and the conventional EOQ lot size needs to be reduced by an amount which is substantial if damage rates are high. Again a further model has been developed by Joshi (1990), which considers the disproportionately high warehousing costs involved when holding stocks of bulky, inexpensive, low risk items.

Variables	States
Order lot size	Ordered equal to received Ordered not equal to received
Demand rate	Constant, Variable, Known, Unknown, Step Function
Order lead time	Constant, Variable, Known, Unknown, Step Function
Replenishment time	Constant, Variable, Known, Unknown, Step Function, Instantaneous, Non-instantaneous
Number of items	Single item, Multi-items
Number of periods	Single period, Multi-periods
Environmental factors	All items do/do not incur the same level of such costs as: Obsolescence, Storage, Pilferage, Damage
Annual demand	Constant, Variable, Trends Turning points on the market life cycle
Usage and Supply cycles	Repetitive, Non-repetitive
Stock-outs	Allowed, Not allowed, Safety stock required
Stock-out costs	Incurred, Not incurred, excessive, low, constant, variable
Item price	Constant, Quantity discounts, Varies over period of time
Importance of individual cost elements	Equal, Variable
Replenishment and demand periods	Simultaneous, Non-simultaneous
Demand rate(DR) versus supply rate (SR)	DR > SR, DR < SR, DR = SR
Order costs	Different for each item, supplier, machine
Holding costs	Different for each item
Constraints	Various, Known, Unknown

Table 3.7 Variables Influencing The EOQ

Two factors that have the greatest influence on throughput time for cell manufacture are the total set-up time for all machines in use and the machining time for that batch. These factors are not considered in the conventional EOQ equation. Boucher (1984) overcomes this limitation by defining the optimum batch size for group technology as shown in Equation 3.4.

$$Q = \frac{((2 \cdot A \cdot D + D \cdot S)(i \cdot S \cdot R))}{(M + m \cdot R) + 2 \cdot i \cdot D \cdot m(M + m \cdot \frac{R}{2})}$$ Equation 3.4

Where:

Q = optimum batch size,

A = set-up cost per set-up,

D = annual demand for parts in units,

S = the sum of set-up times for all machines in the cell i,

i = the carrying cost rate,

R = rate charged per unit of cell production rate,

M = raw material, and

m = total machining time per unit product.

Boucher (1984) found that this order quantity would be useful when annual demand was high and/or machining times were substantial. As a guide to usefulness, he determined that if the product of D and m is greater than 0.5 then Equation 3.4 would be applicable only if m was expressed in years.

Karmarke and Kekre (1989) found that the batch size associated with each kanban signal i.e. the card (or container size) had a significant effect on the performance of a kanban system and that there were interactions between card numbers and batch size.

4. Decision Making Methods

4.1 Introduction

In order to evaluate a proposed manufacturing system design prior to its implementation, a model is normally used (Ravindran, Phillips and Solberg (1987)). Models may be classified as either physical or mathematical types where physical models are simply two or three dimensional scale models of the proposed system and have limited ability in enabling optimum designs to be established.

Mathematical models are developed using equations or logical relationships to describe the real system. In these models, parameters of the real system, such as production times and batch sizes, are incorporated as variables into the model.

Mathematical models, themselves, maybe classified as either, analytical or experimental models. Analytical models normally employ queuing theory or mathematical programming. Experimental models are essentially simulation based that are capable of mimicking the real system under investigation.

4.2 Simulation Techniques

The simplest form of simulation involves the use of queuing networks, Solberg (1980), Suri and Diehl (1985) and Whitt (1983). Solberg (1976) developed CAN-Q (Computer Analysis of Networks of Queues) to model and analyse queuing networks, that assumes that service and transport times are exponentially distributed and that the service discipline is First In First Out (FIFO). Solot and Bastos (1988) further developed a queuing network to allow multiple pallet types to be modelled. Suri and Hildebrant (1984) developed MVA-Q, an extension of CAN-Q that enabled a variety of part types to be modelled. It can be seen from this work that the use of queuing models is inflexible since unique queuing equations need to be developed for each individual problem.

Simulation modelling, using software packages such as Witness (1991), ProModel (1993) and Simfactory (1990), has a wide variety of applications within manufacturing industry including inventory control, design of distribution systems, maintenance scheduling, design of queuing systems and the scheduling and design of manufacturing systems. In manufacturing, simulation is frequently viewed as an extension to operational research (OR) techniques since it allows problems to be modelled dynamically. The use of simulation models has increased since many variables within manufacturing systems prevent the use of OR based analytical models since they are stochastic in nature and complex interactions take place with other variables.

Although simulation has advantages, developing models can be time consuming and complex. In addition, the results of simulation experiments are output in the form of selected performance measures that are difficult to analyse in terms of identifying improvements to system designs.

4.3 Optimisation

Some of the most widely used models are those of maximisation and minimisation, normally referred to as optimisation. An optimisation problem consists of two parts; an objective function and constraints. The objective function is the function which has to be minimised or maximised. Constraints place limitations on the variables that must be satisfied. Optimisation methods are used to explore the region of operation and predict the way that the system parameters should be adjusted to bring the system to optimum. In manufacturing, the optimum for a problem could be the minimisation of the cost for the production of a part where the cost depends on a large number of interrelated parameters in the manufacturing process.

Although a variety of modelling tools exist to aid the manufacturing system designer, the information provided by these tools must be subjectively analysed in order to decide if improvements can be made. This analysis is often time consuming and often it is not clear from the data supplied by the modelling tools which change in input parameters will yield the maximum increase in the overall performance of the system.

In recent years, research has been carried out into the potential for incorporating intelligent decision making at this design stage in order to reduce the amount of subjective decision making required by the user. A general purpose decision making method should; produce the optimal solution, be robust, solve large problems, allow subjective decisions, allow constraints, allow conflicting constraints, allow for changes in circumstances, allow for uncertainty, allow integration of other problems, and produce alternative solutions.

The research in this area has examined the use of expert systems which capture the knowledge of design engineers and translates this knowledge into rules. Shannon et al (1985) and Shannon (1988) also identified the benefits of linking expert systems with simulation for solving manufacturing problems. Wang and Bell (1991) developed an intelligent user interface for a knowledge-based modelling system used for the design of flexible manufacturing systems.

A limitation when using expert systems is that they require knowledge of the problem domain and this knowledge must be extracted from domain experts. This process of knowledge elicitation is time consuming and expensive. A further limitation to the use of expert systems exist since they are unable to deal with problems outside their knowledge domain. Knowledge domains are also restricted by the amount of knowledge that experts have acquired about specific problems.

A number of alternatives such as Genetic Algorithms (Section 4.6) which require little or no knowledge of the problem domain are currently being investigated for use in solving manufacturing problems.

4.3.1 Dynamic Programming

Dynamic Programming (DP), formulated by Bellman (1957) is an optimisation method that has been used to solve a wide variety of problems. However, unlike linear programming, there is no standard formulation for dynamic programming that allows a single DP algorithm to be used for solving a wide variety of problems. The method is essentially composed of a series of stages with decisions concerning the final solution being made at each stage. A decision made at a particular stage of the DP process affects the state of the problem and the possible decisions that can be made at the next stage.

The method relies on the principle of optimality, which states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. Although the concept of dynamic programming is relatively simple, difficulties arise in formulating multi-stage decision problems such that the dynamic programming methods employed comply with the principle of optimality.

As the number of stages and state variables increase, the amount of calculation required to solve the problem increases. This is known as the "curse of dimensionality" which makes the use of dynamic programming computationally impractical when there are a wide variety of state variables involved. This problem has hindered the application of dynamic programming within manufacturing. For example, Lin and Chen (1995) and Vanhoesel et al (1994) applied the technique to scheduling and reported that the method performed well only on small problems. Vanhoesel et al (1994) also reported that constraints had to be placed on the objective function for the method to achieve acceptable performance. Chen et al (1994) and Odanaka (1994) used dynamic programming to determine lot sizes. They reported that the algorithm could find good solutions to the problem but as the problem size increased, the performance of the method decreased. Dynamic programming was applied to the machine layout problem by Karp and Held (1967) and Picard and Queyranne (1981) with both sets of researchers reporting computational time problems.

4.3.2 Search methods

Search methods are problem solving techniques that systematically explore a space of alternative solutions, until an optimum or suitable solution is found. The ideal search method would examine all potential solutions. However, for many problems this is unrealistic due to the large number of potential solutions that could exist. Existing search techniques, therefore, are designed such that only a small random portion of feasible solutions are considered, with the remaining solution space being accounted for implicitly. The basic types of search methods are random, enumerative and calculus based.

Calculus based methods are of two basic types, i.e. direct and indirect. Direct search methods have been developed primarily for single-variable functions and require the initial identification of a solution space that is known to include the optimum. This interval is then systematically reduced in a manner that guarantees that the optimum will be found. The use of both direct and in-direct search methods are limited in that the optimised function is assumed unimodal over the search interval. This results in only one local optimum being found. In addition, no finite intervals exist in which the slope of the function is zero, i.e. with this additional assumption, the optimised function may be referred to as strictly unimodal.

4.4 Simulated Annealing

An optimisation procedure of current interest to researchers is that of Simulated Annealing (Metropolis (1953)). The simulated annealing procedure involves individual changes being made from a current solution and for each change a value δ in the objective function is calculated. When $\delta < 0$, an improvement to the current solution is said to be possible and the change is automatically carried out. Otherwise the move is accepted with a probability of $P(\delta) = e^{-\delta/kT}$, where T is the current objective function value and k is a constant adapted to the application. The heuristic adaptation of k at different levels of T is referred to as "creating an annealing schedule". This provides the algorithm with the ability to move the solution out of a potentially local optimum and search for better solutions in other regions of the search space.

Simulated Annealing (SA) has many successful applications in manufacturing including the areas of scheduling and machine layout. Yamada et al (1994), for example, applied simulated annealing to a job shop scheduling problem and found that the method could find near optimal schedules. Pei-Chann and Ru-Ching (1994) found that the SA approach performs much better than traditional heuristic approaches and provides competitive solutions when compared to solutions generated using a Branch and Bound approach. Crabtree (1995) compared simulated annealing with constraint programming for resource scheduling. It was found that as the "number of constraints" increased the performance of the SA algorithm decreased and the performance of an alternative constraint programming method increased.

Heragu and Alfa (1992), Kouelis et al (1992) and Zegordi et al (1995) successfully applied simulated annealing to the machine and facilities layout problems. They identified that the method was parameter sensitive and required greater computational effort to produce high quality solutions when compared with traditional methods. In addition, they also identified that computation time was higher when the number of facilities was greater than 50.

4.5 Tabu Search

Tabu Search (TS) was originally developed by Glover (1986) and can be described as a metaheuristic technique since the approach undertakes to transcend local optimality by a strategy of forbidding moves that have recently occurred. Instead of terminating upon reaching a point of local optimality, Tabu Search structures the operation of its embedded heuristic in a manner that permits it to continue. This is accomplished by forbidding moves with certain attributes (i.e. making them tabu), and choosing moves from those remaining. In this respect, the method is a constrained search procedure, where each step consists of solving a secondary optimisation problem, admitting only those solutions, i.e. moves that are not excluded by the currently reigning tabu conditions.

Although the Tabu List holds a set of solutions which have recently been evaluated and cannot be evaluated again for a set period of time, the tabu status of a solution can be overridden by aspiration criteria. It is the tabu list and the aspiration criteria which are the basic mechanisms which prevent the search being trapped in local optimum.

Tabu Search has been successfully used to solve a number of manufacturing problems. For example, Kuik et al (1993) and Hindi (1995) applied Tabu Search to the lot-sizing problem and found that the technique was capable of reaching the optimal solution for a large number of lot sizing situations. Kuik et al (1993) found that individually Tabu Search and Simulated Annealing out performed Linear Programming, and that the performance of both Tabu Search and Simulated Annealing could be enhanced by combining elements of both methods. Laguna et al (1991), Laguna and Velarde (1991), Widmer (1991), Barnes and Chambers (1995), Hertz and Widmer (1996) and Reeves (1993) solved scheduling problems with Tabu Search and confirmed that Tabu Search is a more efficient search paradigm than Simulated Annealing.

Glover et al (1995) reported that although Tabu Search and Genetic Algorithms (Section 4.6) have significant differences, the independent success of Genetic Algorithms and Tabu Search in a variety of applications suggests that each has features that are valuable for solving complex problems.

4.6 Genetic Algorithms

Genetic Algorithms (GA's) are essentially search algorithms, the mathematical principles of which are based on the mechanics of survival of the fittest and natural selection in the biological world. Genetic Algorithms were first established by Holland (1975) at the University of Michigan. Since their initial development they have been used to solve problems involved in a variety of areas including control, financial services and VLSI design. Within manufacturing they have been successfully applied to problem areas, which include scheduling, line balancing and simulation.

Cleveland and Smith (1989) and Davis (1985) used genetic algorithms to schedule job shops. Davis (1985) limited the search space of the GA to legal schedules, whereas Cleveland and Smith demonstrated the use of a GA on a number of variations of the scheduling problem.

Minagawa and Kakazu (1992) have successfully applied the genetic algorithm to the line balancing problem, in which the objective was to minimise the cycle time. The advantage of using Genetic Algorithms to solve these types of problems arises when the problem size becomes complex in terms of the number and variety of constraints involved. These complexities reduce the effectiveness of other alternative methods such as linear programming and dynamic programming.

Tenga et al (1988) used genetic algorithms to optimise the design of manufacturing systems, where the parameters examined included length of conveyor, the work rate of robots, the number of pallets and the size of buffer stocks.

4.6.1 Genetic Algorithm Structure

Figure 4.1 is a diagrammatical representation of the structure of a Genetic Algorithm. The GA process requires the initial creation of a group (termed "population") of alternative solutions (termed "strings") to the problem being examined. Next an appropriate objective function is used to determine for each solution a quantitative value that can be used to compare alternative solutions in terms of their relative optimality. The GA process then repeats a sequence of processes in which solutions are modified using operators such as crossover and mutation. Solutions are re-evaluated using the objective function in order that a new population of solutions can be created from the preceding one. This sequence of operations continues until pre-defined criteria are achieved. In order to use GA's, therefore the solution to a specific problem must be reformatted into a genetic code, and a suitable objective function developed that is capable of comparing alternative solutions. The GA process then uses the basic operators of selection, crossover and mutation to create improved individual solutions.

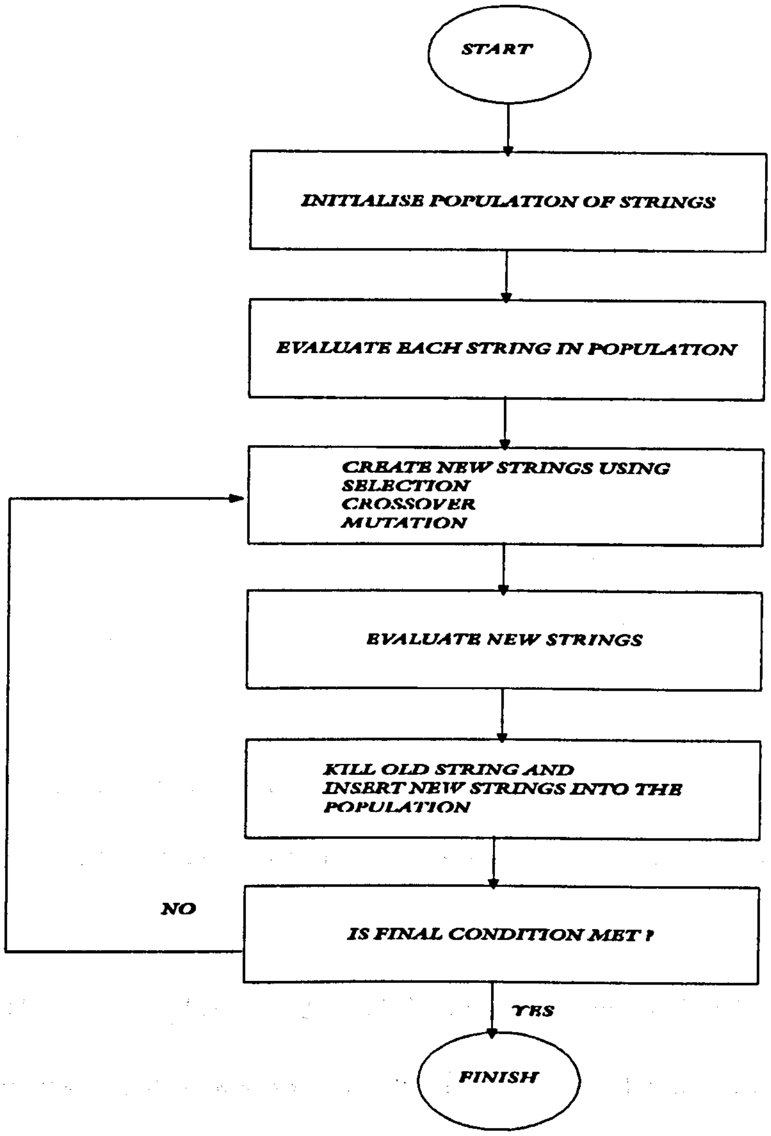


Figure 4.1 Flow Diagram of a Genetic Algorithm

4.6.1.1 GA Coding

Generating a genetic code that represents the problem being examined is normally achieved by representing the variables of the problem as a sequence of digits.

For example, consider a problem which has three variables; x1, x2, x3. where:

$$x1 = 5$$

$$x2 = 1$$

$$x3 = 2$$

A typical solution would be to code the variables using binary numbers as follows:

Variable	x1	x2	x3
Binary	101	001	010

Here the first 3 digits represent the value of variable x1, the next 3 digits represent the value of variable x2 and the last 3 digits represent the value of the variable x3.

The binary alphabet illustrated above is the simplest alphabet that can be used. Although this type of alphabet allows a variety of problems to be coded it also limits the options open to the user. In order to overcome this, a variety of coding types have been developed to aid the identification of suitable codings. Goldberg (1989) has provided the following rules to be used when selecting coding types, i.e.

1. The user should select a coding so that short, low-order schemata are relevant to the underlying problem and relatively unrelated to schemata over fixed positions.

2. The user should select the smallest alphabet that permits a natural expression of the problem.

The initial population is usually chosen at random or alternatively can contain heuristically chosen initial points. In either case the population should contain a wide variety of possible solutions, in order to provide a suitable representation of the entire solution space.

4.6.1.2 Selection Operators

The solution strings for inclusion within a succeeding population are chosen from the previous population by a randomised selection procedure that ensures that the expected number of times a string is selected is proportional to the relative values of each solution's objective function value relative to the rest of the population.

A number of methods are available for selecting strings for crossover, the simplest being the roulette wheel technique, (Goldberg (1989)), i.e., each string is allotted according to the value of its objective function a proportion of a roulette wheel. Brindle (1981) examined further schemes for the selection of strings, i.e.

1. deterministic sampling,
2. remainder stochastic sampling without replacement,
3. stochastic sampling without replacement,
4. remainder stochastic sampling with replacement,
5. stochastic sampling with replacement, and
6. stochastic tournament.

Booker (1982) demonstrated the superiority of "stochastic remainder selection without replacement" over "stochastic sampling without replacement".

4.6.1.3 Crossover and Mutation

To allow other points in the solution space to be searched, variation is introduced into the new population by means of genetic operators, the principle ones being crossover and mutation.

Crossover is carried out by initially selecting two solutions and then exchanging elements of one solution with those of the other selected solution. For example, assume that the two solutions selected are x1 and x2 and their genetic codes are as follows:

x1 = 101:0001 x2= 010:1100

A crossover point initially needs to be chosen, i.e. this is indicated by the position of the colon. The last four digits of x1 are then exchanged for those of x2, with x2 receiving the last four digits of x1. The result is the formation of two new solutions y1 and y2.

y1 = 101:1100 y2 = 010:0001

The principle underlying "crossover" is that if two good solutions undergo crossover, then a better solution may be created. If a poorer solution is generated through crossover then this will have less change of being represented in the next generation. A number of forms of crossover exist, including single point crossover, two point and uniform.

The mutation operator creates a new string by altering one or more bits in a string. If only the crossover operator is used in a genetic algorithm, the algorithm will tend to result in the majority of solutions within a population possessing the same structure. When this occurs, information may be lost that could later prove useful. The mutation operator is, therefore, used to help prevent this situation arising, i.e. mutation guards against such irrecoverable loss. Hence mutation, if used infrequently can be regarded as an insurance policy against premature loss of important information.

A genetic algorithm that utilises only mutation would be equivalent to a random search process. A low probability of mutation would search slowly as it would often leave the string unchanged. It is, therefore, applicable to a mutation probability that restores lost information. Hence the probability value of mutation affects the operation of the algorithm. A number of methods have been proposed for the selection of the mutation probability value. It is normal for the value to be constant for the entire search but recently it has been identified that it is beneficial to vary the probability of the mutation during the search process, Fogarty (1989).

4.6.1.4 Objective Function

The objective function is used to compare alternative solutions within a population. The objective function is the link between the GA and the problem to be solved. It takes a string from the GA and returns a quantitative value which is a measure of that string's performance in resolving the problem being examined. It is, therefore, important to avoid the use of a poorly designed objective function since these may fail to identify good solutions. Objective functions may be designed to include penalty values which can be used to prevent the development of non-feasible solution strings.

4.7 Genetic Algorithm Software used in the Research

A number of Genetic Algorithm software packages have been used during the course of the research; GENESIS, GA workbench, GAME, and XpertRule.

The GA Workbench, developed by Hughes (1990), is an interactive programme created to allow trials to be carried out to test the efficiency of the GA procedure in identifying the optimum point on target functions.

Genesis (Grefenstette (1984)) is a software package containing functions for the manipulation of Genetic Algorithms. The package contains the fundamental procedures for genetic selection, crossover and mutation with the user providing a suitable objective function.

XpertRule (Attar Software(1994))is a software tool for the graphical development and maintenance of knowledge based systems. An XpertRule application is constructed graphically as a hierarchy of chained tasks. A task can consist of a decision tree representing a flow chart controlling procedures or alternatively a set of pattern rules representing knowledge.

The applications built can be run within XpertRule under windows or alternatively the rules, logic and procedures can be generated as source code files in high level languages, for example, C. Xpertrule can also use information in other packages by using Dynamic Data Exchange (DDE).

XpertRule allows parameters to be optimised using genetic algorithms. Solution strings are created graphically by specifying the type of parameters and the range of acceptable values that can be used. The objective function can be generated using either XpertRule's own language, C code or data linking facilities to packages such as spreadsheets.

GAME (Kingdon and Dekker (1994)), is a Genetic Algorithm package that runs under the Windows operating system. The objective functions can be coded in either a spreadsheet or by using a DDL. GAME is also available in a parallel processing version.

5. Application of Genetic Algorithms

This chapter investigates the use of the Genetic Algorithm in solving the manufacturing system design problems identified in Chapters 2 and 3. The chapter initially identifies how each problem type may be represented as a GA code then examines the ability of the GA in solving each problem type. The results obtained using GA procedures are compared with those obtained when using other existing solution techniques.

5.1 Assortment Problem

In order to examine the ability of genetic algorithms in providing acceptable solutions to the assortment problem, case study materials collected, by Stockton (1983), during the development of a new range of electric wire rope hoists was used. Here it was necessary to identify the number of basic frame capacities to develop as standards and the safe working load (SWL) capacities of each standard size. Initial data used, i.e. Table 5.1, included the range of frame sizes that were being considered, forecast sales quantities for each frame size and the prime costs of each standard size. The maximum and minimum frame sizes considered in Table 5.1 were determined from a knowledge of the size range used within the total market.

Using the Minaddition technique, (Section 2.2.1.2), the minimum prime costs had been determined, by Stockton (1983), for successive numbers of standard sizes. These costs are listed, with the corresponding optimum standard frame sizes to manufacture, in Table 5.2. This table also lists the cost savings incurred in increasing the number of standard sizes, i.e. Csv.

Hoist Unit Frame Size (Tonnes)	Sales Forecast (Units per Year)	Prime Costs/Unit (£'s)
0.16	21	241
0.20	10	252
0.25	89	266
0.32	10	284
0.40	24	306
0.50	383	333
0.63	74	367
0.80	170	449
1.00	184	466
1.25	142	533
1.60	192	627
2.00	94	734
2.50	289	868
3.20	79	1016
4.00	131	1270
5.00	74	1515
6.30	24	1886
8.00	8	2368

Table 5.1 Hoist Block Sales and Cost Data

Number of Standard □Frame Sizes	Optimum Frame Sizes (Tonnes)	Prime Costs (Ce)(£'000s)	Cost Savings (Csv)(£'000s)
1	8.0	3434	
2	1.25 8.0	1010	2424
3	1.25 3.2 8.0	519	491
4	1.0 2.5 5.0 8.0	281	238
5	0.5 1.25 2.5 5.0 8.0	191	90
6	0.5 1.0 1.6 2.5 5.0 8.0	129	62
7	0.5 1.0 1.6 2.5 4.0 5.0 8.0	87	42

Table 5.2 Optimum Product Data Range

Using the information listed in Table 5.2, the curve shown in Figure 5.1 was constructed which illustrates the rapid rate at which cost savings are incurred when adding extra standard sizes and indicates, therefore, that only rough estimates of the additional costs, (Ca), incurred are needed to determine the optimum number of standard sizes. It was estimated during the development of the new electric hoist range that the extra fixed costs incurred in adding an additional standard lay between £50,000 and £100,000.

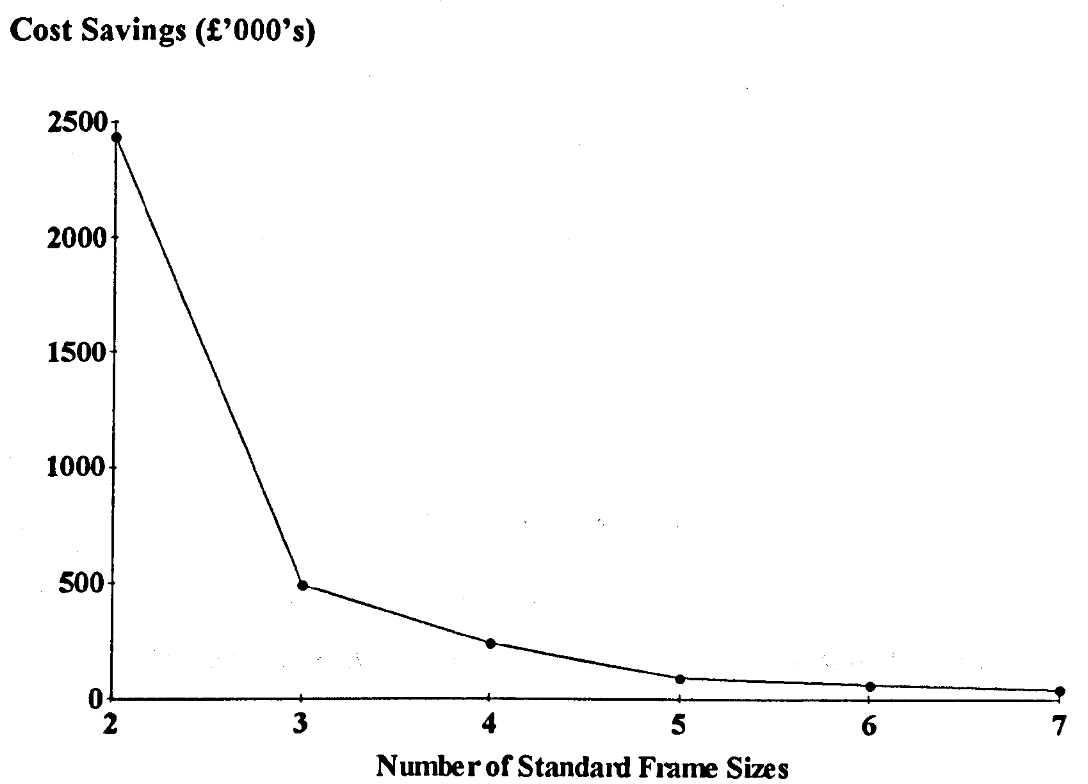


Figure 5.1 Cost Savings Incurred From Increasing Number of Standard Sizes

5.1.1 Coding

When coding the problem into a format compatible for a GA, two principles were taken into consideration i.e., the principle of meaningful building blocks and the principle of minimal alphabets.

The assortment problem is concerned with identifying whether a standard should be produced or not, i.e. either Yes (produce) or No (not produce). It is, therefore, possible to represent solutions to this problem type using a binary alphabet, 1 and 0 to represent Yes and No respectively. This alphabet would agree with the two principles of coding, i.e. meaningful building blocks and minimal alphabets.

The problem was, therefore, formulated using a binary alphabet with each digit representing a standard that is being considered. This produces a string length of 18 digits, i.e. each digit represents a potential standard. For example, Table 5.3 shows that standards 0.16, 0.25, 0.32 would be selected amongst others since their corresponding element in the GA string had a value of 1.

GA String	1	0	1	1	1	0	1	. . .	0	0	1	1	0	1
Frame Capacity (Tonnes)	0.16	0.2	0.25	0.32	0.4	0.5	0.63	. . .	2.5	3.2	4	5	6.3	8

Table 5.3 Genetic Algorithm Representation for the Assortment Problem

5.1.2 Objective Function

The fitness of each solution was calculated using the formula shown in Table 5.4 and was developed as a Microsoft Excel spreadsheet application with the GAME software communicating with the spreadsheet to calculate the fitness of solutions, i.e. GAME passes the current string to the spreadsheet and the spreadsheet firstly decodes the string then calculates costs for the string. This procedure was repeated for each solution string within a population.

GA Solution Code	Frame Sizes in Solution (Tonnes)	Prime Cost (£'s) (A)	Expected Production Volume for Frame Size (B)	Total Cost (A * B)
1	8.00	2368	8 + 24 = 32	75776
0				
1	5.00	1515	74 + 131 +79 = 284	430260
0				
0				
1	2.50	868	289 + 94 = 383	332444
0				
1	1.60	627	192+ 142 = 334	209418
0				
1	1.00	466	184 + 170 + 74 = 428	199448
0				
0				
1	0.50	333	383 + 24 + 10 +89 + 10 + 21 = 537	178821
0				
0				
0				
0				
0				
			Total Prime Costs	1426167
			Fixed Costs @ £100,000 per Standard	600000
			Total Costs	2026167

Table 5.4 Total Cost Model for GA Method

To determine the number of generations that the algorithm would be required to be run for to identify good solutions, a number of experiments were carried out to determine a rough cut value. A GA with roulette wheel selection, single point crossover with an arbitrary probability value of 0.6 and basic mutation with an arbitrary mutation rate of 0.0001 was run for 100 generations. The results are shown in Figure 5.2 where it can be seen that for this problem no significant improvement in the solution occurs after 50 generations. Hence, the maximum number of generations that the algorithm should run for in each replication was set at 50 generations.

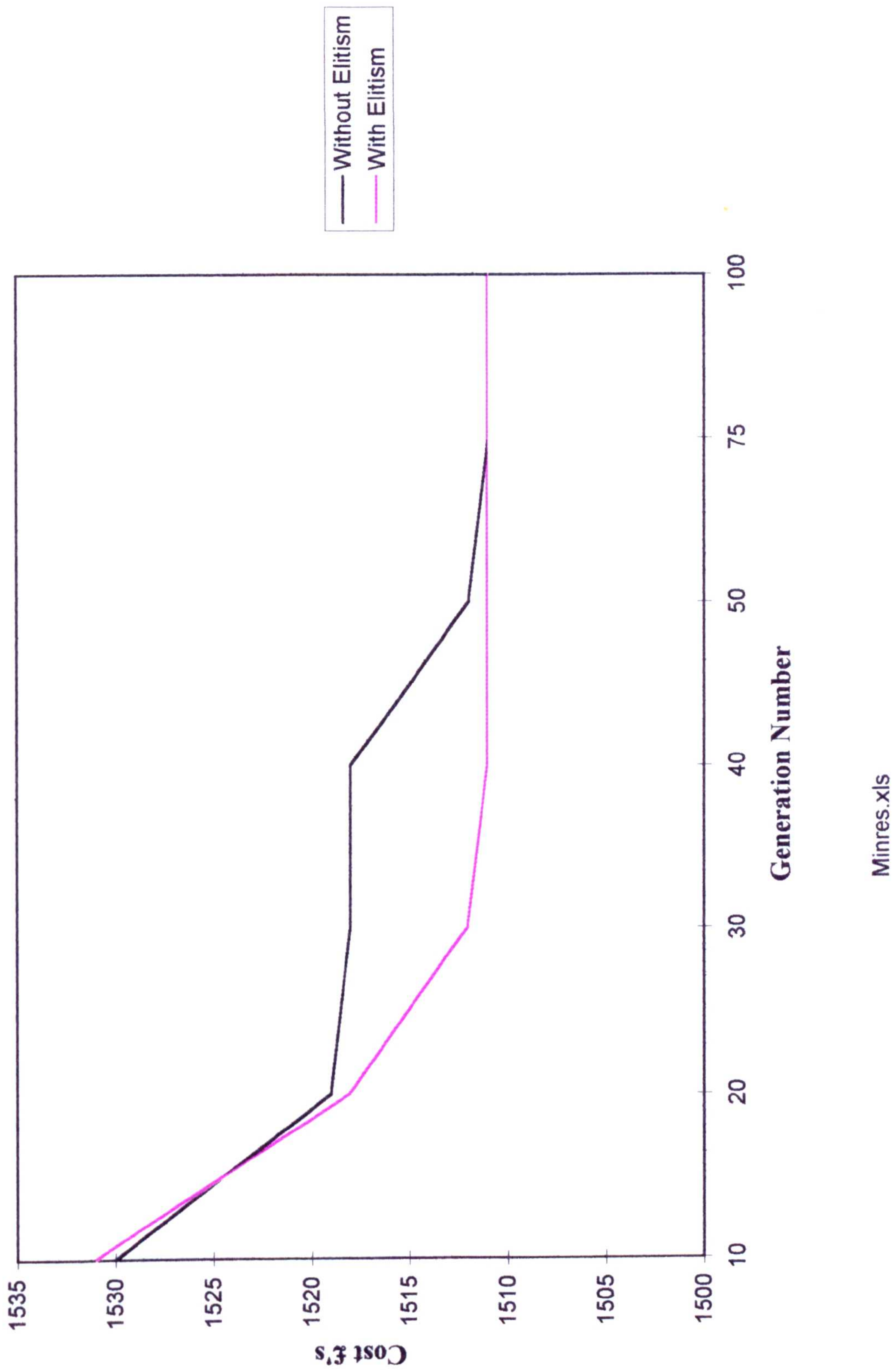


Figure 5.2 Effect of Number of Generations on Best Solutions

5.1.3 Selection

The alternative selection operators investigated were roulette wheel, roulette wheel with elitism, tournament and truncated. The efficiency of each selection operator was identified using experiments in which other operator types and their values remained constant, i.e.:

a. number of replications set at 10,

b. single point crossover used with a probability value set at 0.6,

c. mutation probability rate set at 0.0001, and

d. population size set at 100.

Figure 5.3 shows how well each method performs with respect to each other.

5.1.4 Crossover

The crossover operators investigated were single point crossover, two point crossover and uniform crossover. The efficiency of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator employed,

c. mutation probability rate set at 0.0001, and

d. population size set at 100.

The results are shown in Figure 5.4.

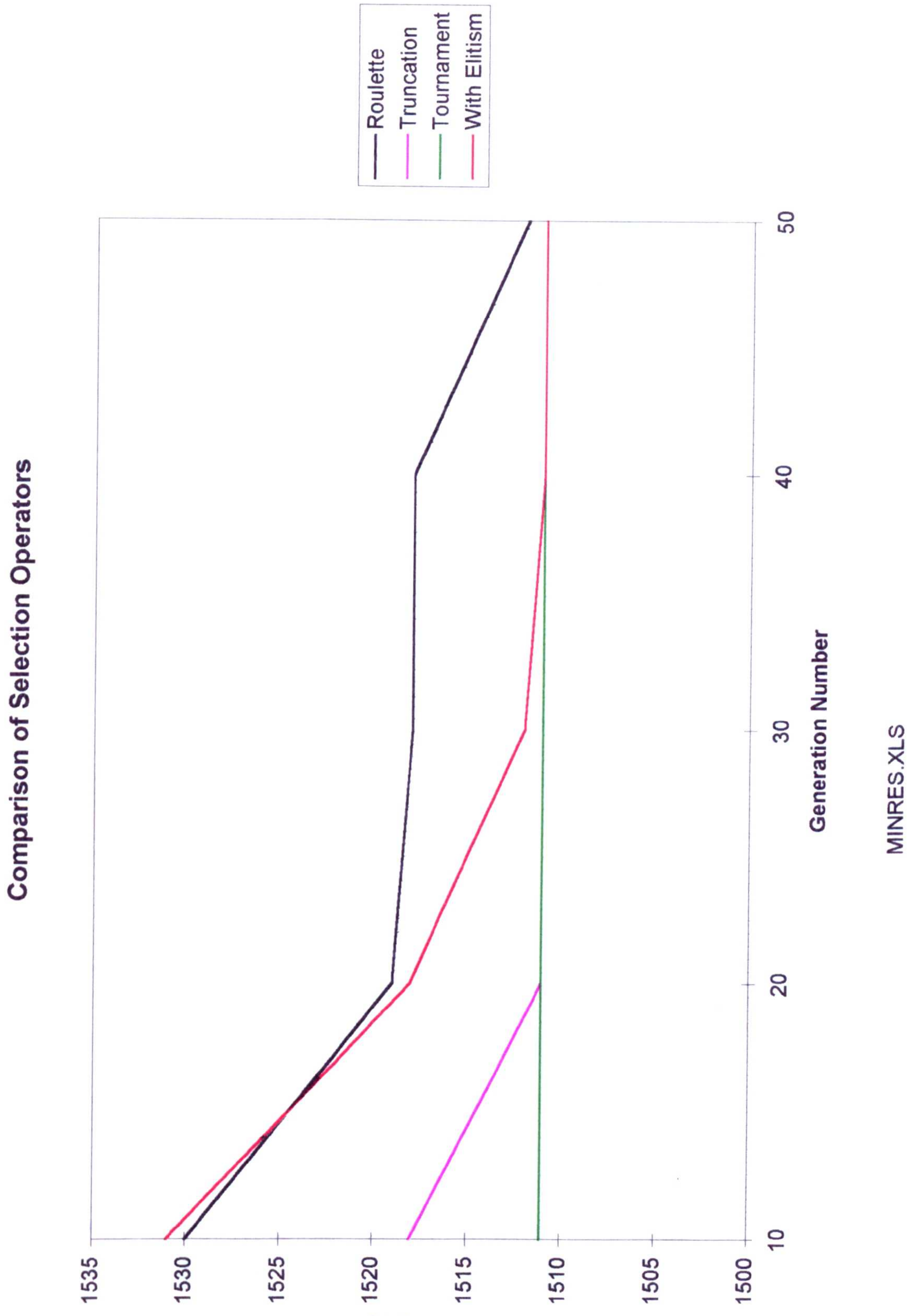


Figure 5.3 Comparison of Selection Operators for the Assortment Problem

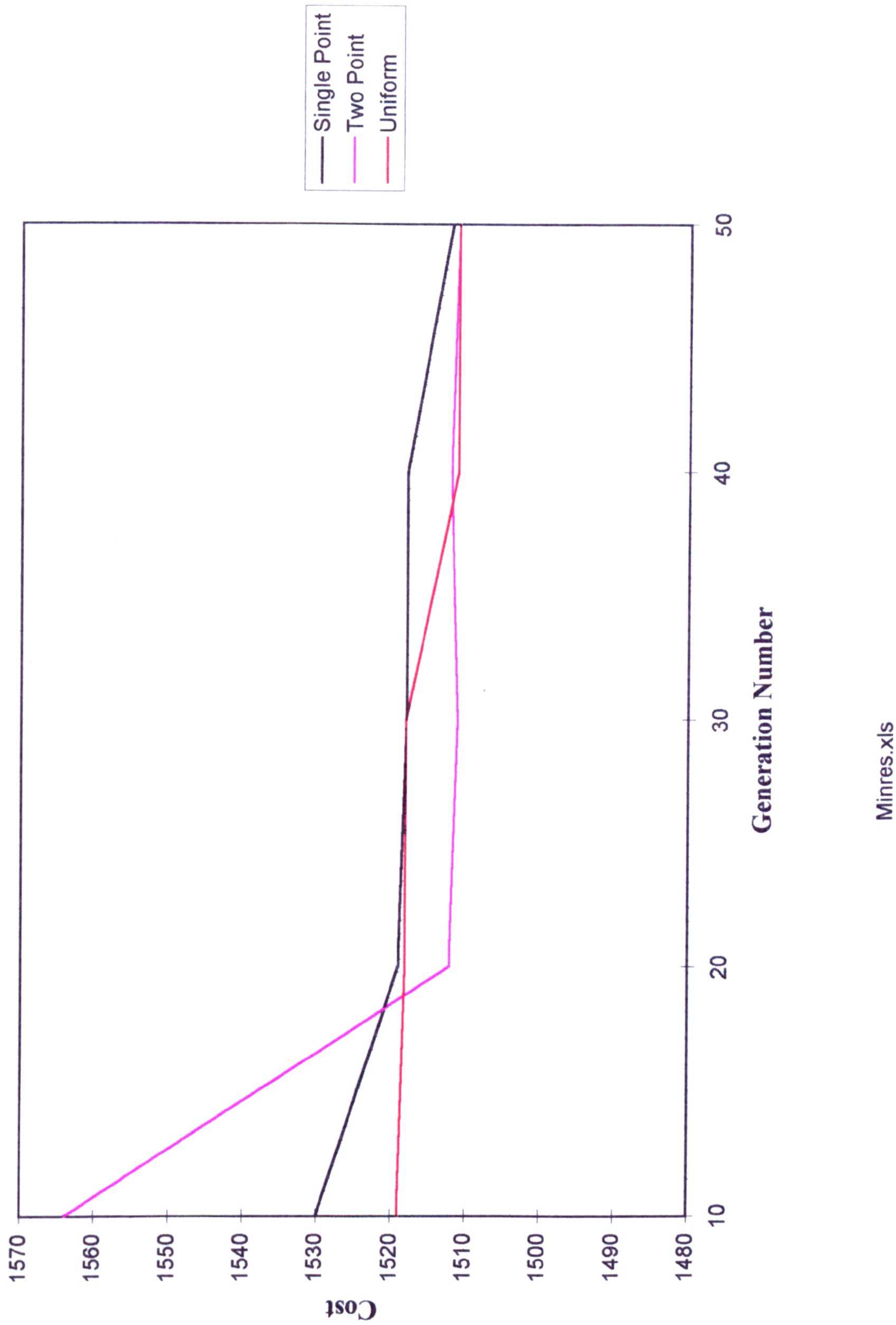


Figure 5.4 Comparison of Crossover Operators for the Assortment Problem

5.1.5 Comparison of GA and Minaddition Results

In terms of the assortment problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. truncation selection operator,

b. mutation probability rate of 0.0001,

c. population size of 100, and

d. two point crossover operator with a crossover probability of 0.6.

The results of the GA and Minaddition procedures, shown in Table 5.5, indicate that the GAME software can quickly identify the least cost solution to the hoist block assortment problem.

	Minaddition		Genetic Algorithm	
Fixed Costs Per Standard Frame Size (£'s)	**Frame Sizes**	**Total Cost (£'s)**	**Frame Sizes**	**Total Cost (£'s)**
50,000	8, 5, 2.5, 1.6, 1.0, 0.5	1, 726,167	8,5,2.5, 1.6, 1.0, 0.63	1,691,683
100,000	8,5,2.5,1.0	2,178,082	8,5,2.5, 1.25, 0.63	1,758,053

Table 5.5 Comparison of Minaddition and GA Results

5.2 Aggregate Planning

A Genetic Algorithm has been used to determine an aggregate plan of the type shown in Table 5.6. This type of plan indicates the additional units that must be achieved in order to meet forecast demand using overtime, sub-contracting, finished goods inventories and late deliveries. Each capacity management method has a specific cost attached to it, for example, overtime premiums are normally paid to employees hence units manufactured using overtime incur higher costs. The traditional objective is to ensure which ever combination of capacity management methods are used the overall costs involved in implementing the aggregate plan are minimised.

Period	Opening Stock	Forecast Demand	Normal Production	Overtime	Sub Contract	Late Deliveries	Finish Stock
January	0	1400	1000	90		310	-310
February	-310	900	1000	90	200		80
March	80	800	1000	90			370
April	370	1500	1000	90	200		160
May	160	1500	1000	90	200	50	-50
June	-50	800	800	90			40
July	40	500	800	90			430
August	430	1200	800	90	300		420
September	420	600	1000	90	300		1210
October	1210	2000	1000	90	300		600
November	600	500	1000	90	300		1490
December	1490	1500	900	90	300		1280

Table 5.6 Aggregate Plan

The plan shown in Table 5.6 is one of many alternatives that can be identified. It is necessary to represent this and other alternative plans in the form of a genetic code.

5.2.1 Coding

Coding the AP problem required determining the maximum number of units that could be produced using overtime and subcontractors and in addition the quantities in which overtime and subcontract units could be allocated. In order to achieve this each of the quantities listed in the plan have been converted from their current decimal base to their equivalent binary base, i.e. as illustrated in Table 5.7. The Genesis package codes the GA in binary alphabet which minimises its alphabet. To minimise the size of solution strings and search space, the number of batches were coded not the number of individual units to be produced. Hence, the overtime units were coded in the range 0 - 3 as a maximum of 90 units could be produced in batches of 30, (i.e. 3 x 30 = 90) and sub-contract units were coded in the range 0 - 3 as a maximum of 300 could be produced in batches of 100, (i.e. 3 x 100 = 300). The binary numbers listed in Table 5.7 have then been connected into a single string, as illustrated in Figure 5.5, to produce the necessary genetic code for an AP solution. In order to decode such a string it is split into two digit sections and each section converted back to its equivalent decimal number. Figure 5.5 indicates which area of the aggregate plan each section of the binary string applies to. It can be seen that each binary number within the string represents a specific aspect of the problem solution, i.e. a potential AP quantity.

Period	Overtime	Sub Contract
January	00	00
February	01	01
March	11	00
April	00	11
May	00	11
June	00	00
July	10	00
August	11	01
September	01	01
October	00	10
November	01	01
December	10	10

Table 5.7 Aggregate Plan Represented as Binary Numbers

00	01	....	01	10
Overtime	Overtime		Sub Contract	Sub Contract
January	February		November	December

Figure 5.5 Genetic Code for Aggregate Plan

The GENESIS software was used to randomly generate an initial population of solution strings. Subsequent generations of populations did not have random solutions introduced. However, non-feasible solutions were allowed to be reproduced into subsequent generations.

5.2.2 Objective Function

The overall costs associated with each string in the population were determined by decoding each string into its respective aggregate plan and using the cost model shown in Table 5.8. In this example, the value 0.0000031666 represents the "fitness" of this particular solution. Individual solutions within a population were then compared using their fitness values with those aggregate plans that resulted in high costs of implementation receiving correspondingly low fitness values.

Unit Costs	£10	£50	£200	£100	
Period	**Overtime**	**Sub Contract**	**Late Deliveries**	**Finishing Stock**	**Additional Costs (£)**
January	90		310		62900
February	90	200			10900
March	90				900
April	90	200			10900
May	90	200	50		20900
June	90				900
July	90				900
August	90	300			15900
September	90	300			15900
October	90	300			15900
November	90	300			15900
December	90	300		1280	143900
				Total Costs	315800
				Fitness value = 1/315800	0.0000031666

Table 5.8 Calculation of Fitness Value

The fitness value of an individual solution was then used to determine a solution's probability of surviving into the next generation as illustrated in Table 5.9.

Solution	Fitness Value (£'s)	% of Total
1	0.0018	52.49
2	0.00021	6.12
3	0.000011	0.32
4	0.000021	0.61
5	0.000014	0.41
6	0.0000178	0.52
7	0.000454	13.24
8	0.000011	0.32
9	0.00089	25.96
Total	0.0034289	100.00

Table 5.9 Comparison of Solutions within a Population

The "% of Total" is the probability that a solution will be represented in the next generation, e.g. solution 8 has approximately a 0.3% chance of being represented in this next population because it resulted in such large implementation costs when compared with other solutions within the population.

In order to test the ability of GA functions to generate aggregate plans, trials were carried out using the data shown in Table 5.10. This data set was selected with care to include the main problems that must be overcome by the aggregate plan, i.e. the data contains:

(a) a wide variety in sales demand between planning periods,

(b) sudden large changes in sales demand between adjacent planning periods,

(c) planning periods where demand is greater than normal production capacity,

(d) planning periods where normal production capacity is greater than demand,

(e) a period when demand is very much greater than that of normal production capacity, and

(f) a period when normal production capacity is very much greater than that of demand.

Aggregate Planning Period	Forecast Demand	Normal Production
1	1400	1000
2	900	1000
3	800	1000
4	1500	1000
5	1500	1000
6	800	800
7	500	800
8	1200	800
9	600	1000
10	2000	1000
11	500	1000
12	1500	900

Table 5.10 Aggregate Planning Data

An evaluation function was developed that calculated only the additional costs of adding extra units through the use of overtime and subcontracting. The following constraints were also included in this function, i.e.:

1. Units gained through the use of overtime incurred a £10 per unit additional charge.

2. The maximum units that can be gained through the use of overtime was set at 90.

3. In order to reduce the potential number of solutions, overtime units could only be added in units of 30.

4. Units gained through the use of subcontracting incurred a £50 per unit additional charge.

5. The maximum units that can be gained through the use of subcontracting was set at 300.

6. In order to reduce the potential number of solutions, subcontracted units could only be added in units of 100.

7. A penalty of £200 per unit was set for stock-outs in order to force the GA procedures to search for solutions that avoided such problems occurring. In this way organisations can identify aggregate plans that provide a high level of customer service.

8. Stock remaining at the end of the last planning period (that is December) incurred a cost of £100 per unit.

5.2.3 Selection

To compare the efficiency of the roulette wheel and the roulette wheel with elitism selection operators experiments were undertaken with the following parameters held constant, i.e.

a. number of replications set at 10,

b. two point crossover with a probability value set at 0.75,

c. mutation probability rate set at 0.001, and

d. population size set at 100.

The results are shown in Figure 5.6

5.2.4 Crossover

Using the two point crossover method the effect of crossover rate was examined using rates of 0.6, 0.65, 0.7 and 0.75. It is considered acceptable to test the effect of the crossover rate in increments of 0.05, i.e. due to the stochastic nature of the crossover operation smaller increments would have relatively insignificant effects. This interval of crossover rates has been widely reported to give the best results for the GA. The efficiency of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator with and without elitism,

c. mutation probability rate set at 0.001,

d. population size set at 100, and

e. sigma scaling set at a value of 2.

The results are provided in Figures 5.7 and 5.8.

5.2.5 Mutation

The effect of the mutation rates, 0.0001, 0.001 and 0.005 were investigated using the following parameter settings, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator with elitism,

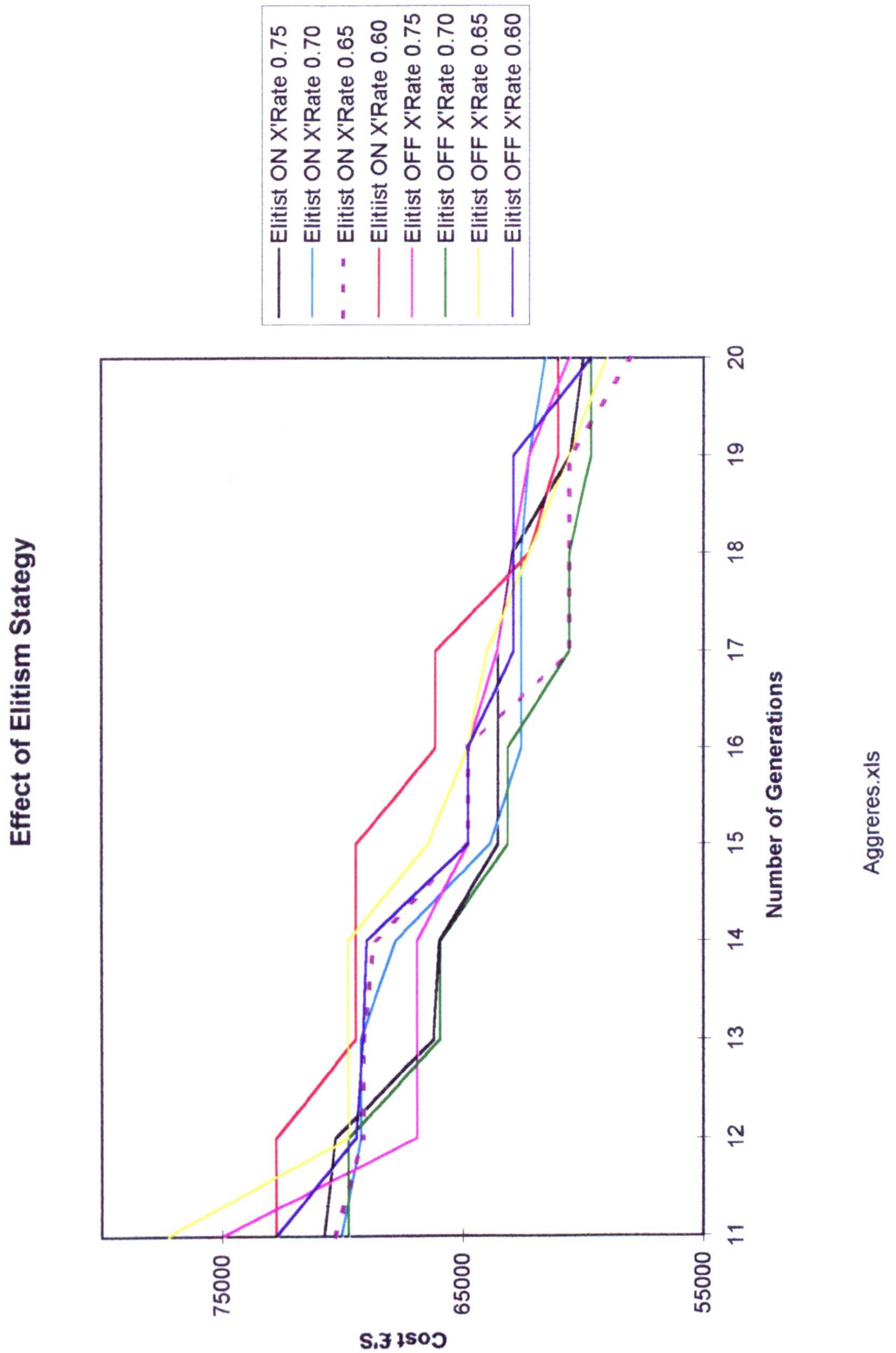


Figure 5.6 Effect of Elitism for the Aggregate Planning Problem

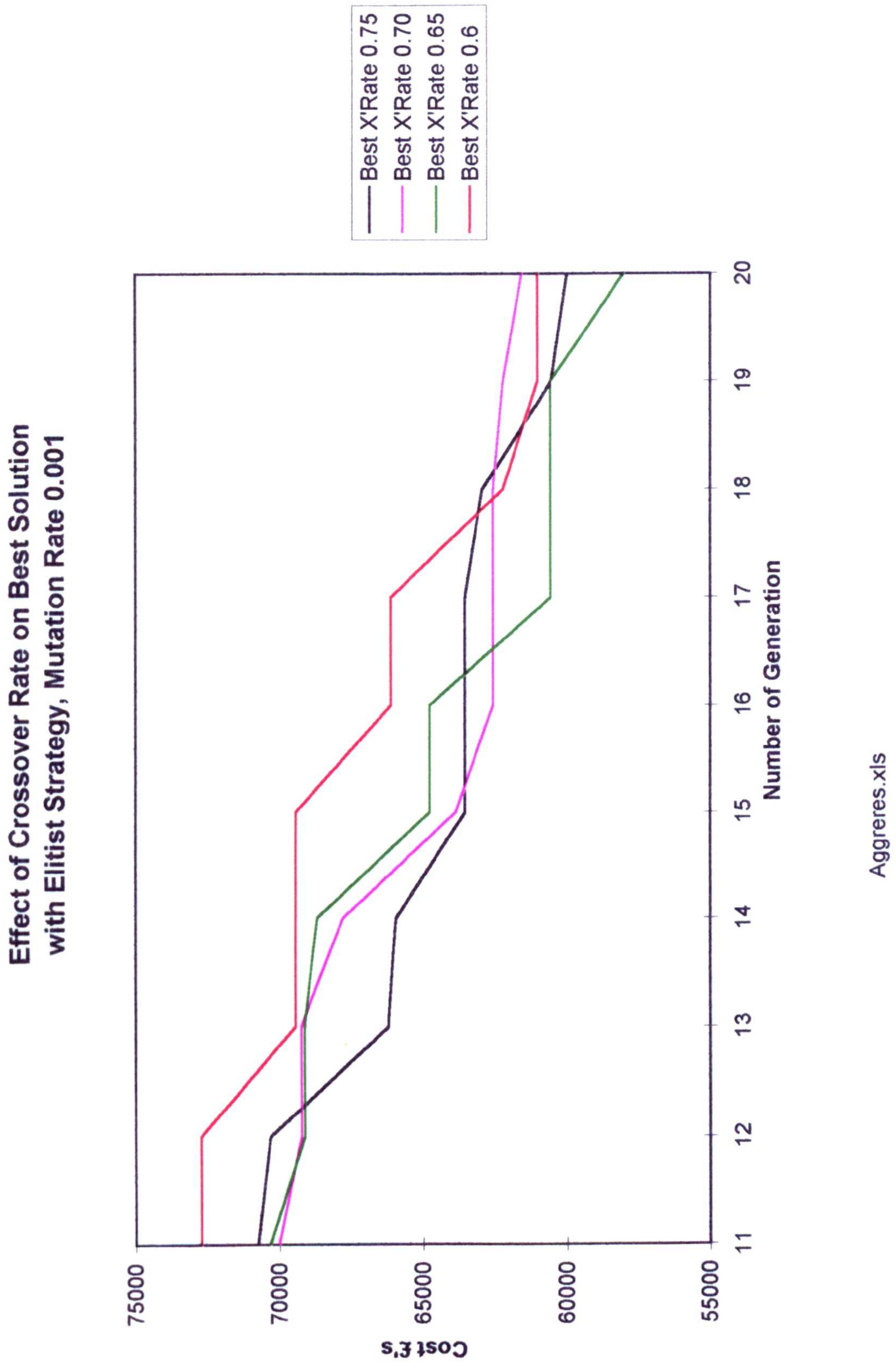


Figure 5.7 Effect of Crossover Rate with Elitism Strategy

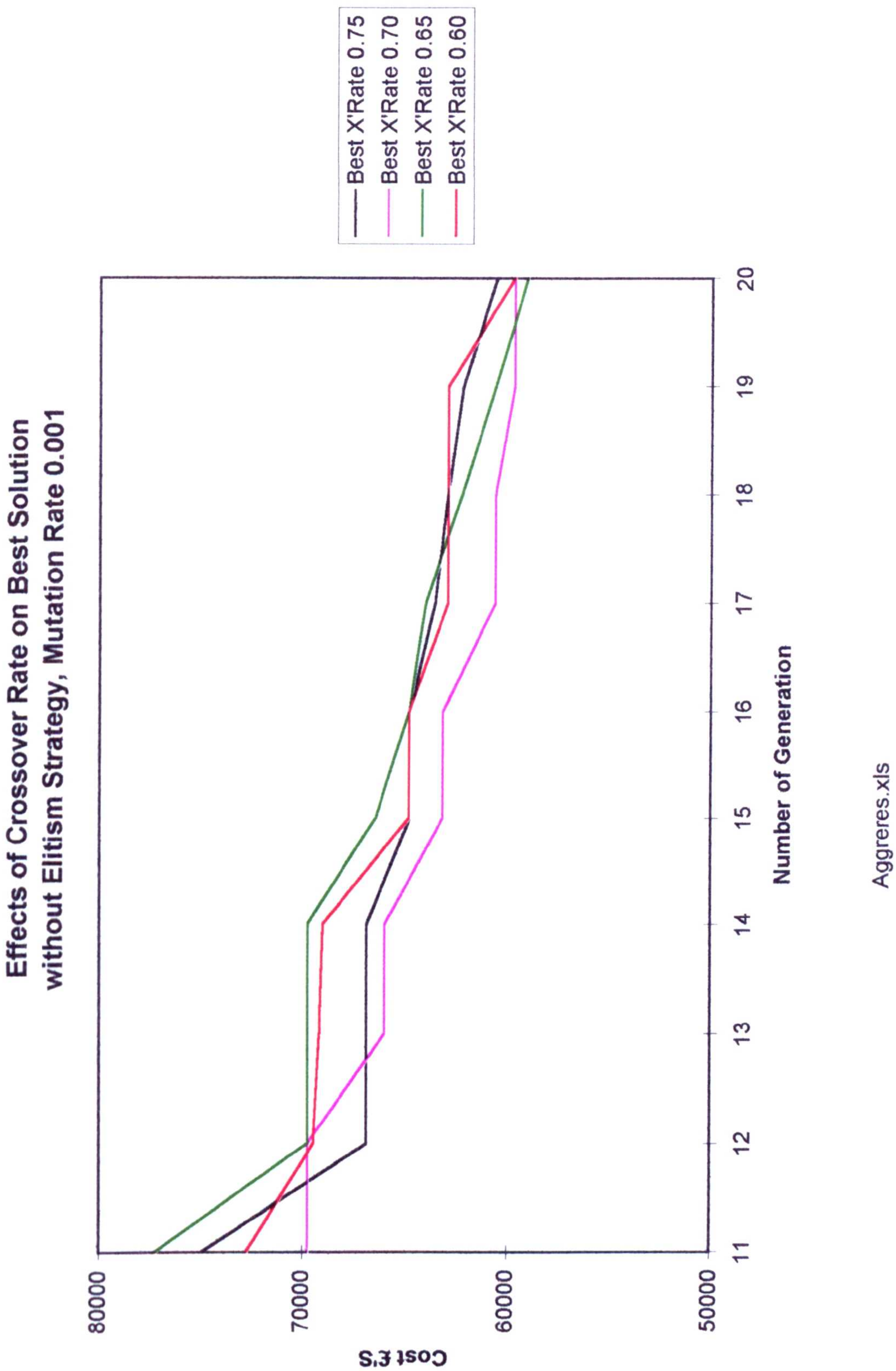


Figure 5.8 Effect of Crossover Rate without Elitist Strategy

c. crossover probability rate of 0.60,

d. population size set at 100, and

e. Sigma scaling set at a value of 2.

The results are shown in Figure 5.9.

5.2.6 Scaling

The Genesis software provides two types of scaling functions that can be used in conjunction with the GA, i.e. sigma scaling and window scaling.

Sigma scaling experiments were conducted using values of sigma equal to 1, 2, 3, 4 and 5. Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. In both sets of experiments, the following parameters were set as constants i.e.

a. roulette wheel selection with elitism,

b. two point crossover with a probability of 0.60.

c. mutation with a probability of 0.001, and

d. population size of 100.

The results are shown in Figures 5.10 and 5.11 respectively.

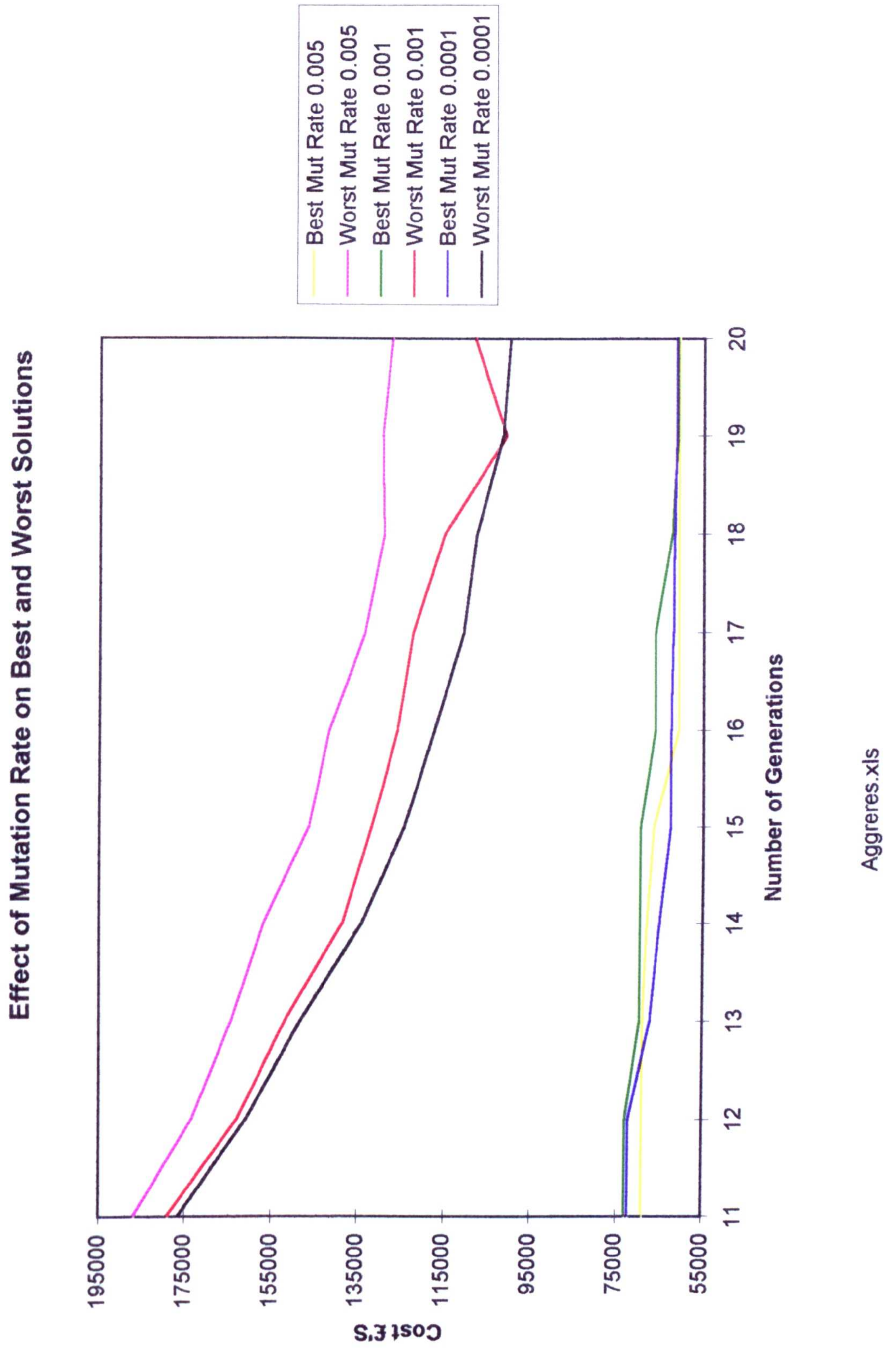


Figure 5.9 Effect of Mutation Rate for Aggregate Planning Problem

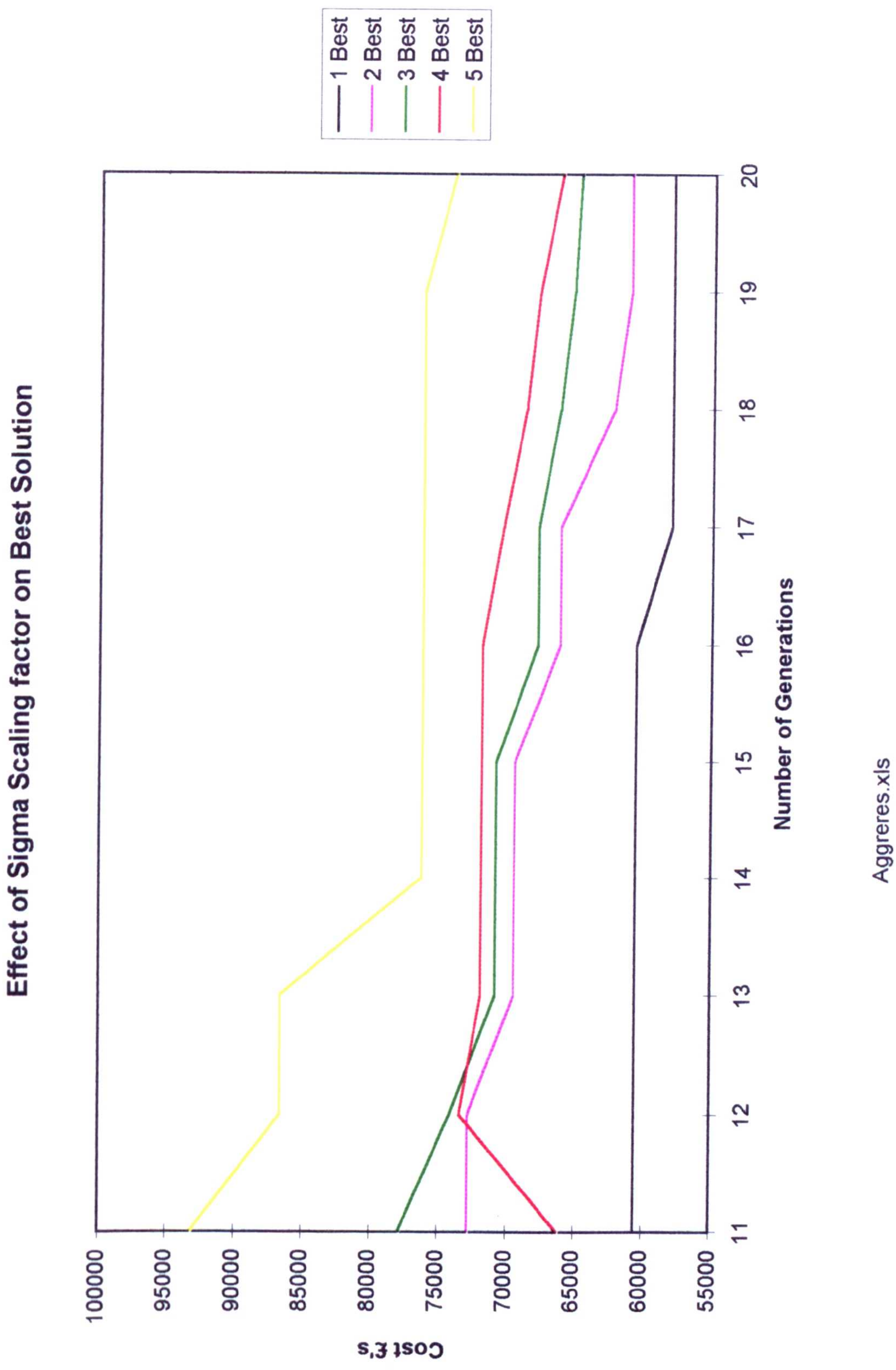


Figure 5.10 Effect of Sigma Scaling for the Aggregate Planning Problem

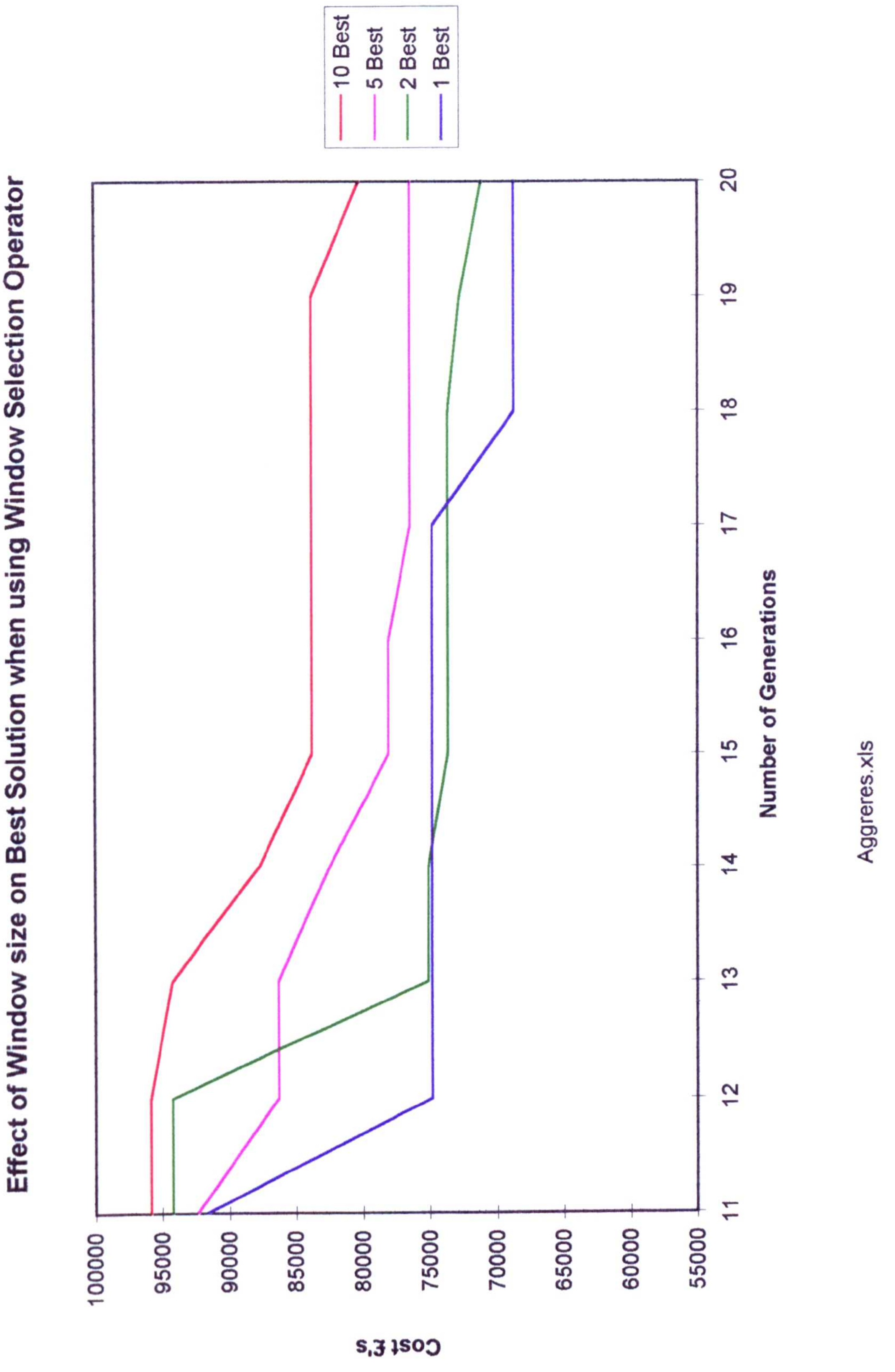


Figure 5.11 Effect of Window Size for the Aggregate Planning Problem

5.2.7 Comparison of GA and Manually Derived Solution Results

In terms of the aggregate planning problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. roulette wheel selection operator with elitism,

b. mutation rate of probability of 0.0005,

c. population size of 100,

d. two point crossover operator with a probability of 0.65, and

e. Sigma scaling with a value of 1.

Table 5.11 shows the least-cost aggregate plan generated manually and incurred total costs of £65,800. This plan is the least cost outcome of a subjective manual decision process, the objective of which was to continually seek to reduce the additional costs. Table 5.12 shows a lower cost solution identified by the GA. The experiment were carried out on a Sparc workstation and required a computation time of approximately 780 seconds to find the solution.

Aggregate Planning Period	Forecast Demand	Normal Production	Overtime	Subcontract	Stock Remaining
1	1400	1000	90	300	-10
2	900	1000	90	200	380
3	800	1000	90	200	870
4	1500	1000	90	0	460
5	1500	1000	90	200	250
6	800	800	90	0	340
7	500	800	90	0	730
8	1200	800	90	0	420
9	600	1000	90	0	910
10	2000	1000	90	0	0
11	500	1000	90	0	590
12	1500	900	90	0	80
		Additional Costs =	£65,800		

Table 5.11 Manually Derived Least Cost Aggregate Plan

Aggregate Planning Period	Forecast Demand	Normal Production	Overtime	Subcontract	Stock Remaining
1	1400	1000	90	300	-10
2	900	1000	90	0	180
3	800	1000	90	200	670
4	1500	1000	90	0	260
5	1500	1000	90	200	50
6	800	800	90	0	140
7	500	800	90	0	530
8	1200	800	90	100	320
9	600	1000	90	100	910
10	2000	1000	90	0	0
11	500	1000	0	0	500
12	1500	900	90	0	-10
		Additional Costs =	£58,900		

Table 5.12 GA Solution for the Aggregate Planning Problem

5.3 MRP Lot Sizing

In order to successfully use GA's for MRP lot sizing, they must be capable of providing a planned order release schedule for each item in a BOM structure, an example of which is shown in Figure 5.12. For example, if a three period planning horizon was assumed then the MRP process would be expected to provide a planned order release schedule of the type shown in Table 5.13, in which the values Q1, Q2, ... Q12 are the Planned Order Release, (POR), quantities of each item in specific MRP planning periods. The traditional objective is to ensure that these quantities minimise the overall costs involved in purchasing and holding stocks, hence a possible POR schedule would be as shown in Table 5.14.

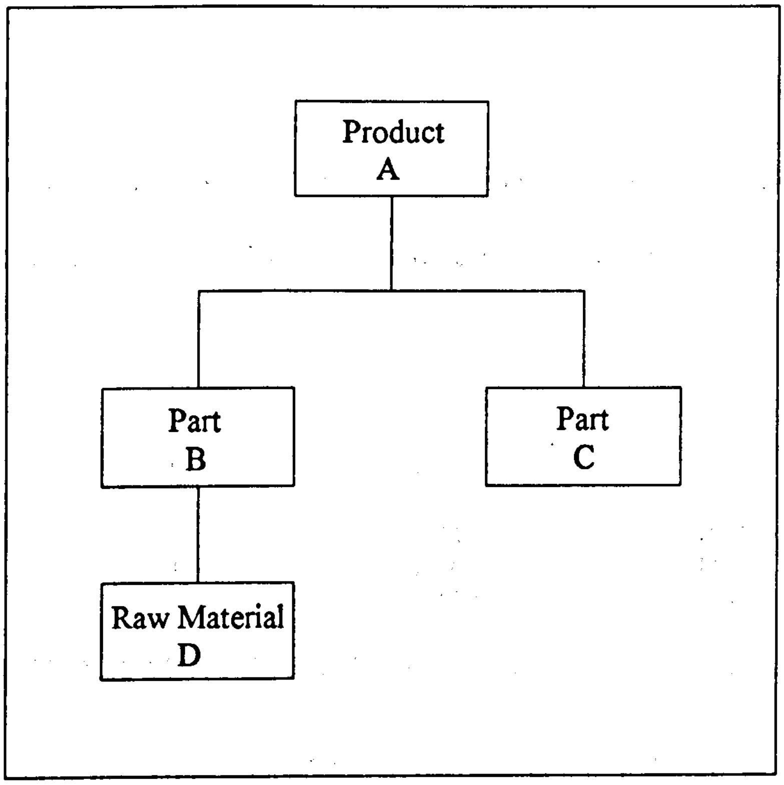


Figure 5.12 Bill of Material Structure

MRP Item	MRP Planning Period		
	Period 1	Period 2	Period 3
A Product	Q1	Q2	Q3
B Part	Q4	Q5	Q6
C Part	Q7	Q8	Q9
D Raw Material	Q10	Q11	Q12

Table 5.13 MRP Planned Order Release Schedule

MRP Item	MRP Planning Period		
	Period 1	Period 2	Period 3
A Product	22	10	21
B Part	16	30	36
C Part	15	25	51
D Raw Material	3	42	22

Table 5.14 Potential POR Schedule for Product A

5.3.1 Coding

In order to represent this POR schedule as a genetic code for use with the Genesis software, each of the batch quantities listed in the schedule was converted into its binary form as shown in Table 5.15.

MRP Item	MRP Planning Period		
	Period 1	Period 2	Period 3
A Product	010110	001010	010101
B Part	010000	011110	100100
C Part	001111	011001	110011
D Raw Material	000011	101010	010110

Table 5.15 POR Schedule Represented as Binary Numbers

The binary numbers listed in Table 5.15 were then connected into a single string as illustrated in Figure 5.13 in order to be compatible with the GA procedure. In order to decode such a string the reverse procedure had to be applied, i.e. split the string into six digit sections and convert each section to its equivalent decimal number. Figure 5.13 indicates which item/MRP period each section of the binary string applies to.

	010110	001010	010101	010000	....	010110
Item	A	A	A	B		D
Period	1	2	3	1		3

Figure 5.13 Genetic Code for POR Schedule

The Genetic Algorithm was applied to the problem shown in Tables 5.16 and 5.17. Table 5.16 shows the Bill of Material for a flashlight and Table 5.17 gives the gross requirements schedule for the flashlight.

The maximum number of items that could be produced was set at 300 units and could only be acquired in batches of 10 units. The requirements for each item in the BOM for each planning period were coded in the range 0-30 since the maximum number of units that could be acquired was 300 in batches of 10, (i.e. 30 x 10 = 300 units). The final coding for the problem is shown in Figure 5.14. This coding requires 5 bits for each parameter in the string. The number of parameters required for the problem is 216, since there are 18 items in the BOM over a planning horizon of 12 periods, (i.e. 18x12= 216). The total length of the string is, therefore, 216 parameters multiplied by 5 bits, i.e. 1080 bits.

Item Code	BOM Level	Description	Holding Cost per Unit (£'s)
1	0	Flashlight	32
2	1	Headlight	10
5	2	Plastic Head	2
16	3	Plastic - Raw Material	1
6	2	Lens	1
7	2	Bulb Sub-Assembly	3
11	3	Bulb	1
12	3	Bulb Holder	1
8	2	Reflector	1
4	1	Body Assembly	15
9	2	Shell Assembly	11
13	3	On/Off Switch	4
17	4	Knob	1
18	4	Metal Slides	1
14	3	Connector Bars	1
15	3	Plastic Shell	4
16	4	Plastic - Raw Material	1
10	2	Spring	1
3	1	Batteries	3

Table 5.16 Bill of Materials for Flashlight

MRP Planning Period	1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirements	10	10	15	20	70	180	250	270	230	40	0	10

Table 5.17 Gross Requirements Schedule for Flashlight

Item	Flashlight	......	Flashlight	Headlight	...	Headlight	...	Batteries	...	Batteries
Period	1	.....	12	1	....	12	.....	1	...	12
Value	01100		10001	10110		00010		11000		00110

Figure 5.14 GA Coding for MRP Problem

5.3.2 Objective Function

Fitness values for each solution within a population were then calculated using the objective function shown in Table 5.18. In the case study data used, the fitness of each solution has been determined on the basis of how well the solution minimises holding and procurement costs. The holding and procurement costs associated with each string in the population was, therefore, determined by decoding each string into its respective POR schedule and using an appropriate model to calculate the costs that would arise from adopting each schedule. Table 5.18 illustrates this process for a typical POR schedule. The value 0.00134 now represents the "fitness" of this particular solution, i.e. solutions with higher total costs would have correspondingly lower fitness values. In practice, multi-objective functions may be used that could contain both quantitative and qualitative fitness criteria. Cost penalties were also added to the fitness function of any solution in which stock-outs occurred, i.e. when lot sizes for lower level BOM items do not meet the requirements of higher level items.

MRP Item	MRP Period	Binary Number	Decimal Number (Q)	Holding Costs (£) (Q/2)*Ch	Purchase Costs (£s) Cp
A	1	010100	20	10.0	50
A	2	001010	10	5.0	50
A	3	010101	21	10.5	50
B	1	010000	16	8.0	50
B	2	011110	30	15.0	50
B	3	100100	36	18.0	50
C	1	001111	15	7.5	50
C	2	011001	25	12.5	50
C	3	110110	54	27	50
D	1	000011	3	1.5	50
D	2	101010	42	21.0	50
D	3	010110	22	11.0	50
				147	600
				Total Costs =	£747
			Fitness Value =	1/747 =	0.001338

Table 5.18 Calculation of the Fitness Value

Using the BOM structure shown in Table 5.16 and the gross requirements schedule illustrated in Table 5.17, the planned order release quantities were determined using genetic algorithm procedures and for comparison purposes McLaren's Order Moment, (MOM),(McLaren (1977)).

McLaren's Order Moment initially generates planned order release schedules for integral numbers of future MRP periods, (e.g. period 1, periods 1 and 2, periods 1, 2, and 3). In order to identify the most suitable schedule from amongst these alternatives the MOM procedure uses the part period accumulation principle, i.e. a part period is equivalent to one unit of stock carried for one MRP period. The selected schedule's accumulated part periods must match the number of part periods that would be incurred if an EOQ batch size had been calculated under conditions of constant demand. Comparison studies, Wemmerlov and Whybark (1984) have found that the MOM procedure produces lower cost order schedules when compared with the main methods in common usage, i.e. Part Period Balancing, Periodic Order Quantity and Economic Order Quantity.

In order to determine the applicability of GA operators for identifying planned order release schedules a series of experiments were performed as follows:

5.3.3 Selection

To identify the use of the roulette wheel selection and the roulette wheel selection with elitism, experiments were undertaken with the following parameters held constant, i.e.:

a. number of replications set at 10,

b. two point crossover with a probability value of 0.75,

c. mutation probability rate of 0.00001, and

d. population size of 1000.

The results shown in Figure 5.15.

5.3.4 Crossover

As with the AP problem (Section 6.2), the crossover operator rate was investigated using two point crossover and crossover rates of 0.6, 0.65, 0.7 and 0.75. The efficiency of crossover rate was identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. mutation probability rate set at 0.00001, and

d. population size set at 1000.

The results are shown in Figure 5.16.

5.3.5 Mutation

The effect of the mutation rates, 0.000001, 0.000005 and 0.00001 were investigated using the following settings, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. two point crossover with a probability rate of 0.75, and

d. population size of 1000.

The results are shown in Figure 5.17.

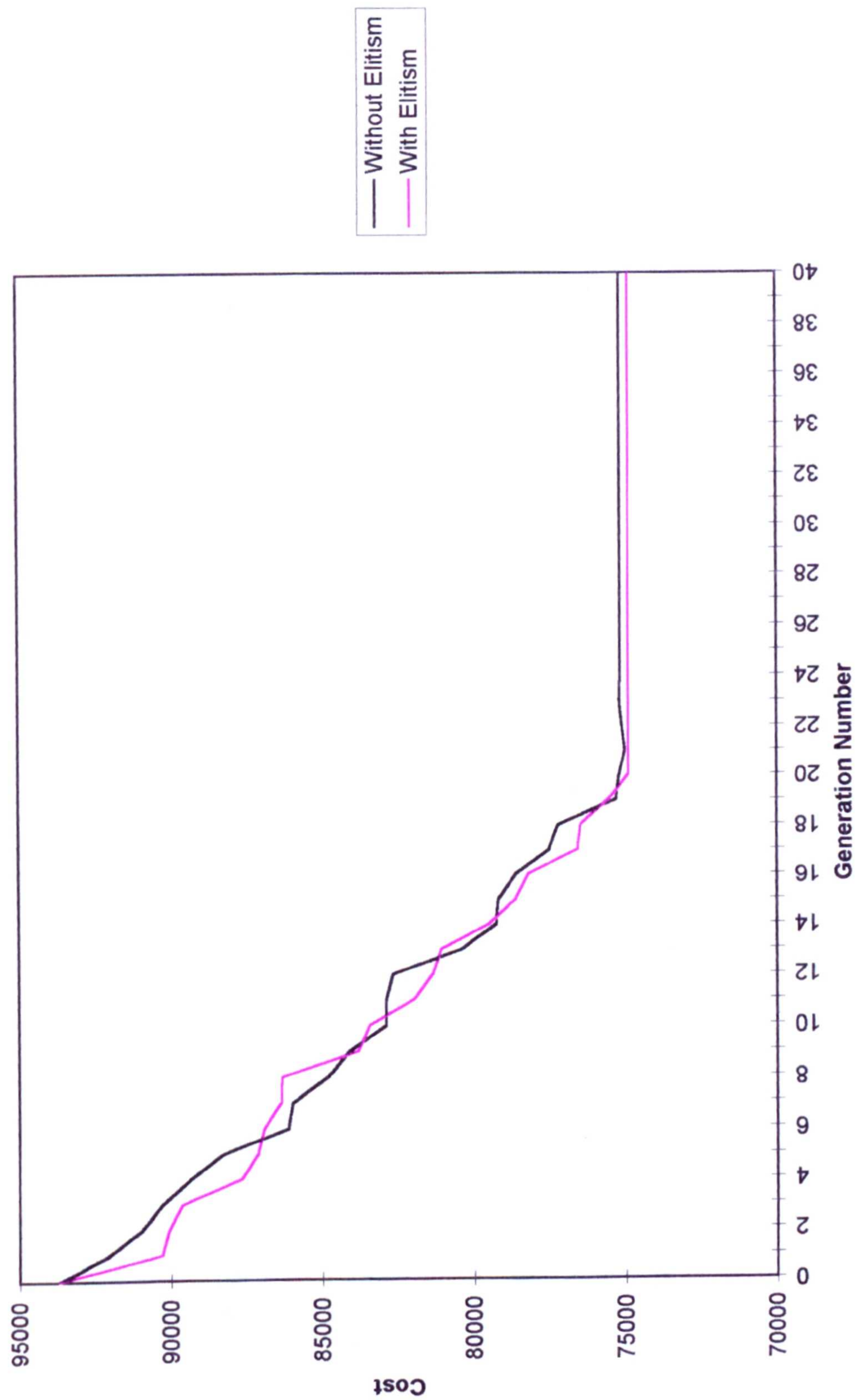


Figure 5.15 Effect of Elitism on MRP Lot Sizing Problem

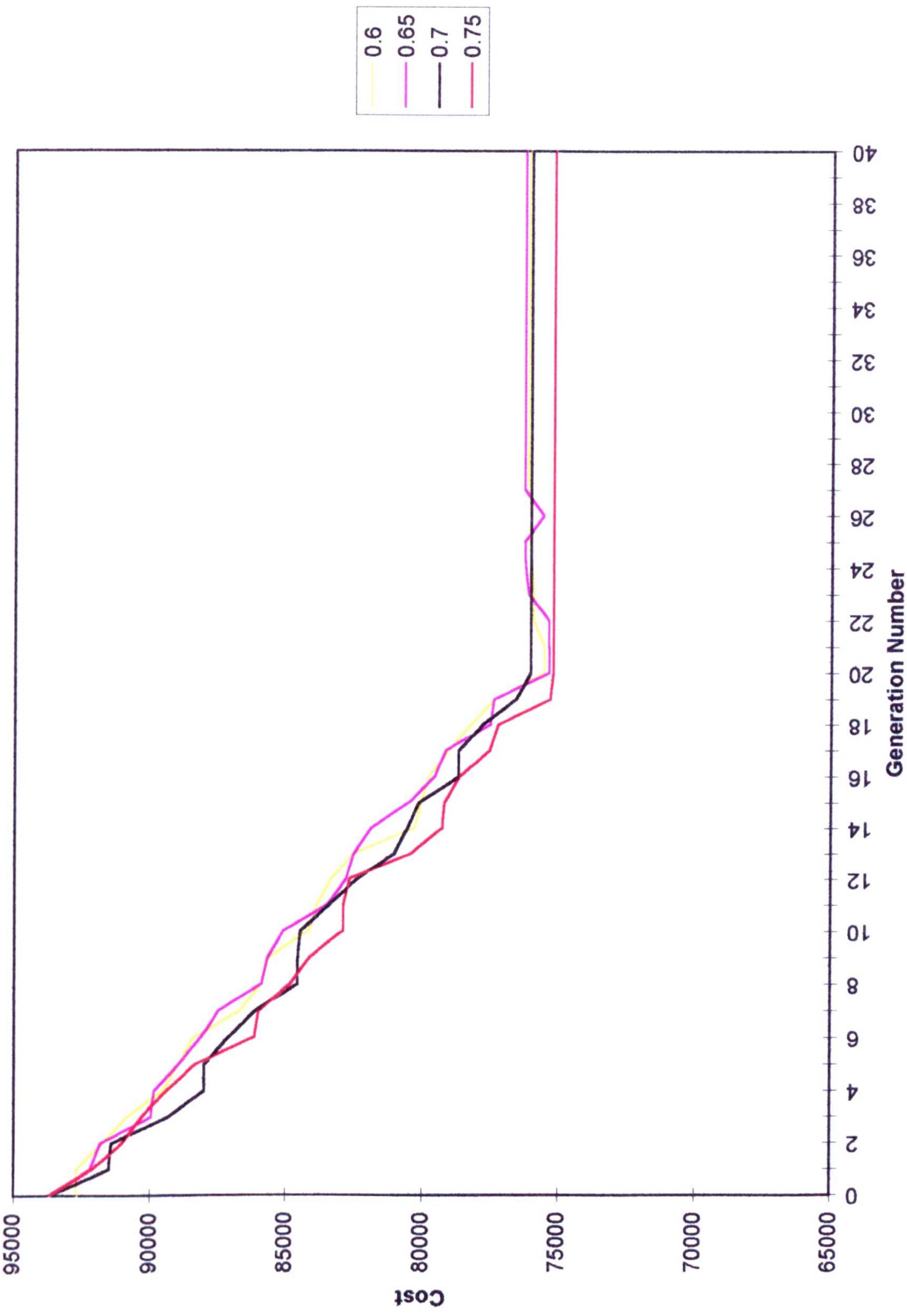


Figure 5.16 Effect of Crossover Rate on the MRP Lot Sizing Problem

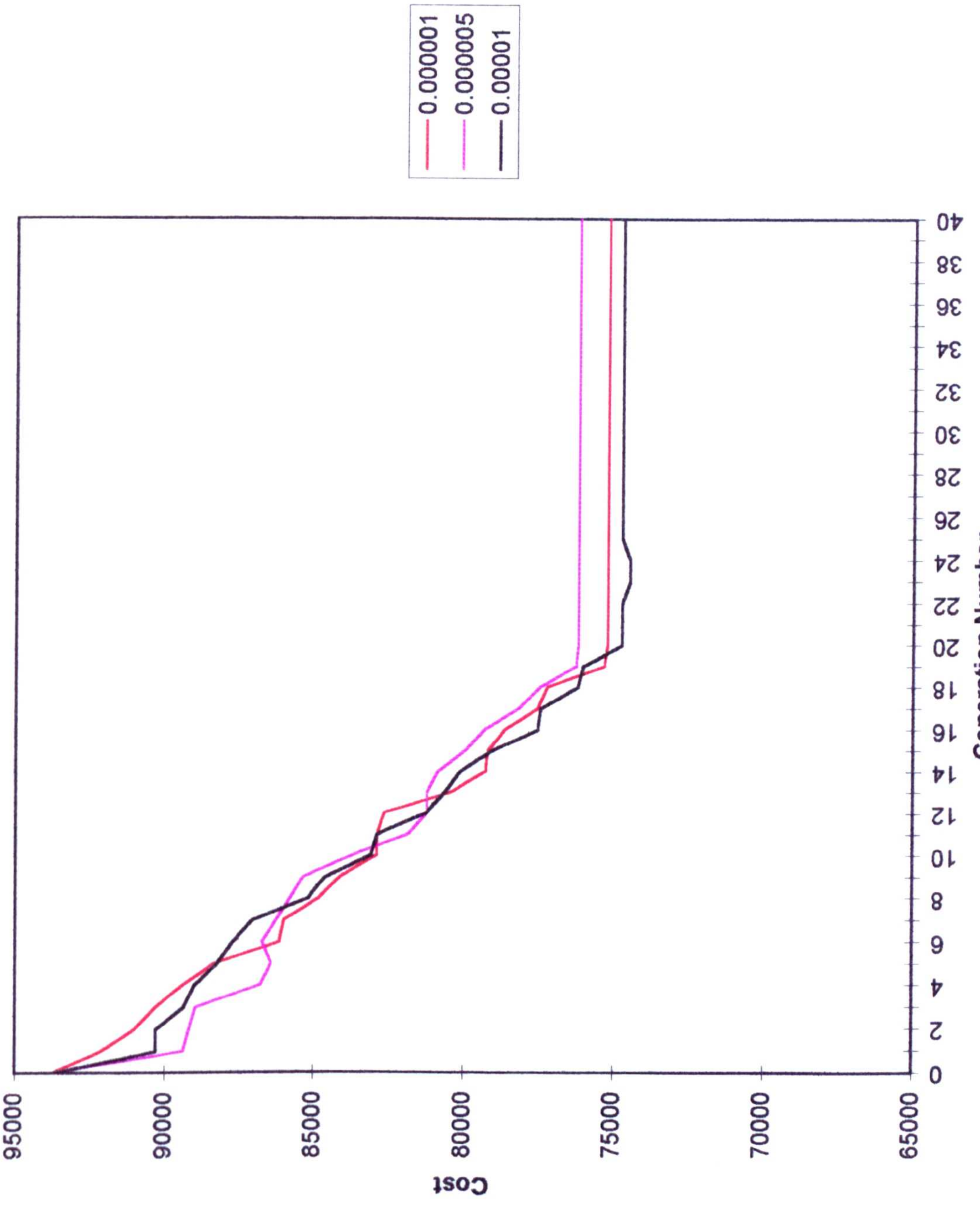


Figure 5.17 Effect of Mutation Rate on the MRP Lot Sizing Problem

5.3.6 Scaling

Sigma scaling experiments were conducted with values of sigma equal to 1, 2, 3, 4 and 5 and Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. In both sets of experiments the following parameters were held constant.

a. roulette wheel selection without elitism.

b. two point crossover with a probability of 0.75.

c. mutation with a probability of 0.000001, and

d. population size of 1000.

The results are shown in Figures 5.18 and 5.19 respectively.

5.3.7 Comparison of GA and MOM Results

The experiments were carried out on a Sparc workstation and required approximately 1200 seconds to find a solution. The solutions found by both the GA and the MOM method are shown in Appendix I.

The GA found better solutions to that calculated by McLaren's Order Moment which calculated a minimum cost of £77,689 in comparison to the best GA solution of £74,861.

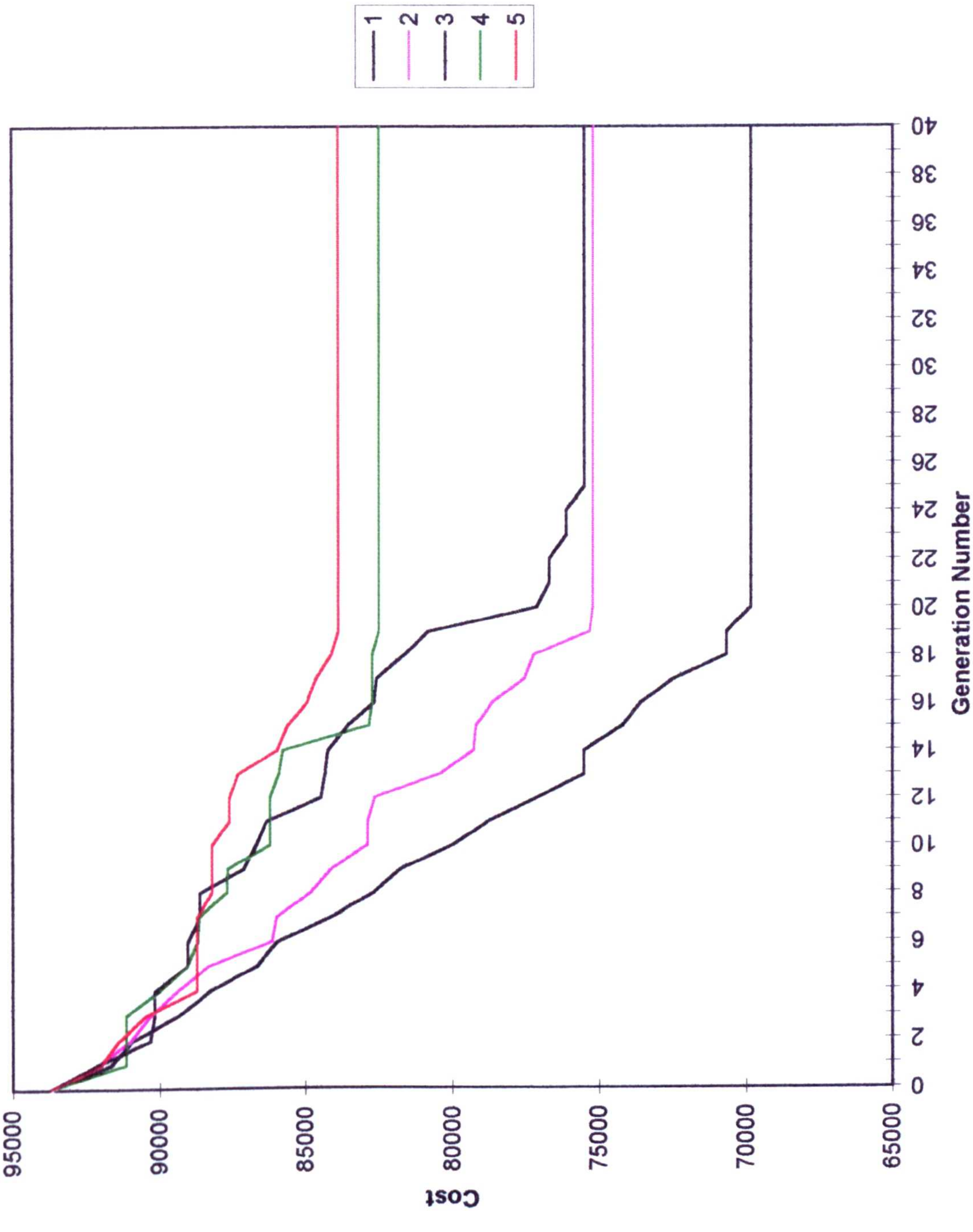


Figure 5.18 Effect of Sigma Scaling on the MRP Lot Sizing Problem

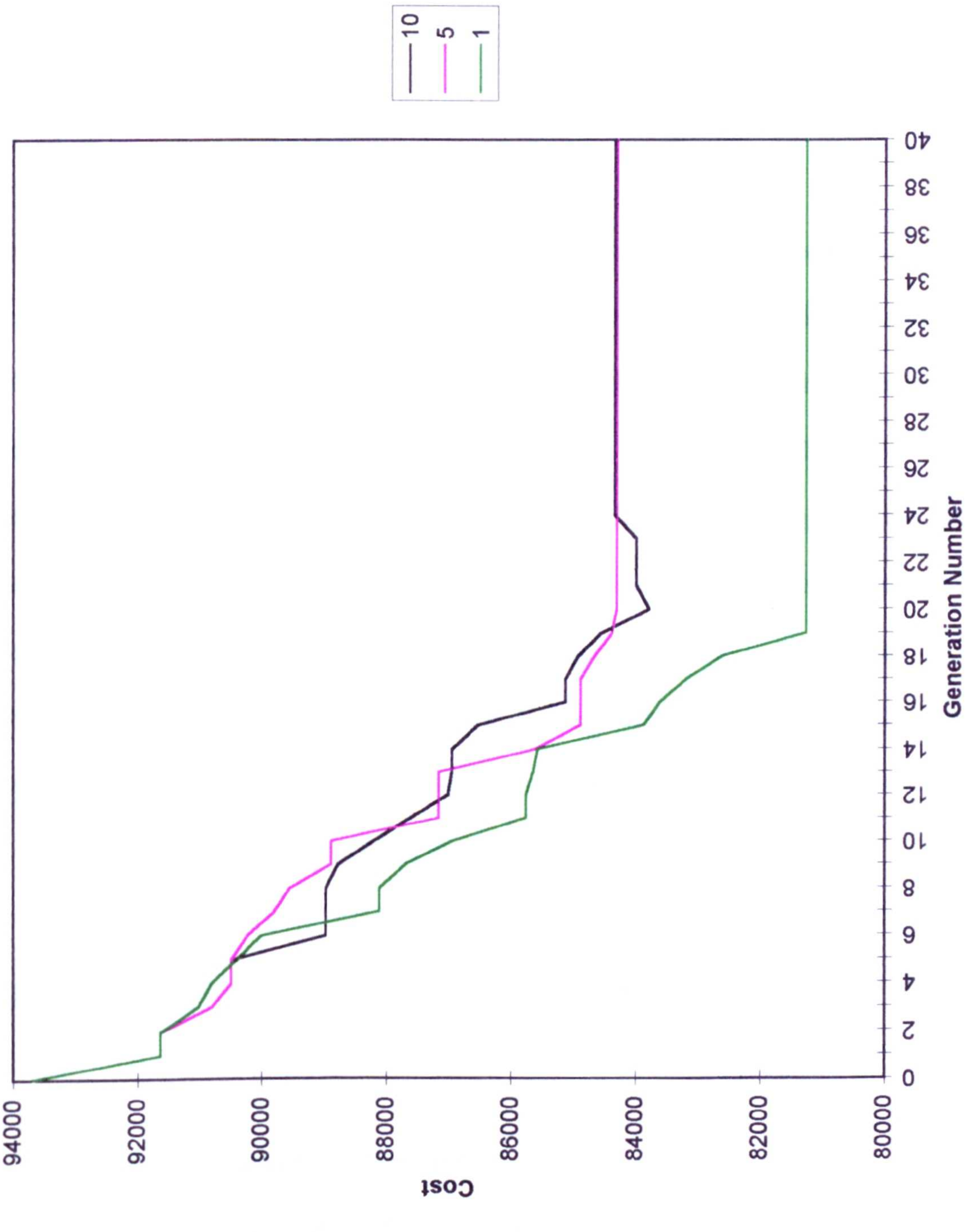


Figure 5.19 Effect of Window Size on the MRP Lot Sizing Problem

5.4 Line Balancing

5.4.1 Coding

In order to identify the ability of genetic algorithms to minimise the balancing loss when designing flow process lines, the problem illustrated in Figure 5.20 was used. The assembly line balancing problem is one of allocating tasks to workstations, in order to minimise the balancing loss (shown in Equation 5.1) with the duration of the operations allocated to each workstation not exceeding the cycle time and no precedence constraints being violated. The solution string was divided into elements in which each element represents an operation to be allocated to a workstation and the value of that element represents the workstation in which that operation is to be undertaken. Figure 5.21 illustrates the GA coding method i.e., operation 1 would be carried out at workstation 3 and operation k-1 at workstation 4. This coding methods produces a solution string of 44 digits in length.

$$\text{Balancing Loss} = \frac{n(c)-T}{n(c)}\cdot 100 \qquad \text{Equation 5.1}$$

Where:

T = Sum of individual operation times,

n = number of workstations required, and

c = cycle time

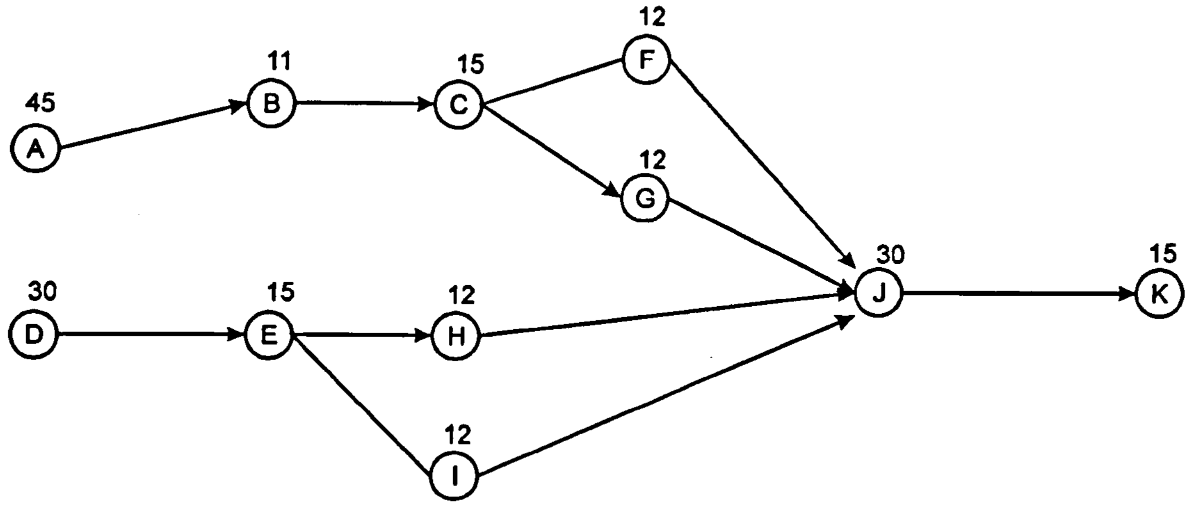


Figure 5.20 Precedence Diagram

operation	1	2	3	......	k-1	k
gene value	3	1	1	......	4	3

Figure 5.21 GA Representation

5.4.2 Objective Function

Equation 5.2 represents the objective function used to measure the fitness of individual solutions. This equation both calculates the balancing loss and adds penalties for solutions that allow workstations to exceed the cycle time and/or allow precedence constraints to be broken.

Fitness Value = BL + (100*Np) + (10*Nt) Equation 5.2

Where:

BL = Balancing Loss

Np = Number of Precedence Constraints Broken

Nt = Number of Workstations exceeding Cycle time

5.4.3 Selection

To compare the relative efficiencies of the 'roulette wheel selection' and the 'roulette wheel selection with elitism' the following parameters held constant,

a. number of replications set at 10,

b. two point crossover with a probability value of 0.60,

c. mutation probability rate of 0.005,

d. population size of 100, and

e. Sigma scaling set at a value of 2

The results are shown in Figure 5.22.

5.4.4 Crossover

In order to identify the optimum rate to employ, experiments were carried out using rates of 0.6, 0.65, 0.7 and 0.75. During the experiments other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. mutation probability rate set at 0.005,

d. population size set at 100, and

e. Sigma scaling with a sigma value set at 2

The results are shown in Figure 5.23.

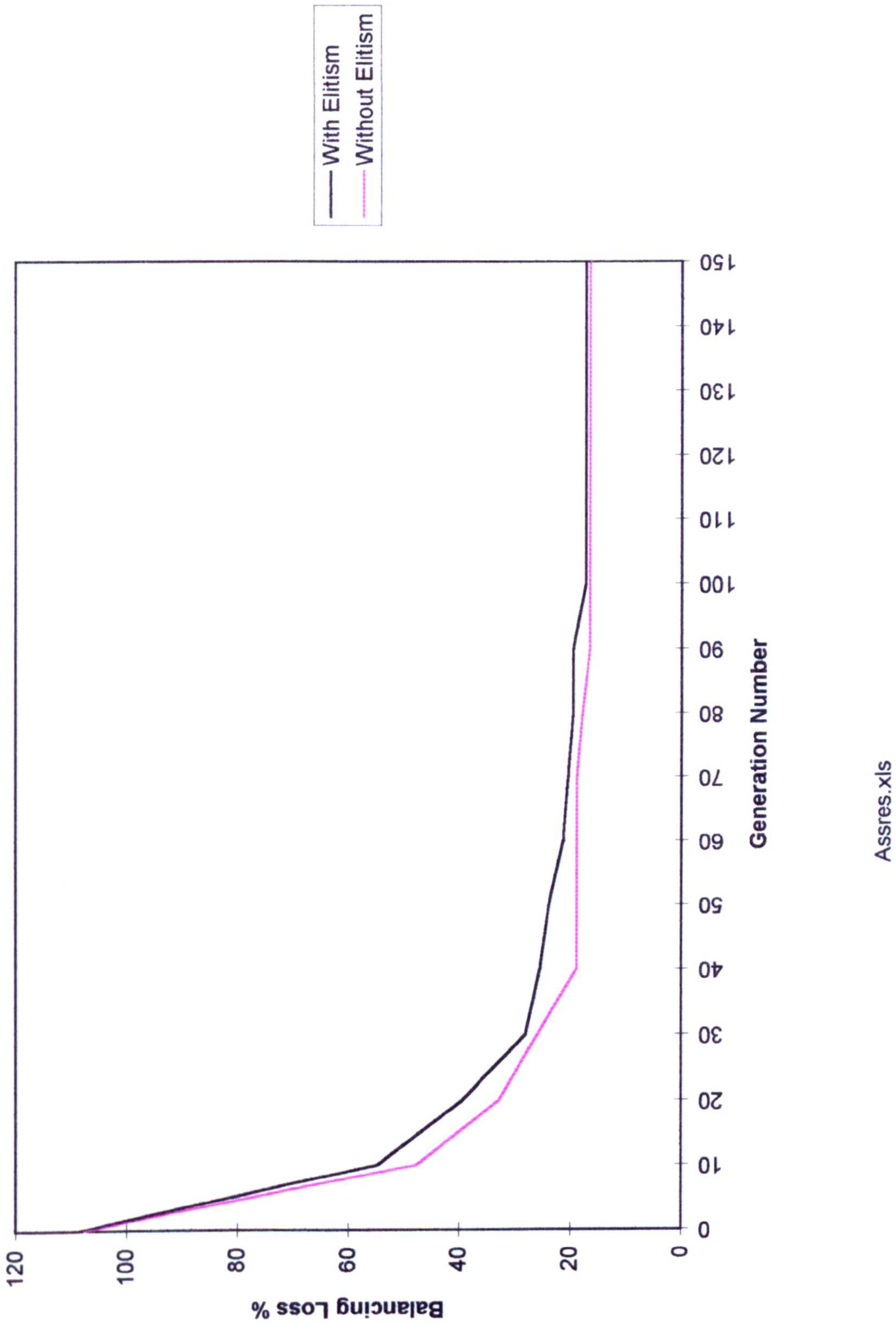


Figure 5.22 Effect of Elitist Strategy for the Line Balancing Problem

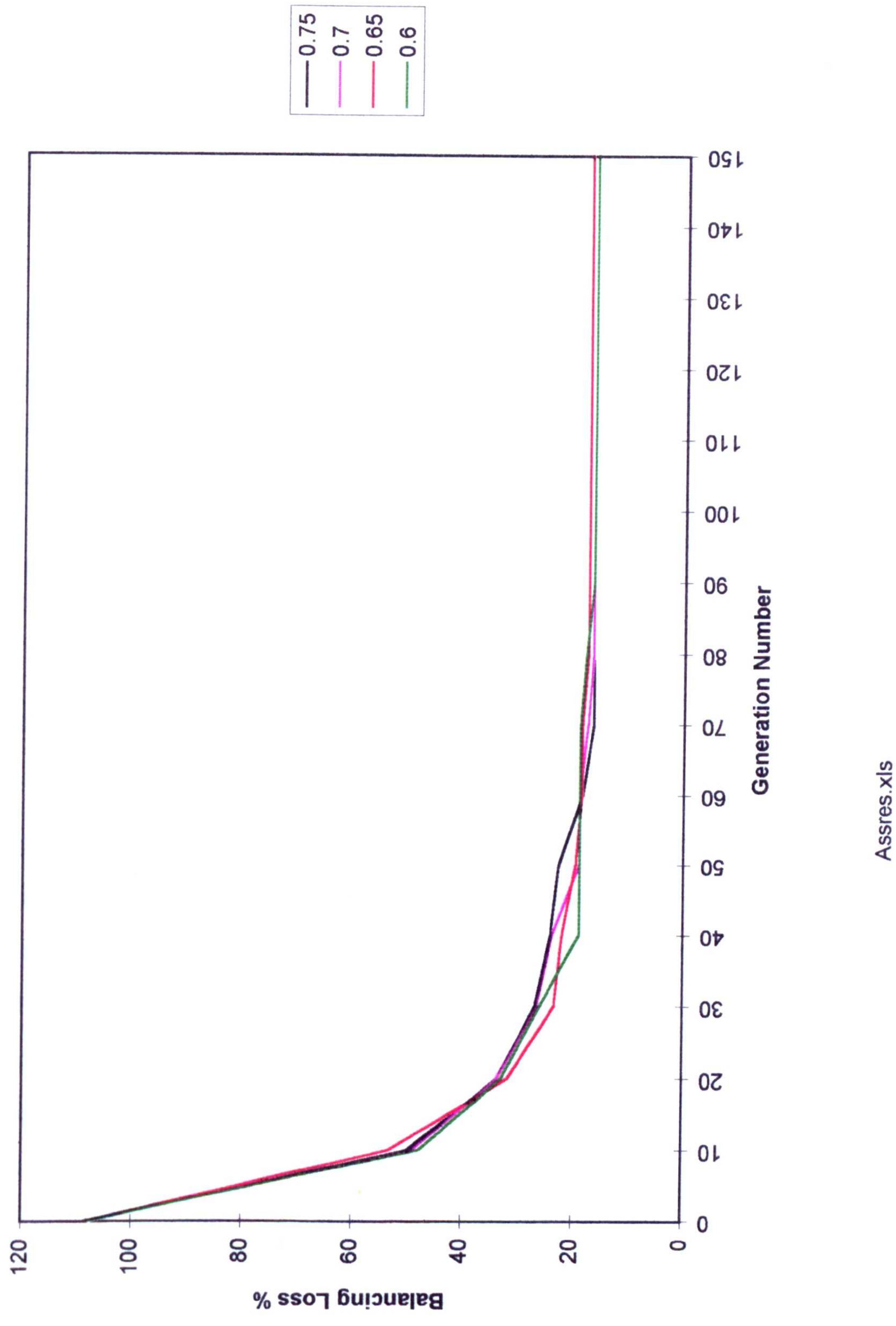


Figure 5.23 Effect of Crossover Rate for the Line Balancing Problem

5.4.5 Mutation

The effect of the mutation rates, 0.005, 0.0022 and 0.0005 were investigated using the following settings:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. two point crossover with probability rate set at 0.60,

d. population size set at 100, and

e. Sigma scaling with a sigma value set at 2.

The Results are shown in Figure 5.24.

5.4.6 Scaling

Sigma scaling experiments were con ducted with values of sigma equal to 1, 2, and 3 and Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. For both sets of experiments the following parameters were set constant, i.e.:

a. roulette wheel selection without elitism,

b. two point crossover with a probability set at 0.60,

c. mutation with a probability rate set at 0.005, and

d. population size set at 100.

Figures 5.25 and 5.26 illustrate the effect of sigma and window size.

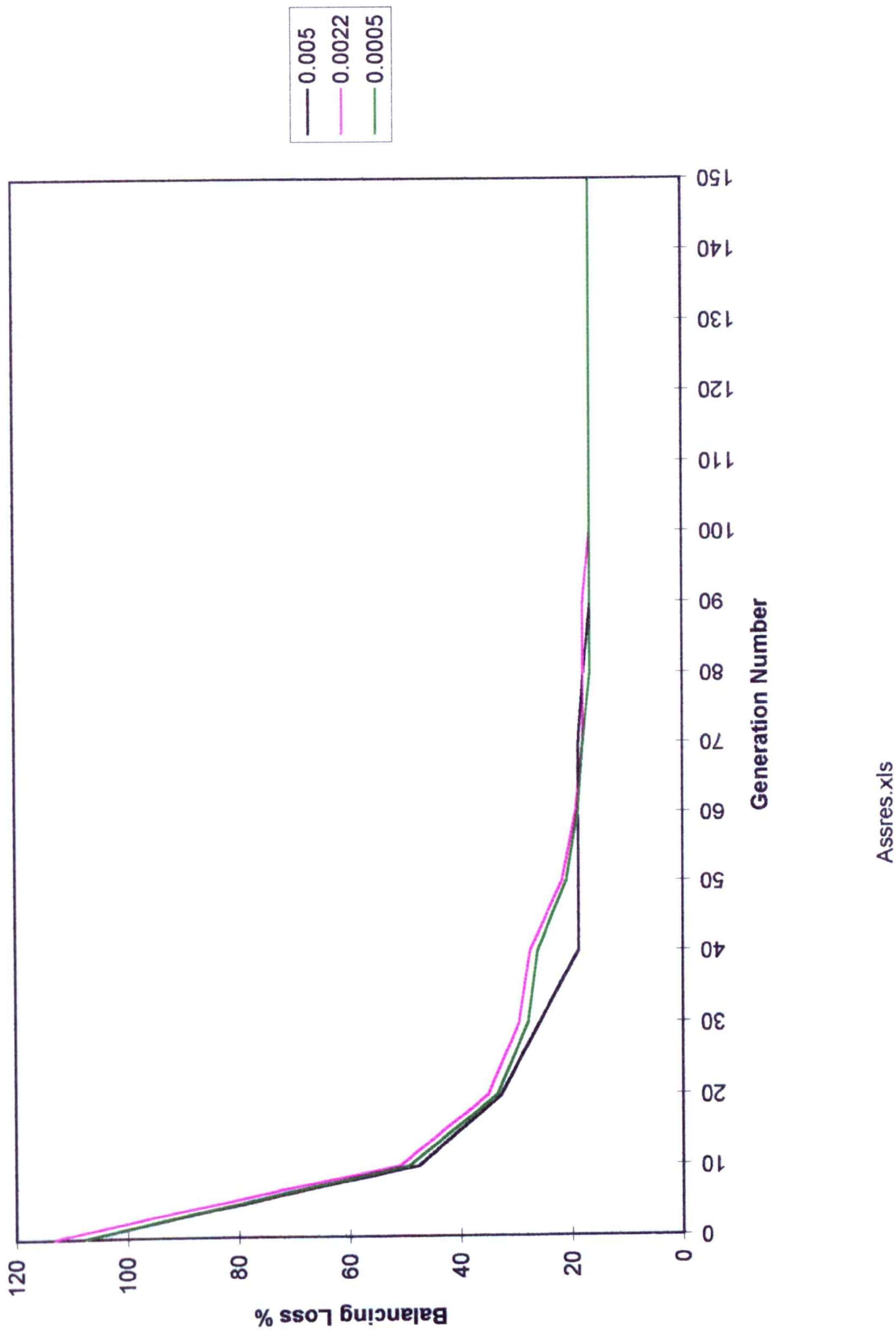


Figure 5.24 Effect of Mutation Rate for the Line Balancing Problem

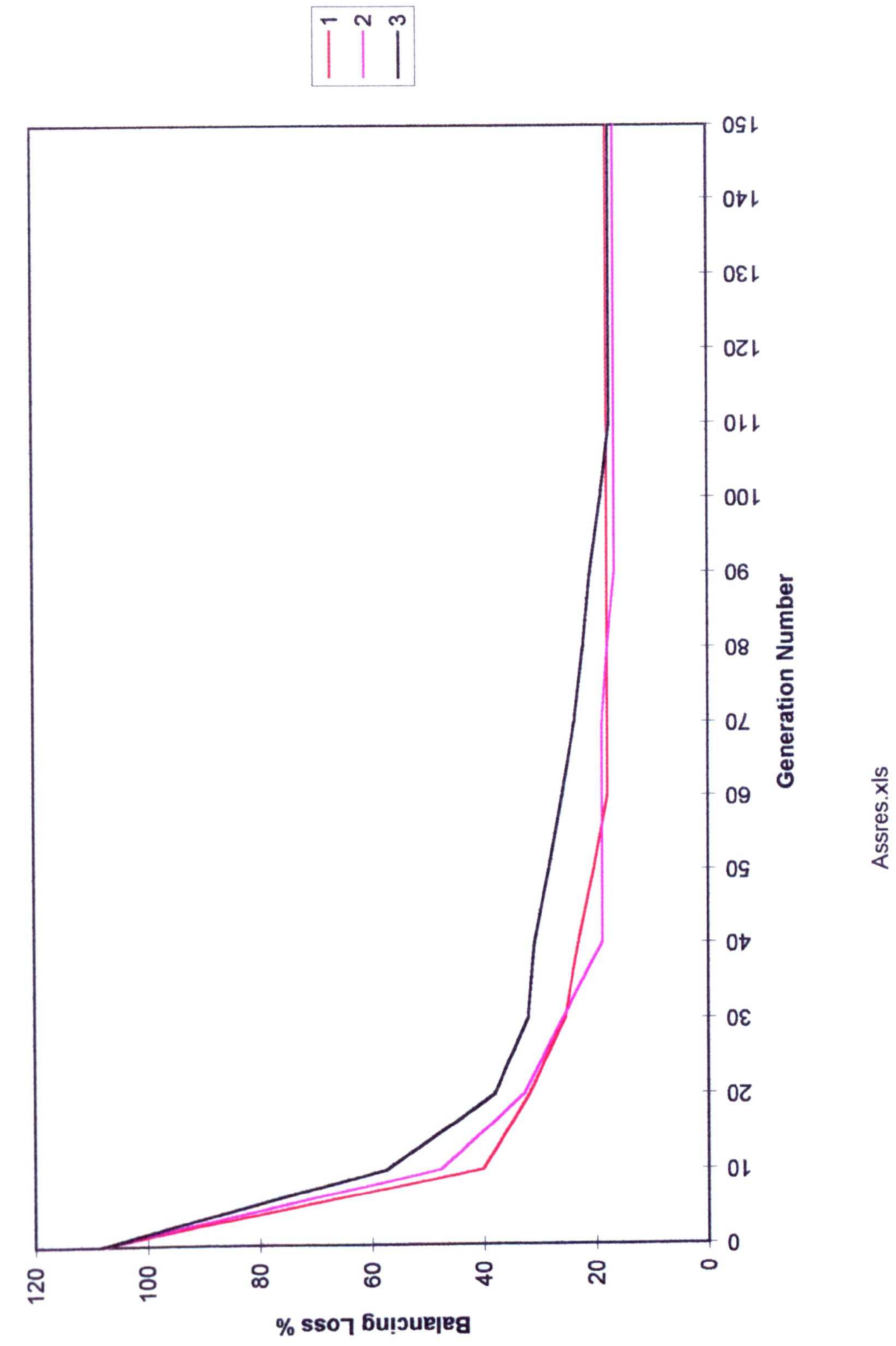


Figure 5.25 Effect of Sigma Scaling for the Line Balancing Problem

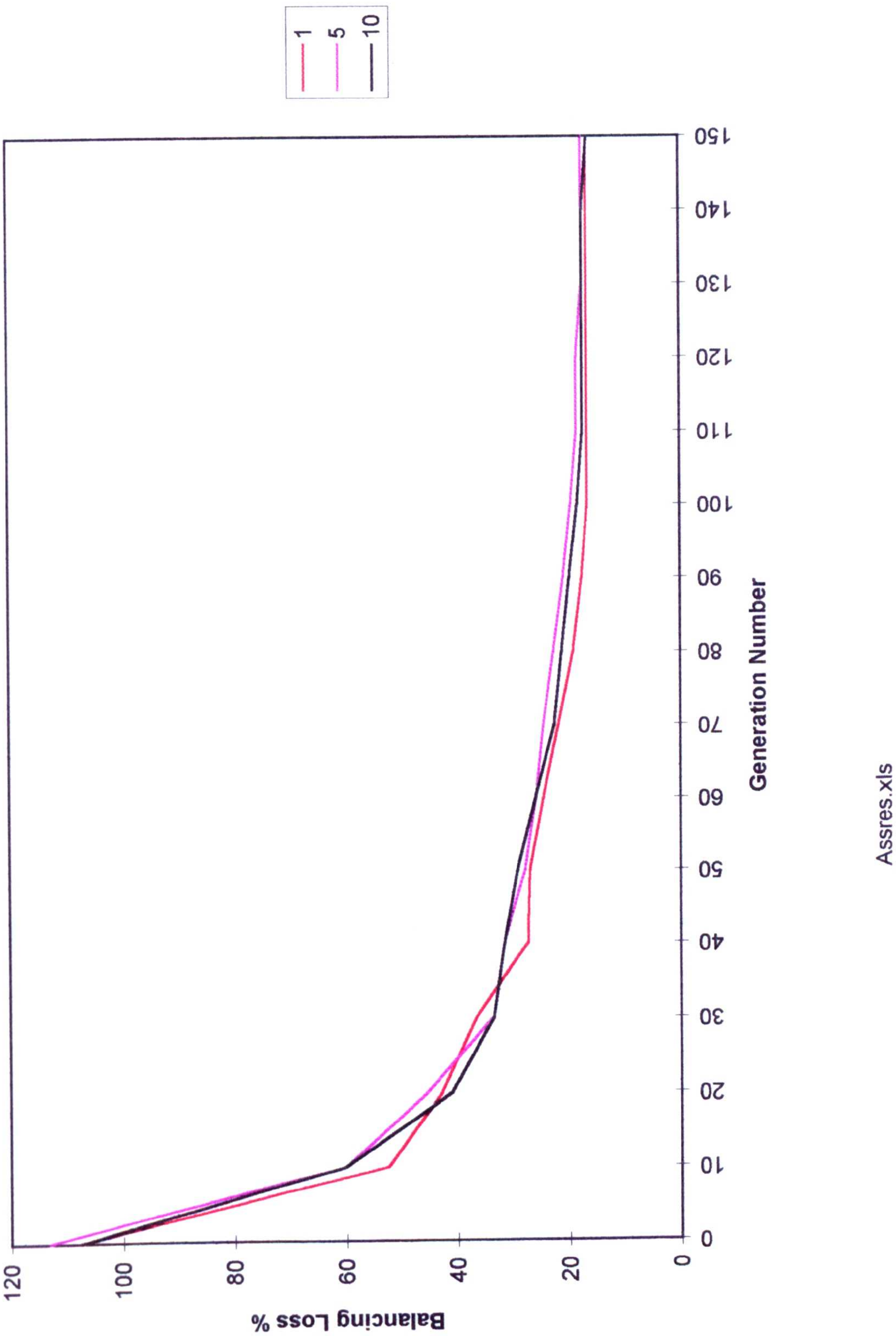


Figure 5.26 Effect of Window Scaling for the Line Balancing Problem

5.4.7 Comparison of RPW and GA Results

The experiment was carried out on a Sparc workstation and required approximately 12 seconds to determine the best solution with a balancing loss of 16.4 %. This solution was also identified by the Rank Positional Weight technique, but the GA identified a number of alternative solutions as shown in Table 5.19. The calculations for the RPW method are shown in Appendix II.

Solution	WS 1	WS 2	WS 3	WS 4	WS 5
1	A	B,D	C,E,F	G,H,I	J,K
2	A	B,D	C,E,G	F,H.I	J,K
3	A	B,D	C,E,H	F,G,I	J,K
4	A	D	B,C,E	F,G,H,I	J,K

GENETIC ALGORITHM SOLUTION

Solution	WS 1	WS 2	WS 3	WS 4	WS 5
1	A	B,D,E	C,F	G,H,I	J,K

RANK POSITIONAL WEIGHT TECHNIQUE SOLUTION

Table 5.19 Solutions Identified by the GA and the RPW method

5.5 Facilities Layout

Here it was necessary to identify the potential for using GA procedures to determine the positions on the shopfloor that individual workstations should be sited. The basic types of layout examined included both single row and multi-row problems.

5.5.1 Coding

The strings for both single and multi-row problems were coded such that they represented the positions on the shopfloor of the co-ordinates of the centre of a machine. For the single row layout problem, each machine needs only to possess one parameter in the string, i.e. a x-axis co-ordinate. Hence, only the x co-ordinate for each machine needs to be included in the GA code. Figure 5.27 shows a diagrammatical representation of the problem and the coding used for the GA. For the multi-row layout problem, each machine possesses two parameters in the string, i.e. a x-axis co-ordinate and a y-axis co-ordinate which are both measured relative to a pre-defined datum as shown in Figure 5.28.

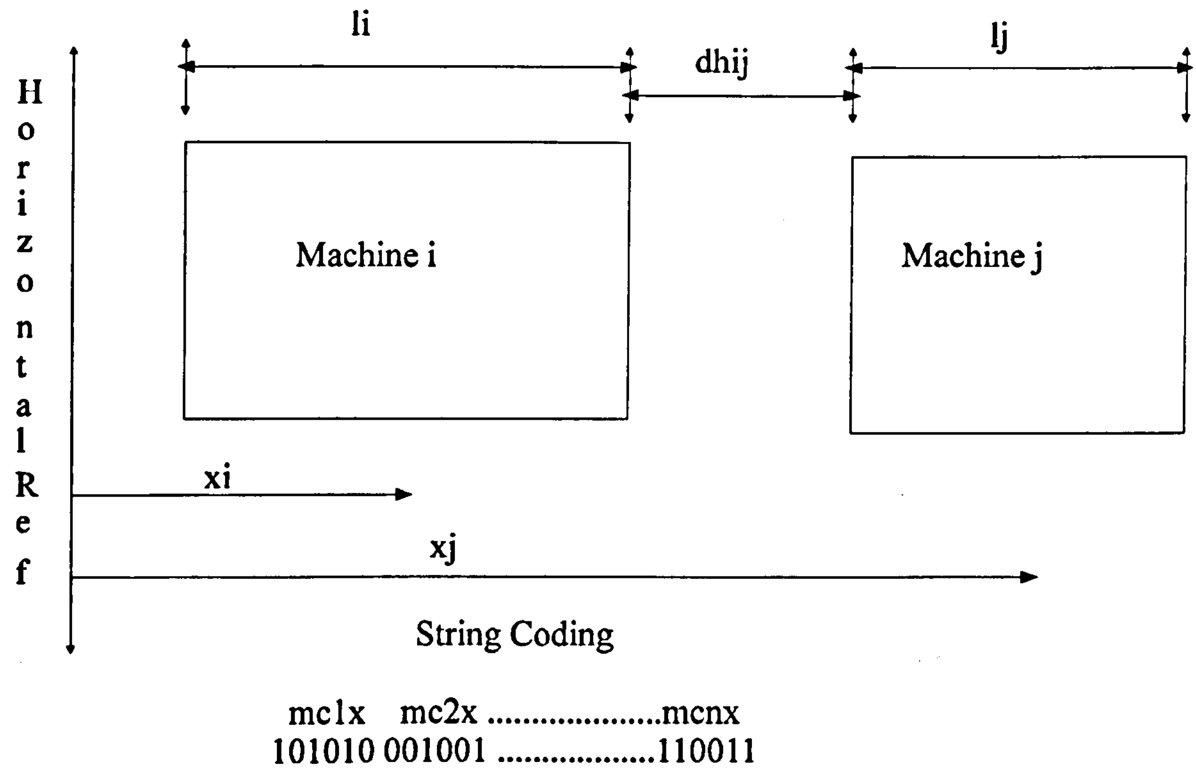
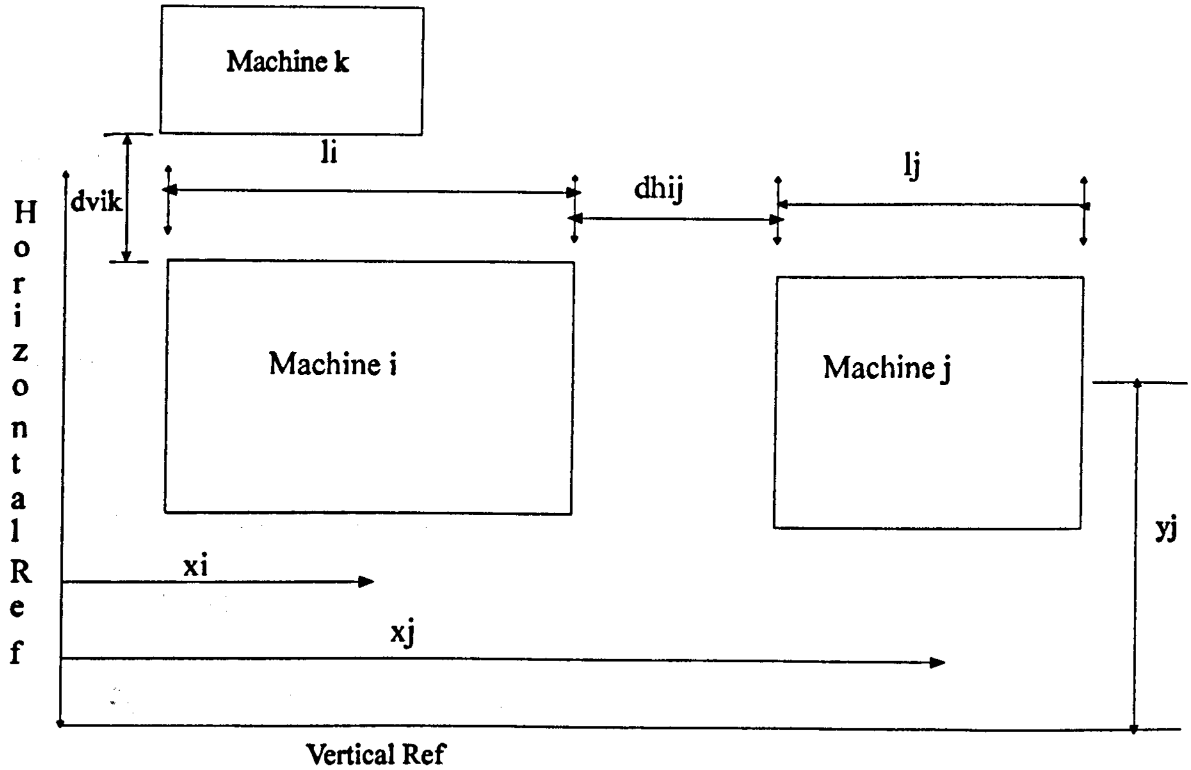


Figure 5.27 Dimensions and String Coding for the Single Row Layout Problem



String Coding

mc1x mc1y mc2x mc2y.................... mcnx mcny
101010 001001 100101 001100..................110011 010100

Figure 5.28 Dimensions and String Coding for Multi-Row Layout Problem

5.5.2 Objective Function

For both single and multi-row layout problems, the objective of the fitness function was to calculate the cost of moving material between workstations. Here, Equation 5.3 was developed to provide a suitable objective function z to determine the cost of moving material for the single row layout problem.

$$z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} |x_i - x_j| \qquad \text{Equation 5.3}$$

subject to:

$$|x_i - x_j| \geq 1/2(l_i + l_j) + d_{hij} \qquad \begin{array}{l} i = 1 \ldots n-1 \\ j = 1 \ldots n \end{array} \qquad \text{Equation 5.4}$$

$$xi \geq 0 \qquad i = 1 \ldots n \qquad \text{Equation 5.5}$$

where

x_i = co-ordinate of machine i in the x direction,

l_i = length of machine i,

c_{ij} = cost of carrying one unit between machines i and j in pounds,

f_{ij} = frequency of movements between machines i and j,

d_{hij} = minimum clearance distance between machines i and j in the x axis in metres,

and

n = number of machines.

For the multi-row layout problem, in order to include the additional y-axis co-ordinate, Equation 5.6 was developed from Equation 5.3. Again the objective function z is used to determine the cost of moving material between workstations.

$$z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} (|x_i - x_j| + |y_i - y_j|) \qquad \text{Equation 5.6}$$

subject to

$$|x_i - x_j| \geq 1/2(l_i + l_j) + d_{hij} \qquad \begin{matrix} i = 1 \ldots n-1 \\ j = 1 \ldots n \end{matrix} \qquad \text{Equation 5.7}$$

$$|y_i - y_j| \geq 1/2(w_i + w_j) + d_{vij} \qquad \begin{matrix} i = 1 \ldots n-1 \\ j = 1 \ldots n \end{matrix} \qquad \text{Equation 5.8}$$

Where:

y_i = co-ordinate of machine i in the y direction,

w_i = width of machine i, and

d_{vij} = minimum clearance distance between machines i and j in the y axis.

5.5.2.1 Penalty Function

When developing GA procedures for the layout problem, it was found necessary to include a condition within the fitness function for penalising infeasible solutions. In this respect, it was essential that the GA procedures did not assign low material handling costs to solutions in which overlapping of workstations occurred on the shopfloor. The GA procedures developed, therefore, were designed to initially test each solution to determine if any constraints had been violated. If the constraints had been upheld then the fitness function was set to the values calculated using Equations 5.3 and 5.6. If a violation of one or more constraints occurred then the solution was assigned a penalty value proportional to the number of constraints violated. Solutions that have violated constraints may contain good information can, therefore, be retained within a solution population. Constraints are included within an objective function using Equations 5.7 and 5.8.

In both the single and multi-row problems, a violation occurs only when the value of the constraint shown in Equations 5.7 and 5.8 is less than zero. The penalty function then simply squares the value by which the constraint is violated. The objective function for the single-row layout problem then becomes as shown in Equation 5.9, where the first part of the equation represents the costs of carrying the material between the machines and the second part represents the penalty cost added due to constraints being violated.

$$z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij}(|xi - xj|) + \beta \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\min(0, (|x_i - x_j| - (\frac{1}{2}(l_i + l_j) + d_{hij}))))^2 \qquad \text{Equation 5.9}$$

With multi-row layouts, only one of the constraints needs not to be broken to prevent the application of the penalty function. The reason for this rule is illustrated in Figure 5.28 where machines k and i do not overlap despite the co-ordinates of machine k overlapping the co-ordinates of machine i in the x direction but not in the y direction. Therefore if the maximum value of A and B in Equation 5.13 is less than zero then the constraints have been violated in both the x and y axes. In order to ensure that both the x-axis and y-axis constraints have been violated the objective function becomes:

$$z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij}(|xi - xj| + |yi - yj|) + \beta \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\min(0, (\max(A, B))))^2 \qquad \text{Equation 5.10}$$

Where:

$$A = (|x_i - x_j| - (\frac{1}{2}(l_i + l_j) + d_{hij})) \qquad \text{Equation 5.11}$$

$$B = (|y_i - y_j| - (\frac{1}{2}(w_i + w_j) + d_{vij})) \qquad \text{Equation 5.12}$$

A number of both single-row and multi-row problems, with up to 12 machines of unequal length and width have been examined using GA procedures. The data used in the problem is shown in Tables 5.20, 5.21 and 5.22.

Machine	Length	Width
MC1	20	20
MC2	20	15
MC3	15	10
MC4	20	20
MC5	30	30
MC6	30	30
MC7	15	10
MC8	15	10
MC9	40	25
MC10	20	15
MC11	25	40
MC12	20	20

Table 5.20 Lengths and Widths of Machines (in Metres)

		To											
	MC	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
F	**1**	0	5	20	1	0	1	1	0	20	5	10	1
r	**2**	5	0	1	30	5	10	20	0	0	1	20	5
o	**3**	20	1	0	5	10	30	15	1	5	0	0	10
m	**4**	1	30	5	0	0	0	1	30	5	10	15	20
	5	0	5	10	0	0	30	10	5	15	1	5	5
	6	1	10	30	0	30	0	20	1	30	5	10	15
	7	1	20	15	1	10	20	0	10	0	30	5	0
	8	0	0	1	30	5	1	10	0	0	0	1	30
	9	20	0	5	5	15	30	0	0	0	20	15	5
	10	5	1	0	10	1	5	30	0	20	0	10	0
	11	10	20	0	15	5	10	5	1	15	10	0	5
	12	1	5	10	20	5	15	0	30	5	0	5	0

Table 5.21 Frequency of Journeys between Machines

		To										
	MC	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
F	**1**	0	20	40	5	30	10	5	5	50	30	40	10
r	**2**	20	0	10	30	15	20	20	20	30	50	40	10
o	**3**	40	10	0	40	50	20	15	30	50	40	20	10
m	**4**	5	30	40	0	30	10	15	20	30	40	50	40
	5	30	15	50	30	0	5	10	30	40	20	15	50
	6	10	20	20	10	5	0	20	40	15	30	50	5
	7	5	20	15	15	10	20	0	5	20	40	30	20
	8	5	20	30	20	30	40	5	0	10	20	50	40
	9	50	30	50	30	40	15	20	10	0	30	10	40
	10	30	50	40	40	20	30	40	20	30	0	20	5
	11	40	40	20	50	15	50	30	50	10	20	0	10
	12	10	10	10	40	50	5	20	40	30	5	10	0

Table 5.22 Cost of Journey/Unit Distance Travelled between Machines (£'s)

Before GA experiments could be carried out it was necessary to determine suitable values for the penalty coefficients, i.e. β in Equations 5.9 and 5.10. If the value chosen for β was too high the algorithm would concentrate on ensuring that the constraints were not violated and ignore the cost of moving materials, whereas if the value of β was set too low the algorithm would concentrate on finding the minimum cost for moving the material and ignore violated constraints. A range of values for β (i.e. between 10 and 50) were examined within a GA and the results are shown in Figure 5.29. These results indicate that increasing the value of β over 35 has little effect on reducing the penalty costs added due to constraints being violated, hence in the GA experiments carried out, the β value was set at 35. Figure 5.29 also shows the relationship between the 'carrying cost' and 'penalty cost' parts of the objective function Equation 5.9, ie as the penalty cost decrease the distances between machines increase and hence carrying costs increase.

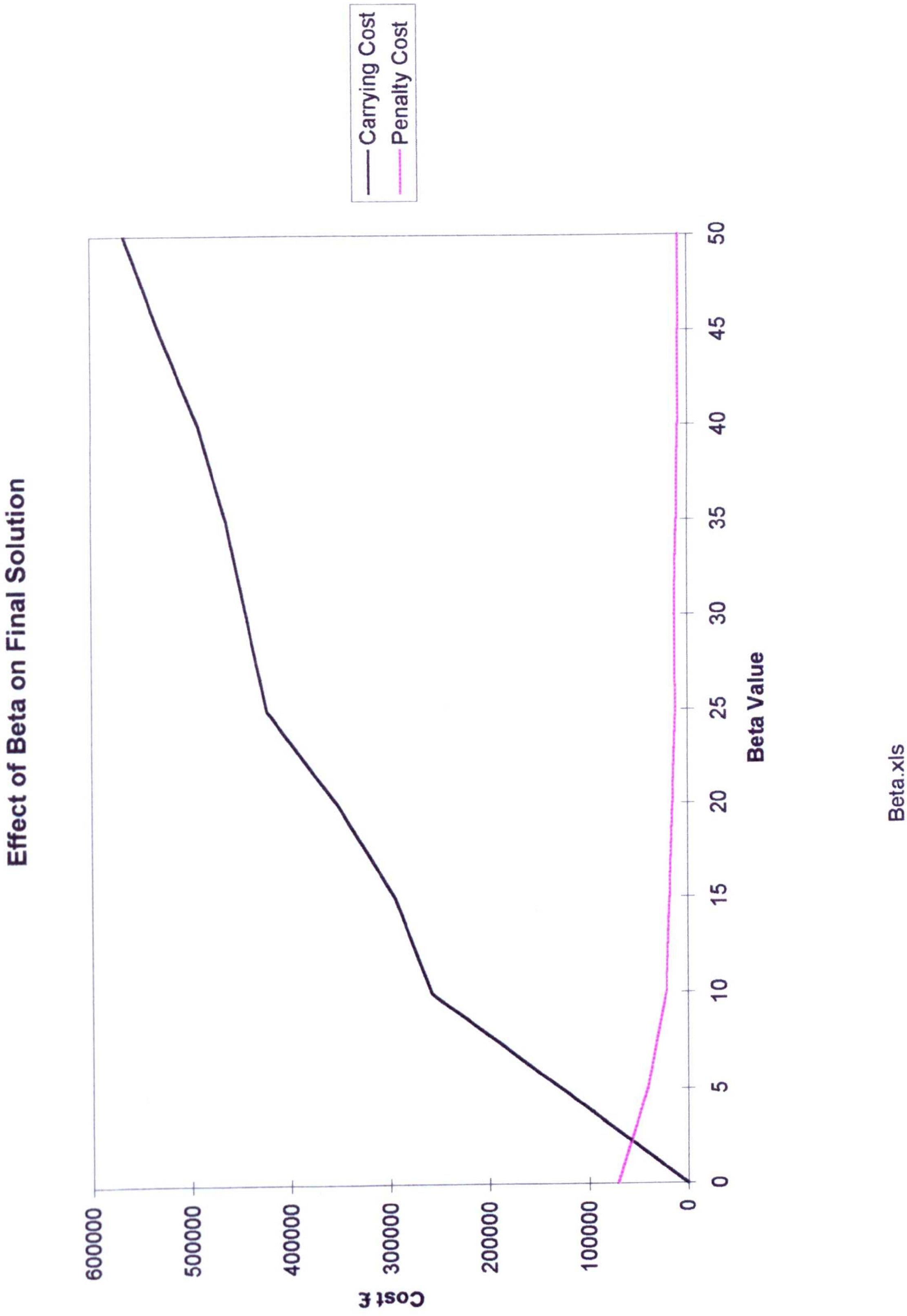


Figure 5.29 Effect of Beta Value

5.5.3 Selection

The alternative selection operators examined were roulette wheel with elitism, tournament and truncated. The efficiency of each selection operator was identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. single point crossover with a probability value set at 0.6,

c. mutation rate with a probability set at 0.0001, and

d. population size set at 100.

The results are presented in Figures 5.30 and 5.31.

5.5.4 Crossover

The crossover operators investigated were single point crossover, two point crossover and uniform crossover. The efficiency of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator,

c. mutation rate with a probability rate set at 0.0001, and

d. population size set at 100.

The results are shown in Figures 5.32 and 5.33.

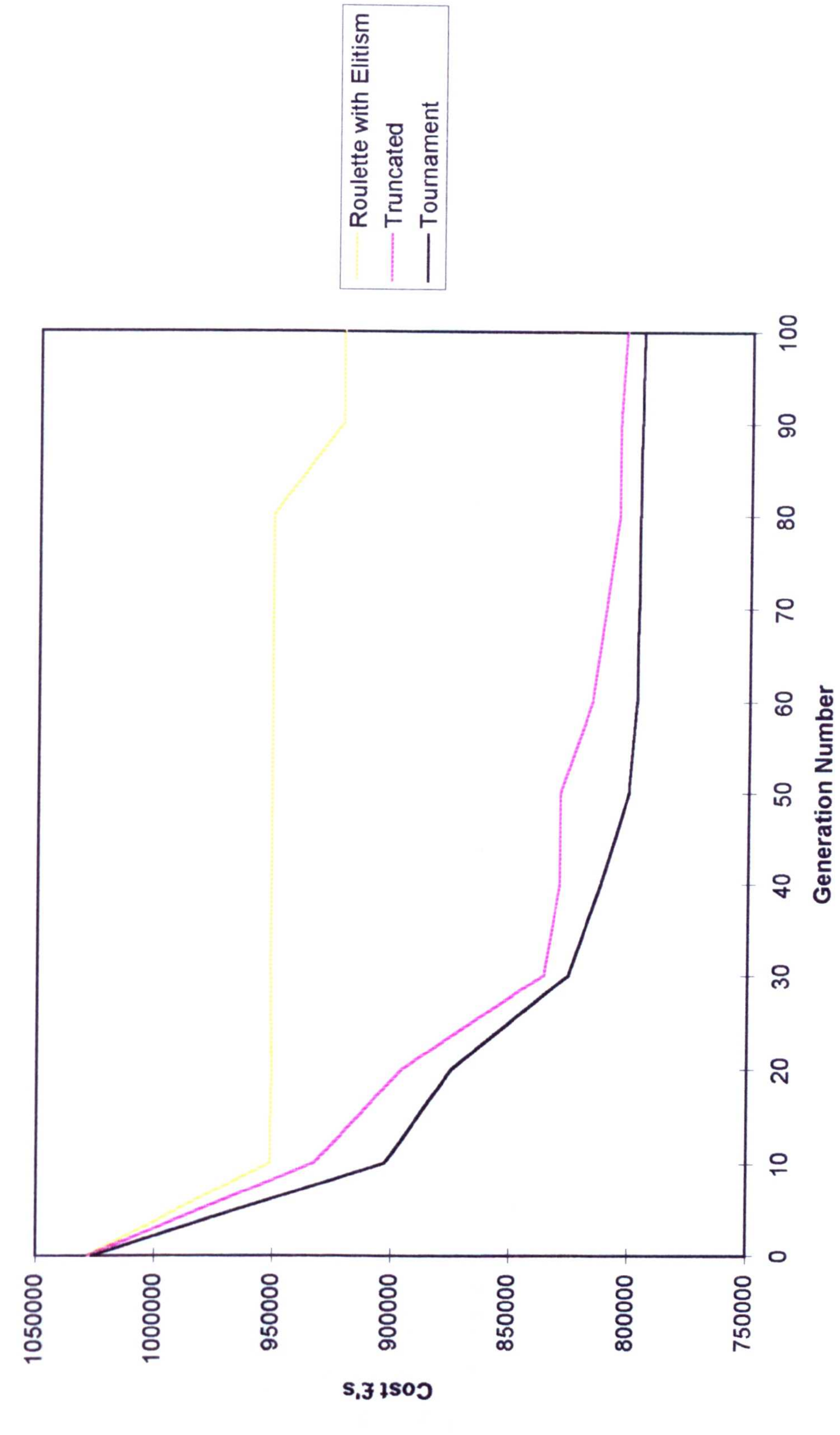


Figure 5.30 Comparison of Selection Operators for the Single Row Layout Problem

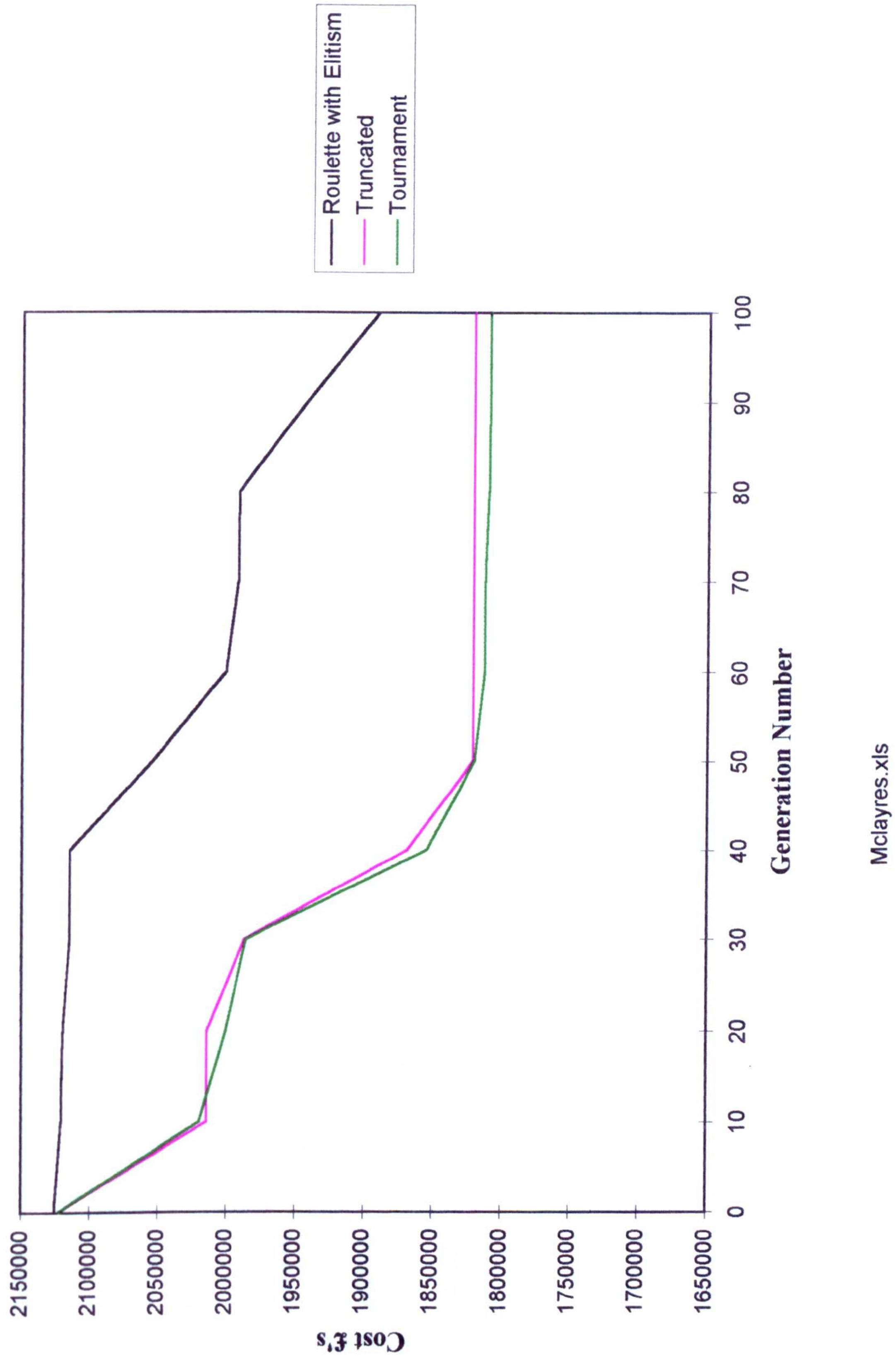


Figure 5.31 Comparison of Selection Operators for the Multi Row Layout Problem

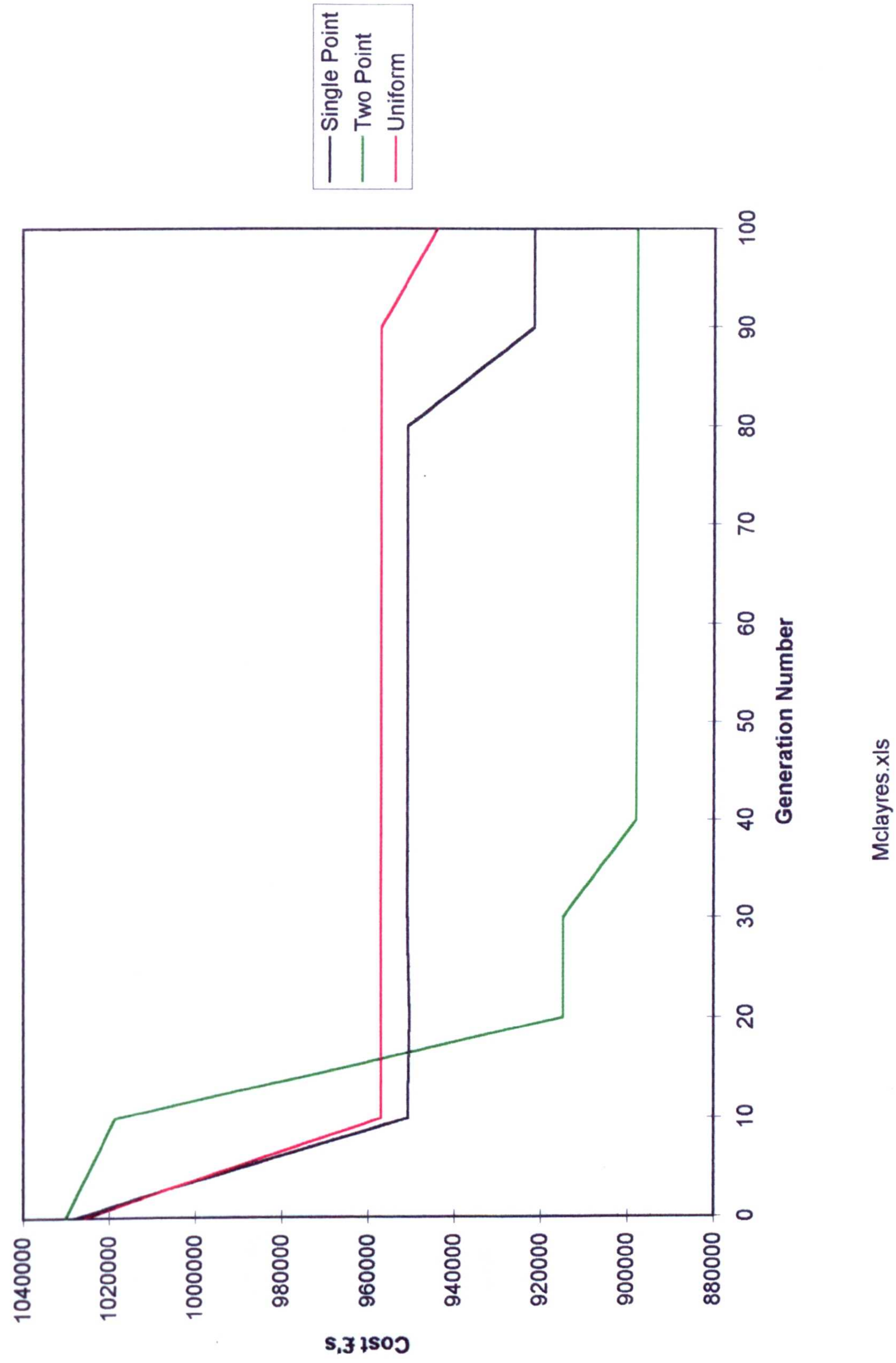


Figure 5.32 Comparison of Crossover Operators for Single Row Layout Problem

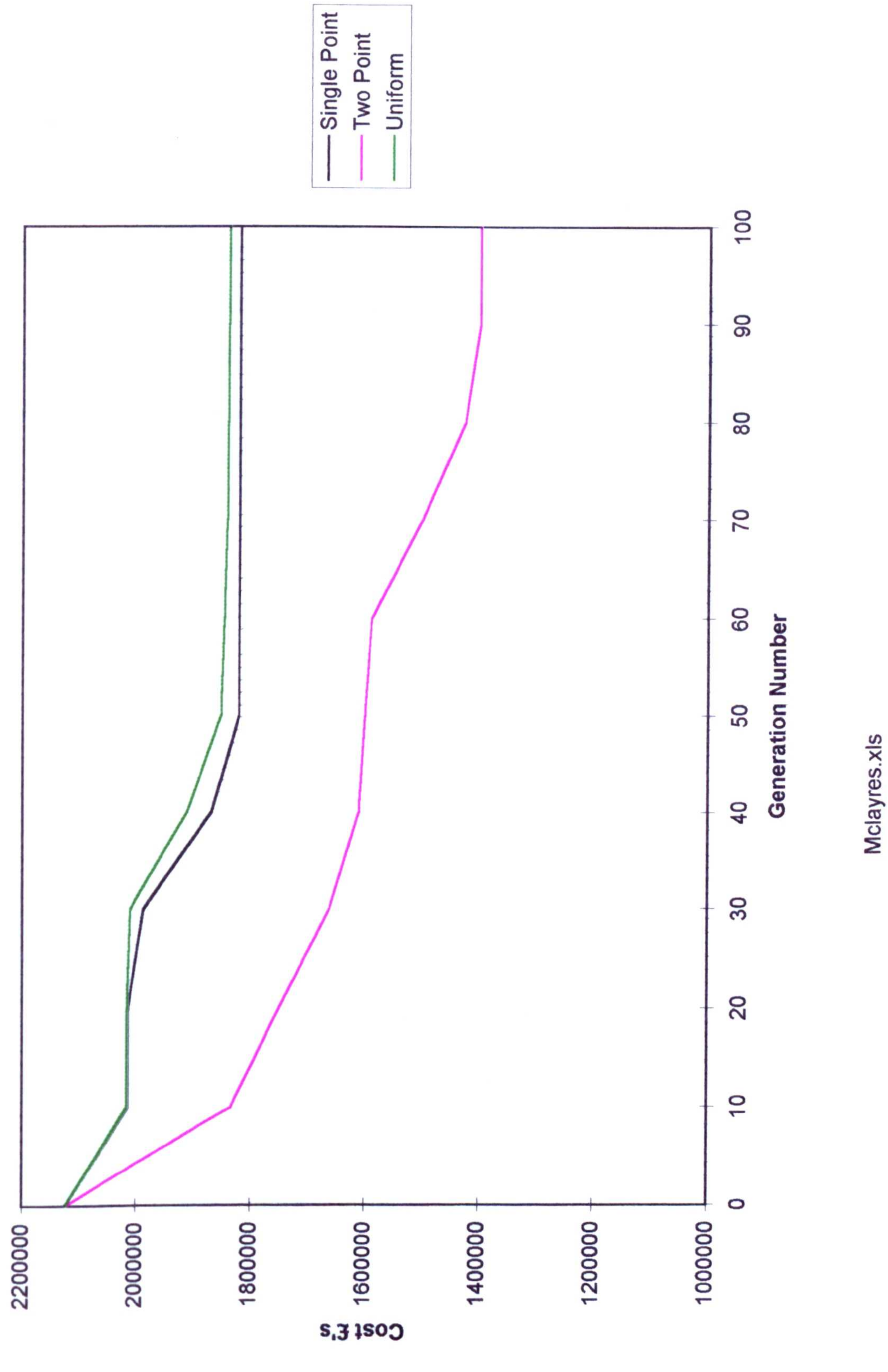
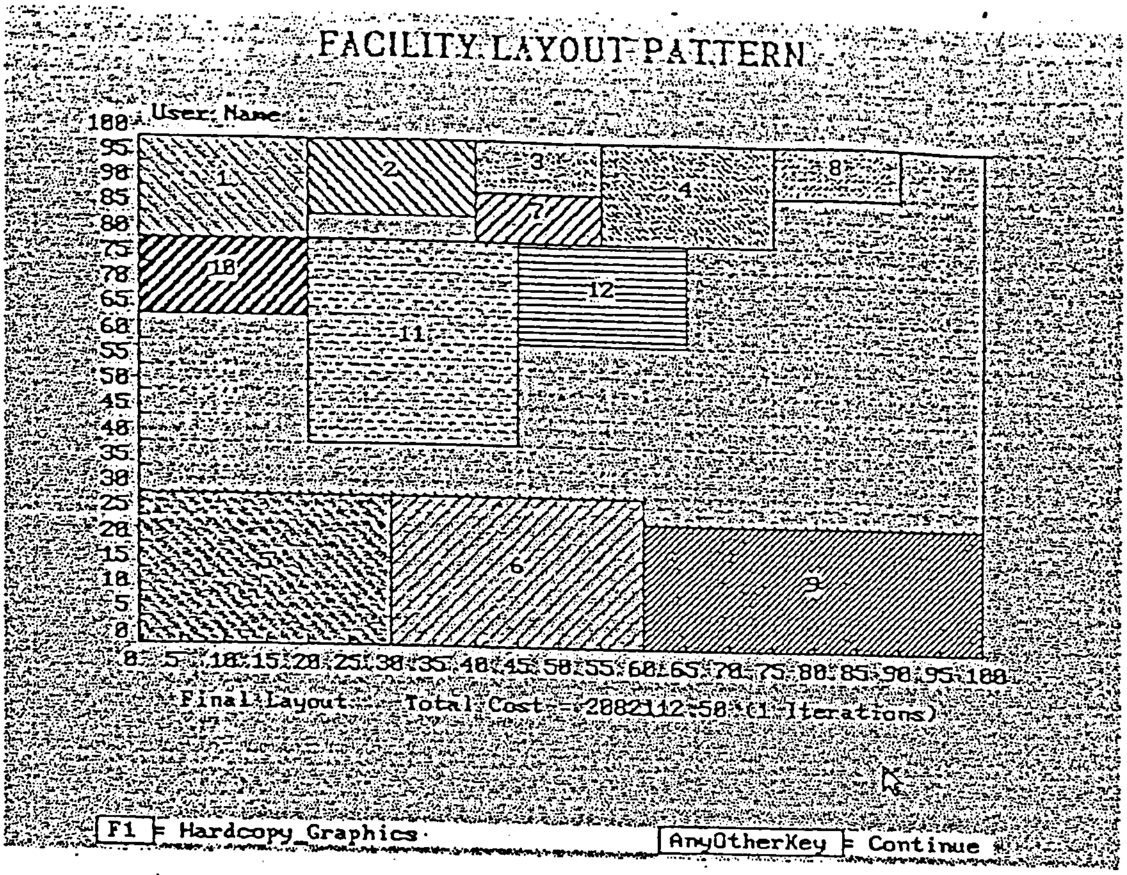


Figure 5.33 Comparison of Crossover Operators for the Multi Row Layout Problem

5.5.5 Comparison of GA and CRAFT Results

The GA has been shown to perform well when compared with the traditional methods of designing plant layouts using the CRAFT method as shown in Figures 5.34 and 5.35 . For the single row problem, the best solution found using the CRAFT method resulted in a handling cost of £ 1,167,600 whereas the lowest cost solution using GA resulted in a cost of £ 802,795, an improvement of 31 %. For the multi row problem, the best solution found using the CRAFT method was £ 2,082,112 compared to the lowest cost solution identified by the GA of £ 1,399,278 by the GA, i.e., an improvement of 32 %.



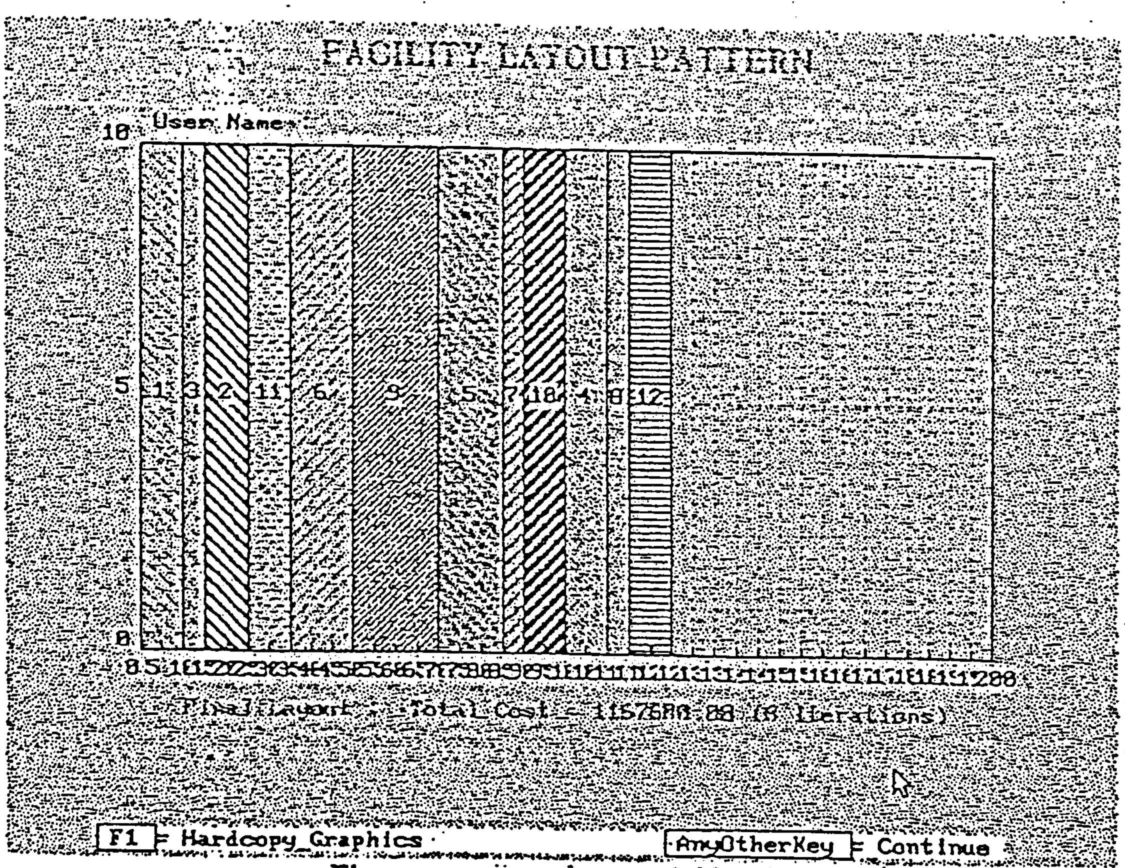
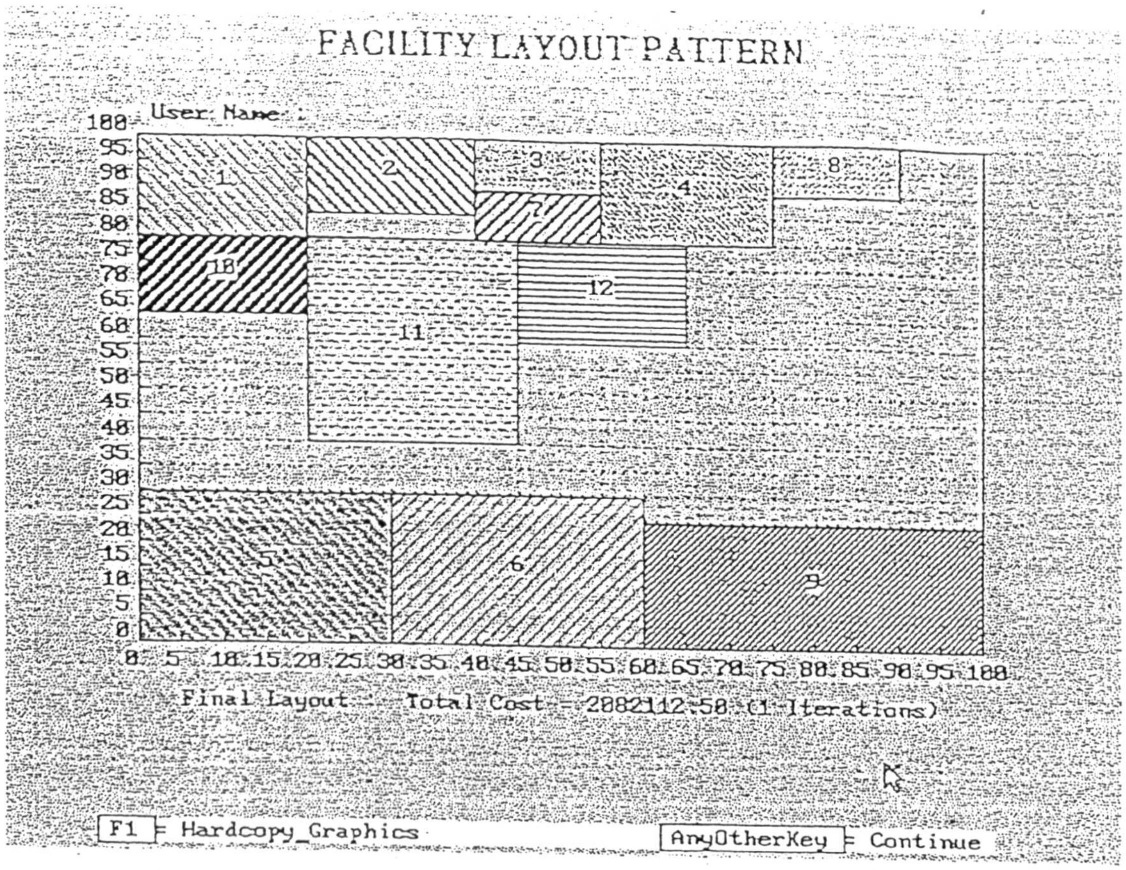


Figure 5.34 Best Solutions identified using CRAFT



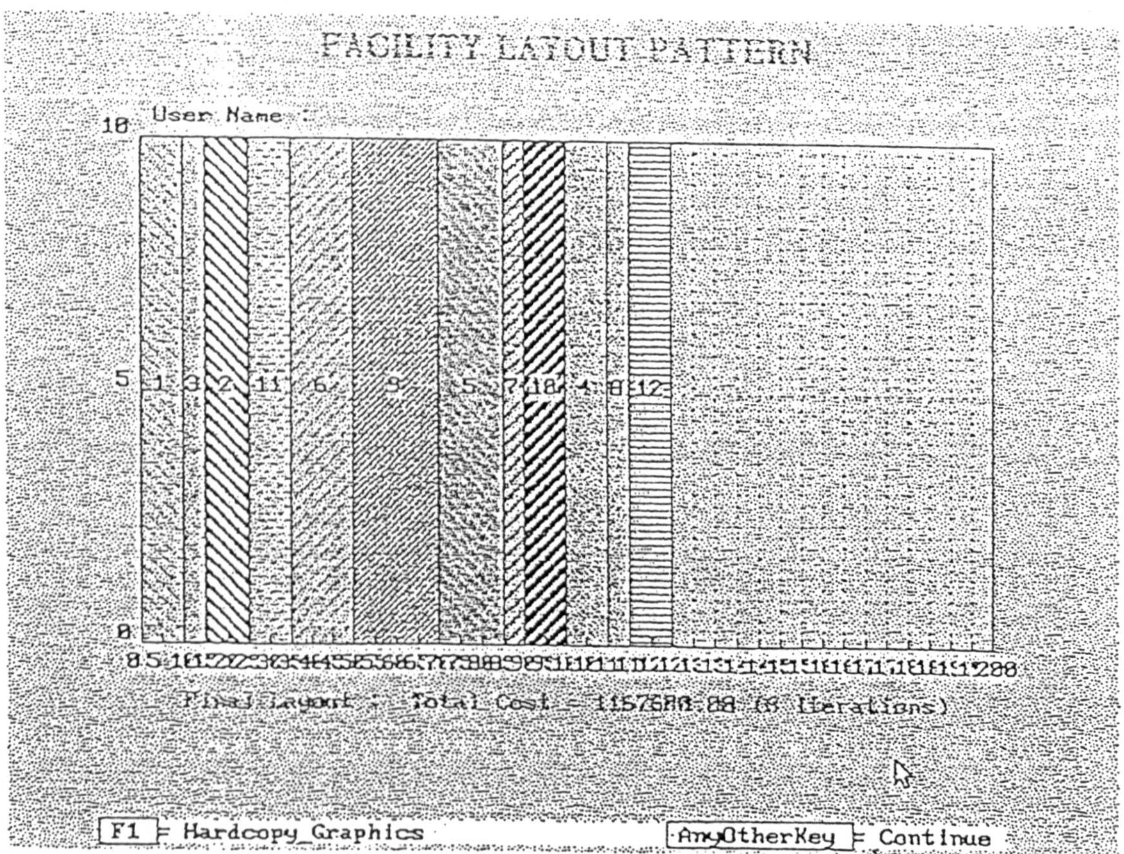
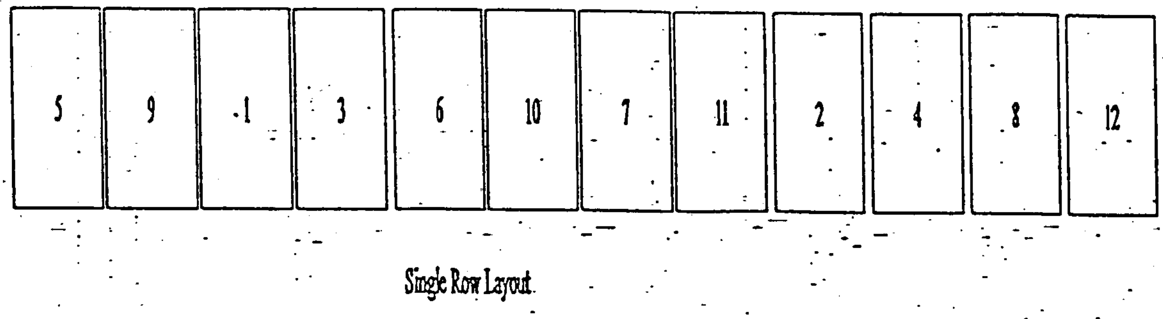


Figure 5.34 Best Solutions identified using CRAFT



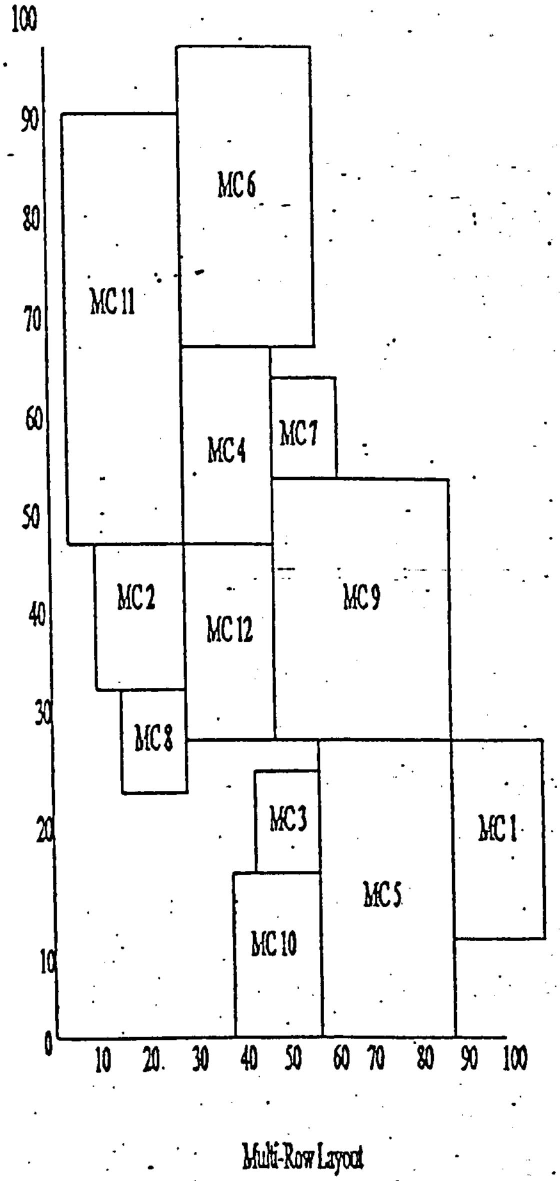


Figure 5.35 Best Solutions identified by the GA

6. Discussion

6.1 Assortment Problem

Existing solution methods for solving the assortment problem have been examined in Section 2.2. These methods are essentially applications of dynamic programming. Dynamic programming has been shown to be of limited use because of the techniques' inability to include, in the decision making process, qualitative variables of the type shown in Table 2.1. In addition, dynamic programming requires the predetermination of each combination of products that could be manufactured. However, this is a decision variable, the value of which can greatly influence the optimality of solutions.

The Minaddition technique has also been used to solve the assortment problem. However, although simple to implement the technique did not identify a lower cost solution than the GA derived solution illustrated in Table 5.5.

The assortment problem was coded into a format suitable for a Genetic Algorithm solution string using a single binary digit to represent each product specification considered as a standard. The decision the GA then had to be make was simply to determine whether or not to include a particular standard in the product range. A binary '1' was used to indicate that a particular standard should be included in the product range and a binary '0' used to exclude it from the range. Using this approach, conformed to the principle of coding, section 4.6.1.1, by maintaining short solution strings lengths and, hence, reducing the search space, i.e. in this case to 262144 search points. The number of digits within the solution string is then determined by the number of alternative standard sizes that are to be considered for inclusion within the product range.

The resulting solution string for the electric hoist problem required 18 digits. This number of digits is relatively small when compared with other problem areas that Genetic Algorithms have been successfully applied to. For example, the MRP problem discussed in Sections 3.3 and 5.3, required 1080 digits to represent the problem. It is not, therefore, expected that this coding method will result in problems when extending the GA solution technique to solving the assortment problem for other product types in which larger numbers of alternative modules or standards need to be considered, e.g. automobile manufacture.

Decisions that need to be made prior to coding are:

a. what will be the largest standard size the company will offer, i.e. this is often determined by manufacturing constraints such as the maximum length that can be machined using existing equipment, and

b. how many standard sizes will there be to choose from, i.e. this could be derived from market research by identifying the standard sizes offered by competitors and/or used by customers.

In order to determine the applicability of individual GA operators for solving the assortment problem a range of experiments was performed as follows:

1. The alternative selection operators investigated were roulette wheel, roulette wheel with elitism, tournament and truncated. The efficiency of each selection operator was identified using experiments in which other operator types and their values remained constant, i.e.:

a. number of replications set at 10,

b. single point crossover used with a probability value set at 0.6,

c. mutation probability rate set at 0.0001, and

d. population size set at 100.

The results are shown in Figure 5.3 and indicate that all types of selection operators investigated were able to find the best solution. However, the efficiency in terms of the number of generations required varies between selection operator types, i.e. both the Tournament and Truncated selection operators allowed the GA to find good solutions in less generations then did the Roulette Wheel and the Roulette Wheel with elitism operators. The inclusion of the elitism option improved the performance of the Roulette Wheel operator by retaining in the next generation the best solution found in the current population. This has the effect of reducing the search space, i.e. variety of solutions within the population. By employing the elitism strategy with the roulette wheel selection operator the search space is prematurely reduced still further hence reducing the algorithm's ability to find the best solutions.

2. The crossover operators investigated were single point crossover, two point crossover and uniform crossover. The efficiency of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator employed,

c. mutation probability rate set at 0.0001, and

d. population size set at 100.

From Figure 5.4 it can be seen that the two point crossover and the uniform crossover operators are more efficient than the single point crossover operator. That is, the two point crossover found the best solution in 40 % less generations than required by the single point crossover and uniform crossover found the best solution in 80 % less generations than required by the single point operator.

In terms of the assortment problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. truncation selection operator,

b. mutation probability rate of 0.0001,

c. population size of 100, and

d. two point crossover operator with a crossover probability of 0.6.

The efficiency with which the GA identified good solutions was found to be dependent on the choice of operators used. This choice was not influenced by the problem size in terms of the solution string length. Operator types are only dependent on the structure of the string, i.e. the number of digits required to represent each basic element of the solution and the type of alphabet used. Applying the algorithm to solve the assortment problem for other product types would not, therefore, be expected to require a change in operator types.

The objective function, used to determine the fitness of individual solutions, calculates the total amount of variable manufacturing costs, (i.e. direct materials and labour), that would be incurred. This involved using market research to forecast the expected demand for each possible standard size and assuming that if any particular standard size was not included in the product range then its forecast demand would be met by providing the next larger size in the range.

For each individual standard size included within the solution, the objective function also contains a fixed cost. These fixed costs were included to represent the effects of increasing the number of standard sizes, within the product range, on the types of qualitative variables listed in Table 2.1. Essentially increasing the number of standards manufactured will increase the complexity involved in planning and controlling the manufacturing facilities and hence result in decreased efficiency and productivity and, therefore, increased fixed costs. The overall costs of these effects are difficult to estimate in cost terms. However, in the case of the electric hoist, management estimated that these costs would lie between £50,000 and £100,000 for each additional standard size. Hence experiments were performed to determine the effect of direct costs on the number and types of standards that would be included in the product range. The results of these experiments, shown in Table 5.5, indicated that higher fixed costs increase the costs of manufacture and force the inclusion of fewer standards within the product range.

The objective function uses variable manufacturing costs, forecasts of demand and fixed costs all of which need to be estimated and could, therefore, be of limited accuracy. In order to apply GA's efficiently, the effect of inaccurate data on the optimal solution would need to be investigated for each individual application. However, from the results it can be seen that the factors that have a significant effect are the large forecast demands for a specific product size, i.e. this tends to force the GA into selecting these as standards, and the size of the fixed costs added for each additional standard size, i.e. this determines the number of sizes selected as standards. When attempting to obtain accurate estimates of data or performing sensitivity analyses to determine the likely effects of inaccurate data it would, therefore, be beneficial to concentrate in these two areas.

The objective function can be considered to be 'practical' in terms of its ability to determine optimal solutions since it takes into consideration the main factors affecting the assortment decision. The qualitative factors involved, Table 2.1, are considered by introducing the fixed cost penalty for each additional standard included in the product range. When using fixed costs, the qualitative variables listed in Table 2.1 would need to be examined in order to determine those that have, for a specific organisation, the greatest impact on business efficiency. For example, the effects on the fixed costs of marketing and sales, may need to be estimated, of including a size in the product range that was not considered a standard in the market place, i.e. greater marketing and sales costs may arise through the extra effort required to sell such a product size. Although estimates of the effect on fixed costs would then need to be made it can be seen from the electric hoist example that detailed accuracy may not be required.

6.2 Aggregate Planning

Existing methods of aggregate planning have been examined in Section 3.2. Although the literature has shown that a wide range of techniques have been developed, including simple graphical techniques, linear programming, transportation models, linear decision rules and decision support systems, several researchers have commented on their current lack of use within industry. This is attributed to the following:

a. each method has been observed to be "situation dependent", hence, identifying the most appropriate method to use presents problems to users,

b. none of the existing aggregate planning techniques can identify optimal or near optimal plans for real world problems that involve a range of planning variables,

c. those techniques that can identify optimal plans do so by achieving only cost related objectives, whereas many other non-cost objectives are often sought by managers,

d. within many organisations the cost relationships used by these methods do not adequately represent those that actually exist,

e. the mathematical procedures used by existing methods, that seek optimal solutions, are complex and difficult for manufacturing management to understand, hence management are reluctant to use such techniques, and

f. models require specific types of data items that are difficult to collect and quantify and often, therefore, this information does not exist in a readily used format.

In order to develop a suitable coding method for the construction of GA solution strings several alternative methods were considered, i.e.:

a. the use of binary numbers to represent actual quantities of product to be obtained from a particular source of capacity, e.g. overtime, sub-contracting,

b. the use of binary numbers to represent the number of batches to obtain from a particular capacity source, in this case batches would be of a pre-determined fixed quantity,

c. the use of decimal numbers to represent actual product quantities from a particular source of capacity, and

d. the use of decimal numbers to represent the number of batches from a particular source of capacity, again batches would be of a pre-determined fixed quantity.

Within high volume manufacturing environments a suitable AP problem would need to deal with high levels of sales demand. The use of GA strings to represent actual quantities could, therefore, result in excessively large search spaces. Additional capacity is also normally achieved in discrete amounts through overtime, sub-contracting and by extra staff.

The method chosen, therefore, employed the use of binary numbers to represent the numbers of batches of a pre-determined fixed quantity. This enabled solution strings to be generated of a practical size for the GA software used to perform the experiments. If greater accuracy was required in terms of product quantities then the solutions obtained using the fixed batch sizes could be used to focus the search area. This concept is discussed in greater detail in Sections 5.3 and 6.3.

In order to obtain a suitable aggregate plan the GA solution string had to identify, for product groups, the capacity requirements from different sources of capacity within specific time periods. Using the approach adopted the user would, therefore, need to decide:

a. the number and length of the planning periods,

b. fixed lot sizes, and

c. the maximum and minimum number of batches.

The maximum limit on overtime was set at 90 units produced in lot sizes of 30 and the maximum limit on sub-contracting was 300 units produced in lot sizes of 100. The allowed range for overtime and subcontracting was between 0 to 90 and 0 to 300 respectively. Using binary numbers to represent actual quantities of products would have resulted in 5.53×10^{29} search points. However, by using fixed lot sizes, the size of the search space is reduced to 2.81×10^{14} points.

The aggregate planning problem investigated in this research required 24 decisions (12 decisions on overtime and 12 decisions on subcontracting) to be taken. The string length for the example problem used in the current work was 96 digits (each decision was coded as 4 digits, 24 x 4 = 96 digits) in length. This is similar to the string lengths expected in practice. String lengths may be larger since companies may wish to include other sources of capacity, for example they may wish to differentiate between alternative sub-contractors. However, such string lengths are well within the capabilities of GA procedures as demonstrated by the MRP problem, Sections 5.3 and 6.3, in which the string length was 1080 digits.

The GA methodology is blind to the type of product applied to and the method of coding can be considered applicable to a wide variety of manufacturing environments, i.e. the fixed batch size could be varied depending on actual demand levels. For example, in high volume manufacturing environments fixed batch sizes could be large since additional capacity would be obtained in larger blocks.

The objective function (Section 5.1.2) was developed using the language C and compiled along with the Genesis software to create an executable file. This allowed an objective function that was representative of those used in practice to be developed. The variables within the objective function included the main sources of providing additional capacity, i.e. overtime, subcontracting, back ordering and the use of stockholding.

The literature search also identified that other relevant factors included the costs of hiring, laying off, scrap and storage. In practice, many factors in an objective function would be constrained by specific company policy or constraints, for example the policy of laying off staff may prevent this being a feasible option. In addition, there could be restrictions on the amount of finished stock that can be held in storage due to limitations in storage space or the high costs of holding finished goods. Identifying relevant costs could also be subjective, e.g. those related to the use of new employees since the time taken to train staff to undertake a task will vary.

In practice, therefore, the objective function used within the aggregate planning problem would need to contain many penalty functions to constrain the GA solution search to feasible areas, e.g. restrict overtime and restrict backorders. Each objective function would need to be developed for individual companies because of the variety of factors that exist and the variation in importance of individual factors between companies.

Although limited, the objective function is of a practical nature since it contains the main methods of varying capacity and includes penalty functions to prevent excess use of specific capacity management methods. In addition, the forecast data (Table 5.10) used to test the GA has been carefully selected to include all the major types of demand changes.

In order to investigate the effects of GA operators and their individual parameter settings, on the ability of GA's to identify optimal solutions, a range of experiments, (i.e. Sections 5.2.3, 5.2.4 and 5.2.5.) were conducted as follows:

1. To compare the efficiency of the roulette wheel and the roulette wheel with elitism selection operators experiments were undertaken with the following parameters held constant, i.e.

a. number of replications set at 10,

b. two point crossover with a probability value set at 0.75,

c. mutation probability rate set at 0.001,

d. population size set at 100, and

e. Sigma scaling with a sigma value set at 2.

The results are provided in Figure 5.6 and show that the roulette wheel with elitism selection strategy was more efficient in terms of the number of generations required to identify best solutions, i.e. identify least cost solutions, than the basic roulette wheel selection. This is contrary to that found in the assortment problem, Section 5.1 and 6.1, where the use of the elitism strategy prevented the GA from finding better solutions. The use of the elitism strategy on this problem allows better solutions to be found as the search space is considerably larger when compared to that in the assortment problem, 2.81×10^{14} search points compared to 262144 search points.

2. Using the two point crossover method the effect of crossover rate was examined using rates of 0.6, 0.65, 0.7 and 0.75. It is considered acceptable to test the effect of the crossover rate in increments of 0.05, i.e. due to the stochastic nature of the crossover operation smaller increments would have relatively insignificant effects. This interval of crossover rates has been widely reported to give the best results for the GA. The efficiency

of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator with elitism,

c. mutation probability rate set at 0.001,

d. population size set at 100, and

e. sigma scaling set at a value of 2.

From Figure 5.7 it can be seen that a crossover rate of 0.65 found lower cost solutions to the AP problem. Although, increasing the crossover rate allows new search spaces to be investigated, when applied to a relatively small search space increasing the rate may reduce the GA ability to find good solutions if no form of insurance, is included such as mutation. For the MRP problem, a crossover rate of 0.75 was found to achieve the lowest cost results. There is, therefore, a relationship between problem size and the ideal crossover rate to employ, i.e. as the problem size increases there is a need to increase the crossover rate in order to ensure that the GA techniques can search new spaces to identify good solutions.

3. The effect of the mutation rates, 0.0001, 0.001 and 0.005 were investigated using the following parameter settings, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator with elitism,

c. crossover probability rate of 0.60,

d. population size set at 100, and

e. Sigma scaling set at a value of 2.

Figure 5.8 illustrates how each of the mutation rates affected the performance of the Genetic Algorithm, i.e. it can be seen that the effect of the mutation rate was limited although. For this problem type a mutation rate of 0.0005 achieved the best results. Higher mutation rates degraded the GA procedure to that of a random search whereas lower mutation rates resulted in lost information through the crossover operator.

When minimising an objective function using a GA, it is common to define the performance variable u(x) of a string x as u(x) = F - f(x), where F is a large baseline function value. By setting F to the maximum value that f(x) can take in the search space, fmax, negative values of u(x) can be avoided or zeroed. Frequently it is impossible to determine the maximum value that fmax can take and in such cases it is normal for F to be set to the maximum value of any string evaluated so far. Using either choice of F, makes good values of x hard to distinguish. For example, if fmax = 100, after a number of generations the current population might only contain strings x for which $5 < x < 10$. At this point no structure in the population has a performance which deviates much from the average. This situation reduces the selective pressure towards better structures and the search stagnates. In this case it is better to update the baseline F as the algorithm progresses. The difficulty arises in how to automatically update this baseline value. If this is left unattended, extraordinary individuals would take over a significant proportion of the population in a single generation. This situation is undesirable as it would lead to premature convergence with the algorithm unable to move outside of a local minima. Later in the GA process, there may be a significant diversity within the population but the average value of the population may be close to the value of the best string in the population. This allows strings of average fitness values and strings with the best fitness values to have the same probability of being selected into the next generation. Hence, the algorithm search for better solutions becomes a random search. Scaling the fitness values can help prevent this situation occurring.

The Genesis software provides two types of scaling functions that can be used in conjunction with the GA, i.e. sigma scaling and window scaling.

Window scaling allows the user to control how aggressively the baseline is updated via the scaling window W. If W > 0, the system sets F to the greatest value of f(x) which has occurred in the last generations. A value of W = 0 indicates an infinite window, i.e. F = f(xmax).

Sigma scaling experiments were conducted using values of sigma equal to 1, 2, 3, 4 and 5. Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. In both sets of experiments, the following parameters were set as constants i.e.

a. roulette wheel selection with elitism,

b. two point crossover with a probability set at 0.60.

c. mutation with a probability set at 0.001, and

d. population size of 100.

Figures 5.10 and 5.11 show clearly that as the values of sigma and window size are lowered, better solutions can be found. These relationships also agree with those existing for the MRP problem, Section 6.3.

In terms of the aggregate planning problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. roulette wheel selection operator with elitism,

b. mutation rate with probability of 0.0005,

c. population size of 100,

d. two point crossover operator with a probability of 0.65, and

e. Sigma scaling with a value of 1.

The operators listed above have been tested on and are, therefore, suitable for use on problems of a practical size.

A prime benefit of using GA's are that they are 'blind' to the type of product under examination hence they are applicable to a wide range of product types.

With low volume/high variety manufacturing environments the sales demand could be expected to fluctuate in an unpredictable manner. However, the sales forecast data has indicated that the GA procedure can cope with such changes.

The GA method when compared with the existing AP methods has been shown to provide the following benefits, i.e.

1. The method is not "situation dependent", hence, overcoming the need to identify the most appropriate method to use under specific conditions.

2. Using the example data a low cost solution has been identified. The optimality of this solution is not known for certain but has improved on an extensive search using manual procedures. In addition, the ability of GA's to identify optimal or near optimal solutions is well proven on much larger problem areas. Hence, it can be assumed that the use of GA's can overcome the limitations of existing aggregate planning techniques by identifying optimal or near optimal plans for real world aggregate planning problems that involve a greater range of planning variables.

3. Non-cost objectives have been included in the analysis, e.g. the inclusion of constraints to limit the maximum amount of capacity that can be gained from specific sources.

4. GA procedures are not complex and, therefore, not difficult for manufacturing management to understand.

6.3 MRP

In Section 3.3 the variety of attempts made to develop effective MRP lot sizing techniques have been examined. Approaches to the development of suitable techniques currently include:

a. the use of dynamic programming,
b. simple decision rules,
c. complex algorithms that attempt to develop optimum planned order release schedules,
d. techniques based on Economic Order Quantity models,
e. techniques based on Periodic Order Quantity models, and
f. single pass and adjusted single pass rule based methods.

Although there are a variety of techniques available, it is widely recognised that each technique will only yield acceptable results under a limited range of demand and inventory cost conditions. In this respect the effectiveness of each individual technique is strongly dependent on such factors as the variability in the sizes of individual material requirements, the variability in the frequency of requirements and the relative values of inventory holding and purchasing costs. Choosing which technique to use from the wide variety available has, therefore, been identified as a major problem.

In addition, other limitations to the use of existing lot sizing methods have been identified from the literature as:

a. limitations on the length of the MRP planning horizon over which optimal order schedules can be found, i.e. usefulness in practical situations is questionable since large numbers of alternative schedules would need to be considered and, in addition, optimal short term schedules would not necessarily result in optimisation of inventory over the long term,

b. limited use in manufacturing industry due to the complexity of the procedures required to generate optimal or near optimal schedules, i.e. these have often been found to be difficult for operating personnel within manufacturing organisations to understand,

c. existing methods treat the lot sizing problem as a single stage process, however, MRP is a multi-stage process and, hence, any lot sizing techniques must consider all items whose demand is related, both horizontally and vertically, throughout the BOM structures,

d. lack of sufficient resources to continuously monitor the effective selection and use of appropriate methods, i.e. in order to achieve good results it is necessary to select with care the most appropriate lot sizing method for each item/demand period within the MRP process and to re-select lot sizing methods when product structures or demand patterns change,

e. all existing methods use costs to measure how effective a specific lot sizing policy is with many methods placing restrictions on the types of costs that are considered, i.e. it is uncommon for non-cost variables or constraints such as the availabilities of working capital or warehouse space to be considered, and

f. lack of understanding when changing lot sizing techniques within existing MRP software can often lead to disastrous consequences such as excessive stockholding costs being incurred.

In order to generate a suitable GA coding structure for the MRP lot sizing problem it was necessary to examine a typical solution, i.e. a planned order release schedule. The GA coding structure adopted needed to enable decisions concerning planned order sizes to be made for each planning period/component combination, hence the variables present in such a schedule type were identified as:

a. the size of the individual planning periods, i.e. normally these 'time buckets' represent either weekly or monthly production periods,

b. the number of individual planning periods within the planning horizon,

c. the number of products manufactured by the company, and

d. the number of components and assemblies within the BOM structure of each product.

As with the AP coding problem, Section 5.2, several alternative methods were considered, i.e.:

a. the use of binary numbers to represent actual order quantities,
b. the use of binary numbers to represent the number of batches to order, i.e. batches would be of a pre-determined fixed quantity,
c. the use of decimal numbers to represent actual order quantities, and
d. the use of decimal numbers to represent the number of batches to order, i.e. again batches would be of a pre-determined fixed quantity.

When choosing which coding method to use, the main constraint that had to be considered was the length, in terms of the number of digits, of each 'planned order quantity' section of the GA code. Here it was considered essential that the size of the complete solution string for a MRP problem of a practical size needed to be minimised since the potential existed for strings to be generated that were many thousands of binary digits in length. For example, in high variety/low volume manufacturing environments, large product ranges could exist each of which could contain many hundreds of individual assemblies and components. The need to manipulate large solution strings could have serious consequences in terms of the ability of the GA to search a sufficient amount of the total solution space such that an optimal or near optimal solution could be generated.

In order to reduce the size of solution strings the coding method was adopted of allocating batch sizes to each component and using the solution string to code only the number of batches required for a particular planning period/component order size. It could be argued that restricting order quantities to specific batch size intervals could affect the ability of the GA to identify the optimal solution. However, the use of fixed batch sizes in this manner is not felt to greatly hinder the effectiveness of the GA procedure since in many instances companies do in practice buy in fixed lot sizes. In addition, the possibility exists of using this procedure as a form of 'coarse sieving' to provide a pool of initial solutions on which a more detailed search can be carried out by reducing the batch size intervals. The effectiveness of using an initial pool of solutions in this manner, known as 'seeding', has been well documented in the literature by Davis(1991).

The size of the example MRP problem used to determine the effectiveness of the GA procedure was limited by the capabilities of the software available for carrying out the GA search procedures. However, it can be seen by comparing the results obtained from the MRP and Assortment example problems that there is a relationship between size of problem, the population size and number of generations required to efficiently reach the lowest cost solution. The assortment problem had a string size of 18 digits and found the lowest cost solution using a population size of 100 in 20 generations, whereas the MRP problem possessed a string size of 1080 and found the lowest cost solution using a larger population size of 1000 in 20 generations.

In practice, string lengths could be many tens of thousands of digits in length, e.g. assume a company uses 1000 different component types, requires the MRP system to plan 10 periods in advance and 5 digits are required to represent each batch size the resulting string length would be 1000 x 10 x 5 = 50,000 digits. In order to reduce this number, the types of components included in the GA planning process could be restricted to 'A' and 'B' class items in the Pareto distribution. This procedure is accepted MRP practice in many companies.

In practice, MRP runs are normally not performed more frequently then on a weekly basis with all computing being carried out overnight. The computing time required to achieve an efficient solution may not, therefore, represent a problem area. Should it do so then the current development of parallel GA computing facilities (Lewis and El-Rewini (1992)) would help in this direction.

An objective function that was representative of those used in practice was developed using the language C and compiled along with the Genesis software to create an executable file. The variables within the objective function included the costs of holding inventory and the costs associated with generating and processing purchase orders. It can be seen from the literature search, Section 3.3, that many existing lot sizing models have been developed to include other relevant factors such as scrap and storage costs. If the range of application areas examined by the current research is examined it can be seen that prime costs, subcontracting, overtime, holding and order costs have all been successfully used within objective functions. Hence, there appears no reason why such costs as scrap and storage could not also be included since in manufacturing these are frequently determined as a direct percentage of holding or ordering costs. Of interest to manufacturing organisations would be the inclusion of constraints such as the amount of 'capital available for materials purchasing'. Here the objective function would need to simply calculate the total purchasing costs of each solution and place appropriate penalty functions on those solutions that exceeded the available capital. In a similar manner, other constraints identified as potentially important, such as the 'amount of storage space available in existing material stores', could be introduced into the objective function.

The objective function developed in the current research also includes penalty functions to prevent the GA seeking the lowest cost solution by not placing any orders at all, i.e. maintaining zero stock levels.

The factors that determine the relative effect on the optimum solution of each variable are:

a. relative size of inventory holding costs,

b. relative size of procurement costs, and

c. relative size of batch sizes.

The effects on the efficiency of a GA of the following operators were investigated, i.e.

1. Selection operators

    roulette wheel selection,

    roulette wheel selection with elitism,

2. Scaling operators

    window scaling

    sigma scaling

3. Mutation rate

4. Crossover operator

    two point crossover.

In order to determine the applicability of GA operators for identifying planned order release schedules a series of experiments were performed as follows:

1. To identify the use of the roulette wheel selection and the roulette wheel selection with elitism, experiments were undertaken with the following parameters held constant, i.e.:

a. number of replications set at 10,

b. two point crossover with a probability value of 0.75,

c. mutation probability rate set at 0.00001, and

d. population size set at 1000.

The results shown in Figure 5.15 indicate that the roulette wheel with the elitism strategy identified better solutions than the basic roulette wheel. The use of the elitism strategy on this problem type allowed improved solutions to be found more efficiently since it quickly reduced the search space from the original $(30^{19})^{12}$ search points.

2. As with the AP problem (Section 6.2), the crossover operator rate was investigated using two point crossover and crossover rates of 0.6, 0.65, 0.7 and 0.75. The efficiency of crossover rate was identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. mutation probability rate set at 0.00001, and

d. population size set at 1000.

From Figure 5.16 it can be seen that the higher crossover rates gave the better performances. Increasing the crossover rate allows new search spaces to be investigated although increasing this rate in problems which have a relatively small search space may reduce the GA ability to find good solutions if no form of insurance is included, e.g. mutation.

3. The effect of the mutation rates, 0.000001, 0.000005 and 0.00001 were investigated using the following settings, i.e.:

a. number of replications set at 10,
b. roulette wheel selection operator without elitism,
c. two point crossover with a probability rate of 0.75, and
d. population size of 1000.

Figure 5.17 illustrates how each of the mutation rates affected the performance of the algorithm in terms of identifying the best solutions. It can be seen that the effect of the mutation rate was limited although in this problem a mutation rate of 0.00005 achieved the best results in terms of identifying the lowest cost solution.

Experiments using both sigma and window scaling were carried out to determine their relative performance with respect to MRP problem types.

Sigma scaling experiments were conducted with values of sigma equal to 1, 2, 3, 4 and 5 and Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. In both sets of experiments the following parameters were held constant, i.e.

a. roulette wheel selection without elitism,
b. two point crossover with a probability of 0.75,
c. mutation with a probability of 0.000001, and
d. population size of 1000.

Figures 5.18 and 5.19 indicate that as the value of sigma and window size decrease, improved solutions can be found. This relationship agrees with the results identified in Section 6.2 and 6.4 for the aggregate planning problem.

In terms of the MRP lot sizing problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. roulette wheel selection operator,

b. mutation rate of probability of 0.00005, and

c. population size of 1000.

d. two point crossover operator with a probability of 0.75.

e. Sigma scaling with a value of 1.

The GA method when compared with the existing MRP lot sizing methods has been shown to provide the following benefits, i.e.:

1. The objectives of the MRP problem are similar to those of Aggregate Planning (AP), i.e. both techniques involve identifying item quantities for specific planning periods generated by various capacity management methods. As such the advantages of AP can be directly related to the MRP situation. In this respect, the variety of demand types used to test the GA based aggregate planning methodology proved no problem to the identification of low cost solutions. Hence it can be expected that a GA based MRP procedure would not be "situation dependent", i.e. GA procedures can be expected to yield acceptable results under a wide range of inventory demand and cost conditions. Major disadvantages with existing methods would, therefore, not arise, i.e. the current problems of choosing which is the most appropriate method to use, monitoring the efficiency of the methods used to determine when the lot sizing techniques should be changed, inability to chose the most appropriate method for each individual component within the BOM product structures and the need to re-examine the methods used when product structures or demand patterns change.

2. The use of GA procedures can be expected to overcome the current limitations on the length of the MRP planning horizon over which optimal order schedules can be found. In this respect, although overcoming these planning horizon limitations has not been established, in the current research, using the relatively small example problem, the literature has identified the successful use of GA's in large scale problems such as optimisation of manufacturing system design and scheduling. It can be surmised, therefore, that the potential size of the MRP problem will not be detrimental to the use of GA's.

3. GA procedures, when compared with existing MRP lot sizing methods, are relatively simple to understand and use. Hence the limited use in manufacturing industry due to the complexity of the existing procedures could be overcome.

4. Existing methods treat the lot sizing problem as a single stage process although in practice MRP is a multi-stage process. Hence, the possibility exists of establishing local optima for particular component lot sizes which result in a non-optimal solution for the overall MRP process. The GA based MRP procedure developed in the current research simultaneously considers lot sizes for all components at all levels in the BOM structure and for all periods in the MRP planning horizon, hence the procedure is a true multi-stage, multi-period lot sizing technique that has the facility to search for global optimums.

In practice, the GA based MRP procedure would need to be linked to MRP functions that check to ensure that the planned orders for a lower level item in the BOM structure meets the net material requirements for its parent item further up the structure. In this way, the procedure would consider all items whose demand is related, both horizontally and vertically, to each other via the BOM structure.

5. The example problem used to demonstrate the use of GA procedures in identifying optimum aggregate plans has shown the use of constraints in limiting specific aspects of the

solution, i.e. constraints were placed on the maximum number of units that could be obtained using overtime and subcontracting. This is achieved simply by adding a penalty function to the final value of the objective function which drastically reduces the chances of that solution being represented in the next generation.

This facility can be extended to the MRP objective function which can be modified to contain a wide variety of constraints including the availabilities of working capital and warehouse space. Hence the limitations, in terms of the type and variety of factors that can be considered, with current lot sizing methods could no longer apply.

6. Lack of understanding when changing lot sizing techniques within existing MRP software has often lead to disastrous consequences such as excessive stockholding costs being incurred. Using a GA technique there will no longer be the need to change methods, hence removing this problem.

6.4 Line Balancing

Existing solution methods for the line balancing problem have been examined in Section 2.3.2.1. Linear programming, a solution method proposed by Bowman(1962) has been found to be limited in the size of problems that can be solved. Although, heuristic methods (Kilbridge and Wester (1961), Arcus (1966) and Helgerson and Birnie (1961)) have been developed to overcome this size problem, these methods according to Wild (1989) yield non-optimum solutions.

The Genetic Algorithm (GA) has been applied to the line balancing problem, Section 5.4, and the results compared to those achieved using the Rank Positional Weight method with both methods finding the best solutions.

In order to develop a suitable GA code, the following method was adopted, i.e.

a. each work task was represented in the string,

b. each element represented each task to be placed,

c. each workstation was represented as a binary number, and

d. the value of each element represented the workstation that the task would be assigned to.

Using this coding approach allowed, the length of the string to be minimised thus reducing the search space, and a coding to be used that is a direct representation of the problem.

In order to determine the applicability of GA operators for identifying work station layouts a series of experiments were performed as follows:

1. To compare the relative efficiencies of the 'roulette wheel selection' and the 'roulette wheel selection with elitism' the following parameters held constant,

a. number of replications set at 10,

b. two point crossover with a probability value of 0.60,

c. mutation probability rate of 0.005,

d. population size of 100, and

e. Sigma scaling set at a value of 2

The results in Figure 5.22 show that the 'roulette wheel' selection operator performed more efficiently without the use of the elitist option. This is in contrast to the results found in the AP problem (Section 6.2) and the MRP problem (Sections 6.3). where the removal of the elitism option prevented the GA from finding improved solutions. A reason for this is, by employing the elitism strategy on small problems search spaces may be prematurely discarded as discussed in section 6.1

2. In order to identify the optimum crossover rate to employ, experiments were carried out using rates of 0.6, 0.65, 0.7 and 0.75. During the experiments other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. mutation probability rate set at 0.005,

d. population size set at 100, and

e. Sigma scaling with a sigma value set at 2

From Figure 5.23 it can be seen that as crossover rates decreased the number of generations required by the GA to converge also decreased. However, as crossover rate increased the ability of the GA to find the best solution decreased. In this situation, increasing the crossover rate allows new search spaces to be investigated. However, increasing the crossover rate for problems that have a relatively small search space could reduce the GA's ability to find good solutions if no form of insurance is included, e.g. mutation. This is the reason why the GA with the higher crossover rates took longer to find good solutions, i.e., any good solutions found were selected for crossover and that information in the string was lost in the generation.

3. The effect of the mutation rates, 0.005, 0.0022 and 0.0005 were investigated using the following settings:

a. number of replications set at 10,

b. roulette wheel selection operator without elitism,

c. two point crossover with probability rate set at 0.60,

d. population size set at 100, and

e. Sigma scaling with a sigma value set at 2.

From Figure 5.24, it can be seen that variations in the mutation rate had little effect on the ability of the GA to identify good solutions. Mutation rate however, did effect the performance of GA in the early generations with a mutation rate of 0.005 achieving better results in initial generations. Using a higher mutation rate reduced the possibility that good strings were lost through the crossover operator. In this instance, the higher mutation rate has reduced the number of good strings being lost in the early number of generations. Although the experiments using lower mutation rates eventually found the good solutions they did so due to the ability of the crossover operator to eventually generate good solutions.

Sigma scaling experiments were conducted with values of sigma equal to 1, 2, and 3 and Window scaling experiments were conducted with values of window sizes equal to 1, 5 and 10. For both sets of experiments the following parameters were set constant, i.e.

a. roulette wheel selection without elitism,

b. two point crossover with a probability set at 0.60,

c. mutation with a probability rate set at 0.005, and

d. population size set at 100.

Figures 5.25 and 5.26 show clearly that as the value of sigma and window size decrease the ability of the GA to find better solutions increases. This is in agreement with the results found during experiments on the aggregate planning problem (Sections 5.2 and 6.2) and the MRP problem (Sections 5.3 and 6.3).

In terms of the line balancing problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. roulette wheel selection operator,

b. mutation rate of probability of 0.005,

c. population size of 100,

d. two point crossover operator with a probability of 0.60, and

e. Sigma scaling with a value of 1.

The coding approach adopted did not prevent infeasible solutions from being created. Hence to overcome this problem a penalty function was included within the objective function that:

a. prevented any precedence constraints being violated, and

b. prevented the accumulated time in any workstation exceeding the cycle time.

The resulting objective function shown in Equation 5.2 initially calculates a value for the 'balancing loss' and then adds a value for the penalty function depending on the type of constraint violated. In this respect, a penalty value of 100 was assigned to each broken precedence constraint and a penalty value of 10 assigned to each workstation that exceeded the required cycle time.

The GA method when compared with existing line balancing solutions methods i.e. the Rank Positional Weighting method has been shown in this problem to match the results obtained.

6.5 Facilities Layout

Section 2.3 critically examines methods for solving the facilities layout problem. Kusiak and Heragu (1987) have classified these methods as either construction methods, improvement methods, hybrid methods or graph theoretic methods. Finke at al (1985) stated that using existing methods optimal solutions cannot be found for facilities layout problems greater than 15 machines x 15 machines.

In order to code the problem into a suitable format for a GA, a number of alternative methods were considered, i.e.:

1. The use of string positions to represent the relative positions of the machines on the shopfloor, e.g. if machine 2 lay between machines 5 and 7 in the GA solution string then this machine would be positioned between machines 5 and 7 on the shopfloor.

2. Binary code to represent the actual positional co-ordinates of the centre of each machine on the shopfloor.

The decision was taken to use method 2 since this method:

a. represented the problem in a more realistic manner,

b. allowed constraints on the positions of the machines on the shopfloor to be incorporated more accurately,

c. allowed material handling costs to be estimated in terms of the unit/distance travelled, and

d. allowed minimum clearance distance constraints between machines to be incorporated easily and realistically.

The following method was therefore adopted, i.e.:

a. all machines were represented in the string,

b. a specific section of the string represented a machine to be placed on the shopfloor,

c. for single row layout problems each section of the solution string represented the x positional co-ordinate of that machine on the shopfloor, and

d. for multi row layout problems each section of the solution string represented both the x positional co-ordinate and the y positional co-ordinate of that machine on the shopfloor.

A GA solution procedure was developed using the ExpertRule software which enabled the use of the following operators to be investigated, i.e.

1. Selection operators
   - Roulette Wheel with Elitism
   - Truncated
   - Tournament
2. Crossover operators
   - Single Point
   - Two Point
   - Uniform

In order to identify the ability and the suitability of each of the above operators for solving the facilities layout problem, a number of experiments were carried out on both single row and multi row layout problems.

1. The alternative selection operators examined were roulette wheel, roulette wheel with elitism, tournament and truncated. The efficiency of each selection operator was identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. single point crossover with a probability value set at 0.6,

c. mutation rate with a probability set at 0.0001, and

d. population size set at 100.

The results shown in Figures 5.30 and 5.31 indicate that the Tournament and Truncated selection operators investigated were both able to find lower cost solutions than did the Roulette Wheel with Elitism selection operator.

2. The crossover operators investigated were single point crossover, two point crossover and uniform crossover. The efficiency of each crossover operator was again identified using experiments in which other operator types and values remained constant, i.e.:

a. number of replications set at 10,

b. roulette wheel selection operator,

c. mutation rate with a probability rate set at 0.0001, and

d. population size set at 100.

From Figures 5.32 and 5.33, it can be clearly seen that the two point crossover out performs the single point crossover and the uniform crossover in identifying lower cost solutions. In the assortment problem investigated, Sections 5.1 and 6.1, the problem was relatively small in relation to the problems investigated during this research. In the assortment, the effect of the different crossover operators was limited as can be seen in Figures 5.4 as all operators eventually identify the best solution although the two point crossover operator identified the lowest cost solution in the least number of generations. When the problem size increases the effect of crossover operators become more apparent as can be seen in Figures 5.32 and 5.33. This findings agree with the findings of other researchers (Davis (1985)) that the two point crossover operator out performs the single point crossover operator.

In terms of both the single row and multi row layout problem, the experiments determined that the most suitable types and values for GA operators are as follows:

a. tournament selection operator,

b. mutation rate of probability of 0.001,

c. population size of 100, and

d. two point crossover operator with a crossover probability of 0.6.

The GA has been shown to perform well when compared with the traditional methods of designing plant layouts using the CRAFT method. For the single row problem, the best solution found using the CRAFT method resulted in a handling cost of £ 1,167,600 whereas the average lowest cost solution using GA resulted in a cost of £ 802,428, an improvement of 31 %. For the multi row problem, the best solution found using the CRAFT method was £ 2,082,112 compared to the average value of the solutions by the GA of £ 1,387,239 an improvement of 33 %. The resulting layouts are shown in Figures 5.34 and 5.35.

The benefits of using GA's to solve the layout problem are:

a. the solution is given by the actual position on the shopfloor.

b. the objective function can be developed to meet the individual needs of each problem and organisation.

c. the GA is capable of solving large problems which could not be solved using conventional methods.

6.6 Guidelines for Parameter Value Selection

The results obtained from the range of GA experiments carried out in Chapter 5 have been examined in order to determine the relationships that exist between 'solution string size' and the 'most suitable operator parameter values' to adopt. Here 'solution string size' is determined by the size of the problem to be solved.

The results obtained from Chapter 5 have been classified into the basic categories shown in Table 6.1. Within this table, problem areas have been sequenced according to problem size.

Problem	**Size**	**Crossover Rate**	**Mutation Rate**	**Elitism**
Assortment	Small	Low	Low	No
Line Balancing	Small	Low	Low	No
Aggregate Planning	Medium	Medium	Medium	Yes
MRP Lot Sizing	Large	High	High	Yes
Machine Layout	Large	High	High	Yes

Figure 6.1 Summary of Findings

Since the objectives of this analysis are to develop only general guidelines for parameter value selection, the use of general classifications rather then specific parameter values has been adopted. Only general guidelines are needed since the optimality with which solutions can be generated is dependent on the accuracy of data input into the GA. In terms of manufacturing design problems, much of the input data needs to be estimated or is of a qualitative nature and, hence, is of limited accuracy. Hence if more precise selection rules for parameter value selection were identified the accuracy with which optimal solutions can be found would still be limited by the accuracy of input data. In addition, GA's tend to be robust in terms of the range of parameter values over which acceptable solutions can be generated, i.e. they are not parameter value sensitive.

From Table 6.1 it can be seen that as the problem size increases, i.e. in terms of the size of individual solution strings, then in order to find improved solutions:

1. Crossover rates should be increased, i.e. this increases the number of new search spaces examined by the GA and hence increases the probability of identifying optimal or near optimal solutions.

2. Mutation rates should also be increased. Here the effect of high mutation rates balance the negative effects of a high crossover rate, i.e. a high mutation operator prevents the loss of good information from the population through the generation of large numbers of new solutions that occurs when using a high crossover rate.

3. Employing the elitism strategy during selection appears beneficial only for large to medium solution string sizes, i.e. with large to medium string sizes elitism enables good solutions to be maintained within future populations. With small string sizes, elitism can cause premature convergence by preventing a sufficient variety of solutions to be maintained in the population.

7. Conclusions

Genetic Algorithm solution procedures have been successfully applied to a variety of decision areas involved in the design of manufacturing systems, i.e. these are:

1. Determining the number and type of standards that are included in a product range, i.e. the Assortment problem. For a problem of a practical size, GA based procedures identified a lower cost solution than that found by the alternative Minaddition technique. (Section 5.1)

2. Determining monthly capacity requirements, i.e. aggregate planning. Here the GA found low cost solutions and successfully coped when a wide variety of sales demand patterns existed i.e. sudden large changes in sales demand, planning periods where demand was greater than normal production capacity and planning periods where normal production capacity was greater than demand. (Section 5.2)

3. Determining MRP lot sizes, i.e. here the ability of GA procedures compared favourably with those of the McLaren's Order Moment technique. It can be concluded that for larger problems, the efficiency of GA procedures would not decrease. GA procedures are also capable of overcoming inherent problems involved in the use of existing MRP lot sizing methods. The GA method also represents a true multi-level, multi-period lot sizing technique as opposed to existing single level techniques. (Section 5.3)

4. Workstation balancing of flow process lines, i.e. here GA procedures successfully generated balanced lines and found many more solutions than those obtained using the Rank Positional Weighting method hence providing management with flexibility in decision making.

(Section 5.4)

5. Solving the facilities layout problem, i.e. the GA solution string developed was able to both minimise material handling costs and identify the actual positions of individual items of processing equipment on the shopfloor. (Section 5.5)

6. For each problem type, the choice of operators, i.e. selection, crossover and mutation, needs careful selection since they have a significant effect on the performance of the algorithm.

(Sections 5.1 to 5.5)

7. The individual operator parameter values have been found to affect the efficiency of the Genetic Algorithm. Guidelines have, therefore, been identified for their selection. (Section 6.6)

8. Recommendations for Further Work

1. For each decision area investigated detailed work now needs to be carried out to establish detailed procedures for the use of GA solution procedures. This is particularly relevant to MRP lot sizing where the need to integrate GA procedures into existing MRP software would be essential.

2. GA's have been shown to provide a common solution technique for a wide range of manufacturing design decision areas. The opportunity now exists to examine the potential for integrating these decisions with a single GA solution code, using the parallel processing and multiple chromosome facilities of the GAME software.

3. The potential needs to be investigated for using hybrid systems, e.g. GA's in conjunction with Tabu Search, in order to make use of the strengths of each technique to offset their individual weaknesses.

9. References

Arcus, AL., 1966, COMSOAL: A COmputer Method for Sequencing Operations for Assembly Lines, International Journal of Production Research, 4(4), 259-277.

Armacost, R. L., Penlesky, R.J. and Ross S.C.,, "Avoiding Problems Inherent in Spreadsheet-Based Simulation Models: An Aggregate Planning Application2, Production and Inventory Management Journal, 1990, 31/2, 62-68.

Armour, G.C. and Buffa, E.S., A heuristic algorithm and simulation approach to relative allocation of facilities, Management Science, 1963, Vol 9, No2, 294-300

Ashworth C.M. (1988), Structured Systems Analysis and Design Method (SSADM), Information and Software Technology, vol 30, no 3.

Attar Software (1994), Xpertrule Manual, Attar Software, Newlands Road, Leigh, Lancashire, U.K.

Aucamp, D.C., "A variable demand lot-sizing procedure and a comparison with various well-known strategies", Production and Inventory Management, 26,2, 1985, pp 1-20.

Axsater, S., "Evaluation of lot sizing techniques", International Journal of Production Research, 24,1, 1986, pp 51-57.

Azoza, M. A. and Bonney, M. C., "Disaggregating Aggregate Production Plans to Smooth Workload Variation", Engineering Costs and Production Economics, 1990, 19/1-3, 125-131.

Bahl, H.C. and Zionts, S., "Lot sizing as a fixed-charge problem", Production and Inventory Management, 27,1, 1986, pp 1-10.

Barman, S., Tersine, R.J. and Burch E.E., "Performance Evaluation of the LDR and the PSH with Forecast Errors", Journal of Operations Management, 1990, 9/4, 481-499.

Barnes, J.W., and Chambers, J.B., 1995, Solving the Job Shop Scheduling Problem with Tabu Search, IIE Transactions, Vol 27, No 2, pp 257-263.

Bellman, R., (1957), Dynamic Programming, Princeton University Press.

Benton, W.C., "Multiple price breaks and alternative purchase lot size procedures in material requirements planning systems", International Journal of Production Research, 23,5, 1985, pp 1025 1047.

Berry, W.L., "Lot sizing procedures for requirements planning systems: A framework for analysis", Production and Inventory Management, 2nd Quarter, 1972, pp 19-34.

Black J.T., Cellular manufacturing systems reduce setup time, make small lot production economical. Industrial Engineering, 33-48, November 1983.

Block, T.E., A new construction algorithm for facilities layout, Journal of Engineering Production, 1978, 2, 11

Bongers C.(1980) Standardization: Mathematical Models in Assortment Determination, Martin Nijhoff, Boston.

Booker, L.B., Intelligent behavior as an adaptation to the task environment. Dissertations Abstracts International, 43(2), 469B, University of Michigan Microfilms No. 8214966, 1982.

Boucher, T.O., Lot sizing in group technology production systems, International Journal of Production Research, 22, 85, 1984.

Bowman E. H., "Production Planning by the Transportation Method of Linear Programming", Journal of Operations Research Society, February 1956, Vol. 4,100 - 103.

Bowman E. H., "Consistency and Optimality in Managerial Decision Making", Management Science, 1963, 9/2 , 310-321.

Brindle, A., Genetic algorithms for function optimisation, University of Alberta, Edmonton, 1981.

Burbidge J.L., Introduction of Group technology, John Wiley, New York, 1975.

Buxey, G., "Production Planning and Scheduling for Seasonal Demand", International Journal of Operations and Production Management, 1993, 13/7, 4-21.

Callarman, T.F. and Hamrin, R.S., "A comparison of dynamic lot sizing rules for use in a single stage MRP system with demand uncertainty", International Journal of Operations and Production Management, 4,2, 1984, pp 39-48.

Chen, h.D., Hearn, D.W. and Lee, C.Y., A new dynamic programming algorithm for the single item capacitated dynamic lot-size model, Journal of Global Optimisation, Vol 4, No 3, pp 285-300, 1994.

Chyr, F., Ming Lin, T., and Ho, C., "An Extension of the EOQ Production Model Based on Damage Costs", Int. Journal of Production Management, 10,5, 1990, pp. 71-76.

Cleveland G.A. and Smith S.F. (1989) Using genetic algorithms to schedule flow shop releases, Proceedings of the Third International Conference on Genetic Algorithms, 160 - 169.

Coleman, B.J. and McKnew, M.A., "An improved heuristic for multilevel lot sizing in material requirements planning", Decision Science Journal, 22,1, 1991, pp 136-156.

Crabtree, I.B., (1995), Resource scheduling-comparing simulated annealing with constraint programming, PACT 95. Practical Applications of Constraint Technology Conference, pp 55-61.

Davis, L, Job Shop scheduling with Genetic Algorithms, Proceedings of the International Conference on Genetic Algorithms, 1985, pp 136-140.

Deisenroth, M.P. and Apple, J.M., A computerised plant layout analysis and evaluation technique, 1972, Annual AIIE Conference, Norcross, Ga.

Doumeingts G, Dumora E., Chabanas M, and Huet J.F., (1987), Use of GRAI for the design of Advanced Manufacturing Systems, IFS Flexible Manufacturinf Systems, 6th International Conference, Turin, Italy, November 4-6 1987, p 341.

Edwards, H.K, Gillett, B.E. and Hale, M.E., Modular allocation technique (MAT), Management Science, 1970, 17 (3), 161-169

Edwards G.A.B., Readings in Group Technology, Machinery Publishing Company, London, 1971.

Evans, J. R., Anderson, D.R., Sweeney, D.J. and Williams T.A., Applied Production and Operations Management, 3rd Ed., West Publishing Co., St Pauls, New York, 1990.

Fagan, M. L., "A Guide to Global Sourcing", The Journal of Business Strategy, March/April, 1991 pp. 21-25.

Feiring, B. R., and Sastri, T., "Improving Production Planning by Utilizing Stochastic Programming", Computers and Industrial Engineering, 1990, 19/1-4, 53-56.

Finch B.J. and Cox, J.F. 1986, An Examination of Just-In-Time Management for the Small Manufacturer: With an Illustration, International Journal of Production Research, 24, 329-342.

Finke G. and Kusiak A., 1985, Network Approach to Modelling of Flexible Manufacturing Modules and Cells, Rairo-Automatique-Productique Informatique Industrielle Automatic Control Production Systems, Vol 19, No4, pp359-370.

Flanders R.E., Design Manufacture and Production Control of a Standard Machine. Transactions of ASME, vol. 46, 1925.

Fogarty Terence C, Varying the probability of mutation in the genetic algorithm, Proceedings of the International Genetic Algorithm Conference 1989

Freeland, J.R. and Colley, J.L., "A simple heuristic method for lot sizing in a time phased reorder system", Production and Inventory Management, 23,1, 1982, pp 14-22.

Gaither, N., "A near-optimal lot-sizing model for material requirements planning systems", Production and Inventory Management, 4th Quarter, 1981, pp 75-89.

Gaither, N., "An improved lot-sizing model for MRP systems", Production and Inventory Management, 3rd Quarter, 1983, pp 10-20.

Gallagher and Knight : Group Technology Production methods for manufacturing. John Wiley, 1986.

Gen, M., Tsujimura, Y. and Ida, Y., "Method for Solving Multiobjective Aggregate Production Planning Problems with Fuzzy Parameters", Computers and Industrial Engineering, 1992, 23/1-4, 117-120.

Gilbert, K. C. and Madan, M. S., "A Heuristic for a Class of Production Planning and Scheduling Problems", I.I.E. Transactions, 1991, 23/3, 282-289.

Gilgeous, V., "Modelling realism in aggregate planning: a goal-search approach", International Journal of Production Research, 1989, 27/7, 1179-1193.

Glover, F., Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research, Vol 13, No. 5, pp 533-549, 1986.

Glover. F., Kelly, J.P. and Laguna, M., 1995, Genetic Algorithms and Tabu Search - Hybrids for Optimisation, Computers and Operations Research, Vol 22, No1, pp 111-134.

Grefenstette, J.J., 1984, "GENESIS: A System for Using Genetic Search Procedures", Proceedings of the 1984 Conference on Intelligent Systems and Machines, pp 161-165.

Groff, G., "A lot-sizing rule for time-phased component demand", Production and Inventory Management, 20,1, 1979, pp 47-53.

Goldberg David E, Genetic Algorithms in search, Optimization, and Machine Learning, Addison-Wesley, 1989

Ham I., Introduction to Group Technology, SME Technical paper, MMR-76-03, Feburary 1975.

Ham Inyong, Hitomi, Katsundo Yoshida Teruhiko, Group Technology: Applications to production management, Kulmer-Nijhoff, 1985.

Han Chingong, Ham Inyong, MultiObjective cluster analysis for part family formations. International Journal of Manufacturing Systems, vol. 5, no 4, 1986.

Huaung P.Y. and Houck B.L.W., Cellular manufacturing: an overview and bibliography, Production and Inventory Management, vol. 26, pp 83-93, 1985.

Helgerson, N.B. and Birnie, D.P., Assembly line balancing using the ranked positional weight technique, Journal of Industrial Engineering, XII (6), p 394, 1961.

Heragu, S. and Alfa, A.S., (1992) Experimental analysis of simulated annealing based algorithms for the layout problem, European Journal of Operational Research, Vol 57, pp 190-202.

Hertz., A. and Widmer M., 1996, Improved Tabu Search Approach for the Job Shop Scheduling Problem with Tooling Constraints, Discrete Applied Mathematics, Vol 65, No1-3, pp 319-345.

Hillier, F.S. and Connors, M.M., Quadratic assignment problem algorithms and the location of indivisible facilities, Management Science, 1966, Vol 13, 42-57

Hindi, K.S., Solving the single item, capacitated dynamic lot sizing problem with startup and reservation costs by Tabu Search, Computers abd Industrial Engineering, Vol 28, No 4, pp 701-707, 1995.

Hitomi, K. (1979), Manufacturing Systems Engineering, Taylor and Francis, London.

Holland J H Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press 1975

Holt, Modigliani, Muth and Simon, Planning, Production, Inventories and Work Force, Prentice-Hall, 1960.

Hughes, M., Good Design by Breeding, Professional Engineering, pp 14-15, June 1990.

Iwata K., Oba F. and Yasuda K. (1984) Simulation for Design and Operation of Manufacturing Systems, Annlals of the CIRP, Vol. 33, No 1, pp 335-339.

Jackson B.M. and Zerbe D.R. (1968) Determination of standard sizes to be manufactured using dynamic programming, Journal of Industrial Engineeering. 19/8, 401 - 406.

Joshi, K., "Storage Space Costs and the EOQ Model", Journal of Purchasing and Materials Management, Summer, 1990, pp.37-41.

Kamien, M. I. and Li, L., "Subcontracting, Coordination, Flexibility and Production Smoothing in Aggregate Planning", Management Science, 1990, 36/11, 1352-1363.

Karmarkar U.S., Kerke Sham, Batching Policy in KanbanSystems, International Journal of Manufacturing Systems, vol. 8, no 4, pp 317-327, 1989.

Karp, R.M. and Held, M., Finite-state processes and dynamic programming, SIAM Journal of Applied Mathematics, Vol 15, pp 693-718, 1967.

Kilbridge, K. and Wester, L., A heuristic method of assembly line balancing, Journal of Industrial Engineering, VXII (4), p 292, 1961.

Kingdon J. and Dekker L. (1994) Development needs for diverse genetic algorithm design, IEE Colloquium of "Applications of Genetic Algorithms", London.

Kouvelis, P, Chiang, W and Fitzsimmons, J, (1992), Simulated annealing for machine layout problems in the presence of zoning constraints, European Journal of Operational Research, Vol 57, pp 203-223.

Kuik, R., Salmon, M, Vanwassenhove and Maes, J., Linear Programming, Simulated Annealing and Tabu Search Heuristics for Lot sizing in bottle neck assembly systems, IIE Transactions, Vol 25, No 1, pp 62-72, 1993.

Kusiak, Andrew, Chow, Wing, S., Efficient solving of the group technology problem. International Journal of Manufacturing Systems, vol. 6, no 2, 1987.

Kusiak, A. and Heragu, S.S., 1987, The facility layout problem, European Journal of Operations Research, Vol. 11, No. 2, pp 229-253.

Kusiak, A. Intelligent Manufacturing Systems,Prentice-Hall, UK, 1990.

Laguna, M and Velarde, J.G., 1991, A Search Heuristic for Just-In-Time Scheduling in Parallel Machines, Journal of Intelligent Manufacturing, Vol 2, No4, pp 253-260.

Laguna, M., Barnes, J.W., and Glover F.W., 1991, Tabu Search Methods for a Single Machine Scheduling Problem, Journal of Intelligent Manufacturing, Vol 42, No2, pp 75-82.

Lee R.C. and Moore J.M., CORELAP - Computerized relationship layout planning, Journal of Industrial Engineering, 1967, 18(3), 195-200

Lewis, T.G and El-Rewini, H., Introduction to parallel computing, Prentice-Hall, 1992.

Lin, Z.C. and Chen, S.C. Measuring sequence planning with Dynamic-Programming, International Journal of Computer Integrated Manufacturing, Vol 8, No 4, pp 237-246, 1995

Love, C. E. and Turner, M., "Note on Utilizing Stochastic Optimal Control in Aggregate Production Planning", European Journal of Operational Research, 1993, 65/2, 199-206.

Lupton T., The management of change to advanced manufacturing systems, Human factors, 1986

Marca, D.A. and McGowan, C.L. (1988), SADT - Structures Analysis and Design Technique, Prentice Hall.

Martin M.J.C. and Piff C.I. (1971) Case Study: Determining the range of standard sizes of capital goods to be manufactured, Int J of Prod Res., 9/4, 443 - 456.

Metropolis, N., Rosenbluth, A., Teller, A. and Teller E., Equation for state calculations by fast computing machines, Journal of Chemical physics, 21, pp 1087-1092, 1953.

Miller, T. C., "Integrating Current End-Item Inventory Conditions into Optimization-Based Long-Run Aggregate Production and Distribution Planning", Production and Inventory Management Journal, 1991, 32/4, 74-80.

Minagawa, M. and Kakazu, Y. 1992, A genetic approach to the line balancing problem, Human Aspects in Computer Integrated Manufacturing, 737-743.

Monden, Y. 1983, Toyota Production System: Practical Approach to Production Management, Norcross, Georgia: Industrial Engineering and Management Press.

Moore, J.M., Computer aided facilities design: An international survey, International Journal of Production Research, 1974, vol 12, No1, 21-44

Muther R. and McPherson, K., Four approaches to computerized layout planning, Journal of Industrial Engineering, 1970, 2 (2), 39-42

Muther, R. Practical Plant Layout, McGraw-Hill, 1955

McAuley, J, , Machine Grouping for efficient production, Production Engineering, 51, 53, 1972.

McLaren, B.J., A study of Multiple level lot sizing techniques for material requirements planning systems, Phd dissertation, Purdue University, 1977.

McManus, J., Some lessons from a decade of Group technology, Production Engineer, vol. 59, no 11, pp 40-42, 1980.

Nam, S-J. and Logendran, R., "Aggregate Production Planning - A Survey of Models and Methodologies", European Journal of Operational Research, 1992, 61/3, 255-272.

Nydick, R.L. and Weiss, H.J., "An evaluation of variable-demand lot-sizing techniques", Production and Inventory Management Journal, 4th Quarter, 1989, pp 41-44.

O'Brien, C. and Abdel Barr, S.E.Z., An interactive approach to computer aided facility layout, International Journal of Production Research, Vol 29., No. 2, pp37-391, 1980.

O'Grady, P., Deleersnyder, J., Hodgson, T.J. and Muller, H., Kanban Controlled Pull Systems: An Analytic Approach, Management Science, Vol 35, No 9, September, pp 1079-1091, 1989.

O'Grady, P., Putting the Just-In-Time philosophy into practice, Nichols, New York, 1988.

Odanaka, T, Dynamic-programming and optimal inventory processes, Computers and Mathematics with Applications, Vol 27, pp 213-217, 1994.

Parnaby J. Concept of a manufacturing system, International Journal of Production Research, 1979, 17, 123-35

Pei-Chann C. and Ru-Ching, H, (1994), A simulated annealing approach to parallel processor scheduling problems with precedence relations, Journal of the Chinese Institute of Engineers, Bvopl 17, Iss 4 pp 485-97

Pentico, D.W., (1988), The Discrete Two-Dimensional Assortment Problem, Operations Research, Vol 36, No 2, pp 324-332.

Philipoom, P.R., Rees, L.P., Taylor, B.W. and Huaung, P.Y. 1987, An Investigation of the factors Influencing the Number of Kanbans Required in the Implementation of the JIT Technique with Kanbans, International Journal of Production Research, 25, 457-472.

Picard, J. and Queyranne, M, On the one-dimensional space allocation problem, Operations Research, Vol 29, No 2, pp 371-391, 1981.

Promodel (1993), Promodel version 1.05 User Guide, Promodel Corporation, 1875 South State, Suite 3400, Orem, Utah 84058, U.S.A.

Ravindran A., Phillips D. and Solberg J., 1987, Operations Research Principles and Practice,John Wiley and Sons.

Reeves, C.R, 1993, Improving the efficiency of Tabu Search for machine scheduling Problems, Journal of the Operational Research Society, Vol 4, No 4, pp 375-382.

Riggs, J. L., Production Systems: Planning, Analysis and Control, 4th Ed., 1987, John Wiley and Sons.

Saad, G. H., "Hierarchical Production-Planning Systems: Extensions and Modifications", Journal of the Operational Research Society, 1990, 41/7, 609-624.

Sahni, S. and Gonzalez, T., NP-complete approximation problem, Journal of Association for Computing Machinery, Vol. 23, No. 3, 1976, pp 555-565.

Salveson, M.E., The assembly line balancing problem, Transactions of the Americian Society of Mechanical Engineering, 77, August, p 939, 1955.

Satyadas, A. and Chen, H. C., "Multicriteria Multigoal Decision Making - The Fuzzy Paradigm", Computers and Industrial Engineering, 1992, 23/1-4, 393-396.

Shannon, R.E, Mayer, R. and Adelsberger, H.H., Expert systems and simulation, Simulation, 1985, 44:6, 275-284

Shannon, R.E., Knowledge based simulation techniques for manufacturing, International Journal of Production Research, 1988, vol 26, no 5, 953-973.

Seehof, J.M. and Evans, W.O., Automated layout design program, Journal of Industrial Engineering, Vol 18, No. 12, pp 690-695, 1967.

Silver, E.A. and Meal, H.C., "A heuristic for selecting lot-sizing quantities for the case of deterministic time-varying demand rate and discrete opportunities for replenishment", Production and Inventory Management, 14,2, 1973, pp 64-74.

Silver E.A. and Kelle P.(1989) Stocking of prescribed sizes of and item.
J. Opl Res Soc., 40/8, 719 - 727.

Simfactory (1990), Simfactory II.5 Reference Manual Release 2, CACI, Palm Court, 4 Heron Square, Richmond, Surrey, UK.

Sims E.R. (1975) Precision costing of manufacturing operations, Society of Automotive Engineers, Report No. 680575, 1 - 16.

Solberg, J.J., Optimal design and control of computerised manufacturing systems. Proccedings of AIIE Systems Engineering Conference, pp 138-144, Boston, Massachusetts, 1976.

Solberg, J.J., CAN-Q user's guide, Purdue University, West Lafayette, 1980.

Solot, P and Bastos, J.M., MULTIQ: A queueing model for FMSs with several pallet types. Journal of the operatopnal research society, 39(9):811-821, 1988

St John, R., "The evils of lot sizing in MRP", Production and Inventory Management, 4thQuarter, 1984, pp 75-85.

Stevenson, W. J., Production/Operations Management, 4th Ed., 1993, Irwin.

Stockton D.J. (1983) Improving the new product development process, PhD. Thesis, Loughborough University of Technology.

Suri, R and Hildebrant, R.R., Modelling flexible manufacturing systems using mean value analysis, Journal of manufacturing systems, 3(1):27-38, 1984.

Suri, R. and Diehl, G.W., Manuplan - a precursor to simulation for complex manufacturing systems, Proceedings of the Winter Simulation Conference, pp 411-420, 1985.

Swanson E.(1970) Determination of standard sizes to be manufactured using dynamic programming - An extension, AIIE Transactions, September, 191 - 202.

Taha, H. A., Operations Research: An Introduction, 2nd Ed., 1971, Macmillan Publishing.

Taubert W. H., "A Search Decision Rule for the Aggregate Scheduling Problem", Management Science, 1968, 14/6, 343-359.

Tenga, R.F., Faccenda, J.F. and Rhoads, D.G., Use of a genetic Algorithm for economic optimisation of a manufacturing system, Simulators V Proceedings of SCS Simulators Conference, pp 282-287, 1988.

Teresko. J. , Can robots integrate manufacturing plants. Industrial week, 204, 99, 1980.

Tompkins, J.A. and White, J.A., Facilities Planning, Wiley, New York, 1984.

Tryfos P. (1985) On the optimal choice of sizes, Operations Research., 33, 678 - 684.

Vanhoesel, S., Kuik, R, Salomon, M. and Vanwassenhove, L.N., The Single-Item discrete lotsizing and scheduling problem- optimization by linear and dynamic programming.

Vercellis, C., "Multi-Criteria Models for Capacity Analysis and Aggregate Planning in Manufacturing Systems", International Journal of Production Economics, 1991, 23/1-3, 261-272.

Vollman, T.E., Berry, W.L. and Whybark, D.C., Manufacturing Planning and Control Systems, 3rd Edition, 1992, Irwin.

Wagner, H.M. and Whitin, T.M., "Dynamic version of the economic lot size model", Management Science, October, 1958, pp 89-96.

Wang, W and Bell, R., An intelligent user interface in a knowledge-based modelling system for the design of flexible manufacturing facilities, International Journal of Computer Integrated Manufacturing, 1991, vol 4, no 6, 364-374.

Wemmerlov, U and Whybark, (1984) Assemble to order manufacture: Implications for materials management. Journal of operations management, 4(4):347-368 1984.

Whitt, W., 1983, The Queueing Network Analyser, Bell System Technical Journal, Vol 62, No9, pp2779-2815.

Widmer, M., 1991, Job Shop Scheduling with Tooling Constraints, A Tabu Search Approach, Journal of the Operational Society, Vol 42., No 1., pp 75-82.

Wild, R., Production amd Operations Management, Cassell Education Limited, pp 97-129, 1989.

Williams D.J., Manufacturing Systems, An introduction to the technologies, Chapman & Hall, 1994

WITNESS (1991), WITNESS user manual, AT&T ISTEL Visual Interactive Systems lTd, Highfield House, Headless Cross Drive, Redditch, Worcs., B97 5EQ, U.K., 1991.

Wolfson M.L. (1965) Selecting the best lengths to stock, Operations Research., 13, 570 - 585.

Woolsey, G., "A Requiem for the EOQ: An Editorial", Production and Inventory Management Journal, 3rd Quarter, 1988, pp.68-72.

Wu, H. L., Venugopal, R., and Barash, M. M., Design of cellular manufacturing system. A syntatic pattern approach, International Journal of Manufacturing Systems, vol. 5, no. 2, 1986.

Yamada, T, Rosen, B.E. and Nakano, R, (1994), A simulated annealing approach to job shop scheduling using critical block transition operators, 1994 IEEE International Conference on Neural Networks, July 1994, ISBN 0 7803 1901 X, pp 4687-92.

Young, H.H., Optimsing models for production lines, Journal of Industrial Engineering, XVII(1), pp 70-78, 1967.

Zegordi, S.H, Itoh, K, Enkawa, T and Chung, S, (1995), Simulated annealing scheme incorporating move desirability table for facility layout problems, Journal of the Operations Research Society of Japan, Vol 38, Iss 1, pp 1-20.

10. Bibliography

Intelligent Manufacturing Systems
Kusiak A., Prentice-Hall, 1990

Productions and Operations management
Wild R., Cassel, 1989

Manufacturing Systems Design and Analysis
Wu B., Chapman and Hall, 1992

Manufacturing Systems: An introduction to the technologies
Williams D.J., Chapman and Hall, 1991

Queueing Methods for Services and Manufacturing
Hall R.W., prentice and Hall, 1991

Production Management Systems: A CIM perspective
Browne J., Harhen J. and Shivnan J., Addison-Wesley 1988

Artificial Intelligence: structures and Strategies for Complex problem Solving
Luger G.F. and Stubblefiel W.A., Benjamin Cummins, 1993

Operations Research
Taha H.A., *MacMillian*, 1982

Introduction to optimisation methods
Adby P.R. and Dempster M.A.H., Chapman and Hall 1974

Operations Research Principles and Practice
Ravindran A., Phillips D. and Solberg J., John Wiley and Sons, 1987

Handbook of Genetic Algorithms
Davis L., Van Nostrand reinhold, 1991

Modelling and Analysis of Manufacturing Systems
Askin R.G. and Standridge C.R., John Wiley and Sons, 1993

Thinking with Models
Saaty T.H. and Alexander J., Pergamon, 1981

Queueing Theory in Manufacturing Systems Analysis and Design
Papadopoulos M.T., Heavey C. and Browne J., Chapman and Hall, 1993

11. Appendix I MOM and GA Calculations and Solutions for MRP lot Sizing

PART FLASHLIGHT

Ordering Cost £100
Holding Cost £32

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	250	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	250	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	0	0	0	0	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/32) = 23.99
TBO = 23.99/92.1 = 0.26
T^* = 0
OMT = 92.1[0 + (0.26 - 0)(0)] = 0

Ordering Cost = £1,100
Inventory Carrying Cost = 17,680
Total Cost = 18,780

┼ Measured in the following Period

PART HEADLIGHT

Ordering Cost £100
Holding Cost £10

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	250	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	250	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	0	0	0	0	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/10) = 42.92
TBO = 42.92/92.1 =0.466
T^* = 0
OMT = 92.1[0 + (0.466 - 0)(0)] = 0

Ordering Cost = £1,100
Inventory Carrying Cost = 5,525
Total Cost = £6,625

┼ Measured in the following Period

PART BODY ASSEMBLY

Ordering Cost £100
Holding Cost £15

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	250	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	250	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	0	0	0	0	0	0	0	0	0	0	0	0

Average D	= 92.1			
EOQ	= (2(100)(92.1)/15) = 35.04			
TBO	= 35.04/92.1 =0.38		Ordering Cost =	£1,100
*T**	= 0		Inventory Carrying Cost =	8,288
OMT	= 92.1[0 + (0.38 - 0)(0)] = 0		Total Cost =	£9,388

† Measured in the following Period

PART BATTERIES

Ordering Cost £100
Holding Cost £3

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	250	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	250	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	0	0	0	0	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/3) = 78.35
TBO = 78.35/92.1 =0.85
T^* = 0
OMT = 92.1[0 + (0.85 - 0)(0)] = 0

Ordering Cost = £1,100
Inventory Carrying Cost = 1,658
Total Cost = £2,758

† Measured in the following Period

PART PLASTIC HEAD

Ordering Cost £100
Holding Cost £2

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	20		35		70	180	250	270	230	40	0	10
Beginning Inventory	20	10	35	20	70	180	250	270	230	40	0	10
Ending Inventory	10	0	20	0	0	0	0	0	0	0	0	0
Part Periods	0	10	0	20	0	150	270	230	0	0	0	0
Exceed Target?	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	-	20	0	40	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/2) = 95.97
TBO = 95.97/92.1 = 1.04
T^* = 1
OMT = 92.1[0 + (1.046 - 1)(1)] = 4.2366

Ordering Cost = £ 900
Inventory Carrying Cost = 1,175
Total Cost = £2,075

† Measured in the following Period

PART LENS

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	250	270	230	50	0	0
Beginning Inventory	55	45	35	20	70	180	250	270	230	50	10	10
Ending Inventory	45	35	20	0	0	0	0	0	0	10	10	0
Part Periods	0	10	40	100	0	0	0	0	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	0	0	0	60	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost =	£ 700
Inventory Carrying Cost =	688
Total Cost =	£1,388

+ Measured in the following Period

PART BULB SUB-ASSEMBLY

Ordering Cost £100
Holding Cost £3

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	150	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	150	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	150	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	-	-	-	-	-	-	-	-	-	-	-	-

Average D	= 92.1
EOQ	= (2(100)(92.1)/3) = 78.36
TBO	= 78.36/92.1 =0.85
T^*	= 0
OMT	= 92.1[0 + (0.85 - 0)(0)] = 0

Ordering Cost =	£1,100
Inventory Carrying Cost =	1,658
Total Cost =	£2,758

† Measured in the following Period

PART REFLECTOR

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	150	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	55	45	35	20	70	180	150	270	230	50	10	10
Ending Inventory	10	35	20	0	0	0	0	0	0	10	10	0
Part Periods	0	10	40	100	0	0	0	0	0	0	0	0
Exceed Target?	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$C_H(K-1)D_K$	-	-	-	60	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.7
TBO = 135.7/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART SHELL ASSEMBLY

Ordering Cost £100
Holding Cost £11

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	150	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	150	270	230	40	0	10
Beginning Inventory	10	10	15	20	70	180	150	270	230	40	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	No	No	No	No	No	No	No	No	No	No	No	No
$C_H(K-1)D_K$	0	110	0	0	0	0	0	0	0	0	0	0

Average D	= 92.1
EOQ	= (2(100)(92.1)/11) = 40.92
TBO	= 40.92/92.1 =0.44
*T**	= 0
OMT	= 92.1[0 + (0.44 - 0)(0)] = 0

Ordering Cost =	£1,100
Inventory Carrying Cost =	6,078
Total Cost =	£7,178

┼ Measured in the following Period

PART SPRING

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	150	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	55	45	35	20	70	180	150	270	230	50	10	10
Ending Inventory	45	35	20	0	0	0	0	0	0	10	10	0
Part Periods	0	10	40	100	0	0	0	0	0	0	0	0
Exceed Target?	No	No	No	Yes	No	No	No	No	No	No	No	No
$C_H(K-1)D_K$	-	-	-	60	0	0	0	0	0	0	0	0

Average D	= 92.1
EOQ	= (2(100)(92.1)/1) = 135.72
TBO	= 135.72/92.1 = 1.47
*T**	= 1
OMT	= 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost =	£ 700
Inventory Carrying Cost =	688
Total Cost =	£ 1,388

† Measured in the following Period

PART PLASTIC

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	20		35		70	180	250	270	230	50		
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	55	35	35	0	70	180	150	270	230	50	0	0
Ending Inventory	35	35	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	70	0	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	70	-	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART BULB

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	55	45	35	20	70	180	150	270	230	50	10	10
Ending Inventory	45	35	20	0	0	0	0	0	0	10	10	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART BULB HOLDER

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	55	35	35	0	70	180	150	270	230	50	0	0
Ending Inventory	35	35	0	0	0	0	0	0	0	0	0	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	20
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	0	0	0	0	0	0	0	0

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.29

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART ON/OFF SWITCH

Ordering Cost £100
Holding Cost £4

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	10	10	15	20	70	180	150	270	230	40		10
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	0	0	0	0	0	0	0	0
Exceed Target?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	-	-	-	-	-	-	-	-	-

Average D	= 92.1
EOQ	= (2(100)(92.1)/4) = 67.86
TBO	= 67.86/92.1 = 0.73
*T**	= 0
OMT	= 92.1[0 + (0.73 - 0)(0)] = 0

Ordering Cost =	£ 1,100
Inventory Carrying Cost =	2,210
Total Cost =	£ 3,310

† Measured in the following Period

PART CONNECTOR BARS

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	-	-	-	-	-	-	-	-

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.62

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART PLASTIC SHELL

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	-	-	-	-	-	-	-	-

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.62

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART KNOB

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	-	-	-	-	-	-	-	-

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.62

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART METAL SLIDES

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	10	10	15	20	70	180	250	270	230	40	0	10
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	10	40	100	180†	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	60	-	-	-	-	-	-	-	-

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.62

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

PART PLASTIC

Ordering Cost £100
Holding Cost £1

Period	1	2	3	4	5	6	7	8	9	10	11	12
Requirements	55				70	180	250	270	230	50		
Order Quantity	55				70	180	150	270	230	50		
Beginning Inventory	10	10	15	20	70	180	150	270	230	50	0	10
Ending Inventory	0	0	0	0	0	0	0	0	0	0	0	0
Part Periods	0	0	0	0	280	250†	270†	230†	0	0	0	0
Exceed Target?	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No
$C_H(K-1)D_K$	-	-	-	-	240	-	-	-	-	-	-	-

Average D = 92.1
EOQ = (2(100)(92.1)/1) = 135.72
TBO = 135.72/92.1 = 1.47
T^* = 1
OMT = 92.1[0 + (1.47 - 1)(1)] = 43.62

Ordering Cost = £ 700
Inventory Carrying Cost = 688
Total Cost = £ 1,388

† Measured in the following Period

ITEM	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10	Period 11	Period 12
Flashlight	10	10	20	20	70	180	250	270	240	40	0	10
Headlight	20	10	20	20	70	200	250	270	250	40	0	10
Bodyassembly	10	10	10	20	70	190	260	270	240	40	0	10
Batteries	10	10	10	20	70	180	250	270	230	40	0	10
Plastic Head	40	0	20	0	70	180	250	270	230	40	0	10
Lens	70	0	0	0	60	180	260	280	240	50	0	0
Bulb Subassembly	10	10	20	20	70	190	250	280	240	40	0	10
Reflector	60	0	0	0	70	180	250	270	240	60	0	0
Shell Assembly	10	10	20	20	80	180	260	270	240	50	0	0
Spring	70	0	0	40	250	0	250	280	290	0	0	0
Plastic	70	0	0	0	260	0	240	260	250	50	0	0
Bulb	60	0	0	0	80	190	250	270	250	60	0	0
Bulb Holder	70	0	0	0	70	190	260	290	260	50	0	0
On Off Switch	20	20	20	10	70	180	260	270	240	50	0	0
Connector	60	0	0	0	70	180	250	270	240	60	0	0
Plastic Shell	70	0	0	50	230	0	250	280	290	0	0	0
Knob	60	0	0	0	80	190	250	270	250	60	0	0
Metal Slides	50	0	0	20	70	180	250	270	240	50	0	0
Plastic	130	0	0	0	0	180	250	270	250	60	0	0

GA SOLUTION

12. Appendix II Rank Positional Weight methods calculations and RPW and GA solutions

<table>
<tr><th>Task</th><th>Task Duration</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th><th>I</th><th>J</th><th>K</th><th>Positional weight</th></tr>
<tr><td>A</td><td>45</td><td></td><td>|</td><td>+</td><td></td><td></td><td>+</td><td>+</td><td></td><td></td><td>+</td><td>+</td><td>140</td></tr>
<tr><td>B</td><td>11</td><td></td><td></td><td>|</td><td></td><td></td><td>+</td><td>+</td><td></td><td></td><td>+</td><td>+</td><td>95</td></tr>
<tr><td>C</td><td>15</td><td></td><td></td><td></td><td></td><td></td><td>|</td><td>|</td><td></td><td></td><td>+</td><td>+</td><td>84</td></tr>
<tr><td>D</td><td>30</td><td></td><td></td><td></td><td></td><td>|</td><td></td><td></td><td>+</td><td>+</td><td>+</td><td>+</td><td>114</td></tr>
<tr><td>E</td><td>15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td>|</td><td></td><td></td><td>84</td></tr>
<tr><td>F</td><td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>57</td></tr>
<tr><td>G</td><td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>57</td></tr>
<tr><td>H</td><td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>57</td></tr>
<tr><td>I</td><td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>57</td></tr>
<tr><td>J</td><td>30</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>|</td><td>45</td></tr>
<tr><td>K</td><td>15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>15</td></tr>
</table>

Task	**A**	**D**	**B**	**C**	**E**	**F**	**G**	**H**	**I**	**J**	**K**
Task Duration	45	11	15	30	15	12	12	12	12	30	15
PW	140	114	95	84	84	57	57	57	57	45	15
Immediate Predecessors	-	-	A	B	D	C	C	E	E	F,G,H ,I	

Workstation	Task	PW	Immediate Predecessor	Task Duration	Cumulative Station Time Z	Unassigned Station Time C - Z
1	A	140	-	45	45	5
2	D	114	-	11	11	39
	B	95	A	15	26	24
	E	84	D	15	41	9
3	C	84	B	30	30	20
	F	57	C	12	42	8
4	G	57	C	12	12	38
	H	57	E	12	24	26
	I	57	E	12	36	14
5	J	45	F,G,H,I	30	30	20
	K	15	J	15	45	5

BALANCING LOSS = ((5*50 - 209)/(5*50))*100 = 16.4 %

Solution	WS 1	WS 2	WS 3	WS 4	WS 5
1	A	B,D	C,E,F	G,H,I	J,K
2	A	B,D	C,E,G	F,H.I	J,K
3	A	B,D	C,E,H	F,G,I	J,K
4	A	D	B,C,E	F,G,H,I	J,K

GENETIC ALGORITHM SOLUTION

Solution	WS 1	WS 2	WS 3	WS 4	WS 5
1	A	B,D,E	C,F	G,H,I	J,K

RANK POSITIONAL WEIGHT TECHNIQUE SOLUTION