AN INTEGER PROGRAMMING MODEL OF CELL FORMATION FOR CELLULAR MANUFACTURING

L. R. Foulds, Department of Management Systems, University of Waikato, New Zealand

K. Neumann, Institute of Economics and Operations Research, University of Karlsruhe, Germany

Abstract. It has long been recognized that productivity in manufacturing plants can often be increased by producing similar products in dedicated manufacturing cells. This involves: (i) assigning parts to individual machines and (ii) forming machines into cells. These two activities have traditionally been carried out separately. However most solution procedures for (i) above, utilize a solution to (ii), and vice versa. Here we present, for the first time, a unified integer programming model that deals with (i) and (ii) simultaneously.

1. INTRODUCTION

Over 70 years ago it was first proposed by Flanders that transportation could be reduced in factories by manufacturing similar items in product-oriented manufacturing departments. This theme was further developed in the then Soviet Union by Sokolovski and Mitrofanov who suggested that similar parts should be manufactured together by standardized operations. Skinner [1974] put forward his concept of a focused factory, in which small manufacturing systems operated independently within large production plants. This idea works best for medium-variety, medium-volume situations, i.e. batch production. The focused factory is constructed using the notion of group technology (GT), which is based on the precept that certain activities should be dedicated to a family of related parts in a manufacturing cell. Later Burbidge [1975] developed and popularized a systematic approach to this concept, which has subsequently seen widespread adoption through the advanced manufacturing countries of the world.

Locating machines in close proximity in a manufacturing cell, where a family of related parts are produced, usually results in a reduction of: transport requirements, conveyance times, setup times, and inventory. Moreover, the relatively large autonomy of a cell leads to extra motivation of the workers (who are responsible for "their products"), often resulting in higher productivity and product quality. These, and other advantages, have been discussed by Shunk [1985] and Hadley [1996].

There are two main types of cellular manufacturing systems : Group Technology (GT) and Flexible Manufacturing Systems (FMS). GT is related to FMS in so far as both are sub-systems which represent "islands" within the production process consisting of a group of machines (possibly including a material handling system and workers) which produce a family of items. The main difference between those two systems is that an FMS represents a fully automated system, whereas in GT, conventional technology generally predominates. Most of the recent major results in the cellular manufacturing literature have been surveyed by Chu [1995], Kusiak & Heragu [1987], Vakharia [1993], and Wemmerlov and Hyer [1987].

Suppose that a number of different products have to be manufactured using certain machine types. It is known from the process plans of the parts which machine types are required for producing the individual parts, and the routing (machine ordering) for each part is given. We wish to assign the different parts to the individual machines of the types required and to group the machines so that each group may form a dedicated manufacturing cell. This leads to several interesting problems, such as:

(a) Assign part families to groups of machine types,
(b) Find lot sizes of the parts produced,
(c) Determine the minimum number of machines needed of each machine type,
(d) Assign parts to individual machines,
(e) Form machine cells, and
(f) Compute job schedules for the machines.

The so-called machine type-part incidence matrix specifies which parts are to visit the different machine types. It is desirable that the machine type-part matrix should be transformed into a block-diagonal form to solve problem (a) (cf. Hadley [1996] or Kusiak and Chow [1988]). Each block then shows which family of parts is to be processed by which machine cell. This is reviewed in Section 2 of this paper.

If such a block-diagonal clustering cannot be obtained, problems (b) to (e) have to be solved. Well-known methods from inventory control can be used to solve problem (b). (See, for example, Neumann [1996] or Askin and Chiu [1990].) A method that includes specific information relevant to group technology has been proposed by Askin & Chiu [1990]. Given the lot sizes for all parts, we can compute the utilization of each machine type, which also provides the number of machines needed of each type, ie, the solution to problem (c). Problems (b) and (c) are discussed briefly in Section 3.

In the literature, problems (d), (e) and (f) are generally solved separately. (See, for example, Hadley [1996], Neumann [1996] or Askin and Chiu [1990.) Problems (d) and (e) are reviewed in Section 4. However, most solution procedures for problem (d) utilize a solution to problem (e) and vice versa. In Section 5 we present a unified approach that deals with problems (d) and (e) simultaneously.

In Section 6 we shall show that, given the solution to problems (d) and (e), the remaining job-shop problems can be solved using well-known methods from the literature (cf. Brucker [1995] and Pinedo [1995]). Section 7 summarizes our conclusions. We shall now briefly discuss these problems and sketch methods for solving them approximately.

2. FORMATION OF FAMILIES OF PARTS AND THEIR ASSIGNMENTS TO CELLS OF MACHINE TYPES

Assume that n parts, numbered 1, 2, ..., n are processed on m machine types : $M_1, M_2 ..., M_m$, which are to be grouped into cells. The information as to which parts are to visit the individual machine types is given by the so-called machine type-part matrix, with elements:

$$a_{ij} := 1, \text{ if part } j \text{ is processed on machine type } M_i, \text{ and}$$
$$0, \text{ otherwise}, i = 1, 2, ..., M_i; j = 1, 2, ..., n.$$

We attempt to reorder the machine type rows and part columns of the machine type-part matrix to obtain a block diagonal structure. The term "block diagonal" implies that we can partition the matrix such that the boxes on the main diagonal contain as many 1's as possible, but the off-diagonal boxes contain only 0's. If such a block diagonal structure is obtained, the items which correspond to columns of one block (constituting a family of parts) are processed only on those machine types which correspond to the rows of that block (group of machine types). Each block is a candidate for a cell. To order the rows and columns of the machine type-part matrix, we can use, for instance, the binary ordering algorithm described Neumann [1996].

3. COMPUTATION OF LOT SIZES AND MINIMUM NUMBER OF MACHINES

For part j let : d_j be the demand per period or unit of time, K_j be the setup cost, and h_j be the inventory holding cost per unit and period. If the demand does not vary much over time and the grouping of machines (that is, the forming of cells) does not affect the lot sizes very much, we can choose the *lot size* or *batch size* q_j to be the *economic order quantity*:

$$q_j = \sqrt{\frac{2K_jd_j}{h_j}}, j = 1,2,\ldots,n. \qquad (1)$$

For (1) we refer to Neumann [1996]. Askin & Chiu [1990] have modified the economic order quantity cost function by assuming that the total throughput time for a part is a multiple of the total processing time (including setup time) for a lot of that part.

Next, for the individual machine types: $M_1, M_2, \ldots, M_m$, we determine the minimum number of machines needed. Let C_i be the capacity of a machine of type M_i (measured by its running time including setup) available per period. Let s_{ij} be the required setup time per batch or lot of part j for a machine of type M_i (termed the M_i machine"), and let t_{ij} be the processing time (without setup time) for one unit of part j on an M_i machine. We do not consider conveyance times because they can be neglected within a single machine group forming a cell. The processing time p_{ij} of a job j (that is, a batch of size q_j of part j) on the M_j machine is then

$$p_{ij} = s_{ij} + q_jt_{ij}\ , i = 1, 2, ..., M_i; j = 1, 2, ..., n. \qquad (2)$$

The *utilization* u_{ij}, of machine type M_i by part j is given by

$$u_{ij} = \frac{\frac{d_j s_{ij}}{q_j} + d_j t_{ij}}{C_i},$$

$$i = 1, 2, ..., M_i; j = 1, 2, ...,n. \quad (3)$$

We set $u_{ij} = 0$ if part j is not processed on M_i. The numerator in (3) represents the running time of machine type M_i per period required for part j. If $u_{ij} > 1$, say, $1 < u_{ij} < 2$. we introduce a dedicated M_i machine to process part j, which will be assigned to the same cell as the first M_i machine that processes part j. Thus, we can assume, without loss of generality, that $u_{ij} \leq 1$. That is, one M_i machine is sufficient for processing part j.

Given the utilization u_{ij} of machine type M_i by part j ($i \leq j \leq n$) the *minimum number* μ_i, of M_i machines required for producing all parts, can be computed as:

$$\mu_i = \left\lceil \sum_{j=1}^{n} u_{ij} \right\rceil, i = 1, 2, ..., M_i; j = 1, 2, ...,n \quad (4)$$

where $\lceil c \rceil$ is the smallest integer greater than or equal to c (rounding up). The *average utilization* $\overline{u}_i$ of an M_i machine is

$$\overline{u}_i = \frac{\sum_{j=1}^{n} u_{ij}}{\mu_i}, i = 1, 2, ..., M_i; j = 1, 2, ...,n \quad (5)$$

We now go on to use the concepts we have just defined, in the assignment of parts to actual machines.

4. MACHINE-PART ASSIGNMENT AND GROUPING OF MACHINES

If part j is processed on machine type M_i and $\mu_i > 1$, we must specify on which of the μ_i machines of type M_i part j is to be manufactured. Thus, for each machine type M_i with $\mu_i > 1$, we have to solve *a machine-part assignment problem* whose objective function (to be minimized) should be a measure of the material flow or material handling cost between different cells. In other words, the solving of the machine-part assignment problem requires some preliminary knowledge of the solution to the machine grouping problem.

The machine-part assignment problem can be modelled as a *graph-partitioning problem* where the nodes of the graph correspond to the parts processed on machine type M_i and we seek to determine a minimum cost graph partition into μ_i subgraphs (cf. Neumann [1996]). If $\mu_i = 2$, and no limits are imposed on the number of nodes of the subgraphs, the graph-partitioning problem can be solved as a multi-terminal network in polynomial time (See, for example, Nagamochi & Ibaraki [1992].) Otherwise, the graph-partitioning problem is known to be NP-hard. In the machine-part assignment problem there is a maximum number of parts which can be processed on a single machine of type M_i due to the limited capacity of that machine. Thus, the corresponding graph-partitioning problem is NP-hard for any $\mu_i \geq 2$.

To reduce the computational effort for solving the graph-partitioning problem, it is recommended to use some heuristic method, for example, the Kernighan-Lin heuristic (Kernighan & Lin [1970]), preceded by some partition construction procedure (cf. Neumann [1996] or Askin and Chiu [1990].)

The *construction procedure* determines an assignment of the parts to the machines so that machine capacities are not exceeded. Consider a machine type M_i with $\mu_i > 1$ let $J_i \subseteq \{1,2,..., n\}$ be the set of parts processed on machine type M_i. To construct a feasible solution, assign the parts $j \in J_i$ successively to the first M_i machine, with capacity available. If the capacity of the first M_i machine is exceeded, we use a second machine, and proceed analogously. Only if none of the μ_i machines of type M_i have sufficient capacity to process the entire operation for some part j, can the operation be divided into partial operations on several machines.

The solution to the assignment problem for all machine types can be summarized in a *machine-part incidence matrix* with elements:

b_{rj}	=	if part j is assigned to actual machine number r,
	=	0, otherwise.

where the actual machine number r stands for some machine M_i^k $(1 \leq k \leq \mu_i)$.

Given a machine-part incidence matrix, the *machine grouping problem* (that is, the formation of cells) can be solved in a number of different ways. Kusiak & Chow [1998], and others, have devised decomposition procedures for the machine-part matrix. Methods proposed by Neumann [1996], or Askin and Chiu [1990], and others, are based upon a machine graph. Each edge [r, s], is assigned a positive value if some part j, has to be processed on both machines r and s and if the machine sequence (or routing) of part j contains the subsequence (r, s) or (s, r). That is, if part j is moved from machine r to machine s, or vice versa. The weight of edge [r, s] is assigned a value equal to the sum of the demand rates of the parts moved from machine r to machine s or vice versa. Again, a graph-partitioning problem can be formulated and solved by the methods mentioned above, where the resulting subgraphs correspond to the cells.

We now present a unified approach, in the form of a single model, for the formation of cells which incorporates into one model, the assignments of parts to individual machines and the grouping of machines into cells.

5. A UNIFIED APPROACH TO THE FORMATION OF CELLS

We now develop a multicommodity, min-cost, network circulation flow model for the formation of cells. The nodes in the network in the model can be classified in the following way. The first set of nodes: $P_1, P_2, ..., P_n$ represent the n parts. The second set of nodes $M_1, M_2, ..., M_m$ represent the m machine types. An arc is created to join directly node P_j to node M_i if, and only if, $a_{ij} = 1$. The capacity of this arc is set as $d_{ij} = 1$, with unit cost set as $c_{ij} = 0$.

The third set of nodes: $M_1^1, M_1^2, \ldots, M_m^{\mu_m}$ represent the actual machines themselves, with node M_i^k representing the k^{th} machine of type M_i (termed the M_i^k machine) for i = 1, 2,..., m; k = 1, 2,..., μ_i. For each i, i = 1, 2, ..., m, an arc is created to join directly node M_i to each node M_i^k. That is, the node for each machine of type M_i, is joined to the nodes representing the M_i^k machines. The capacity of the arc from M_i to M_i^k, is set as $d_i^k = 1$, with unit cost set as $c_i^k = 0$.

The fourth set of nodes: $C_1, C_2, ..., C_p$, represent the cells that can be formed to group the machines into (up to) p cells. As p is unknown, the following estimate $\hat{p}$, of p is used:

$$\hat{p} = \left\lceil (1/\overline{u}) \sum_{i=1}^{m} \mu_i \right\rceil, \ i = 1, 2, ..., M_i \tag{6}$$

where

$$\overline{\mu} = \left\lceil \frac{(\mu_{min} + \mu_{max})}{2} \right\rceil. \tag{7}$$

An arc is created to join directly each node M_i^k to each node C_q i = 1, 2, ..., m; k = 1, 2, ..., μ_i; and q = 1, 2, ..., p. That is, the node of each actual machine is connected to the node of each cell, allowing for the possibility that any machine can be allocated to any cell. The capacity of the arc, connecting node M_i^k to node C_q is set as $d_i^{kq} = \infty$, with unit cost set as $c_j^q = f(L_j, K_q)$, where L_j = the inter-cell materials handling cost for part j, K_q = the set up cost for the q^{th} cell, and $f(L_j, K_q)$ = the combined unit cost incurred, related to L_j and K_q for a single part j being assigned to cell q.

The endeavour to minimize the cost of the travel of all flow in this network model is subject to constraints of two types:

(i) **Classical network flow constraints:**
- Conservation of flow at all nodes,
- Arc capacity limits must be observed, and
- Nonnegative flow on all arcs.

(ii) **Side constraints:**
- After any dedicated machines have been removed, each part must be assigned to exactly one actual machine of each type.
- Each actual machine must be assigned to exactly one cell,
- Each cell must be allocated a number of machines between a given lower and upper bound, and
- Logical links between the utilization variables and the assignment variables.

In order to formulate the necessary constraints we first define some notation:

Given parameters:

μ_{ij} = the utilization of machine type M_i by part j, (the demand at node P_j.),

Utilization decision variables:

x_{ij}^{k} = the utilization of machine type M_1 by part j that is assigned to the actual machine M_i^k,

x_{ij}^{kq} = the utilization of actual machine M_i^k by part j that is assigned to the cell q, and

x_j^q = the total utilization, over all machines, in cell q of part j.

Assignment decision variables:

$$y_{ij}^{kq} = \begin{cases} 1 & \text{if part j is assigned to machine } M_i^k \text{ in cell q,} \\ 0 & \text{otherwise.} \end{cases}$$

The conservation of flow constraints are:

$$\sum_{j=1}^{n}\sum_{k=1}^{\mu_i} x_{ij}^{k} = \sum_{j=1}^{n} u_{ij},\ i = 1,2,\ldots,m.$$

(Conservation of flow at each M_1 node.)

$$\sum_{j=1}^{n}\sum_{q=1}^{p} x_{ij}^{kq} = \sum_{j=1}^{n} x_{ij}^{k},\ i = 1,2,\ldots,m;\ k = 1,2,\ldots,\mu_i$$

(Conservation of flow at each M_i^k ,.. node.)

$$\sum_{j=1}^{n} x_j^q = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{\mu_i} x_{ij}^{kq}\ ,\ q = 1,2,\ldots,p\,.$$

(Conservation of flow at each C_q node.)

$$\sum_{q=1}^{p} x_j^q = \sum_{i=1}^{m} \mu_{ij},\quad j = 1,2,\ldots,n\,.$$

(The total machine utilization in all cells equals the total machine utilization of all parts.)

$$\sum_{j=1}^{n} x_j^k \le 1,\ i = 1,2,\ldots,m\ ;\ k = 1,2,\ldots,\mu_i\,.$$

Each machine M_i^k, cannot be over-utilized.)

The nonnegativity conditions are:

$$x_{ij}^{k} \ge\ 0,\ i = 1, 2, \ldots, m\ ;\ j = 1, 2, \ldots, n\ ;$$
$$k = 1, 2, \ldots, \mu_i\,,$$

$$x_{ij}^{kq} \ge\ 0,\ i = 1, 2, \ldots, m\ ;\ j = 1, 2, \ldots, n\ ;$$
$$k = 1, 2, \ldots, \mu_i\,,\ ;\ q = 1, 2, \ldots, p.$$

The side constraints are:

$$\sum_{q=1}^{p}\sum_{k=1}^{\mu_i} y_{ij}^{kq} = 1,\ i = 1,2,\ldots,m\ ;\ j = 1,2,\ldots,n.$$

(After any dedicated machines have been removed, each remaining unassigned part must be assigned to exactly one actual machine of each type. That is, its assignment cannot be split among more than one actual machine of the same type.)

$$\sum_{q=1}^{n}\sum_{q=1}^{p} y_{ij}^{kq} = 1,\ i = 1,2,\ldots,m\ ;\ k = 1,2,\ldots,\mu_i\,.$$

(Each actual machine must be assigned to exactly one cell.)

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{\mu_i} y_{ij}^{kq} \ge \mu_{min}\ ,\ q = 1,2,\ldots,p\,.$$

(Each cell actually created must contain no less than a given, minimum number of machines)

$$x_{ij}^{kq} \le y_{ij}^{kq}\ ,i = 1,2,\ldots,m; j = 1,2,\ldots,n;$$
$$k = 1,2,\ldots,\mu_i\ ;q = 1,2,\ldots,p.$$

(If a part j is not assigned to the actual machine M_i^k then its utilization of that machine must be zero.)

$$0 \le x_{ij}^{k} \le 1, i = 1,2,\ldots, m; j = 1,2,\ldots, n;$$
$$k = 1,2,\ldots, \mu_i$$

$$y_{ij}^{kq} = 0,1, \text{for } i = 1,2,\ldots,m; j = 1,2,\ldots,n;$$
$$k = 1,2,\ldots,\mu_i\ ; q = 1,2,\ldots,p$$

(The total assignment of parts to each individual machine must not exceed its capacity of one unit. Either a part j is assigned to the k^{th} individual machine of type i, which is in cell q, or it is not.)

The objective is to

Minimize

$$\sum_{j=1}^{n} \sum_{q=1}^{p} f\left(L_j, K_q\right) x_i^q .$$

subject to the preceding constraints.

This model can be solved for practical numerical instances by the techniques for multicommodity network flow models with side constraints given by, among others, Ahuja et al. [1993].

6. JOB-SHOP SCHEDULING

After the formation of cells, some *job-shop problems* have to be solved. A job corresponds to a lot of some part. For each set of cells with some inter-cell material flow (briefly called a *cell system*), the makespan, that is, the maximum completion time of all jobs, is to be minimized. We seek to determine the job sequence for each machine of the cell system and the job schedules, which specify the start and completion times of the jobs. These job-shop problems can be solved by well-known methods (cf. Brucker [1995] and Pinedo [1995]).

7. SUMMARY AND CONCLUSIONS

We have reviewed the issues in the formulation of cellular manufacturing cells of: (i) assigning parts to individual machines and (ii) forming individual machines into cells. We have presented an integer programming model which combines these two activities in one model for the first time. We believe that the resulting multicommodity network flow model that results will become a useful tool for production planners.

References

- R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows*, Prentice Hall, Englewood Cliffs, 1993.
- R.G. Askin and K.S. Chiu, A graph partitioning procedure for machine assignment and cell formation in group technology, *Int. J. Prod. Res.*, **28**, 1555 - 1572, 1990.
- J.L. Burbidge, *The Introduction of Group Technology*, John Wiley, New York, 1975.
- P. Brucker, *Scheduling Algorithms*, Springer, Berlin, 1995.
- C.H. Chu, Recent advances in mathematical programming for cell formation, in : *Planning, Design and Analysis of Cellular Manufacturing Systems* A.K. Kamrani, H.R. Parasari, D.H. Liles (eds), Elsevier, Amsterdam, pp 3-46, 1995.
- S.W. Hadley, Finding part-machine families using graph partitioning techniques, *Int. J. Prod. Res.*, **34**, 1821-1839, 1996.
- B.W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, *Bell Systems Technical Journal*, **49**, 291 - 307, 1970.
- A. Kusiak and W.S. Chow, Decomposition of Manufacturing Systems, *J. Robotics and Automation*, **4**, 457 - 471, 1988.
- Kusiak and S.S. Heregu, Group Technology, *Computers and Industry*, **9**, 83-91, 1987.
- H. Nagamochi and T. Ibaraki, Computing Edge-connectivity in Multigraphs and Capacitated Graphs, *SIAM J. Disc. Math.*, **5**, 54 - 66, 1992.
- K. Neumann, *Produktions-und Operations-Management*, Springer, Berlin, 1996.
- M. Pinedo, *Scheduling*, Prentice -Hall, New Jersey, 1995.
- W. Skinner, The Focused Factory, *Harvard Business Review*, May – June, 113 – 121, 1974.
- D.L. Shunk, Group Technology provides an oerganized approach to realized benefits of CIM's, *Industrial Engineering*, **17**, (1985) 74-81, 1985.
- Vakharia, Methods of cell formation in group technology: a framework for evaluation, J. *Operations Management*, **6**, 257-271, 1993.
- U. Wemmerlov and N.L. Hyer, Research issues in cellular manufacturing, *Int. J. Prod. Res.*, **25** , 412-422, 1987.