Generating Optimised MRP Lot Sizes Using Genetic Algorithm: Considering Supplier Deals

J. A. D. N Thanura[1], R. M. U. J. D Rathnayake[1], S. D. Dewasurendra[1], L. Rajakaruna[2]

[1]Department of Computer Engineering, Faculty of Engineering,
University of Peradeniya, Sri Lanka

[2]IFS R & D Internationals Pvt. Ltd., Sri Lanka.

***Abstract*— Material requirement Planning (MRP) is the process of determining the detailed order release policy for parts, components and material needed to produce end products to a required production plan at minimum total cost.**

The objective of this paper is to present a method to generate optimised MRP solutions using the genetic algorithm to multi-product, multi-period lot-sizing problems with supplier discounts/deals and storage capacity limitations. Demand for products is assumed known.

Keywords—MRP; Genetic Algorithm; GA; Lot Sizing; Capacity Constraints; Chromosome; Heuristic Search; POR; ERP

I. INTRODUCTION

Material requirement Planning (MRP) is a production planning, scheduling, and inventory control system used to manage dependent demand in manufacturing processes. It provides detailed schedules specifying when raw material and components should be ordered so that the final product could be delivered to meet a given demand profile.
The popularity and effectiveness of MRP systems have created much interest in extending these methods to constrained multi-layer production systems. Lot sizing problems become difficult in multi-layer, multi-product planning when capacity and supplier related cost constraints like Discounts/Deals are introduced. Currently available MRP software and systems lack the flexibility to handle these complex production lot-sizing problems effectively. The present paper describes a method to handle these situations effectively in an MRP system, thus helping production companies to increase profit and reduce waste.

Genetic algorithm (GA) is a heuristic search algorithm that mimics natural evolution to find near-optimum solutions to complex problems with large solution spaces. Genetic Algorithms adopt the properties of natural evolution and "survival of the fittest".

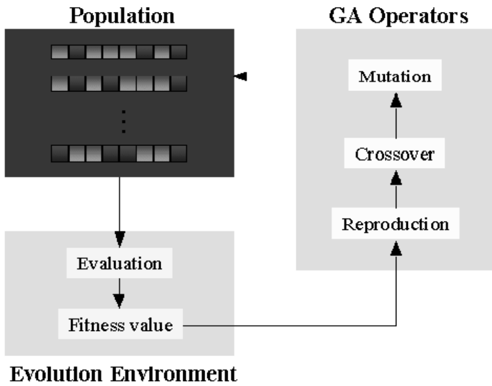

Fig. 1. Genetic Algorithm Flow

In genetic algorithm (GA), a population of potential solutions (chromosomes) is subjected to a sequence of manipulations (crossovers and mutations). These solutions are striving for survival. Some of the fittest solutions are selected to survive in the next generation. Since the successor generation consists mostly of the fitter members of the previous generation (ones with better/stronger characteristics) solutions with better characteristics tend to survive. After several generations, the algorithm either converges to an optimal solution or terminates based on termination conditions.
GA has been used to solve lot sizing problems of different complexities related to MRP [1].
The main objective of this paper is to generate an MRP solution to minimise the total cost by deciding suitable suppliers and lot-sizes using the genetic algorithm in the presence of supplier discounts/deals. A secondary objective is to parallelise the algorithm.

II. LITERATURE REVIEW

The advantages of integrating Genetic Algorithms to MRP have been reported by several researchers. Some of the work has direct correlation with our project. Those are the following. In [2] *an MRP Optimisation Algorithm Based on*

Multi Objective Genetic Evolution is proposed. Here they discuss the advantages of integrating MRP and capacity requirement planning based evolutionary algorithms. They design an optimisation algorithm using GA and prove the superiority of evolutionary algorithms (GA being an evolutionary algorithm) over conventional MRP algorithms. [3] Proposes heuristic Genetic Algorithms for General Capacitated Lot-Sizing Problems. They design a domain specific encoding scheme for lot-sizes and provide a heuristic shifting procedure as the decoding schedule. The main contribution of these genetic algorithms is the presentation technique that encodes only the binary variables for the setup patterns but derives other decision variables by making use of the problem-specific knowledge. Some results from the computational experiments are also given.
In [4] GA is applied for inventory lot-sizing problems with supplier selection under storage capacity constraints. This work has high relevance to ours. They discuss the possibility of calculating optimal inventory lot-sizing considering suppliers/providers with joint purchase and transportation costs. They also propose using genetic algorithms when maximum storage capacities over given time periods are given. The proposed method has been validated giving detailed computational results that are superior to those from LINGO (an optimisation software package), themselves close to optimum for realistically sized problems. Additionally, the computation time when using GAs is also short, making it a very practical means for solving the multiple products and multi-period inventory lot-sizing problem with supplier selection under storage capacity. The work reported in [1] describes initial efforts to use genetic algorithms to develop a multi-level MRP lot sizing technique. Initially the limitations of existing lot sizing techniques are explained: such methods are seen to determine lot sizes for individual items at only a single level in BOM structure and hence by default assume that demand for the items is independent. The essential procedures used in genetic algorithms are then explained using an example MRP lot sizing problem. This example clearly illustrates the multi-level lot sizing potential of GA. Order release schedules are then determined using GA and compared with McLaren Order Moment method [5]. The results are then used to identify the effectiveness of GA for determining MRP lot sizes. We continue from this latter work and propose some improvements which are then validated.

Looking at commercial MRP scheduling software such as, Fishbowl Manufacturing, E2 Shop System, Global Shop Solutions and One-System ERP, none of these has used GA in their algorithms. Most of them use some mathematical programming or heuristic lot-sizing methods which only consider some simple demand schemes. Therefore they lack in capability to handle dynamic and complex MRP problems involving many constraints. Even though commercial software such as AiPlAN has implemented GA in their algorithms, none of them supports supplier selection based on discounts/deals in Material Requirement Planning.

III. Methodology and Implementation

A. *Planned Order Release (POR)*

Planned orders generated by MRP list suggested quantities and, release and receipt dates necessary to cover net requirements. The quantities are based on the pre-determined lot size rules, and the release date is lead-time offset of planned receipt date. The planned order differs from a scheduled receipt in that the former is a plan and the letter is a released purchase or shop order. The planned order release is also affected by the BOM structure.

period / parts	1	2	3	4	5	6	7	8	9
A					Q	Q		Q	
B			Q		Q				
C		Q	Q	Q	Q				

Fig. 2. Planned Order Release

B. *Design of the Chromosome*

An integer string was used as the chromosome since it is more effective and convenient. The data from the planned order release (Fig. 2) is mapped to chromosome (Fig. 3) as follows. Here, only the structure is mapped, numerical values are not copied.

Fig. 3. Design of the Chromosome

The chromosome is an integer array as shown in the figure 3

- A, B and C are separate parts.
- S_x is the supplier for each part. It is a variable.
- Q_n is the quantity for each period.

As one can observe, the chromosome size can be varied from order to order depending on the demand quantities and periods.

C. *Initial Population*

Ninety percent of the initial population is created randomly with random integers within the range of suppliers and max order size.

- Q_n = rand() % Max order size
- S_x = rand() % Number of Suppliers

The other ten percents of the populations are created based on the Planned Order Release (POR).

D. *Bill of Materials (BOM)*

The Bill of Materials is the list of the raw materials, sub-components or parts and the quantities of each required to create a product: see Figure 4.

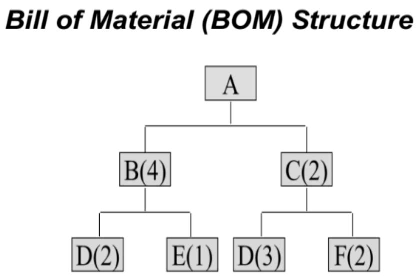

Fig. 4. BOM Structure

Since the Planned Order Release (POR) is also a valid solution (see Fig. 2.), including it in the initial population may provide a little nudge to the population towards a better solution. It is more efficient than starting with a completely random population. When affecting 10% of the population with POR solution, only the quantities are selected from POR, the suppliers are randomly selected since POR doesn't have suppliers. The fitness of every solution in the initial population is set to zero.

E. *Calculating the Fitness*

First the cost for quantities of separate parts with relevant supplier is calculated. Then the total cost is calculated as the sum of each cost for each part quantity. Then the inverse of the cost is taken as the fitness of the solution. The higher the fitness, the better the solution is.

F. *Cost Calculation*

The cost functions are saved as strings in the MySQL database, even though the user enters them as arithmetic expressions. Therefore, the cost functions are returned as strings when they are queried, and turning them back to arithmetic expressions is little difficult. A third party library is used for evaluating expressions: evaluateExpression [6]

G. *Validation*

Each solution is checked to see whether the solution can fulfil the demand on time. If any particular solution is not capable of delivering the demand, its fitness is set to zero. Then the whole population is sorted by the fitness to find the best solutions. This is done by a simple insertion sort.

H. *Elitism*

In some cases the best-fitted solutions from the previous generation may be altered or lost after the genetic operations are applied: the next generation's best solutions may be weaker than the predecessors. Normally these solutions may be rediscovered in future generations but there is no guarantee. To avoid this, the feature known as Elitism is used. A portion of the best-fitted solutions is sent to the next generation without any alteration, guaranteeing the best solutions are preserved. In this project best fitted 10% is sent to the next generation without any change to prevent re-discovering earlier solutions.

I. *Crossover*

In genetic algorithms, the crossover is the most crucial part. This is the part where genetic algorithms adopt natural reproduction. Much like mating, it takes two solutions as the parents to create offspring. Single Point Crossover method is used in this project.

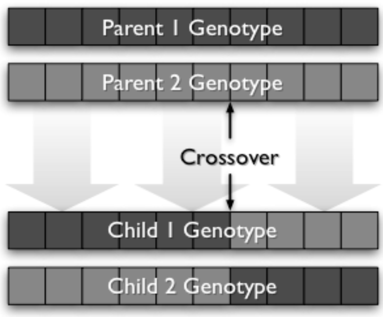

Fig. 5. Single Point Crossover

In the Single Point Crossover, a random point in both parent chromosomes is selected. Then all the data beyond that point are swapped between two parents. The resulting two solutions are considered as offspring (children).

J. *Mutation*

Mutation is the operator that is used to keep the diversity in the solution space. Much like mutations in natural evolution, some mutations lead to better-fitted solutions. Without the mutations, the solutions may converge to or get stuck in a local optimum. A solution may change completely with a single element mutation and may lead to a better optimum.

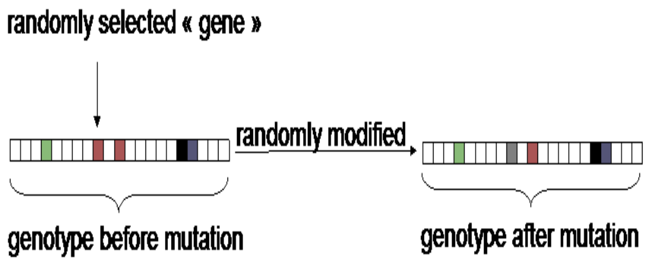

Fig. 6. Mutation

In the Genetic algorithms, the mutation probability is user-definable and should be kept low. The search will be turned into a random search if the mutation rate is too high.
The probability of the mutation is set to 0.01 in this algorithm.

IV. Results and Analysis

In our system, we checked our algorithm using different generation sizes, population sizes and recorded the time it took to get to an optimal solution and how cost varied with the generation sizes and population sizes.

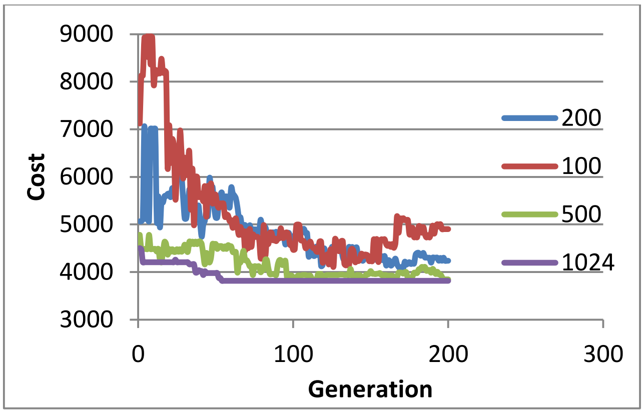

Fig. 7. Cost with population

In figure 7, the cost of the best solution of each generation is mapped.
We used a constant generation size of 200 and different population sizes (100, 200, 500 and 1024).
Here one can observe that when the population size is too small it does not reach the lowest cost. The solution just fluctuates without converging to a better solution. However, as the population size increases, the cost of the best solution moves towards a minimum. When the population size is at 1024, costs of the best solutions are converged to a lower value in about 50-60 generations.

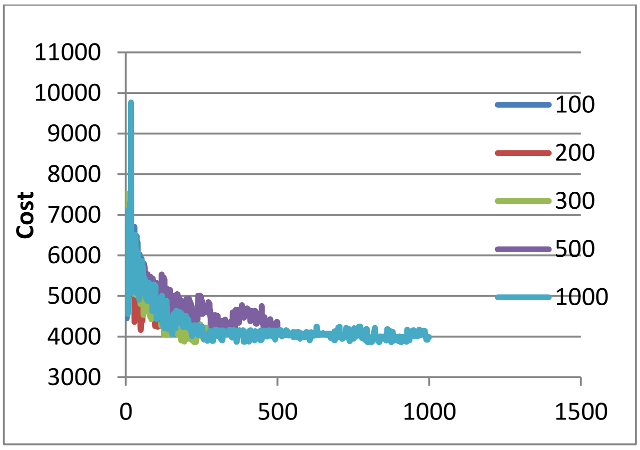

Fig. 8. Cost with different generation sizes

In figure 8 we considered the cost of the best solution for each generation for fixed population size of 100 different generation sizes (100, 200, 300, 500 and1000).

Here one can also observe that when the generation size becomes larger cost become lower. When the generation size is small, the algorithm does not have enough time to converge to an optimal solution. Normally solutions of the first generations are scattered throughout the solution space. It takes several generations to converge towards a better solution.

V. CONCLUSIONS

In this paper, we present a method to get an optimized MRP solution to a multi-level, multi-item dynamic lot sizing problem using GA when supplier discount/deals, Storage constraints and holding/transportation costs are considered in reaching the solution. The solution answers the problems of; from whom to buy when to buy and how much should be bought.
The experiments are done for small problems that normally converge to optimal solutions within few minutes. However, the real world industry level problems are complex and considerably huge. To reduce computational time, parallel programming is being considered in the implementation of the algorithm. The sorting algorithm is a simple insertion sort and can be replaced with a better one. Using GA to get to a solution could be more time consuming than using mathematical programming. However, mathematical programming cannot address highly constrained problems with supplier selection such as these (Huge solution space).
This method already addresses various cost constraints such as holding and transportation. Also, one can always expand the algorithm by adding more cost functions directly to the database without altering the algorithm. This method also can be used for ERP problems with some alterations.

REFERENCES

[1] L. Q. D.J Stockton, "MRP Lot Sizing Using Genetic Algorithms" BPICS CONTROL, 1993
[2] Qin Liao Wangyu Chen, "A New MRP Optimisation Algorithm Based on MultiObjective Genetic Evolution" ISKE, 2007
[3] Jinxing Xie And Jiefang Dong, "Heuristic Genetic Algorithms for General Capacitated Lot-Sizing Problems", Computers and Mathematics with Applications, 2002
[4] Chirawat Woarawichai, Kritsana Kuruvit, Paitoon Vashirawongpinyo, "Applying Genetic Algorithms for Inventory Lot-Sizing Problem with Supplier Selection under Storage Capacity Constraints" IJCSI, 2012
[5] B.J. McLaren "A Study of Multiple Level Lot Sizing Techniques for Material Requirements Planning Systems", Ph.D. Thesis, Purdue University, 1977
[6] http://www.parsifalsoft.com/examples/evalexpression/