SOFTWARE FOR MANUFACTURING CELL FORMATION: ISSUES AND EXPERIENCES

B Mahadevan
Indian Institute of Management, Bangalore, INDIA.
mahadev@iimb.ernet.in
and
G Srinivasan
Indian Institute of Technology, Madras, INDIA.
gsri@iitm.ac.in

Abstract

The Cell formation problem in Group Technology (GT)/ Cellular Manufacturing Systems (CMS) is being researched for the past forty years and many solution methodologies have been developed for solving this “hard” problem. One of the areas where there is little progress is in software development for cell formation. A software provides an opportunity to appropriately apply theoretical algorithms to meet practical requirements in cell formation.

In practice the degree of completion is more important than maximizing similarity of parts. Inter-cell moves have to be calculated in terms of percentage volume of work carried in other cells as against number of exceptions. It is also important that user has options such as number of cells, cell size and budget restriction (for purchasing new machines). Further it is necessary to provide unconstrained solutions that act as starting point in cell formation.

We have developed MASCOT, a windows based software for cell formation considering the above issues. It is currently undergoing tests using real-life data. This paper addresses various issues in cell formation to be considered in developing software and presents experiences on the development, testing and implementation of the software.

1. INTRODUCTION

Cellular Manufacturing is a popular methodology used in mid-volume mid-variety production systems to achieve the benefits of mass production in a batch environment. The main emphasis is on creating machine cells and product/component families such that each cell is independent of the other resulting in ownership and responsibility for the cell to make the product(s). The key benefits are improved quality, less lead-time for production, reduced set-up time and reduced material travel times.

Over the last forty years organizations have reported use of CMSs in reorganizing their production systems. They have also reported the benefits mentioned above (Burbidge, 1992, Wemmerlov and Hyer, 1989 and Wemmerlov and Johnson, 2000).

Many researchers have developed algorithms for cell formation and have tested them using real-life data. Most of these algorithms have used a binary machine-component incident matrix to represent the production situation. More recently authors and researchers have started using interval data in the form of production volumes, processing times and set up times to represent a production system that is reorganized into manufacturing cells (see for example Harhalakis et al., 1996, Lee and Chen, 1997 and Zhao and Wu 2000).

There have been a few studies where the authors discuss progressive, incremental and sequential cell formation where cells are formed progressively to make high yielding products (Marsh et al., 1999). There have been studies where Cellular Manufacturing systems have been classified into Product Oriented Plants, Manufacturing Oriented Plants and Turnover Oriented Plants depending on the volume-variety characteristics (Mahadevan, 1999).
While the theory and practice have advanced considerably in the field of cellular manufacturing independently, studies show that the algorithms developed by researchers are not widely used in practical cell formation (Marsh et al., 1999). However, some reports indicate the use of early algorithms such as Production Flow Analysis (Burbidge, 1977) in practical cell design.

We believe that amongst other things, non-availability of these algorithms in the form of a “user friendly” software for practical applications and not addressing several requirements of the practitioners could be the reasons for this phenomenon. Keeping these issues in mind, we have developed a software that accepts interval data and can be used for practical cell formation considering several alternatives and dimensions. This paper addresses the features of the proposed software and its use in cell formation.

The software for Cellular Manufacturing, discussed in this paper is called MAnufacturing Shop COnfiguration Tool (MASCOT). This is an interactive tool to restructure manufacturing systems using principles of GT/CMS. Although the software is primarily intended to design a CMS, we envisage alternative uses to the software. Some of them include the following:

- Split a manufacturing system into cells
- Check if existing cells in a manufacturing system are designed appropriately and if not to try alternative cell configurations
- Evaluate the usefulness of the software to a cell design problem by trying a sample cell design
- Educate potential production personnel on some issues related to cell design & operation
- Enable decision support by analyzing alternative scenarios pertaining to key design parameters such as investment and cell size

In the current version, the software can model up to 20 products, 200 machine types and 1000 components. These values are larger than most shops that we have come across in real life.

The software is designed to have powerful import and export features. Data entry could be done either through MASCOT or through MS Excel. Direct uploading of input data created using MS Excel into MASCOT is possible. All input data is stored in a format readable in MS Excel.

The software has several menus that cater to Data management, Algorithms for Cell Design, Output & Reports, Ageing Analysis and Decision Support. In addition, there are other standard features for file and directory maintenance. The software has been developed in VC ++ and runs in a Win 32 environment. We now discuss the salient features of the software in detail.

2. DATA MANAGEMENT

MASCOT views CMS design data in a hierarchical fashion consisting of three levels. At the highest level is the set of products being manufactured in an organization. The products may have a group of product lines each having some projected aggregated demand during the planning horizon for which the shop is to be designed. At the second level, there are several components. Each product is related to a set of components through a Bill of Material (BOM). At the last level there are machines of various types. Each component is related to the machine through a process plan. The process plan will indicate the order in which the machines are visited by the component and for each visit other details including process time, set-up time and batch quantities. The data management menu in MASCOT uses this as the basis for capturing data.

CMS design in real life situations often involves very large data sets (Miltenburg and Montazemi, 1993). Typically, large amount of information needs to be culled out of routing data, process plans, BOM and master files for key resources such as machines. Further, over time, changes in process plan, addition/deletion of parts due to outsourcing and design changes are common. Therefore data management is a critical requirement in any software for CMS design. Furthermore, interface with commonly used software such as MS office will enhance both ease of use and user acceptance.

The Data Management menu in MASCOT incorporates all these features. The data management menu offers the following features:

- User friendly screen for data capture
- Maintenance of data (add/delete/modify)
- Ease of import & export of data
- Data integrity checks
  - All input data verified before loading
  - Real-time error prevention routines to eliminate illogical entries

In the current version, the software can model up to 20 products, 200 machine types and 1000 components. These values are larger than most shops that we have come across in real life.

The software is designed to have powerful import and export features. Data entry could be done either through MASCOT or through MS Excel. Direct uploading of input data created using MS Excel into MASCOT is possible. All input data is stored in a format readable in MS Excel.

3. CELL DESIGN

Classical research in cell design in the last four decades has brought out several key insights in cell design. First is the issue of the grouping itself. Two schools of thought exist in identifying groups. One believes that groups exist naturally and it is the role of the cell designer to detect them (Burbidge, 1977). On the other hand, others believe that several practical considerations would influence the cell designer to pre-
determine the number of groups. The cell designer usually limits the number of cells or the cell size. In this case, a cell designer needs to devise methods to partition existing machines in the best possible manner (Boctor, 1991).

Second is the issue of goodness of design. A cell designer often needs to make a critical trade-off between utilization and independence. While independent cells (with no inter-cell moves) will offer numerous benefits arising from better planning and control, it can increase investment in additional machines. On the other hand, cells with better machine utilization may induce excessive inter-cell moves threatening to nullify the benefits of a CMS. Researchers have coined alternative measures such as grouping efficiency (Chandrasekharan, and Rajagopalan, 1989), and Quality Index, (Seifoddini and Djassemi,1996) to assess the cell design on the basis of inter-cell moves. Practitioners are not yet clear how to resolve this trade-off and may need to experiment with alternative weights of importance to these attributes.

Third, is the data structure employed in the cell design problem. Earlier surveys show that there has been a preference to use binary data by the researchers (Reisman et al., 1997, Mahadevan and Venkataramanaiah, 2003). However, use of complete production data is preferred to make meaningful cell designs.

In MASCOT, several of these considerations have been addressed. Algorithms that cater to both the schools of thought with respect to forming groups are included. For instance, cell designs with no restriction in cell size will cater to the first school of thought while designs involving some restrictions in cell size will cater to the later. Similarly, cell designers are given the opportunity to choose either cell independence or utilization or a suitable tradeoff of both these considerations through a relative weighing scheme. Finally, the entire production information captured in the data menu is put to use in the design of cells.

These considerations have increased the complexity of the problem and the solution and have denied the opportunity of using a vast number of solution methodologies already available. Nevertheless, they have minimized the level of abstraction in the modeling process and provided a tool in the hands of practitioners to study alternative cell designs.

The overall scheme of cell design used in MASCOT considers known algorithms such as PFA, algorithms based on own research and user defined logic. Further it also considers cell design restrictions pertaining to cell size (number of machines) and additional investment.

The “Algorithms” menu in MASCOT offers nine different algorithms for cell design arising out of three configurations and three options for machine capacity in each of these configurations. We explain them below

(a) Three types of configurations:

1. Individual Product Cells
2. User Defined Product Cells
3. Component Based Cells

In the individual product cells category, we create as many cells as the number of products. The user defined product cells create as many cells as the user desires. The user chooses the products that are to be included in each cell.

While designing component based cells, we have initially used binary data in Production Flow Analysis (Burbidge, 1977) to create initial modules. However, we have incorporated interval data and other details such as cell size restrictions to merge, split or synthesize initial modules into final cells.

The first two options represent POP (Product Oriented Plant) architecture and the third option MOP (Manufacturing operations Oriented Plant) architecture as defined in (Mahadevan, 1999).

When product based cells are formed, we may have few very large cells. These large cells have to be further decomposed to smaller cells due to cell size restrictions. Here, the issue of assigning machine with limited copies becomes significant. We use PFA and another algorithm proposed by Mahadevan and Venkataramanaiah (2002).

(b) Three options for machine capacity:

1. Cells with existing machine capacity
2. Cells with additional investment
3. Independent cells with no restriction on investment (no capacity constraints)

When the software creates cells with existing capacity it ensures that enough capacity is available in the system. The software progressively allocates machines to cells based on requirement until all available machines are exhausted.

When the software creates machine cells with restriction on additional investment, it first allocates available machines based on requirement (computed as machine loads after incorporating all relevant data such as unit processing time set uptime, batch size, bill of
material and production volume). After all machines are allotted, the algorithm assumes that the next most required machine is bought from the amount indicated by the user and progressively buys machines till all the money is exhausted.

When the software creates cells without restrictions, it assumes infinite resources and creates a system with no inter-cell moves. It also computes the number of additional machines required and the total investment. Clearly, options 1 and 3 in machine capacity represent the two extremes. Using alternative choices of investment restrictions, the CMS designer can in effect study a vast number of intermediary configurations using option 2.

In all these cell formation methods, the inter-cell moves are calculated as machine hours required in a different cell.

4. PERFORMANCE MEASURES

In practice, cell designers have considered alternative objectives for assessing the goodness of cell design. We have identified four frequently encountered objectives in CMS design:

1. Cost minimization
2. Cell independence
3. Flexibility
4. Ease of control

As discussed already, cost minimization and cell independence involve critical trade-offs in the choice of one design over the other. Measures such as additional investment required and total inter-cell moves are used to capture these two objectives. We also introduce a measure “product ownership” to assess cell independence from a product’s perspective. The percentage of the total processing (in hours) of a product in the various cells is the product ownership. The cell where the product has the maximum value of this measure is the parent cell for the product.

However, it is well known that cells age over time due to changes in product demand (Marsh, 1995). Flexibility objective will indicate how well cells could handle changes in production volume over time. MASCOT performs an ageing analysis on the basis of the input given by the user regarding demand changes in the planning horizon and computes an ageing index. Finally, ease of control is related to the number of cells and within cell and across cell variations in cell size and machine utilization.

The “Output” menu in MASCOT generates a set of reports indicating the performance of the chosen algorithms with respect to the four objectives mentioned above. The following outputs are available from MASCOT:

- Cell Design Details
  - Cell Design Logic
  - Machines in the cell
  - Products/Components in the cell
  - Product Ownership in the cell
- Measures - Cell level
  - Cell Size
  - Machine Utilisation (Max, Min, Avg.)
  - Inter-cell moves
- Measures – Overall level
  - Cell Size variations (Max, Min, Avg.)
  - Cell Load variations (Max, Min, Avg.)
  - Additional Investment
  - Total Inter-cell moves across cells
  - Ageing of cells

Among these measures “product ownership” has been specifically introduced in the software. We assume that in the case of product based cells, each cell acts as a profit centre and each product has to be owned by only one cell. We have deliberately used a time measure instead of a cost measure so that it can be generalized to any application.

5. DECISION SUPPORT FOR MANAGERS

The real strength of a software for CMS lies in its ability to provide the user scope for obtaining additional insights about the problem by performing a series of carefully designed experiments. Often, the user undergoes an intensive learning in this process and is better equipped with making informed and better decisions.

The “Decision Support” menu of MASCOT caters to this requirement. A user could perform additional experiments on three dimensions. First, she could rate a sub-set of the nine algorithms for their relative performance in any sub-set of the four objectives identified in section 4. For instance, she would like to rate six of the nine algorithms on their relative performance in cost, flexibility and ease of control. She would also identify the relative importance of these three objectives using a weighing scheme. On the basis of these inputs, MASCOT will analyze the six algorithms and provide a ranking.

Second, the CMS designer could perform a sensitivity analysis of the model parameters viz., cell size and additional investment. The designer will define a range of values for investment restriction and cell size restriction. MASCOT evaluates the algorithms for the user defined range of values in a pre-determined step
size and provides an output of the sensitivity of the chosen parameters. For example, if the user chooses a cell size restriction from 8 to 12, the algorithms are run for all the five values between 8 and 12 and results are given.

Finally, the user could perform an ageing analysis to assess the impact of future changes on the quality of the cell design. When ageing analysis needs to be performed, the user inputs the growth rate in demand for the products over the next six periods. Using these growth rates, MASCOT projects how well the current design accommodates these changes. It can indicate to the user when the present design will run out of capacity to meet the demand.

6. IMPLICATIONS FOR RESEARCH

Our experiences in developing a software for CMS design resulted in several implications for research. First is the issue of portability of research into practice. CMS design often involves handling a large set of data, ill-structured matrices, and numerous computations involving alternative algorithms. Hence, providing these functionalities through a user-friendly software will improve the portability of research into practice. In the absence of a software, users are not motivated to apply much of the research.

Developing computer-based applications for solving real-life CMS design problems involves several trade-offs. Miltenburg and Montazemi (1993) reported that even simple algorithms could not be directly used on account of large problem sizes. Developing modified versions of known algorithms and providing simple heuristics for grouping will prove to be very valuable. For instance, we could solve the problem initially with known algorithms such as PFA involving binary data and then refine the solutions with heuristics. In this process, we might have lost our ability to obtain optimal solutions as well as to judge how far the proposed solution is away from the optimal. One way to overcome this limitation is to provide features for generating a number of solutions.

Since good heuristics play an important role in cell design, software for CMS design should actively involve the cell designer in the cell formation process. For instance, the cell designer may be allowed to suggest alternative cell configurations and preferences for cell size and investment restrictions. These critical inputs could be incorporated into the cell design logic built into the software. It should also provide features for performing carefully designed experiments over the base case data.

7. CONCLUSIONS

Despite considerable work in CMS design, developing software for CMS design is an under-researched area. This paper reports the development of MASCOT, a Windows-based software for CMS design. The software provides a choice of nine algorithms for CMS design. It also permits the user to perform a set of carefully designed experiments for decision support. A researcher faces several trade-offs while developing software for CMS design. These include the choice of algorithms to be included, the need to obtain optimal solutions and the extent of user involvement in the cell design process. Nevertheless, a software for CMS design is likely to improve the portability of research into practice.

There are several areas that need further research in development of software for CMS design. For instance, linking the software with a simulation tool will help the designer study the efficacy of the solution in a dynamic setting and arrive at an appropriate layout. Further, the software could have features that will enable the user to study the impact of several short-term planning issues such as worker allocation on a chosen design. In the current study, other resources have not been considered for cell design. The most significant among them is labor. Incorporating these constraints in the design could be explored.

ACKNOWLEDGEMENTS

This research is funded by the Centre for Software & Information Technology Management at the Indian Institute of Management, Bangalore. We acknowledge the key role played by Mr. N Balaji in developing the software in VC++.

REFERENCES

5. HARHALAKIS, G., T. LU, I. MINIS, and R. NAGI, 1996, A practical method for design of


**Figure 2 – Parameters for rating.**

**Figure 3 - Algorithms for cell design in MASCOT**