DESIGN OF THE CELL SYSTEM OF PRODUCTION

BY

GAJENDRA KUMAR ADIL

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Industrial Engineering
University of Manitoba
Winnipeg, Manitoba

ISBN 0-612-12946-2

Name GAJENDRA KUMAR ADIL

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

INDUSTRIAL ENGINEERING — SUBJECT TERM

0546 — SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

Nom ____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.

____ ☐☐☐☐ U·M·I

SUJET CODE DE SUJET

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS

Architecture ... 0729
Beaux-arts ... 0357
Bibliothéconomie ... 0399
Cinéma ... 0900
Communication verbale ... 0459
Communications ... 0708
Danse ... 0378
Histoire de l'art ... 0377
Journalisme ... 0391
Musique ... 0413
Sciences de l'information ... 0723
Théâtre ... 0465

ÉDUCATION

Généralités ... 515
Administration ... 0514
Art ... 0273
Collèges communautaires ... 0275
Commerce ... 0688
Économie domestique ... 0278
Éducation permanente ... 0516
Éducation préscolaire ... 0518
Éducation sanitaire ... 0680
Enseignement agricole ... 0517
Enseignement bilingue et multiculturel ... 0282
Enseignement industriel ... 0521
Enseignement primaire. ... 0524
Enseignement professionnel ... 0747
Enseignement religieux ... 0527
Enseignement secondaire ... 0533
Enseignement spécial ... 0529
Enseignement supérieur ... 0745
Évaluation ... 0288
Finances ... 0277
Formation des enseignants ... 0530
Histoire de l'éducation ... 0520
Langues et littérature ... 0279
Lecture ... 0535
Mathématiques ... 0280
Musique ... 0522
Orientation et consultation ... 0519
Philosophie de l'éducation ... 0998
Physique ... 0523
Programmes d'études et enseignement ... 0727
Psychologie ... 0525
Sciences ... 0714
Sciences sociales ... 0534
Sociologie de l'éducation ... 0340
Technologie ... 0710

LANGUE, LITTÉRATURE ET LINGUISTIQUE

Langues
- Généralités ... 0679
- Anciennes ... 0289
- Linguistique ... 0290
- Modernes ... 0291

Littérature
- Généralités ... 0401
- Anciennes ... 0294
- Comparée ... 0295
- Médiévale ... 0297
- Moderne ... 0298
- Africaine ... 0316
- Américaine ... 0591
- Anglaise ... 0593
- Asiatique ... 0305
- Canadienne (Anglaise) ... 0352
- Canadienne (Française) ... 0355
- Germanique ... 0311
- Latino-américaine ... 0312
- Moyen-orientale ... 0315
- Romane ... 0313
- Slave et est-européenne ... 0314

PHILOSOPHIE, RELIGION ET THÉOLOGIE

Philosophie ... 0422
Religion
- Généralités ... 0318
- Clergé ... 0319
- Études bibliques ... 0321
- Histoire des religions ... 0320
- Philosophie de la religion ... 0322

Théologie ... 0469

SCIENCES SOCIALES

Anthropologie
- Archéologie ... 0324
- Culturelle ... 0326
- Physique ... 0327

Droit ... 0398
Économie
- Généralités ... 0501
- Commerce-Affaires ... 0505
- Économie agricole ... 0503
- Économie du travail ... 0510
- Finances ... 0508
- Histoire ... 0509
- Théorie ... 0511

Études américaines ... 0323
Études canadiennes ... 0385
Études féministes ... 0453
Folklore ... 0358
Géographie ... 0366
Gérontologie ... 0351
Gestion des affaires
- Généralités ... 0310
- Administration ... 0454
- Banques ... 0770
- Comptabilité ... 0272
- Marketing ... 0338

Histoire
- Histoire générale ... 0578
- Ancienne ... 0579
- Médiévale ... 0581
- Moderne ... 0582
- Histoire des noirs ... 0328
- Africaine ... 0331
- Canadienne ... 0334
- États-Unis ... 0337
- Européenne ... 0335
- Moyen-orientale ... 0333
- Latino-américaine ... 0336
- Asie, Australie et Océanie ... 0332

Histoire des sciences ... 0585
Loisirs ... 0814
Planification urbaine et régionale ... 0999
Science politique
- Généralités ... 0615
- Administration publique ... 0617
- Droit et relations internationales ... 0616

Sociologie
- Généralités ... 0626
- Aide et bien-àtre social ... 0630
- Criminologie et établissements pénitentiaires ... 0627
- Démographie ... 0938
- Études de l' individu et de la famille ... 0628
- Études des relations interethniques et des relations raciales ... 0631
- Structure et développement social ... 0700
- Théorie et méthodes ... 0344
- Travail et relations industrielles ... 0629

Transports ... 0709
Travail social ... 0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES

Agriculture
- Généralités ... 0473
- Agronomie. ... 0285
- Alimentation et technologie alimentaire ... 0359
- Culture ... 0479
- Élevage et alimentation ... 0475
- Exploitation des péturages ... 0777
- Pathologie animale ... 0476
- Pathologie végétale ... 0480
- Physiologie végétale ... 0817
- Sylviculture et faune ... 0478
- Technologie du bois ... 0746

Biologie
- Généralités ... 0306
- Anatomie ... 0287
- Biologie (Statistiques) ... 0308
- Biologie moléculaire ... 0307
- Botanique ... 0309
- Cellule ... 0379
- Écologie ... 0329
- Entomologie ... 0353
- Génétique ... 0369
- Limnologie ... 0793
- Microbiologie ... 0410
- Neurologie ... 0317
- Océanographie ... 0416
- Physiologie ... 0433
- Radiation ... 0821
- Science vétérinaire ... 0778
- Zoologie ... 0472

Biophysique
- Généralités ... 0786
- Medicale ... 0760

SCIENCES DE LA TERRE

Biogéochimie ... 0425
Géochimie ... 0996
Géodésie ... 0370
Géographie physique ... 0368
Géologie ... 0372
Géophysique ... 0373
Hydrologie ... 0388
Minéralogie ... 0411
Océanographie physique ... 0415
Paléobotanique ... 0345
Paléoécologie ... 0426
Paléontologie ... 0418
Paléozoologie ... 0985
Palynologie ... 0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT

Économie domestique ... 0386
Sciences de l'environnement ... 0768
Sciences de la santé
- Généralités ... 0566
- Administration des hipitaux ... 0769
- Alimentation et nutrition ... 0570
- Audiologie ... 0300
- Chimiothérapie ... 0992
- Dentisterie ... 0567
- Développement humain ... 0758
- Enseignement ... 0350
- Immunologie ... 0982
- Loisirs ... 0575
- Médecine du travail et thérapie ... 0354
- Médecine et chirurgie ... 0564
- Obstétrique et gynécologie ... 0380
- Ophtalmologie ... 0381
- Orthophonie ... 0460
- Pathologie ... 0571
- Pharmacie ... 0572
- Pharmacologie ... 0419
- Physiothérapie ... 0382
- Radiologie ... 0574
- Santé mentale ... 0347
- Santé publique ... 0573
- Soins infirmiers ... 0569
- Toxicologie ... 0383

SCIENCES PHYSIQUES

Sciences Pures

Chimie
- Genéralités ... 0485
- Biochimie ... 487
- Chimie agricole ... 0749
- Chimie analytique ... 0486
- Chimie minérale ... 0488
- Chimie nucléaire ... 0738
- Chimie organique ... 0490
- Chimie pharmaceutique ... 0491
- Physique ... 0494
- PolymÇres ... 0495
- Radiation ... 0754

Mathématiques ... 0405
Physique
- Généralités ... 0605
- Acoustique ... 0986
- Astronomie et astrophysique ... 0606
- Electronique et électricité ... 0607
- Fluides et plasma ... 0759
- Météorologie ... 0608
- Optique ... 0752
- Particules (Physique nucléaire) ... 0798
- Physique atomique ... 0748
- Physique de l'état solide ... 0611
- Physique moléculaire ... 0609
- Physique nucléoire ... 0610
- Radiation ... 0756

Statistiques ... 0463

Sciences Appliqués Et Technologie

Informatique ... 0984
Ingénierie
- Généralités ... 0537
- Agricole ... 0539
- Automobile ... 0540
- Biomédicale ... 0541
- Chaleur et ther modynamique ... 0348
- Conditionnement (Emballage) ... 0549
- Génie aérospatial ... 0538
- Génie chimique ... 0542
- Génie civil ... 0543
- Génie électronique et électrique ... 0544
- Génie industriel ... 0546
- Génie mécanique ... 0548
- Génie nucléaire ... 0552
- Ingénierie des systömes ... 0790
- Mécanique navale ... 0547
- Métallurgie ... 0743
- Science des matériaux ... 0794
- Technique du pétrole ... 0765
- Technique minière ... 0551
- Techniques sanitaires et municipales ... 0554
- Technologie hydraulique ... 0545

Mécanique appliquée ... 0346
Géotechnologie ... 0428
Matières plastiques (Technologie) ... 0795
Recherche opérationnelle ... 0796
Textiles et tissus (Technologie) ... 0794

PSYCHOLOGIE

Généralités ... 0621
Personnalité ... 0625
Psychobiologie ... 0349
Psychologie clinique ... 0622
Psychologie du comportement ... 0384
Psychologie du développement ... 0620
Psychologie expérimentale ... 0623
Psychologie industrielle ... 0624
Psychologie physiologique ... 0989
Psychologie sociale ... 0451
Psychométrie ... 0632

DESIGN OF THE CELL SYSTEM OF PRODUCTION

BY

GAJENDRA KUMAR ADIL

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Abstract

An important design problem encountered in the implementation of the cell system of production is cell formation. Cell formation consists of identifying part families and machine groups. The basic cell formation procedure consists of rearranging the part machine matrix in a block diagonal form so that the part families and machine groups can be easily identified. A perfect decomposition of the part machine matrix to form exclusive cells (no inter cell movement) is not possible in most manufacturing situations. However, by considering alternate process plans and additional units of same machines as available, the groupability can be enhanced. A motivating factor for introducing a cell system is reduction in material handling. As the number of cells increases the within cell or intra cell material handling decreases but the inter cell material handling increases. By balancing the intra and inter cell material handling costs, it is possible to determine the optimal number of cells and cell sizes. Moreover, the material handling depends on production quantity, sequence of operations of parts and multiple visits to the same machine. In process industries involved in repetitive manufacturing, the parts are processed by the same set of machines and in the same order. Cell formation in this case is affected by the investment and operational costs. In this thesis, mathematical models for cell formation are developed progressively considering the issues discussed above. The models simultaneously identify part families and machine groups and do not require any manual intervention. By varying the weights in the model the designer can form large loose cells or small tight cells. For

the efficient solution of large problems iterative and simulated annealing based solution procedures are developed. The results obtained from these procedures compare favourably with the existing procedures. The iterative solution procedure is very simple and less computer intensive; and large problems of 400 parts and 240 machines are solved in less than a minute on a Sun Sparc 2 station. Simulated annealing procedure gives more consistent results but require considerably higher computation time. It is recommended that simulated annealing procedure be used for solving smaller problems. For larger problems, the iterative procedure should be used a few times with different initial machine assignments. The integer programming models developed in the thesis can be solved using commercial softwares.

Acknowledgements

I express my deep sense of gratitude and appreciation to my thesis supervisors Dr. D. Rajamani and Dr. D. Strong for their inspiring guidance, valuable suggestions, constructive criticisms and constant encouragement during various stages of the research. My thanks are also due to my thesis committee members Dr. C. R. Bector and Dr. S. Balakrishnan for their time, effort and suggestions. I am thankful to Dr. S. Sankaran from University of Regina for being an external examiner.

I wish to acknowledge the financial supports provided by the University of Manitoba in the form of a graduate fellowship and Natural Science and Engineering Research Council of Canada in the form of a research assistantship.

I am thankful to my friends for encouragement and company which has made my stay at Winnipeg pleasant and memorable.

Finally, I would like to dedicate this dissertation to my parents and other family members whose confidence and expectation have been a constant source of support in my career.

Table of Contents

List of Tables

List of Figures

Nomenclature

Indices:

(i) Cells:

$c = 1$ to C

$c = 1$ to n

(ii) Exceptional elements:

$e = 1$ to e_p

(iii) Machines:

$f = 1$ to f_m copy

m & $m' = 1$ to M type

(iv) Operations:

$o = 1$ to O_p

(v) Parts:

$i = 0$ to n

j & $j' = 1$ to n

$l = 1$ to $(n + 1)$

$p = 1$ to P

(vi) Process plans:

$r = 1$ to R_p

(vii) Stage:

$s = 1$ to k

Parameters:

$$a_{pm} = \begin{cases} 1 & \text{if part } p \text{ requires processing on machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$a^{r}_{pm} = \begin{cases} 1 & \text{if part } p \text{ requires processing on machine } m \text{ in process plan } r \\ 0 & \text{otherwise} \end{cases}$$

a_j = arrival time of part j

B = a large positive number

b^0_{mc} = total capacity available in machine m in cell c at the beginning of the assignment

b_m = capacity of a copy of machine m

C = upper limit on number of cells

C^c = cost of cell c

$C^s_{jj'}$ = set-up cost for changeover from part j to j' at stage s

$$\begin{matrix} c^o_{pmm'} \\ (o > 1) \end{matrix} = \begin{cases} 1 & \text{if two consecutive operations, } (o-1) \text{ \& } o, \text{ of part } p \text{ are performed} \\ & \text{on machine types } m \text{ and } m' \text{ respectively } (m \neq m') \\ 0 & \text{otherwise} \end{cases}$$

d_j = due date of part j

HCB_p = inter cell handling cost of part p

HCB^c_p = the unit inter cell handling cost of part p in cell c. This is defined for the cells other than the part's parent cell. For parts in the parent cell it takes a value 0.

HCP^c_p = the unit intra cell handling cost of part p in cell c. This is defined only for the part's parent cell. For other cells it takes a value 0.

$\mathrm{HCW}_p(\mathrm{N}^c)$ = handling cost within the cell (intra cell move cost) for part p in cell c, as a function of the number of machines in the cell c, N^c

$\Delta \mathrm{HCW}_p^c$ = increase in the intra cell move cost for part p in cell c if one machine is added to the cell

h_j^s = inventory holding cost per unit time for part j after processing it at stage s

M = number of machine types

N = capacity level of processor

N_m= number of copies of machine type m available

N_0^c = number of machines in cell c before assignment

n = number of parts and is also specified as the maximum number of cells

P = number of parts

Q_p = quantity of part p produced

$S_{jj'}^s$ = set-up time for changeover from part j to j' at stage s

t_{pm} = total time required by part p on machine m for one or more operations.

t_{pm}^e = time required to process exceptional element e of part p on machine m.

U^s = cost to increase the capacity level of stage s to maximum

w = fraction representing the weight on exceptional elements $(0 \leq w \leq 1)$

$(1-w)$ = fraction representing the weight on voids

w_{pm} = fraction representing the weight on exceptional element corresponding to part p and machine m $(0 \leq w_{pm} \leq 1)$

$(1-w_{pm})$ = fraction representing the weight on void corresponding to part p and machine m

α_j = early finish penalty cost per unit time for part j

$$\alpha_p^e = \begin{cases} 2 & \text{if exceptional element } e \text{ of part } p \text{ corresponds to an intermediate operation} \\ 1 & \text{if exceptional element } e \text{ of part } p \text{ corresponds to the first or the last operation} \end{cases}$$

β_j = late finish penalty cost per unit time for part j

γ^s = machine idle time penalty per unit time at stage s

Chapter 1

Introduction

A traditional manufacturing system employs a process layout in which each department consists of a group of functionally similar machines such as lathe, drill, etc. (Figure 1.1 (a)). Parts must be moved from department to department in order to perform the required operations. This system provides flexibility to process a large variety of parts. However, scheduling, sequencing, material handling and overall managerial control of such a system become difficult because of the size and complexity involved. Moreover, although an individual part may use only a small subset of machines, it spends a large amount of time for completion of its processing due to excessive material handling and waiting (Figure 1.1 (c)).

1.1 The Cell System of Production

The cell system of production or cellular manufacturing (CM) is a system where efficiencies in production are attained by suitably decomposing a larger system into smaller subsystems or cells by exploiting similarities among the parts (Shafer and Roger, 1991). In addition to the simplification of management through the creation of smaller subsystems, CM also leads to reduced material handling, reduced setup time, reduced work-in-process, reduced throughput time and improved sequencing and scheduling on the shop floor (Askin et al., 1991).

1.2 Design of the Cell System of Production

Designing a cell system is a complex undertaking with broad implications for the organization. The number of decisions in a system design process are numerous. They are related to system structure and operations. Decisions related to system structure include part types to be processed in the cell, the type and the number of machines on which these parts are processed, the type and number of material handling equipment, the routings, the system layout, tools, and fixtures. Decisions related to operational procedures include job design, supervisory and support personnel, and production planning and control, (Wemmerlov and Hyer, 1986). Structure related decisions often precede the operational decisions. Cell formation (CF), a structure-related decision process, takes on a special significance since most subsequent decisions are influenced by this choice (Wemmerlov and Hyer, 1986). The focus of this thesis will be on issues related to cell formation (CF).

1.2.1 Cell Formation

Cell formation (CF) consists of identifying part families (PFs) and machine groups (MGs) or cells, such that the part families are processed with minimum interaction with other cells. If the problem is reorganizing the existing facilities, information on machine requirements for each part type can be obtained from the routing cards. This information is often summarized in the form of a binary part machine matrix. In the part machine matrix a, an element a_{pm} is 1 if part p requires machine m for processing, otherwise it is 0 (or blank). Figure 1.1(b) shows a part machine matrix for the 6 parts and 5 machine types system shown in Figure 1.1(a). In Figure 1.1(b), all units of the same type of machine are lumped into one, and the sequence of visits is ignored. Let us consider the cell formation for this given situation. If we wish to form two cells we can do it in a number of ways. For instance, assume that the first three parts and three machines are assigned to the first cell and the remaining to the second cell. The decomposed system and the corresponding part machine matrix are shown in Figure 1.2. In Figure 1.2(b) each diagonal block represents a cell. The cells thus formed indicate that except for part 1 the other parts are required to visit both the cells. This interaction is shown by an entry 1, outside the diagonal blocks and the corresponding operations are called exceptional elements. Also, not all the machines in a cell are used by all the parts assigned to the cell. This occurrence is shown by a 0 entry inside the diagonal blocks referred to as voids (0s outside diagonal blocks are not voids). In an unpartitioned matrix shown in Figure 1.1(b), all parts and machines belong to one cell (block). The number of voids is 15 with no exceptional elements. Thus, we observe that as the manufacturing system is decomposed the number of exceptional elements will increase with a decrease in voids. Rather than decomposing the system arbitrarily as described above, rearranging the rows and columns of the part machine matrix to obtain a block diagonal form leads to identification of better part families and machine groups. Figure 1.3 shows a decomposition of the same system using this approach. In this rearranged matrix, parts: 1, 4 and 5, and machines: saw, grinding machine, and lathe are in cell 1; and parts: 2, 3, and 6 and machines: milling and drill are in cell 2. The number of voids and exceptional elements are 2 each, which is less as compared to 7 each in Figure 1.2. Therefore, a number of heuristic procedures have been proposed to obtain a block diagonal form of the matrix and by manual intervention identification of part families and machine groups such that the number of exceptional elements is minimum.

1.2.2 Alternate Routings

Most manufacturing situations are such that a perfect decomposition of part machine matrix with no voids and exceptional elements is not possible. Two possible approaches to improve the groupability of the data represented by the part machine matrix are by considering alternate process plans for parts and considering additional units of machines as available (Kusiak and Cho, 1992). Alternate process plans are frequently available for the parts (Kusiak, 1987), and in many instances, additional units of the same machine type are also available (King and Nakornchai, 1982). To illustrate the improvement, consider Figure 1.3. If for part 2, the operation performed on the lathe can be re-scheduled on the milling machine, one exceptional element will be reduced. In the same example two units of drills are available. If one unit of drilling machine to each cell is assigned, it will reduce one exceptional element. Thus, considerations of alternate routings (alternate process plans for parts and additional units of machines as available) at the time of partitioning the part machine matrix lead to a better grouping (Figure 1.4).

1.2.3 Material Handling

At the first level it is important to consider the tradeoff between voids and exceptional elements in the cell formation. The anticipated improvement of control within a department, indicated by reduction in voids should not be offset by the increase in cell dependency indicated by an increase in exceptional elements. A motivating factor for cell formation is the reduction in material handling. Smaller cells reduce the physical distance traveled by the parts. Thus, intra cell material handling effort or cost per operation for a part is expected to decrease in a cell shop. In addition, material handling equipment can be used more effectively due to reduced scheduling problems, cluttering and traffic. The unit cost of intra cell material handling to a great extent depends upon the cell size (McAuley, 1972; Sankaran and Kasilingam, 1993). On the other hand, the cells have fewer machines and therefore, parts visit other cells more often, thus increasing the inter cell material handling cost. By balancing the intra cell and inter cell material handling costs one should be able to determine the optimal number of cells and cell sizes.

It should be noted that the material handling cost also depends on production quantity and other factors. For example, some parts may require special handling equipment because of their size, shape and fragility. The presence of an exceptional element represents one or two inter cell movement(s) of a part depending upon the sequence of operations. The part has to make one intercell move if the operation is the first or the last and two inter cell moves if the operation is an intermediate one. For example, in Figure 1.5, part 5 makes two moves for an intermediate operation (drill) while part 2 makes one move for its last operation (lathe). However, the part machine matrix shows one exceptional element for both. Furthermore, there are situations when the same machine is visited more than once by parts for non-consecutive operations which affect the material handling cost. For example, in Figure 1.4, part 2 visits the drilling machine twice. Thus, cell size, part type and quantity, operation sequence, and multiple visits to a machine have significant implications for material handling.

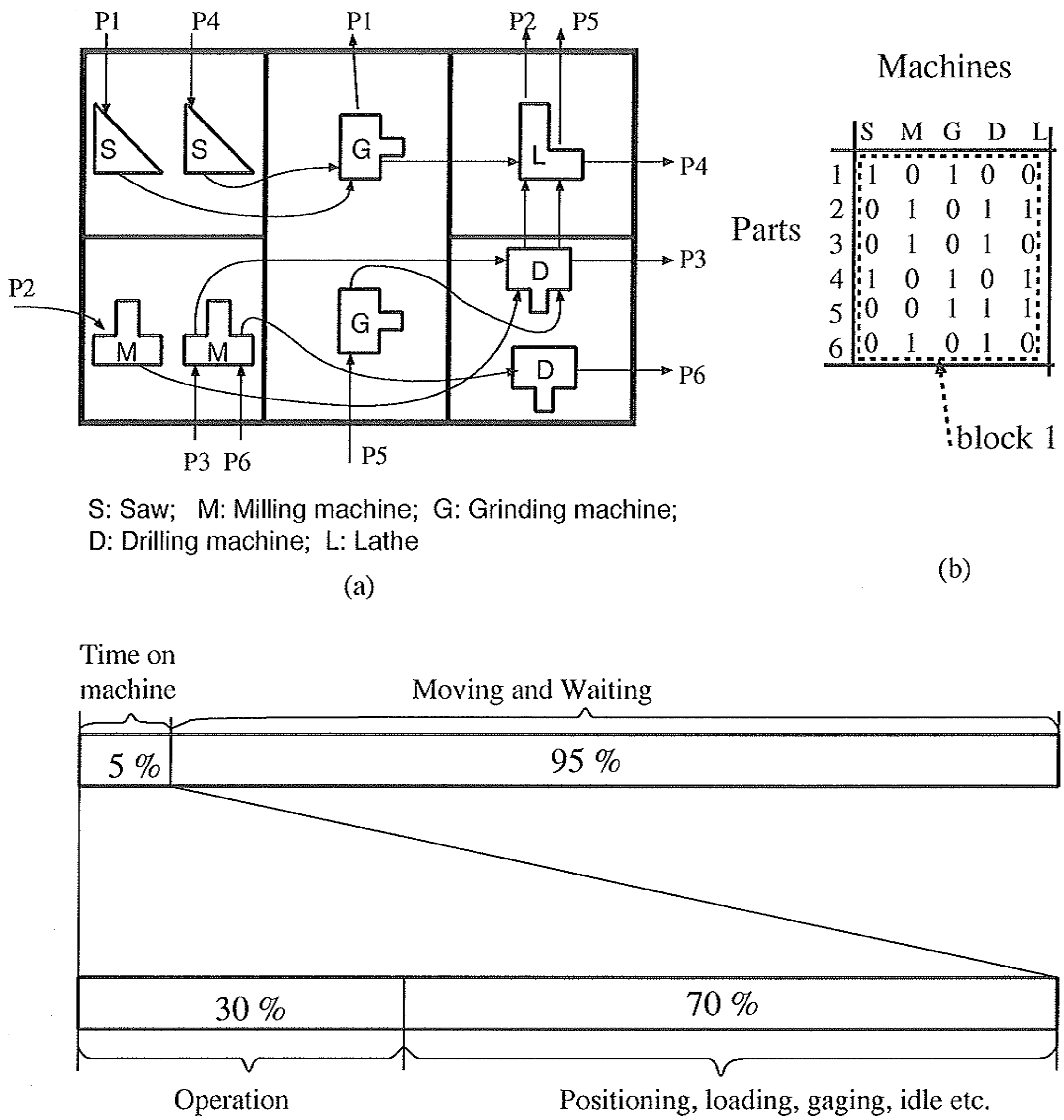
The inter cell moves can be further reduced by considering the following: transfer an extra unit of machine from another cell to the parent cell of the part; or re-allocate those operations which resulted in exceptional element to a different type of machine (capable of processing the operation) in the part's parent cell.

1.2.4 Investment and Operational Costs

The majority of the cell formation methods consider grouping of parts and machines by decomposing the part machine matrix. However, in some process industries involved in repetitive manufacturing, all parts require the same set of machines. The block diagonalization procedure is not applicable in these cases since all the elements of the part machine matrix are 1. The cell formation decision in this case involves, deciding on number of cells to be formed, capacity at each processing stage in a cell, and determining allocation and optimal sequence of part production in the cells. Moreover, the CF decision is affected by investment and operation costs such as sequence dependence setup, machine idle time, part inventory, part early and late finish compared to due date.

1.3 Organization of the Thesis

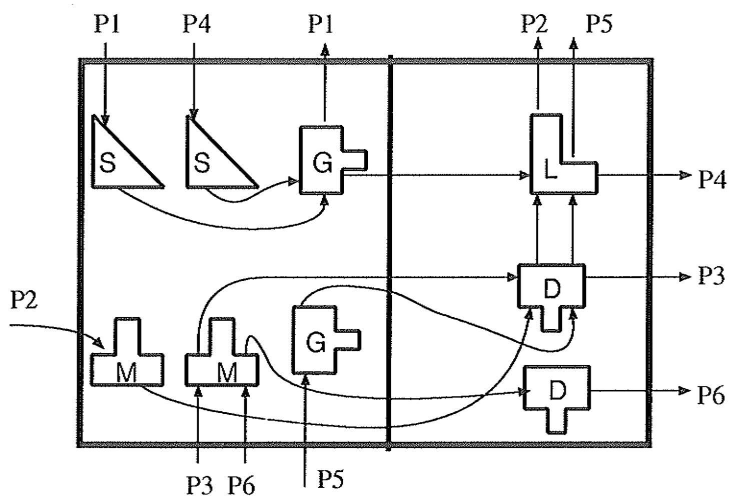
The remainder of the thesis is organized as follows. Chapter 2 provides a review of existing literature on cell formation. In addition, the chapter describes the motivation for the current research and the research objectives. In Chapters 3 and 4, the basic cell formation problem (i.e., to obtain a block diagonal form of the part machine matrix) is addressed. Cell formation considering alternate routings is presented in Chapter 5. The material handling aspects of cell formation are considered in Chapter 6. The influence of investment and operational costs on the cell formation decision is considered for the repetitive manufacturing environment in Chapter 7. The contributions of the research and directions for future research are given in Chapter 8.



(Reproduced from Ham et al., 1985)

(c)

Figure 1.1: Traditional manufacturing system.

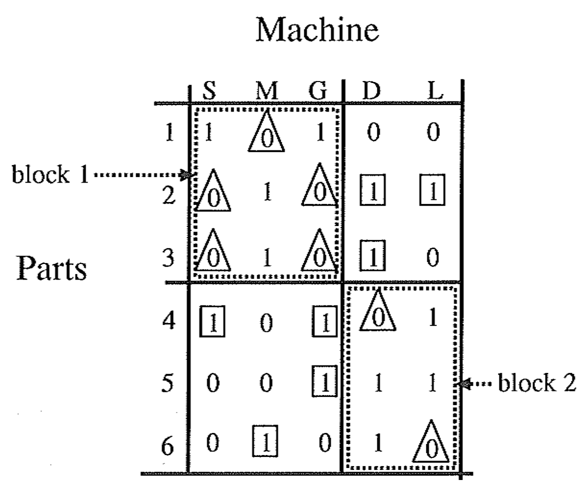


Part Family 1: {P1, P2 , P3}

Part Family 2: {P4, P5, P6}

S: Saw; M: Milling machine; G: Grinding machine;
D: Drilling machine; L: Lathe.

(a)

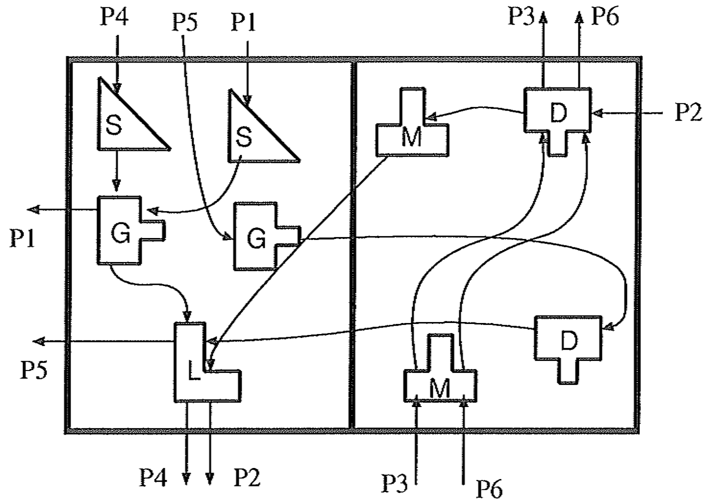


□ Operation performed in other cell (exceptional element)

△ Machine not used by the allocated part (void)

(b)

Figure 1.2: Arbitrary decomposition into 2 cells.



Part family 1: {P1, P4, P5}

Part family 2: {P2, P3, P6}

S: Saw; M: Milling machine; G: Grinding machine;
D: Drilling machine; L: Lathe

(a)

Machine

Parts	S	G	L	M	D
1	1	1	0	0	0
4	1	1	1	0	0
5	0	1	1	0	$\boxed{1}$
2	0	0	$\boxed{1}$	1	1
3	0	0	0	1	1
6	0	0	0	1	1

block 1

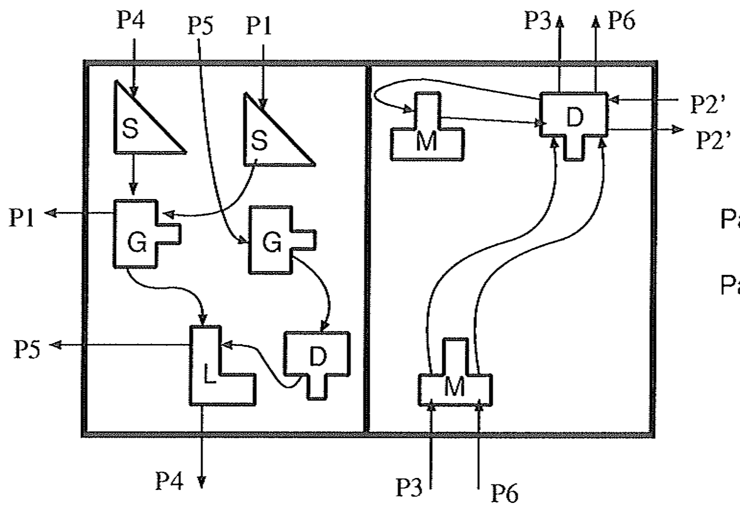
block 2

□ Operation performed in other cell (exceptional element)

△ Machine not used by allocated part (void)

(b)

Figure 1.3: Decomposition after block diagonalizing the part machine matrix.



Part family 1: {P1, P4, P5}

Part family 2: {P2', P3, P6}

S: Saw; M: Milling machine; G: Grinding machine;
D: Drilling machine; L: Lathe

(a)

Machine

Parts

	S	G	L	D	M	D
1	1	1	0	0	0	0
4	1	1	1	0	0	0
5	0	1	1	1	0	0
2'	0	0	0	0	1	1
3	0	0	0	0	1	1
6	0	0	0	0	1	1

block 1

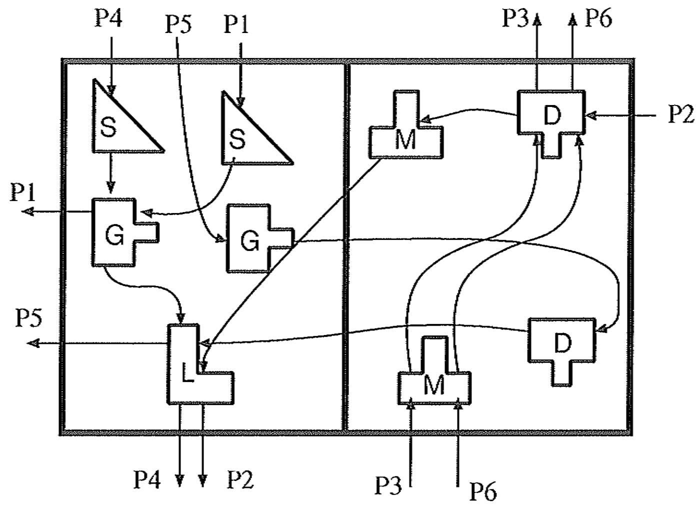
block 2

□ Operation performed in other cell (exceptional element)

△ Machine not used by allocated part (void)

(b)

Figure 1.4: Decomposition into 2 cells considering alternate routings.



S: Saw; M: Milling machine; G: Grinding machine;
D: Drilling machine; L: Lathe.

Part family 1: {P1, P4, P5}

Part family 2: {P2, P3, P6}

(a)

Machine

Parts	S	G	L	M	D
1	1	2	0	0	0
4	1	2	3	0	0
5	0	1	3	0	$\boxed{2}$
2	0	0	$\boxed{3}$	2	1
3	0	0	0	1	2
6	0	0	0	1	2

block 1

block 2

□ Operation performed in other cell (exceptional element)

△ Machine not used by allocated part (void)

(b)

Figure 1.5: Decomposition into 2 cells considering sequence of operations.

Chapter 2

Literature Review

Design of the cell system of production has gained considerable attention from the practitioners as well as academicians. The research efforts in this field have been numerous. Solving the cell formation (CF) problem is the first step towards implementation of the cell system of production. The complexity of cell formation procedure varies with the complexity of the cell formation problem. In this chapter a review of the literature under each category of the cell formation problem introduced in Chapter 1 is provided. First, the literature on the basic cell formation problem, i.e., the partitioning of part machine matrix is reviewed in section 2.1. Next, procedures considering the effects of alternate routings are reviewed in section 2.2, followed by literature on material handling in section 2.3. Finally, cell formation procedures applicable to the repetitive manufacturing environment are reviewed in section 2.4. Section 2.5 provides the motivation for the proposed research and section 2.6 states the objectives of the research.

2.1 Cell Formation

A number of approaches have been reported for cell formation. Extensive review of the techniques are available in the literature (Shafer and Meredith, 1990, Wemmerlov and Hyer, 1986). A mathematical programming statement of a seemingly small CF problem becomes large, combinatorial, and NP-complete and hence, most of the procedures are heuristic in nature. The popular ones are production flow analysis, PFA (Burbidge, 1989), rank order clustering, ROC (King, 1980), modified rank order clustering, MODROC (Chandrasekharan and Rajagopalan, 1986), single linkage clustering, SLC (McAuley, 1972), average linkage clustering, ALC (Seifoddini and Wolfe, 1986), bond energy algorithm, BEA (McCormick et al., 1972), ZODIAC (Chandrasekharan and Rajagopalan, 1987) and GRAFICS (Srinivasan and Narendran, 1991). Comprehensive review of the methods are provided in: Heragu, 1994; Cheng, 1992; Miltenburg and Zhang, 1991 and Wemmerlov and Hyer, 1986.

SLC and ALC require a large amount of data storage and computation of similarity matrices and do not form part families (PFs) and machine groups (MGs) simultaneously. SLC also suffers from the chaining problem. Algorithms such as ROC, MODROC, PFA, and BEA, cluster the rows and columns of the part machine matrix followed by a manual intervention to identify the MGs and PFs. This becomes very difficult for large problems that are not perfectly groupable. ZODIAC and GRAFICS identify PFs and MGs simultaneously without any manual intervention.

A few mathematical models have been proposed for clustering (Boctor, 1991; Kasilingam, 1989; and Kusiak, 1987). These models use indirect measures such as similarity/compatibility indices for identifying part and/or machine grouping. Boctor (1991) and Kasilingam (1989) developed integer programming models for simultaneous grouping of parts and machines. They however considered constraints on number of parts and/or machines assigned to a cell, that may not uncover natural groupings existing in the data. The importance of identifying natural groupings existing in the data is explained in Chapter 5 of this thesis.

Ribeiro and Pradin (1993) proposed a two phase methodology which minimizes the number of exceptional elements given the number of cells and by imposing constraints on number of parts and machines. To obtain a good partitioning, the procedure requires a number of trials in order to select appropriate values of three parameters required in computing the distance matrix. Also, a kernel selection rule needs to be specified (Ribeiro and Pradin, 1993).

None of the above procedures have the flexibility to create large loose cells or small tight cells to provide the designer with alternate solutions in a controlled manner. In addition, the performance for larger problems is most often not reported.

2.2 Alternate Routings

Procedures reported for partitioning the part machine matrix, considering alternate process plans and additional units of same machines, have been very few (Kusiak and Cho, 1992). Assigning additional units of machines as available is often treated subsequent to the cell formation procedure (King and Nakornchai, 1982).

Kusiak (1987) proposed a p-median formulation for grouping parts and illustrated the importance of considering alternate process plans in improving the groupability. He considered the objective as maximizing the similarity. The model requires the number of medians (p) or cells to be specified. In this procedure, one has to experiment on the value of p before a desired grouping is discovered. Moreover, the best objective value need not correspond to the objective value for the best grouping of the data. Subsequently, Kusiak and Cho (1992) developed a branching algorithm. This algorithm requires representations of process plan similarities on a transition graph and then partitioning this graph to obtain part families. The above similarity based methods identify only the part (process) families. Once part families (PFs) are identified by the algorithm (or model) the machine groups (MGs) are formed. Kasilingam and Lashkari (1991) formulated a nonlinear 0-1 integer programming model for simultaneous grouping of parts and machines in the presence of alternate process plans. The objective considered in the model is to maximize compatibility indices between machines and parts, based on tooling requirements and processing time. They also consider the availability of more than one unit of a machine. However, this model assumes that the upper limits on number of machines and parts in each cell are known. This model is appropriate to use in a flexible manufacturing system, where tooling similarity is important and restrictions on cell size can be imposed by production constraints. Imposing restrictions on number of machines and/or parts in a cell as assumed by Kasilingam and Lashkari (1991) or specifying the number of cells as in p-median model will not allow the identification of a diagonal block structure existing in the data. Moreover, indirect measures such as maximization of similarity/compatibility do not necessarily yield the best block diagonal form of the part machine matrix (explained in Chapter 5).

2.3 Material Handling

A motivating factor for introducing the cell system of production is the reduction in material handling. The following papers consider material handling in the cell formation process. Ballakur and Steudel (1987) presented a within cell utilization based algorithm which considers the minimization of the number of exceptional elements that were assumed to be a measure of the total inter cell movement. Song and Hitomi

(1992) developed a nonlinear integer programming model for forming machine groups with the objective of maximizing the number of parts completely processed inside the cell in an attempt to minimize the inter cell moves indirectly. Logendran (1990) proposed a heuristic that considers minimization of a weighed sum of intra cell and inter cell moves. He ignores the sequence of operations, and assumes that a part makes $n-1$ inter cell moves if it visits n different cells. Similarly, the part makes $m-1$ intra cell moves in a cell where m of its operations are performed. Later Logendran (1991) incorporated the effects of sequence of operations on inter cell moves for a given layout of machines and modified his total move equation to compute the exact number of inter and intra cell moves. However, based on his expression the minimization is simply equivalent to the minimization of inter cell moves only (as explained later in Chapter 6). Ahmed et al. (1991) addressed cell formation considering the costs of intra cell and inter cell moves and the cost of voids (i.e., skipping cost). The production quantity is also considered. However, they do not consider the sequence of operations, hence, the number of intra and inter cell moves are not exact. The major problem with all of the above procedures except for Logendran (1991) is that they do not consider the processing sequence and assume either the number of exceptional elements or the number of cell visits is equal to the total inter cell movements. This problem is overcome by the heuristics developed by Okogbaa et al. (1992) and Harhalakis et al. (1990) which attempt to minimize the total inter cell movements considering the sequence of operations and production quantity. However, they assumed that the number of cells and/or cell size is known.

The importance of considering the tradeoff between intra cell and inter cell handling costs is often stated in the literature (McAuley, 1972, Logendran, 1990 & 1991). However, the effect of cell size per intra cell move cost is not considered by any of the above papers. A noteworthy attempt in this direction is made by Sankaran and Kasilingam (1993). The intra cell move cost is assumed to be a stepwise linear function of cell size. They have taken into consideration the sequence of operations, additional units of same machines and the possibility of performing an operation on alternate machine types. They developed a nonlinear integer programming model with an objective of minimizing the sum of the costs due to machine amortization, space use, processing, intra cell handling and inter cell handling. The assumption of a stepwise linear function for intra cell cost, allows them to linearize the model. Owing to the integer programming nature of the linearized model, only very small size problem can be solved optimally. They have also presented a heuristic to solve the model. However, the heuristic assumes that machine amortization and processing costs are considerably higher than the sum of intra and inter cell handling costs. This is not always the case in situations such as the re-organization of the existing shops where machines are already available and the cost of amortization need not be considered.

The following papers consider the elimination of exceptional elements to reduce inter cell material handling. Kern and Wei (1991) consider machine duplication and subcontracting as possible alternatives to eliminate exceptional elements, and rank the exceptional elements based on the tradeoff between costs of machine duplication, the cost of subcontracting, and the reduction in inter cell transfer cost. Shafer et al. (1992) developed an integer programming model for the same objective. These procedures however do not consider allocating an operation to alternate machines.

2.4 Investment and Operational Costs

A number of approaches have been reported for cell formation. However, the literature simply isn't able to determine whether it is the group technology (GT) which brings the benefits or if an improved traditional shop gives the same performance. Some researchers (Flynn and Jacobs, 1986; Morris and Tersine, 1990) studied the performance of GT cells formed by part machine matrix consideration, and compared it with traditional shops using simulation techniques. Their reports indicate that GT cells are inferior in performance to the traditional shops with respect to some measures, such as WIP inventory, average part waiting times and part flow times. However, GT cells exhibited superior performance in terms of average move times and setup times. This questions the use of cell formation methods which only consider part and machine similarities. Wemmerlov and Hyer (1987) have also indicated in their study that there is a need to develop procedures in which structural or investment related variables such as number of machines, are balanced against operational variables, such as machine utilization, throughput time, etc., during the design of CM. All this suggests that there is a need to develop a mathematical framework to consider as many factors as possible, rather than just considering geometrical or processing similarities of parts.

Rajamani et al. (1992) developed a mathematical model for cell formation in repetitive manufacturing. They considered the tradeoff between cost of line (cell) and sequence dependence cost to decide the number of cells and the number of machines at each stage in these cells. However, they have not considered the scheduling aspects such as, the costs of WIP inventory, machine idle time, and part early and late finish. Gupta and Dudek (1971) studied the effects of these costs, viz., operation, job waiting, machine idle, and penalty cost of job late finish, on a flow shop. They used a simulation technique and complete enumeration for small problems, to arrive at an optimal solution. They found that considering all the costs is necessary if one needs to find an economic schedule.

2.5 Motivation for Proposed Research

The cell formation (CF) problem has attracted a great deal of research effort and numerous approaches have been proposed for partitioning the matrix. The procedures are evaluated in terms of two objectives: to maximize the usage of machines within the cell (i.e., minimize the voids) and minimize the interactions between the cells (i.e., minimize the exceptional elements), after obtaining a block diagonal form of part machine matrix. The following are the common characteristics of most of the CF algorithms reported.

- They adopt a sequential approach to CF, i.e, machine groups and part families are formed sequentially.

- Rearrangement of the part machine matrix is often sought by maximizing indirect measures such as the similarity index.

- They do not have the flexibility to create large loose cells or small tight cells to provide the designer with alternate solutions in a controlled manner.

- They require manual/subjective intervention to identify part families and machine groups. For large matrices this is very difficult.

- Performance of these algorithms are often reported for only small structured problems. Computation on large matrices is often not reported.

- Restrictions on number of parts and/or machines in a cell is imposed, which does not allow identification of natural groups in the data.

In many cases, a good decomposition can not be obtained. The groupability of the data can be enhanced by considering alternate process plans for parts and additional units of same machines as available. The survey of literature reveals the following facts.

- Models and procedures considering alternate process plans have been very few.

- Assignment of additional units of same machines to the cell is often sought after the partition is done.

- Available procedures carry most of the drawbacks of the cell formation algorithms described above, including following a sequential approach, not uncovering natural grouping, using indirect measures and requiring subjective judgment.

Cellularization generally reduces the intra cell material handling cost but increases the inter cell material handling cost. Production quantity, sequence of operations, cell size, options of assigning an operation to alternate machine types and additional machine units, have significant effects on the material handling cost. The literature survey indicates the following.

- There does not exist adequate literature to consider the effect of cell size on intra cell material handling cost.

- The tradeoff between intra cell and inter cell material handling cost is not interpreted correctly and considered.

Cell formation in the repetitive manufacturing environment requires consideration of investment and operational variables such as sequence dependence set up costs, inventory and machine utilization costs, part early and late finish costs etc. to arrive at a good cell design. The literature review gives the following picture in this area.

- The tradeoff between investment and operational costs are not considered.

- Although part inventory, early and late finish and machine utilization are important they have not been considered in the literature.

2.6 Objectives of the Research

The following are the objectives of the proposed research.

- Develop mathematical models and solution procedures to provide the simultaneous grouping of parts and machines considering the following features:

  - consider voids and exceptional elements as explicit measures instead of indirect measures;

  - develop a method which does not require any manual intervention or subjective judgment for identifying part families and machine groups;

  - provide the designer with the flexibility for forming large loose cells or small tight cells;

  - develop a robust procedure to provide good solutions for well-structured as well as ill-structured matrices;

  - develop a procedure capable of solving large problems (400x240).

- Consider alternate routings to allow for a better group formation considering the above mentioned points.

- Consider the tradeoff between intra cell and inter cell material handling. In the material handling cost calculation consider the effects of production quantity, sequence of operations, cell size, options of assigning an operation to alternate machine types and additional machine units.

- Develop a mathematical model for cell formation in a repetitive manufacturing environment considering investment, sequence dependence costs, machine utilization costs, part inventory costs, and part early and part late finish costs. Also, identify which variables significantly affect the solution time of the model.

Chapter 3

Cell Formation - Part 1

In this chapter, a nonlinear mathematical programming model is developed for cell formation (CF) which identifies part families (PFs) and machine groups(MGs) simultaneously. The objective of the model is to minimize the weighted sum of voids and exceptional elements. For the solution of large size problems, an iterative procedure called the Assignment Allocation Algorithm (AAA) is proposed. Performance of the AAA has been evaluated against many well known procedures for problems selected from published literature. Large problems with 400 parts and 240 machines are solved.

3.1 Problem Background

The processing requirements of the parts are commonly represented by a two dimensional binary matrix known as part-machine matrix, $a = \{a_{pm}\}$. The typical element of matrix a, a_{pm} is 1 or 0 depending on whether part p, requires processing on machine m or not. Table 3.1(a), shows a part machine matrix a_1, for a system which has 6 parts and 6 machines. Row and column permutation of matrix a_1, yields a block diagonal matrix a_1' shown in Table 3.1(b). In matrix a_1', one can identify two diagonal blocks (submatrices) which correspond to two subsystems or cells. The machine groups are: MG1=$\{1,3,5\}$ and MG2=$\{2,4,6\}$; the corresponding part families are PF1=$\{1,4,5\}$ and PF2=$\{2,3,6\}$.

In the above decomposition there are no 1^s outside the diagonal blocks which implies that the two cells are independent, i.e., each part family is completely processed within a machine group. Also, there are no 0^s inside the diagonal blocks which indicates each part in a part family is processed by every machine in the corresponding machine group. This example illustrates a case when a perfect decomposition of a system into two subsystems (cells) is obtained. However, in most of the situations a perfect decomposition is hardly obtained. This could be either due to the properties of the data or the inadequacies in the algorithm or both (Chandrasekharan and Rajagopalan, 1989). An example is shown in Table 3.2, where dependency between cells cannot be avoided (a_2 is the initial matrix and a_2' is the permuted matrix). In the current partition, parts 2 and 5 require processing by both the machine groups. Also, not all the machines are required for processing by every part assigned to cell 1. In this situation, one would like to obtain a near perfect decomposition considering the following objectives while partitioning the matrix.

1. to have minimum number of 0 entries inside the diagonal block (known as voids); and

2. to have minimum number of 1 entries outside the diagonal block (known as exceptional elements).

A void indicates that a machine assigned to a cell is not required for the processing of a part in the cell. When a part passes a machine without being processed on the machine, it contributes to an additional intra cell handling cost. This leads to inefficient large cells. An exceptional element is created when a part requires processing on a machine not available in the allocated cell. When a part needs to visit a different cell for its processing the inter cell handling cost increases. This also requires more co-ordinating effort between cells. Thus, voids and exceptional elements are undesirable. In a manufacturing situation, for different part/machine combinations the associated costs of voids and exceptional elements may vary and in general they are not the same. For example, if there is any special machine then all the parts requiring processing on this machine should be placed in the same cell (Burbidge, 1993). This can be achieved by giving a high value of weight to the exceptional elements corresponding to this machine for all the parts. Similarly, if there is any special part that should complete all its operations in a single cell then a high value of weight should be given to the exceptional elements corresponding to this part. Thus, there is a need to consider the importance of voids and exceptional elements explicitly. The voids and exceptional elements created are dependent on the number of diagonal blocks and the size of each diagonal block. In general, as the number of blocks decreases the size of the block increases. This results in more voids and fewer exceptional elements. If all parts and machines are grouped as one diagonal block (i.e., the cell is large and loose) we have maximum voids and no exceptional elements. For example, in Table 3.3, if all parts and machines are treated as one diagonal block we have 40 voids and no exceptional elements. For 2, 3 and 4 blocks the corresponding partitions are shown in Tables 3.3(b), 3.3(a) and 3.3(c) respectively. The number of voids and exceptional elements in these cases are (8,0), (0,0) and (0,8) respectively. Thus, as the number of voids are reduced, the number of exceptional elements increases and vice versa. In this chapter we consider the minimization of the weighted sum of the voids and the exceptional elements explicitly. Thus, combining the cell formation and evaluation procedures in one step. The proposed model also identifies parts/machines which if not assigned to a cell (external parts/machines) can enhance the block diagonalization. These parts can be considered to have the potential for subcontracting and the machines would serve as a common resource to the cells. In addition, by providing different values of weight we can generate alternate solutions. This systematic generation is important for further refining the cell structures by incorporating other manufacturing aspects such as: part sequence, processing times and machine capacities (Sule, 1991, Kern and Wei, 1991, Logendran, 1992, Shafer et al., 1992). It is worth mentioning that the objective considered in this chapter is not the ultimate objective of the cell formation. Nevertheless, this generates a first cut solution and the exceptional elements and each group can be individually considered for a more detailed analysis integrating other manufacturing aspects. The importance of integrating these aspects is however acknowledged (Choobineh, 1988; Okogbaa et al., 1992; Rajamani, 1990).

The remainder of the chapter is organized as follows. In section 3.2, a nonlinear mathematical model is developed and subsequently, in section 3.3 the Assignment Allocation Algorithm (AAA) to solve the model is presented. In section 3.4, we illustrate the procedure with numerical examples. Section 3.5 provides a comparison of the performance of AAA with existing methods from the published literature. The performance of AAA for large size (400 parts and 240 machines) problems in which the data range from well structured (groupable) to ill-structured (ungroupable) are also included in this section. Discussions and summary are presented in sections 3.6 and 3.7 respectively.

3.2 Mathematical Model

In this section a nonlinear integer programming model is developed for identification of PFs and MGs simultaneously. The proposed model is based on the objective of minimizing the weighted sum of voids and exceptional elements. The model can be formulated as.

MODEL - M1

$$Min\ \ z \ \ = \ \ w.\sum_p \sum_m \sum_c a_{pm} x_{pc}(1 - y_{mc}) + (1 - w).\sum_p \sum_m \sum_c (1 - a_{pm}) x_{pc} y_{mc} \quad (3.1)$$

subject to

$$\sum_c x_{pc} = 1 \qquad \forall \ p \qquad (3.2)$$

$$\sum_c y_{mc} = 1 \qquad \forall \ m \qquad (3.3)$$

$$x_{pc}, \ \ y_{mc} \ \ \in \ \ \{0, 1\} \qquad \forall \ p, m, c \qquad (3.4)$$

In the objective function given by equation (3.1), the first term captures the contribution of the exceptional elements and the second term represents the contribution of voids. In Table 3.4, these terms have been evaluated for a given part-machine-cell combination (pmc) for all possible values of the parameters and variables. Constraints (3.2) ensure that each part is allocated to one cell. Constraints (3.3) guarantee that each machine is assigned to a cell. Binary restrictions on the x and y variables are imposed by constraints (3.4).

C, the maximum value of cell index c is given as an over-estimate on number of cells. The model will select the appropriate number of cells. Since no upper limit constraints on number of parts or machines assigned in a cell are imposed, the model will identify natural groupings which exist in the data for a given weight.

3.3 An Iterative Algorithm (AAA)

In this section, we develop a solution algorithm for the model developed in the previous section. In model M1, two sets of variables y_{mc} and x_{pc} were defined in which the first set relates to the machines and other to the parts. Solution of model M1 is equivalent to block diagonalization of the part machine matrix minimizing the value of the objective function considered.

A block, c ($c = 1, 2, \ldots C$) represents a cell. y_{mc} will take a value of 1 or 0 depending on whether machine m is assigned to cell c or not. Similarly, x_{pc} will take a value of 1 or 0 depending on whether part p is allocated to cell c or not. The objective function captures the contribution of voids and exceptional elements given the machine assignment and part allocation.

The nonlinearity of the model arises due to the product terms of these two variable sets. The model becomes linear if one of the variable sets has known values. The solution algorithm proposed in this chapter is based on an iterative scheme. First, one set of variables (say, y_{mc}) is fixed and values of the other set of variables (x_{pc}) are obtained. Then using the values of (x_{pc}) so obtained, the algorithm solves for y_{mc} and so on, till a convergence is reached. The step which involves the determination of values of y_{mc} variables is referred to as assignment of machines to the cells and the one which involves determination of values of x_{pc} is referred to as allocation of parts to the cell. The algorithm is thus named Assignment Allocation Algorithm (AAA). Srinivasan and Narendran (1991) developed a similar approach for part machine grouping to minimize the sum of the voids and exceptional elements. However, they provided no mathematical model for the problem. Kasilingam and Lashkari (1991) have proposed a similar procedure for part and machine groupings in the presence of alternate process plans maximizing compatibility indices between part and machines.

In model M1, if y_{mc} variables are fixed at value Y_{mc} for all mc, then we get the following submodel.

SUBMODEL – M1.1 (Allocation submodel)

$$Min \quad z_1 = \sum_p \sum_m \sum_c B_{pmc}.x_{pc} \tag{3.5}$$

where, $B_{pmc} = w.a_{pm}(1 - Y_{mc}) + (1 - w).(1 - a_{pm})Y_{mc}$

subject to

$$\sum_c x_{pc} = 1 \quad \forall p \tag{3.6}$$

$$x_{pc} \in \{0, 1\} \tag{3.7}$$

This submodel is separable by parts. This means the variable x_{pc}, can be solved for each part p $(p = 1, 2, .., P)$ independently. The resulting problem for each part p, can be solved by inspection in the following way. Set $x_{pc^\star} = 1$ such that,

$$B_{pc^\star} = \underset{c}{Min} \quad \underset{m}{\sum} B_{pmc}$$

The remaining x_{pc} $(c \neq c^\star)$, should be set equal to 0.

The above can be interpreted as follows. Assume that machine assignment to the cells is given. Let the number of machines thus assigned in cell c be denoted by NM_c. To find an optimal placement of a part (row) p, one should calculate the objective value contribution for that part (row). The contribution (due to voids and exceptional elements) in the objective function of the part (or row) p depends upon which cell c the part is allocated to. If part p is allocated to cell c the contribution is B_{pc}. The voids and exceptional elements resulting from an allocation can be computed using the information of row p of the part machine matrix. The total number of ones in the row is OP_p. This represents the number of machines required by part (row) p. The number of ones in the intersection of row p with all machine columns assigned to cell c is UP_{pc}. This gives the number of machines in cell c, which can be utilized by part p if allocated to that cell. This allocation will give number of voids $vp_{pc} = NM_c - UP_{pc}$ and number of exceptional element $ep_{pc} = OP_p - UP_{pc}$. Weighing voids and exceptional elements will give $B_{pc} = w\ ep_{pc} + (1-w)\ vp_{mc}$. The optimal allocation of the part is to a cell $c^\star$ for which the contribution B_{pc} is minimum.

In a similar way we get an assignment submodel. It can be shown that

$$w.\sum_p \sum_m \sum_c a_{pm} x_{pc}(1 - y_{mc}) = w.\sum_p \sum_m \sum_c a_{pm} y_{mc}(1 - x_{pc})$$

In order to obtain an expression similar to (3.5) we substitute this relationship for the exceptional element in the objective function of model M1. Then fixing x_{pc} equal to X_{pc} for all pc will give the following submodel.

SUBMODEL – M1.2 (Assignment submodel)

$$Min \quad z_2 = \sum_p \sum_m \sum_c D_{pmc} y_{mc} \tag{3.8}$$

where, $D_{pmc} = w.a_{pm}(1 - X_{pc}) + (1 - w).(1 - a_{pm})X_{pc}$

subject to

$$\sum_c y_{mc} = 1 \quad \forall\ m \tag{3.9}$$

$$y_{mc} \quad \in \quad \{0, 1\} \tag{3.10}$$

This submodel can be solved for each y_{mc} variable as follows. Set $y_{mc^\star} = 1$ such that,

$$D_{mc^\star} = \underset{c}{Min} \ \sum_p D_{pmc}$$

The remaining y_{mc} ($c \neq c^\star$), are set equal to 0. The assignment procedure can also be carried out in a similar way as described for the allocation.

The assignment and allocation submodels are solved iteratively until a convergence is achieved. The detailed algorithm steps are described next.

Assignment Allocation Algorithm (AAA)

Step 1: Input

Read input data w, P, M, C, and $\{a_{pm}\}$ matrix.

Step 2: Initialization

iteration count, $i \rightarrow 0$

OBJM^0 (previous objective value for machine assignment step) $\rightarrow -B$

OBJP^0 (previous objective value for part allocation step) $\rightarrow -B$

where, B is a large positive number.

Step 3: Initial Assignment

[3.1]As a starting solution, we assume that each machine is assigned to an independent cell. Cell C (C = M + 1) is left empty to allow for parts to be external to cell. Assigning machine 1 to cell 1 in iteration, $i = 0$, means, set $Y_{11}^0 = 1$, and $Y_{1c}^0\ (\forall c \neq 1) = 0$

[3.2] Update $i \rightarrow 1$

go to step 4.

Step 4: Allocation

[4.1] Using the values of $y_{mc}^{i-1} = Y_{mc}^{i-1}$ (from the previous iteration), solve for x_{pc}^i i.e., find X_{pc}^i, in the following way.

For p = 1 to P

(a) Count the number of ones in row p (say OP_p), of part machine matrix.

(b) for c= 1 to C

(i) Count the number of machines assigned to cell c (say NM_c).

(ii) In part machine matrix, count the number of ones in the intersection of row p, with all the columns(m) for which $Y^i_{mc} = 1$ or columns corresponding to the machines which are assigned to cell c (say UP_{pc}).

(iii) Set $ep_{pc} = OP_p - UP_{pc}$, $vp_{pc} = NM_c - UP_{pc}$, and $B_{pc} = w\ ep_{pc} + (1\text{-}w)\ vp_{pc}$

(c) Assign part p to cell c (say, c*), for which B_{pc} is minimum, i.e., set $X^i_{pc^*} = 1$ and the remaining values of X corresponding to part p as 0.

[4.2] Set objective value, $OBJ^i = \sum_p B_{pc^*}$.

[4.3] If the objective value in the last two iterations for step 4 and step 5 remain the same (i.e., $OBJP^{i-1} = OBJ^i$ and $OBJM^{i-2} = OBJM^{i-1}$) go to step 6,

else, update the objective value $OBJP^i \rightarrow OBJ^i$ and go to step 5.

Step 5: Assignment

[5.1]Using the current values of $x^i_{pc} = X^i_{pc}$, solve for y^i_{mc} i.e., find Y^i_{mc} in the following way.

For m = 1 to M

(a) Count the number of ones in column m (say OM_m), of part machine matrix.

(b) for c = 1 to C

(i) Count the number of parts assigned to cell c (say NP_c).

(ii) In part machine matrix, count the number of ones in the intersection of column m, with all the rows(p) for which $X^i_{pc} = 1$ or columns corresponding to the parts which are assigned to cell c (say UM_{mc}).

(iii) Set em_{mc} = OM_m - UM_{mc}, vm_{mc} = NP_{pc} - UM_{mc}, and $D_{mc} = w$ em_{mc} + (1-w) vm_{mc}

(c) Assign machine m to cell c(say, c*), for which D_{mc} is minimum, i.e., set $Y^i_{mc^*} = 1$ and rest of the Y variable for part m as 0.

[5.2] Set objective value, $OBJ^i = \sum_m D_{mc^*}$.

[5.3] If the objective value in the last two iterations for step 4 and step 5 remain the same (i.e., $OBJP^{i-1}$ = $OBJP^i$ and $OBJM^{i-1}$ = OBJ^i) go to step 6,

else, update the objective value $OBJM^i \rightarrow OBJ^i$ and iteration count $i \rightarrow i+1$ and go to step 4.

Step 6: Termination

Print the results and stop.

Numerical examples are provided in the next section to illustrate the algorithm.

3.4 Illustrative Examples

In this section, an example problem is solved to illustrate the algorithm steps. Subsequently it is shown how changes in relative weights to exceptional elements and voids affect the part and machine grouping solution. The solution algorithm described in the previous section was coded in Fortran-77 and run on a Sun Sparc 2 station.

3.4.1 Example-1

An example with 6 parts and 5 machines is considered.

Step 1- Input is shown in Table 3.5.

Step 2- The following initializations were made in this step. $i = 0$, OBJM^0= -1000000.0, and OBJP^0= -100000.0.

Step 3- Initially (at $i = 0$), each machine is assigned to a separate cell, machine 1 in cell 1, machine 2 in cell 2, machine 3 in cell 3, machine 4 in cell 4, machine 5 in cell 5. Cell 6 is left empty. This corresponds to the assignment, $Y_{11}^0, Y_{22}^0, Y_{33}^0, Y_{44}^0, Y_{55}^0 = 1$ and all other Y variables are 0. Update the iteration number $i = 1$.

Step 4- In this step, we allocate parts to cells using initial machine assignment obtained from step 3. Consider part 1 first. The objective function contribution of part 1 will be calculated for allocation to all possible cells. To do this for cell 1, we consider row 1 (this corresponds to part 1) and column 1 (since in cell 1 the only machine assigned is machine 1 which corresponds to column 1) of the part machine matrix.

	↓					
cell →	1	2	3	4	5	6
machine (column) →	1	2	3	4	5	-
part (row) ↓						
→ 1	0	1	0	1	1	-
2						
3						
4						
5						
6						

From row 1, the number of machines required by part 1 is $\text{OP}_1 = 3$ (which is the sum of ones in the row). The number of machines assigned to cell 1 is $\text{NM}_1 = 1$. In the intersection of row 1 and column 1 we have no ones, hence the number of machines that can be utilized by part 1 when assigned to cell 1 is $\text{UP}_{11} = 0$. If part 1 is allocated to cell 1, we have the following. There will be three exceptional elements, since none of the three machines required by part 1 is assigned to cell 1. Also, there will be one void as machine 1 assigned in cell 1 is not required by part 1. We can calculate the number of exceptional elements and voids also as:

$$ep_{11} = OP_1 - UP_{11} = 3 - 0 = 3$$

$$vp_{11} = NM_1 - UP_{11} = 1 - 0 = 1$$

Thus, the objective function contribution will be, $B_{11} = 0.5 \text{x} 3 + (1\text{-}0.5) \text{x } 1 = 2$. Similar calculations for allocating part 1 to other cells are:

c=2, OM_2=1, UP_{12}=1, ep_{12}=3-1=2, vp_{12}=1-1=0, B_{12}= 1,

c=3, OM_3=1, UP_{13}=0, ep_{13}=3-0=3, vp_{13}= 1-0=1, B_{13}= 2,

c=4, OM_4=1, UP_{14}=1, ep_{14}=3-1=2, vp_{14}= 1-1=0, B_{14}= 1,

c=5, OM_5=1, UP_{15}=1, ep_{15}=3-1=2, vp_{15}= 1-1=0, B_{15}= 1,

c=6, OM_6=0, UP_{16}=0, ep_{16}=3-0=3, vp_{16}= 0-0=0, B_{16}= 1.5.

Thus, assigning part 1 to cell 2, 4 or 5 will give the minimum value of 1 as the objective function contribution. We break ties arbitrarily by picking the last, i.e., part 1 will be assigned to cell 5 or c^*= 5 which gives $B_{15} = 1$.

The above calculations for other part allocations (p = 2, 3, 4, 5 and 6) are summarized in the Table 3.6.

Parts 2, 5 and 6 are allocated to cell 4 and parts 1, 3 and 4 are allocated to cell 5 at iteration $i = 1$.

Step 5- For the part allocation obtained in step 4, we will find machine assignments to the cells. To illustrate this procedure let us consider machine 1. The objective value contribution for this machine assignment in different cells needs to be calculated. Since parts are assigned to cells 4 and 5 the other cells are empty at this step. We illustrate calculations for the first non empty cell, i.e., cell 4.

Cell 4 has parts 2, 5, and 6 allocated, hence we have to consider rows, 2, 5 and 6. The column corresponding to machine 1 is 1.

		↓				
	machine (column) →	1	2	3	4	5
cell↓	part (row) ↓					
5	1	0				
→4	2	1				
5	3	0				
5	4	0				
→4	5	1				
→4	6	1				

The number of parts requiring machine 1 is $\mathrm{OM}_1 = 3$ (which is sum of ones in column 1). Total number of parts assigned to cell 4 is $\mathrm{NP}_4 = 3$. The number of allocated parts in cell 4 which require processing on machine 1 is the sum of ones in the intersection of column 1, with rows 2, 5 and 6. This will give $\mathrm{UM}_{14} = 3$. Thus number of exceptional elements are $ep_{14} = \mathrm{OM}_1 - \mathrm{UM}_{14} = 3\text{-}3 = 0$, and the number of voids are $vp_{14} = \mathrm{NP}_4 - \mathrm{UM}_{14} = 3\text{-}3 = 0$. The objective function contribution D_{14}, for this assignment will thus be 0.5x0 + (1-0.5)x0 = 0. Remaining calculations for this machine and other machines are tabulated in Table 3.7.

Assignment and allocation steps were carried out iteratively till convergence in the objective function value is obtained after 3 iterations. In Table 3.8, the first line (corresponding to i=0) shows the steps 2 and 3. Then, in the same table steps 4 and 5 for each iteration are shown.

Table 3.9 shows the final output. The solution identifies 2 cells as below.

MG1 = {1, 4}, PF1 = {2, 5, 6}

MG2 = {2, 3, 5}, PF2 = {1, 3, 4}

For the above partition the number of voids and exceptional elements are 1 and 2 respectively.

3.4.2 Example-2

In this section we consider a 22-part, 11-machine problem given in Cheng (1992), to illustrate the effect of changing weights. Cheng (1992) solved the above problem using a number of algorithms. The best grouping obtained for this problem from these algorithms was from ZODIAC which gave 3 cells with 15 voids and 10 exceptional elements.

The algorithm developed in this chapter controls the type of cells, (a large number of tight cells or a smaller number of loose cells) to be formed, by changing the weight w. Table 3.10 shows the solution for different weights. The types and number of cells are altered as w is changed. When the least weight is given to the exceptional element, i.e., $w = 0.0$, no part is assigned to the cell and all the parts are identified as external parts. At the other extreme, when the exceptional elements are given the maximum weight, i.e., $w = 1$ and a zero weight is given to the voids, then one cell is formed to which all the parts are allocated. For weights between 0 and 1, with an increase in value of w the number of exceptional elements decreases and the number of voids increases. The best solution from the existing procedures due to ZODIAC is the same as the solution from the present algorithm with weight for exceptional elements as 0.7 (see Table 3.10). This solution has a grouping measure (defined in the Appendix and discussed in section 3.5) of 0.691. AAA gives a better partitioning at the weight of 0.6 with a grouping measure of 0.696. This solution is given in Table 3.11.

3.5 Computational Experience

In this section, the performance of AAA is evaluated against a number of existing procedures for a variety of problems. Computational experience with large size problems is also reported.

3.5.1 Comparison of AAA with existing algorithms

In section 3.5.1.1, well known CF problems from the open literature are solved. The performance of AAA is compared with several well known algorithms. All necessary data for these problems and results for existing methods are adopted from Miltenburg and Zhang (1991) and Ribeiro and Pradin (1993).

In section 3.5.1.2, we study the effect of data types, ranging between well-structured and ill-structured, on the performance of AAA. For this purpose, we considered data sets available in (Chandrasekharan and Rajagopalan, 1989). Also, this allows us to compare AAA with two more algorithms, namely, ZODIAC and GRAFICS. Results obtained using ZODIAC and GRAFICS for these problems are taken from Srinivasan and Narendran (1991).

3.5.1.1 Data from open literature

Nine well-known algorithms, referred to as A1 - A9, for identifying part families and machine groups are evaluated and compared by Miltenburg and Zhang (1991) for eight well known problems, referred to as P1 - P8, from the literature. The details of the problems and algorithms may be obtained from Miltenburg and Zhang

(1991). The evaluation criterion to judge the goodness of the solutions considered is the "grouping measure" (stated as a primary measure by Miltenburg and Zhang, 1991) and is given in the Appendix. The grouping measure is high if use of machines is greater (fewer voids) and if few parts require processing on machines from more than one cell (fewer exceptional elements). The results obtained using AAA in comparison with the nine algorithms are given in Table 3.12. AAA compares favourably with the nine algorithms.

We also consider 7 of the 12 problems given in Ribeiro and Pradin (1993). The remaining 5 problems use multiple copies of machines and are excluded. Table 3.13 provides the results obtained in comparison with Ribeiro and Pradin (1993). For 5 of the 7 problems (except problems HN90 and KV87) the results are the same in terms of number of cells, total number of voids and exceptional elements and grouping measure. It is worth mentioning that in the procedure by Ribeiro and Pradin (1993) the number of cells is an input parameter, while it is determined by AAA. Also, they emphasize on minimizing the exceptional elements only. Minimizing only exceptional elements may not yield a good partition. This is evident from the results obtained for problems HN90 and KV87. Although the number of exceptional elements are fewer, this is achieved at the expense of substantial increase in number of voids leading to a low value of grouping measure. AAA provides a better partition with a higher value of the grouping measure.

3.5.1.2 Well-structured and ill-structured data

Seven data sets from Chandrasekharan and Rajagopalan (1989) are taken which range from well-structured to ill-structured. All the data matrices consider 40 parts and 24 machines. They vary only in the degree of groupability. The data sets (D1

- D7) are arranged in decreasing order of groupability. The first data set (D1) is perfectly groupable and the last data set (D7) is the least groupable. Two effectiveness measures, grouping efficiency and grouping efficacy, were used to evaluate the performance of ZODIAC (Chandrasekharan and Rajagopalan, 1987) and GRAFICS (Srinivasan and Narendran, 1991). These are defined in Appendix. In order to be consistent, we compute these measures for the solution obtained by AAA on the data sets (D1 - D7). We solved the seven data sets using AAA with a weight of 0.7. A comparison of the performance measures is provided in Table 3.14. Data given in Chandrasekharan and Rajagopalan (1989) for D3 and D4 were found the same and so, we treat them as one problem. It can be observed from the table that AAA compares favourably with the two algorithms on both the measures for all the problems.

3.5.2 Computational experience with large problems

In order to study the performance of AAA on larger problems we replicated rows and columns of the seven problems (D1 - D7) ten times to obtain problems (L1 - L7) of size 400x240. All these problems were solved with the AAA for a weight of 0.7 and the results are summarized in Table 3.15. The table shows the computational time, number of iterations required for the procedure to converge, grouping measure(s), number of groups, number of voids and exceptional elements in the final solution. As expected the objective function value was 100 times that of the smaller problems solved in section 3.5.1.2. The time required to solve these problems on the Sun Sparc 2 station is less than 1 minute in each case. Thus, we can see that the present algorithm is very effective in solving large size problems.

3.6 Discussion

If the part machine matrix is small, the model developed in this chapter can be solved optimally by linearizing the nonlinear terms in the objective function using one of the procedures given in Stecke (1981). For the solution of larger problems we have proposed AAA in this chapter. It is observed in the preceding section that AAA provides a good solution to the part machine grouping problem. AAA offers many other advantages: no manual intervention is required for identifying part families and machine groups, solution for large size problems can be obtained in reasonable amount of computer time and alternative solutions can to be generated in a systematic way. However, a procedure like AAA may be sensitive to the initial solution. We considered large problems (L1 - L7) to study the effect of different initial solutions on the final solution. We randomly generated the initial solution for each of the problems and solved them using AAA. The number of iterations it took to converge, and the grouping measures obtained for the solution are given in Table 3.16. The algorithm converged within 6 iterations. Although it is sensitive to the initial solution it yielded good solutions (compared to the grouping measures obtained for these problems in section 3.5.2 and shown in Table 3.15) in all cases.

The parameter w governs the types of cells to be formed and the value of w is required as input in AAA. Our experimentation indicated that a value of $w = 0.7$ gives a good value of grouping measure considered. However, due to the nature of input data we may obtain superior results in the range 0.5 - 0.7 for a few selected problems. For example in problem P8, the grouping measure was 0.681 for value of w in the range 0.5 - 0.65 and is superior to that obtained when the weight was 0.7. It should be noted that it is not necessary to have all the voids (or exceptional elements) to have the same weights. It is possible to give different weights to different part/machine combination to reflect the scenario when opportunity costs on machines and transportation costs of parts are not the same.

3.7 Summary

In this chapter, a nonlinear mathematical programming model, was developed to identify part families and machine groups simultaneously. The objective of the model is minimization of the weighted sum of voids and exceptional elements. Subsequently, an algorithm called the Assignment Allocation Algorithm (AAA) was proposed to solve the model. The algorithm allows the weighing of voids and exceptional elements differently and thus gives the designer the flexibility to form large loose or small tight cells. The algorithm identifies natural groupings present in the data and does not require any manual intervention or subjective judgment. The results obtained using AAA compares favorably with 12 well known algorithms. AAA is very simple and less computer intensive. The solution to problems of size 400x240 using this algorithm in less than a minute shows the effectiveness of the algorithm in handling large problems.

Table 3.1: Example of a perfect decomposition.

(a) Matrix a_1

	Machines	1	2	3	4	5	6
P	1	1	0	1	0	1	0
a	2	0	1	0	1	0	1
r	3	0	1	0	1	0	1
t	4	1	0	1	0	1	0
s	5	1	0	1	0	1	0
	6	0	1	0	1	0	1

(b) Matrix a'_1

	Machines	1	3	5	2	4	6
P	1	1	1	1	0	0	0
a	4	1	1	1	0	0	0
r	5	1	1	1	0	0	0
t	2	0	0	0	1	1	1
s	3	0	0	0	1	1	1
	6	0	0	0	1	1	1

Table 3.2: Example of an imperfect decomposition.

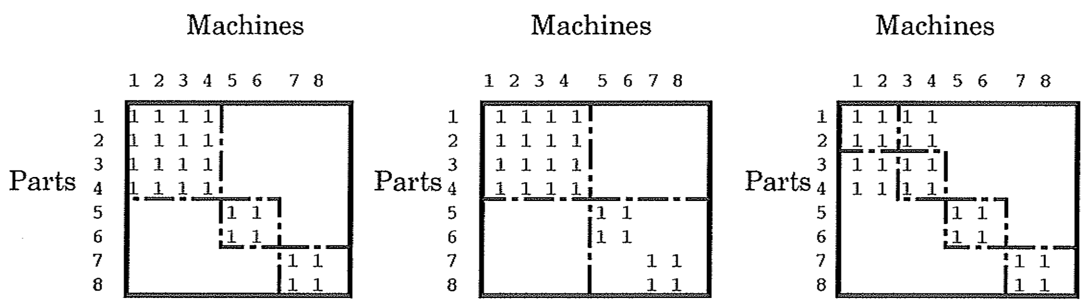
(a) Matrix a_2

	Machines	1	2	3	4	5	6
P	1	1	0	1	0	0	0
a	2	0	1	0	1	1	1
r	3	0	1	0	1	0	1
t	4	1	0	1	0	0	0
s	5	0	1	1	0	1	0
	6	0	1	0	1	0	1

(b) Matrix a'_2

	Machines	1	3	5	2	4	6
P	1	1	1	0	0	0	0
a	4	1	1	0	0	0	0
r	5	0	1	1	1	0	0
t	2	0	0	1	1	1	1
s	3	0	0	0	1	1	1
	6	0	0	0	1	1	1

Table 3.3: Effects of forming different number of cells.



(a) C = C*

(b) C < C*

(c) C > C*

Table 3.4: Contributions of voids and exceptional elements.

Possible values			Value of objective terms	
a_{pm}	x_{pc}	y_{mc}	First term (exceptional)	Second term (void)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	$(1-w)$
1	0	0	0	0
1	0	1	0	0
1	1	0	w	0
1	1	1	0	0

Table 3.5: Step 1 - input.

$w = 0.5$, P = 6, M = 5, C = 6

		Machines				
		1	2	3	4	5
	1	0	1	0	1	1
	2	1	0	0	1	0
Parts	3	0	1	1	0	1
	4	0	1	1	0	1
	5	1	0	0	1	0
	6	1	1	0	1	0

INITIAL UN-ARRANGED MATRIX

Table 3.6: Computation of objective function contribution for part allocation.

Part p $\rightarrow$ Cell c $\downarrow$		1	2	3	4	5	6	$\sum_p B_{pc^*}$
1	ep_{p1}	3	1	3	3	1	2	
	vp_{p1}	1	0	1	1	0	0	
	B_{p1}	2.0	0.5	2.0	2.0	0.5	1.0	
2	ep_{p2}	2	2	2	2	2	2	
	vp_{p2}	0	1	0	0	1	0	
	B_{p2}	1.0	1.5	1.0	1.0	1.5	1.0	
3	ep_{p3}	3	2	2	2	2	3	
	vp_{p3}	1	1	0	0	1	1	
	B_{p3}	2.0	1.5	1.0	1.0	1.5	2.0	
4	ep_{p4}	2	1	3	3	1	2	
	vp_{p4}	0	0	1	1	0	0	
	B_{p4}	1.0	0.5	2.0	2.0	0.5	1.0	
			$\uparrow$			$\uparrow$	$\uparrow$	
5	ep_{p5}	2	2	2	2	2	3	
	vp_{p5}	0	1	0	0	1	1	
	B_{p5}	1.0	1.5	1.0	1.0	1.5	2.0	
		$\uparrow$		$\uparrow$	$\uparrow$			
6	ep_{p6}	3	2	3	3	2	3	
	vp_{p6}	0	0	0	0	0	0	
	B_{p6}	1.5	1.0	1.5	1.5	1.0	1.5	
Solution	c^*	5	4	5	5	4	4	
	B_{pc^*}	1.0	0.5	1.0	1.0	0.5	1.0	5.0

Table 3.7: Computation of objective function contribution for machine assignment.

Machine m / Cell $c \downarrow$	$\rightarrow$	1	2	3	4	5	$\sum_m D_{mc^*}$
1	em_{m1}	3	4	2	4	3	
	vm_{m1}	0	0	0	0	0	
	D_{m1}	1.5	2.0	1.0	2.0	1.5	
2	em_{m2}	3	4	2	4	3	
	vm_{m2}	0	0	0	0	0	
	D_{m2}	1.5	2.0	1.0	2.0	1.5	
3	em_{m3}	3	4	2	4	3	
	vm_{m3}	0	0	0	0	0	
	D_{m3}	1.5	2.0	1.0	2.0	1.5	
4	em_{m4}	0	2	1	1	3	
	vm_{m4}	0	2	2	0	3	
	D_{m4}	0.0	2.0	1.5	0.5	3.0	
		$\uparrow$			$\uparrow$		
5	em_{m5}	3	1	0	3	0	
	vm_{m5}	3	0	1	2	0	
	D_{m5}	3.0	0.5	0.5	2.5	0.0	
			$\uparrow$	$\uparrow$		$\uparrow$	
6	em_{m6}	3	4	2	4	3	
	vm_{m6}	0	0	0	0	0	
	D_{m6}	1.5	2.0	1.0	2.0	1.5	
Solution	c^*	4	5	5	4	5	
	D_{mc^*}	0.0	0.5	0.5	0.5	0.0	1.5

Table 3.8: Steps 2, 3, 4 and 5 - intermediate results.

Iter no. i	Part allocation/ machine assignment	step	Elements (part/machine) of cell						Objective function value	
			1	2	3	4	5	6	Previous $OBJP^i$/ $OBJM^i$	Current OBJ^i
0	machine	2,3	1	2	3	4	5		-100000.0	-
1	part	4	-	-	-	2,5,6	1,3,4	-	-100000.0	5.0
	machine	5	-	-	-	1,4	2,3,5	-	-100000.0	1.5
2	part	4	-	-	-	2,5,6	1,3,4	-	5.0	1.5
	machine	5	-	-	-	1,4	2,3,5	-	1.5	1.5
3	part	4	-	-	-	2,5,6	1,3,4	-	1.5	1.5
	machine	5	-	-	-	1,4	2,3,5	-	1.5	1.5

Table 3.9: Step 6 - final results.

Objective function value = 1.5

Cell #	Machines assigned	Parts allocated
1	none	none
2	none	none
3	none	none
4	1,4	2,5,6
5	2,3,5	1,3,4
6	none	none

$$
\begin{array}{ll|cc|ccc|}
 & & \multicolumn{5}{c}{\text{Machines}} \\
 & & 1 & 4 & 2 & 3 & 5 \\
\hline
 & 2 & 1 & 1 & 0 & 0 & 0 \\
 & 5 & 1 & 1 & 0 & 0 & 0 \\
\text{Parts} & 6 & 1 & 1 & \boxed{1} & 0 & 0 \\
\hline
 & 1 & 0 & \boxed{1} & 1 & (0) & 1 \\
 & 3 & 0 & 0 & 1 & 1 & 1 \\
 & 4 & 0 & 0 & 1 & 1 & 1 \\
\hline
\end{array}
$$

$\boxed{1}$ - exceptional element
(0) - void

FINAL ARRANGED MATRIX

Table 3.10: Effect of weights on cell formation.

Weight w	Number of groups	Number of exceptional elements	Number of voids	Grouping measure
0.0	1[1]	78	0	-1.000
0.1	6	29	0	0.628
0.2	6	29	0	0.628
0.3	5	26	0	0.667
0.4	6[2]	24	2	0.656
0.5	4[3]	13	11	0.688
0.6	3[4]	11	13	0.696
0.7	3	10	15	0.691
0.8	3	9	44	0.495
0.9	2	6	58	0.478
1.0	1	0	164	0.322

[1] all parts were identified as external parts.
[2] part 10 was identified as an external part.
[3] parts 7 and 13 were identified as external parts.
[4] part 13 was identified as an external part.

Table 3.11: AAA solution to example-2 (Cheng, 1992) for weight = 0.6.

Cell number	Machines assigned	Parts allocated
1	2,3,6	5,8,12,19
2	1,4,5,10	1,2,3,7,11,15,16,20,21,22
3	7,8,9,11	4,6,9,10,14,17,18
	-	13 (external part)

Table 3.12: Comparison of AAA with existing algorithms for well known problems.

Prob -lem	η_g from different algorithms									
	Existing algorithms									
	A1	A2	A3	A4	A5	A6	A7	A8	A9	AAA
P1	0.24	0.40	0.35	0.35	-	0.37	0.44	0.39	0.45	0.48 (0.7)*
P2	0.53	0.76	0.76	0.76	-	0.76	0.73	0.76	0.76	0.76 (0.7)
P3	0.18	0.22	0.18	0.18	-	0.18	0.24	0.20	0.25	**
P4	0.66	0.85	0.85	0.85	0.85	0.82	0.85	0.85	0.85	0.85 (0.7)
P5	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57 (0.7)
P6	0.92	0.92	0.92^+	0.92^+	0.92^+	0.92^+	0.92^+	0.92^+	0.92^+	0.92 (0.7)
P7	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81 (0.7)
P8	0.68	0.57	0.63	0.59	0.57	0.59	0.68	0.58	0.64	0.68 (0.5)

- results not available in Miltenburg and Zhang (1991)
*number within bracket represents weight (w) at which the solution was obtained
**modified data for this problem could not be matched with Miltenburg and Zhang (1991)
$^+$ was reported as 0.926 in Miltenburg and Zhang (1991), but acknowledged as typographical error.

Table 3.13: Comparison of AAA with Ribeiro and Pradin (1993).

Problem	Size	Ribeiro & Pradin (1993)				AAA solution for $w = 0.7$			
		Number of cells	e	v	η_g	Number of cells	e	v	η_g
HN90	20x20	5	14	15	0.635	6	16	10	0.660
		4	11	40	0.490				
SN90	10x20	4	0	0	1.000	4	0	0	1.000
KV87	41x30	3	6	313	0.234	13	34	23	0.538
		2	3	526	0.169				
CM81(P8)	10x15	3	0	4	0.920	3	0	4	0.920
KU87	5x4	2	0	1	0.900	2	0	1	0.900
BS87	7x5	2	2	3	0.716	2	2	3	0.716
WS84	7x5	2	2	3	0.699	2	2	3	0.699

e and v are number of exceptional elements and voids respectively

Table 3.14: Comparison of AAA with ZODIAC and GRAFICS for well structured to ill structured problems (Srinivasan and Narendran, 1991).

Prob -lem	Grouping efficiency, η			Grouping efficacy, τ			from AAA	
	ZODIAC	GRAFICS	AAA	ZODIAC	GRAFICS	AAA	e	v
D1	1.000	1.000	1.000	1.000	1.000	1.000	0	0
D2	0.952	0.952	0.952	0.851	0.851	0.851	10	11
D3/D4	0.911	0.911	0.918	0.378	0.735	0.729	23	17
D5	0.773	0.788	0.875	0.204	0.432	0.506	56	17
D6	0.724	0.791	0.860	0.182	0.445	0.445	65	17
D7	0.693	0.791	0.908	0.176	0.416	0.437	70	7

e and v are number of exceptional elements and voids respectively

Table 3.15: Results for large size problems.

Prob -lem	cpu time (sec)	No. of iter-ations	No. of cells	e	v	Measures		
						η	τ	η_g
L1	30.1	3	7	0	0	1.000	1.000	1.000
L2	33.3	3	7	1000	1100	0.952	0.851	0.839
L3/L4	33.7	3	8	2300	1700	0.918	0.729	0.688
L5	33.5	3	11	5600	1700	0.875	0.507	0.388
L6	54.1	5	12	6500	1700	0.860	0.446	0.299
L7	33.0	3	14	7000	700	0.908	0.438	0.357

e and v are number of exceptional elements and voids respectively

Table 3.16: Results for the randomly generated starting solution.

Prob -lem	Number of iterations for different run					η_g for different runs				
	1	2	3	4	5	1	2	3	4	5
L1	3	3	3	3	3	1.000	1.000	1.000	1.000	1.000
L2	3	4	4	4	3	0.839	0.839	0.839	0.839	0.839
L3/L4	4	5	5	5	4	0.695	0.695	0.695	0.694	0.690
L5	5	5	5	5	4	0.424	0.393	0.412	0.412	0.397
L6	4	5	5	5	4	0.407	0.412	0.407	0.356	0.390
L7	4	6	5	4	3	0.388	0.343	0.379	0.315	0.321

Chapter 4

Cell Formation - Part 2

A nonlinear model to identify machine assignment and part allocation simultaneously was developed in Chapter 3. The objective was the minimization of the weighted sum of voids and exceptional elements. An iterative procedure called the Assignment Allocation Algorithm (AAA) was proposed to solve the model. Although it has been observed that the procedure provides a good solution, it can be sensitive to the initial solution, the number of cells specified and the groupability of the input part machine matrix. A more robust simulated annealing approach is presented in this chapter. In section 4.1, we develop a Simulated Annealing Algorithm (SAA). Section 4.2 compares the performance of the Simulated Annealing Algorithm (SAA) with the Assignment Allocation Algorithm (AAA). Section 4.3 presents a case study. Finally, a summary is presented in section 4.4.

4.1 Simulated Annealing Algorithm (SAA)

Simulated annealing is one of the procedures which has been successfully applied by researchers in solving large combinatorial problems such as the parallel machine scheduling problem, the vehicle routing problem etc. Boctor (1991) applied simulated annealing for solving part-machine grouping problems with the objective of minimizing the number of exceptional elements assuming that the upper limit on the number of machines in a cell is known. Unlike the iterative procedure, the solution obtained by simulated annealing does not depend on the initial configuration and has an objective function value closer to the global optimum (Laarhoven and Aarts, 1987). However, the rate of convergence is slower in simulated annealing in comparison to the iterative procedure (AAA). A limit on the number of iterations is often imposed when developing the Simulated Annealing Algorithm (SAA). This may restrict SAA for getting an optimal solution. Thus, the implementation of SAA is a tradeoff between solution quality and computational time.

The following discussion abstracted from Laarhoven and Aarts (1987) introduces the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems. "In condensed matter physics, annealing denotes a physical process in which a solid in a heat bath is heated up by increasing the temperature of the heat bath to a maximum value at which all particles of the solid randomly arrange themselves in the liquid phase, followed by cooling through slowly lowering the temperature of the heat bath. In this way, all particles arrange themselves in the low energy ground state of a corresponding lattice, provided the maximum temperature is sufficiently high and the cooling rate is carried out sufficiently slowly. At each temperature value T, the solid is allowed to reach thermal equilibrium, characterized by a probability function given by the Boltzmann distribution. As the temperature decreases, the Boltzmann distribution concentrates on the states with lowest energy and finally, when the temperature approaches zero, only the minimum energy states have a non-zero probability of occurrence. However, if the cooling is too rapid without allowing the solid to reach equilibrium for each temperature, defects can be frozen into the solid and a metastable amorphous structure is reached. Furthermore, similar to quenching in which the temperature of heat bath is lowered instantly the same result is obtained. To simulate the evolution to the thermal equilibrium of a solid for a fixed value of temperature T, Metropolis et al. (Metropolis et al., 1953) proposed a Monte Carlo method, which generates sequences of states of the solid in the following way. Given the current state of solid a small randomly generated perturbation is applied by a small displacement to a randomly chosen particle. If the difference in energy between the current state and the perturbed states is negative, then the process is continued with the new state. If the difference is positive, then the process is accepted by a probability which is a function of temperature and difference in energy. This acceptance rule is referred to as Metropolis criterion. Following this criterion, the system evolves into thermal equilibrium after a large number of perturbations. The Metropolis algorithm can also be used to generate sequences of configurations of a combinatorial optimization problem. In this case, the configurations assume the role of states of solid while the cost function (objective function) takes a role of energy. Temperature is a control parameter. The Simulated Annealing Algorithm can now be viewed as a sequence of Metropolis algorithms evaluated at a sequence of decreasing values of the temperature." (Laarhoven and Aarts, 1987)

In this section an implementation of the SAA to obtain grouping of parts and machines is proposed. The main steps in this algorithm are as follows. The maximum number of cells to be formed, C is specified. An initial machine assignment is generated. Assignment of machines to the cells is done using a predefined rule. For example, in Chapter 3, it was suggested that the value of C be fixed at one more than the total number of machines and initially each machine is assigned to a separate cell leaving one cell empty. Another way is to assign machines to cells randomly. In this chapter, one of the above two rules will be used to obtain an initial machine assignment. For this machine assignment, an initial part allocation is obtained by solving the allocation submodel (see Chapter 3). Thus, an initial solution (part families and machine groups) and the objective function value are obtained. At each subsequent iteration, one machine is moved from the current cell to another cell in order to get a new machine assignment. The machine to be moved and the cell for this machine are selected randomly (Boctor, 1991). Part allocation is made for this new machine assignment and the objective value is computed. The generated solution (new part families and machine groups) is accepted if the objective function value improves. If the objective function value does not improve the solution is accepted with a probability depending upon a cooling temperature, which is set to allow the acceptance of a large proportion of generated solutions at the beginning. Then, the cooling temperature is modified to reduce the probability of acceptance. This enables the algorithm to escape from a local optimum at an early stage. At each cooling temperature many moves are attempted and the algorithm stops when predefined conditions are met.

The detailed steps of the proposed implementation of SAA are presented below.

Simulated Annealing Algorithm (SAA)

[0] Initialize

0.1 Define the annealing parameters: initial temperature T_0, minimum accepted transition at each temperature AT_{min}, decrementing factor α, maximum number of iterations i_{max}, and final acceptance ratio R_f.

0.2 Initialize iteration counter: $i = 0$

0.3 Generate initial machine assignment and allocate parts by solving the allocation submodel (get SOL^0, OBJ^0).

[1] Execute outer loop, i.e., steps (1.1 - 1.7) until conditions in step 1.7 are met

1.1 Initialize inner loop counter $l = 0$, and accepted number of transitions AT $= 0$

1.2 Initialize solution for inner loop, $\mathrm{SOL}_0^i = \mathrm{SOL}^i$, $\mathrm{OBJ}_0^i = \mathrm{OBJ}^i$

1.3 Execute inner loop, i.e., steps (1.3.1 - 1.3.5) until conditions in step 1.3.5 are met

1.3.1 Update: $l = l + 1$

1.3.2 Generate a neighboring solution by perturbing machine assignment and obtaining part allocation for new machine assignment (get SOL_l^i, OBJ_l^i.)

1.3.3 $\delta = \mathrm{OBJ}_l^i$ - OBJ_{l-1}^i

1.3.4 If $\delta \leq 0$ or random(0,1) $\leq e^{-\frac{\delta}{T_i}}$ then

- SOL_l^i and OBJ_l^i are accepted
- Update AT = AT + 1

else

- solution is rejected, $\mathrm{SOL}_l^i = \mathrm{SOL}_{l-1}^i$,$\mathrm{OBJ}_l^i = \mathrm{OBJ}_{l-1}^i$

1.3.5 If one of the following conditions holds true: $\mathrm{AT} \geq \mathrm{AT}_{min}$ or $l \geq \mathrm{M}^2$ (M = number of machines), then assign L_i (length of Markov chain) $= l$, terminate the inner loop and go to 1.4, else continue the inner loop and go to 1.3.1.

1.4 Update: i = i + 1

1.5 Update: $\mathrm{SOL}^i = \mathrm{SOL}_{L_{i-1}}^{i-1}$, $\mathrm{OBJ}^i = \mathrm{OBJ}_{L_{i-1}}^{i-1}$

1.6 Reduce the cooling temperature: $\mathrm{T}_i = \alpha \cdot T_{i-1}$

1.7 If one of the following conditions holds true: i $\geq$ i_{max}; or the acceptance ratio (defined as AT/L_i) $\leq$ R_f; or the objective function value for the last 10 iterations remains the same, then terminate the outer loop and go to 2, else continue the outer loop and go to 1.1.

[2] Print the best solution obtained and terminate the procedure.

4.1.1 Selection of Simulated Annealing Parameters

The implementation of the Simulated Annealing Algorithm in the previous section resorts to generating a sequence of homogeneous Markov chains of finite length (to ensure the thermal equilibrium is reached) at decreasing values of cooling temperature. The following parameters should be specified (see Laarhoven and Aarts, 1987):

- initial value of temperature, T_0;
- length of Markov chain, L_i (at iteration, i);
- a rule for changing the current value of temperature to the next one;
- criteria to terminate the algorithm.

A choice for these parameters is referred to as a cooling schedule. In this thesis we define the cooling schedule in the following way.

The initial value of temperature, T_0, is taken in such a way that virtually all transitions are accepted. An acceptance ratio, R, is defined as the number of accepted transitions divided by the number of proposed transitions. T_0 is set in such a way that the initial acceptance ratio, R_0 is close to 1. Usually, the value of T_0 is in the order of the expected objective function value. The value of T_0 is increased or decreased to bring the acceptance ratio for the first 10 iterations between 0.95 and 1.0.

Length of the Markov chains L_i, are controlled in such a way that for each value of temperature, T_i, a minimum number of transitions should be accepted, i.e., L_i is determined such that the number of accepted transitions is at least, AT_{min}. However, as T_i approaches 0, transitions are accepted with decreasing probability and thus one eventually obtains $L_i \rightarrow \infty$ for $T_i \downarrow 0$. Consequently, L_i is bounded by some constant L (usually chosen polynomial in the problem size) to avoid extremely long Markov chains for low value of cooling temperature. We define $L = a \, . \, M^2$; where M is the total number of machines and a is a constant. The value of AT_{min} should be high enough to ensure that an equilibrium is reached at each temperature. Higher the value choosen for AT_{min}, the better the expected quality of solution is.

To ensure slow cooling, the decrement for temperature should be gradual. We adopt a frequently used decrement rule given by (Laarhoven and Aarts, 1987)

$$T_i = \alpha . T_{i-1}$$

where α is a constant smaller than but close to 1. Also, if a faster cooling is desired, AT_{min} is given a high value and α is given a lower value. Thus if fast cooling is done the Markov chains at each temperature would be longer.

The stopping criterion most often used in simulated annealing is by the value of the final temperature. In this implementation, the final temperature is not chosen a priori. Instead, the annealing is allowed to continue until the system is frozen by one of the following criteria:

- The maximum number of iterations (temperature), i_{max};

- If the objective function value of the last accepted transition for the temperature is identical for a number of iterations (kept as 10 iterations); or
- The acceptance ratio is smaller than a given value, R_f at a temperature.

The value of simulated annealing parameters used in this chapter are as follows: $T_0 = 10$; $AT_{min} = 25$; $\alpha = 0.90$; $i_{max} = 100$; and $R_f = 0.01$. The maximum number of iterations i_{max} was set at a high value so that the algorithm is terminated by other criterion. For most problems, T_0 is in the order of the objective function value expected. AT_{min} is taken in the order of number of machines. Initially, α was taken as 0.99. The value was reduced to 0.9 since solution quality did not deteriorate. This reduces the computational time. R_f was set to a low value of 0.01. As the value of α decreases, i_{max} can also decrease.

4.2 Computational Experience

First we considered the 8 well known problems (Table 3.12) and problems from Ribeiro and Pradin, 1993 (Table 3.13) from Chapter 3. The initial machine assignment for SAA to solve these problems were obtained by assigning each machine to a separate cell as suggested in Chapter 3. The comparison of SAA and AAA is shown in Tables 4.1 and 4.2. The objective function value obtained by SAA is the same or better than AAA. However, the 'grouping measure' obtained from SAA for the two problems, HN90 and KV87, were worse. This can be attributed to the fact that this measure has not been directly considered in the objective function of the model.

To illustrate the robustness of the SAA in comparison with the AAA we study the effects of initial machine assignment. We considered a perfectly groupable matrix (problem D1 from Chapter 3) that can form 7 perfect clusters. The maximum number of cells C, is varied from 7 to 25. The initial machine assignment is generated by randomly assigning machines to the cells. AAA and SAA are run for 19 (C = 7, 8,.....25) different initial machine assignments. It was found that AAA gave non-optimal solutions for the following 13 cases (the table shows grouping measure, η_g and objective function value, O.V):

$C \rightarrow$	7	8	9	10	11	12	13	14	15	17	20	22
η_g	0.64	0.77	0.64	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
O.V.	22.5	12.0	22.5	10.5	10.5	10.5	10.5	10.5	10.5	10.5	10.5	10.5

For the remaining 6 cases AAA gave the optimal solution with a 'grouping measure' of 1 and the objective function value of 0.0. SAA is not sensitive to the initial solution and gave the optimal solution for all the cases.

We considered problems D2 to D7 from Chapter 3. We generated 10 initial machine assignments for each problem. C was varied to take the values 15, 18, 20, 23 and 25. For each value of C, two random number seeds were used for obtaining the initial machine assignment. These problems were solved using AAA and SAA on a Sun Sparc 2 station. The time taken by AAA was less than 1 second for each problem while the average time taken by SAA was 54.7 seconds. The results from AAA and SAA are summarized in Tables 4.3 - 4.5. The deviations (σ_{n-1}) of the objective function value and grouping measure are more in the AAA than the SAA. However, as the groupability of data reduces, the solution from AAA becomes less sensitive to the initial machine assignment, i.e., deviation in the objective function value reduces. Overall, the SAA gives better (lower value of the mean objective function value and higher value of the mean grouping measure) and more consistent solutions (objective function values and grouping measures close to the mean objective function value and the mean grouping measure).

4.3 A Case Study

In this section the AAA and SAA are used on the data collected from a local manufacturing company. The input in the form of a part machine matrix is shown in Table 4.6. We solved this problem using the AAA and SAA for $w = 0.7$. The initial machine solution was obtained by setting the maximum number of cells, C, to 27 and assigning each machine to a separate cell leaving cell 27 empty. The rearranged part machine matrix from the AAA and SAA are shown in Tables 4.7 and 4.8 respectively. The AAA gave a solution with the objective function value of 45.8 and grouping measure of 0.44. It took 0.17 seconds. The SAA gave a better solution with the objective function value of 41.8 and grouping measure of 0.52. The computation time in this case was 137.5 seconds. We ran the AAA for 3 randomly generated initial machine assignments. The grouping measures (objective function values) obtained are: 0.52 (42.3), 0.50 (42.6) and 0.44 (45.8) respectively. Computational time for each case was less than 0.2 seconds. Thus a good solution can be obtained by running the AAA a few times with different initial machine grouping.

4.4 Summary

A Simulated Annealing Algorithm is proposed to solve cell formation problem. In general, the Simulated Annealing Algorithm gives better and more consistent results than the Assignment Allocation Algorithm. However, the computation time required is significantly higher. In our opinion, for smaller problems, the Simulated Annealing Algorithm is recommended. For large problems, it is preferable to solve the Assignment Allocation Algorithm a few times with different initial machine assignments.

Table 4.1: Comparison of SAA with AAA for well known problems (see Chapter 3).

Problem	From AAA		From SAA	
	Grouping measure	Objective function value	Grouping measure	Objective function value
P1	0.48	31.4	0.48	31.3
P2	0.76	13.0	0.76	13.0
P3	***	***	***	***
P4	0.85	6.3	0.85	6.3
P5	0.57	20.7	0.57	20.7
P6	0.92	1.2	0.92	1.2
P7	0.81	2.3	0.81	2.3
P8	0.68	5.5	0.68	5.5

*** Input data for P3 is not available (see Chapter 3)

Table 4.2: Comparison of SAA with AAA for problems from Ribeiro and Pradin (see Chapter 3).

Problem	From AAA		From SAA	
	Grouping measure	Objective function value	Grouping measure	Objective function value
HN90	0.66	14.2	0.65	14.0
SN90	1.00	0.0	1.00	0.0
KV87	0.54	30.7	0.53	30.4
CM81	0.92	1.2	0.92	1.2
KU87	0.90	0.3	0.90	0.3
BS87	0.72	2.3	0.72	2.3
WS84	0.70	2.3	0.70	2.3

Table 4.3: AAA results for well structured to ill structured problems.

Prob-lem	Grouping measure and Objective function value									
	C†=15		18		20		23		25	
	1‡	2	1	2	1	2	1	2	1	2
D1	1.00⋆ 0.0*	0.79 10.5	1.00 0.0	1.00 0.0	1.00 0.0	0.79 10.5	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0
D2	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.68 18.9	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3
D3/ D4	0.69 21.2	0.69 22.5	0.69 21.2	0.69 21.3	0.69 22.5	0.55 28.5	0.69 22.5	0.69 22.5	0.69 21.2	0.69 21.2
D5	0.38 42.8	0.39 43.1	0.37 44.6	0.47 45.2	0.40 42.0	0.43 45.9	0.44 44.9	0.46 44.5	0.39 45.1	0.44 44.4
D6	0.35 48.3	0.34 47.8	0.37 49.6	0.35 48.7	0.39 48.2	0.28 51.2	0.35 51.4	0.38 48.2	0.40 50.2	0.35 47.5
D7	0.34 49.1	0.30 50.7	0.27 51.1	0.28 50.4	0.31 52.7	0.31 51.2	0.28 50.8	0.37 51.9	0.38 50.5	0.29 51.8

† maximum number of cells
‡ seed 1 for initial machine assignment
⋆ grouping measure
∗ objective function value

Table 4.4: SAA results for well structured to ill structured problems.

Prob -lem	Grouping measure and Objective function value									
	$C^{\dagger}=15$		18		20		23		25	
	$1^{\ddagger}$	2	1	2	1	2	1	2	1	2
D1	$1.00^{\star}$ 0.0^{*}	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0	1.00 0.0
D2	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3	0.84 10.3
D3/ D4	0.69 20.0	0.69 20.0	0.69 20.0	0.69 21.2	0.69 20.0	0.69 20.0	0.69 20.0	0.69 20.0	0.69 20.0	0.69 20.0
D5	0.40 41.1	0.40 41.1	0.41 41.3	0.40 41.1	0.41 41.3	0.39 41.3	0.42 42.7	0.43 42.0	0.42 42.7	0.38 42.5
D6	0.39 46.3	0.34 46.5	0.39 46.3	0.38 46.6	0.39 46.7	0.39 46.7	0.39 46.7	0.40 46.8	0.39 46.3	0.39 46.7
D7	0.32 48.5	0.33 48.2	0.31 48.4	0.35 48.4	0.33 48.2	0.35 48.4	0.33 48.2	0.31 48.4	0.31 48.4	0.35 48.4

$\dagger$ maximum number of cells
$\ddagger$ seed 1 for initial machine assignment
$\star$ grouping measure
$*$ objective function value

Table 4.5: Comparison for well structured to ill structured problems.

Prob -lem	Grouping measure (Objective function value)							
	From AAA				From SAA			
	Min.	Max.	Mean	σ_{n-1}	Min.	Max.	Mean	σ_{n-1}
D1	0.79⋆ (0.0)*	1.00 (10.5)	0.96 (2.1)	0.09 (4.4)	1.00 (0.0)	1.00 (0.0)	1.00 (0.0)	0.00 (0.0)
D2	0.68 (10.3)	0.84 (18.9)	0.82 (11.2)	0.05 (2.7)	0.84 (10.3)	0.84 (10.3)	0.84 (10.3)	0.00 (0.0)
D3/ D4	0.55 (21.2)	0.69 (28.5)	0.68 (22.5)	0.04 (2.2)	0.69 (20.0)	0.69 (21.2)	0.69 (20.1)	0.00 (0.4)
D5	0.37 (42.0)	0.46 (45.9)	0.42 (44.3)	0.04 (1.3)	0.38 (41.1)	0.43 (42.7)	0.41 (41.7)	0.01 (0.7)
D6	0.28 (47.5)	0.40 (51.4)	0.36 (49.1)	0.03 (1.4)	0.34 (46.3)	0.40 (46.8)	0.39 (46.6)	0.02 (0.2)
D7	0.27 (49.1)	0.38 (52.7)	0.31 (51.0)	0.04 (1.0)	0.31 (48.2)	0.35 (48.5)	0.33 (48.4)	0.02 (0.1)

⋆ grouping measure
* objective function value

Table 4.6: Input part machine matrix for the case study.

	Machine																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1												1						1		1						
2								1		1		1					1	1								
3								1		1		1			1	1	1									
4						1		1		1		1		1				1						1		
5		1							1		1		1									1				
6										1							1						1		1	1
7										1										1						
8		1		1						1										1						
9																				1						
10		1		1																1						
11		1				1	1	1		1		1			1	1		1								
12			1		1															1						
13			1		1															1					1	1
14		1							1	1							1			1					1	1
P 15		1		1		1	1			1					1	1			1							
a 16			1					1									1						1		1	1
r 17			1					1									1						1		1	1
t 18								1	1								1									
19				1																1						
20				1																1						
21				1																1						
22						1	1	1												1						
23						1	1	1												1						
24						1	1	1												1						
25						1	1	1												1						
26				1	1	1														1						
27	1			1		1				1					1				1							
28		1							1	1							1					1				
29		1							1	1							1					1				
30		1							1	1							1					1				
31		1							1	1							1					1				
32		1							1	1							1					1				
33		1							1	1							1					1				
34																						1				
35																						1				
36				1																1						
37				1			1													1						

Table 4.7: Rearranged matrix from AAA.

Part \ Machine	8	12	16	18	1	6	15	19	4	7	20	5	21	2	9	10	17	22	3	23	25	26	11	13	14	24
1		1		1							1															
2	1	1		1												1	1									
3	1	1	1				1									1	1									
4	1	1		1		1										1									1	1
11	1	1	1	1		1	1			1				1		1										
15			1			1	1	1	1	1				1		1										
27					1	1	1	1	1							1										
7											1					1										
8									1		1			1		1										
9											1															
10									1		1			1												
19									1		1															
20									1		1															
21									1		1															
22	1					1				1	1															
23	1					1				1	1															
24	1					1				1	1															
25	1					1				1	1															
26	1					1				1	1															
36									1		1															
37									1	1	1															
12												1	1						1							
5														1	1			1					1	1		
14													1	1	1	1	1				1	1				
18	1															1	1									
28														1	1	1	1	1								
29														1	1	1	1	1								
30														1	1	1	1	1								
31														1	1	1	1	1								
32														1	1	1	1	1								
33														1	1	1	1	1								
6																1	1			1	1	1				
13												1	1						1		1	1				
16	1																1		1	1	1	1				
17	1																1		1	1	1	1				
34																		1								
35																		1								

Table 4.8: Rearranged matrix from SAA.

Machine

Part	15	16	19	1	2	17	10	09	22	12	18	14	24	21	5	4	20	23	25	26	3	8	7	6	11	13
3	1	1				1	1			1												1				
15	1	1	1		1		1									1							1	1		
27	1		1	1			1									1								1		
5					1			1	1																1	1
14					1	1	1	1						1					1	1						
18						1	1															1				
28					1	1	1	1	1																	
29					1	1	1	1	1																	
30					1	1	1	1	1																	
31					1	1	1	1	1																	
32					1	1	1	1	1																	
33					1	1	1	1	1																	
1										1	1						1									
2						1	1			1	1											1				
4							1			1	1	1	1									1	1			
12														1	1						1					
7							1										1									
8					1		1									1	1									
9																	1									
10					1											1	1									
19																1	1									
20																1	1									
21																1	1									
36																1	1									
37																1	1						1			
6						1	1											1	1	1						
13														1	1					1	1					
16						1														1	1	1				
17						1														1	1	1				
11	1	1			1		1			1	1											1	1	1		
22																	1					1	1	1		
23																	1					1	1	1		
24																	1					1	1	1		
25																	1					1	1	1		
26																	1					1	1	1		
34									1																	
35									1																	

Chapter 5

Cell Formation Considering Alternate Routings

Many cell formation (CF) procedures use the part machine matrix as an input. A perfect diagonalization of the part machine matrix to form exclusive part families and machine groups is not possible in many instances. Considering alternate routings (i.e., alternate plans for the parts and additional units of same machines) improves this diagonalization. In this chapter, a nonlinear integer programming model is developed for cell formation considering alternate routings. The model is illustrated with numerical examples. The optimal solutions for these examples are obtained by solving the linearized version of the model. For the efficient solution of larger problems a Simulated Annealing Algorithm is implemented.

5.1 Problem Background

In most manufacturing situations, a perfect decomposition of part machine matrix to form mutually exclusive cells is not possible due to the property of the data and/or the inadequacies in the algorithm (Chandrasekharan and Rajagopalan, 1989). Two possible approaches to improve groupability are by considering alternate process plans for parts and additional units of same machines as available (Kusiak and Cho, 1992). In this chapter, we consider alternate routings (alternate process plans and additional units of same machines) during cell formation. The objective of cell formation is to identify a near block diagonal structure existing in the data. By imposing restrictions on number of machines and/or parts in a cell (as assumed by Kasilingam and Lashkari, 1991) or specifying the number of cells (as in p-median model, Kusiak, 1987) will not allow to obtain block diagonal form existing in the data. For example, let us consider Table 5.1. Table 5.1(a) shows a partition into the optimal number of cells ($C^* = 3$). This forms three fully dense diagonal blocks with no exceptional elements. If we specify the number of cells as 2 ($C < C^*$) this will result in an increase in the number of voids. For example, Table 5.1(b) shows a possible partition into two cells resulting in 8 voids and no exceptional elements. Similarly, Table 5.1(c) illustrates a situation when the number of cells is specified as 4 ($C > C^*$). This will result in an increase in the number of exceptional elements to 8 with no voids. We can also observe from these partitions that restricting the maximum number of parts in a cell to 2 will yield a partition as shown in Table 5.1(c) which is not the best grouping for this data. In addition, indirect measures such as maximization of similarity/compatibility do not necessarily yield the best diagonalization of part machine matrix. This will be illustrated by modifying the numerical example in Kusiak (1987). The original matrix is shown in Table 5.2. On solving the p-median model for p = 2, the following part (process) families are identified.

PF1 = {1(2), 3(2)} and PF2 = {2(2), 4(2), 5(2)}.

The unbracketed number shows the part number and bracketed number shows the process plan number.

Using the above part grouping and assigning machines to these parts, the final matrix obtained is shown in Table 5.3 (SOLUTION 1). This solution gives one void and no exceptional elements.

In the above example (PROBLEM 1), let us assume an additional process plan for part 5, say 5(3) is available. The resulting input matrix for PROBLEM 2, is shown in Table 5.4. If this problem is solved using the p-median model, the part (process) families are modified as follows.

PF1 = {1(2), 3(2)} and PF2 = {2(2), 4(2), 5(3)}.

Assigning machines to parts gives a solution (SOLUTION 2) as shown in Table 5.5. This solution gives one exceptional element and no voids.

Both solutions have the same objective function value (similarity index). The only difference in these solutions is in the plan selection for part 5. SOLUTION 1, gives 1 void and 0 exceptional elements; and SOLUTION 2 gives 0 voids and 1 exceptional element. The question arises as to which solution of the two is better. The answer depends upon whether one gives more importance to voids or to exceptional elements. The similarity based methods cannot distinguish between the two solutions. Therefore, we consider the minimization of the weighted sum of the voids and the exceptional elements explicitly in the objective.

The remainder of the chapter is organized as follows. In section 5.2, a nonlinear integer programming model is developed. A linear version of the model is also presented in this section. In section 5.3, we present an iterative solution procedure and develop a Simulated Annealing Algorithm for solving large problems. In section 5.4, we illustrate the model by considering numerical examples. The effect of forming loose cells and tight cells is illustrated by solving the model for different weights. Computational experience with iterative procedure and Simulated Annealing Algorithm is presented in section 5.5. The results obtained are compared with the optimal solutions. A summary is presented in section 5.6.

5.2 Mathematical Models

In this section a nonlinear integer programming model is developed for simultaneous grouping of parts and machines. A linear model obtained by linearizing the objective function terms is also presented in this section.

5.2.1 Nonlinear model

The proposed nonlinear model (NLM) considers the objective of minimizing the weighted sum of the voids and the exceptional elements. The NLM is as follows:

MODEL - NLM (M2)

$$\text{Min } z = \sum_{pmcr} w_{pm}\, a^r_{pm}\, x^r_{pc}\, (1 - y_{mc}) + \sum_{pmcr} (1 - w_{pm})\, (1 - a^r_{pm})\, x^r_{pc}\, y_{mc} \quad (5.1)$$

subject to:

$$\sum_{cr} x^r_{pc} = 1 \qquad \forall\ p \quad (5.2)$$

$$\sum_{c} y_{mc} \leq N_m \qquad \forall\ m \quad (5.3)$$

$$x_{pc},\ y_{mc}\ \in\ \{0, 1\} \qquad \forall\ p, m, c \quad (5.4)$$

In the objective function given by equation (5.1), the first term captures the contribution of the exceptional elements and the second term represents the contribution of the voids. Constraints (5.2) guarantee that each part is allocated to one of the cells and only one process plan is selected for the part. Constraints (5.3) ensure that the total units of a machine type assigned to different cells do not exceed the available units for that machine type. Binary restrictions on x and y variables are imposed by constraints (5.4).

C, the maximum value of cell index c is given as an upper limit on the number of cells. The model selects the appropriate number of cells.

5.2.2 Linear model

The NLM has nonlinear terms in the objective function and linear constraints. The model can be linearized by introducing additional variables and constraints. The following additional variables are defined.

$$\delta^{+}_{pmcr} = \begin{cases} 1 & \text{if part } p \text{ is allocated to cell } c\text{, uses process plan } r \text{ and requires an inter} \\ & \text{cell movement for machine } m \text{ (i.e., an exceptional element)} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta^{-}_{pmcr} = \begin{cases} 1 & \text{if part } p \text{ is allocated to cell } c\text{, uses process plan } r \text{ and does not require} \\ & \text{processing on a machine } m \text{ which is assigned to cell } c \text{ (i.e., a void)} \\ 0 & \text{otherwise} \end{cases}$$

The linear model (LM) is as follows:

MODEL - LM (M2L)

$$\text{Min } z = \sum_{pmcr} w_{pm}\, \delta^{+}_{pmcr} + \sum_{pmcr} (1 - w_{pm})\, \delta^{-}_{pmcr} \tag{5.5}$$

subject to: constraints (5.2), (5.3), (5.4) and the following

$$\delta^{+}_{pmcr} \geq x^{r}_{pc} - y_{mc} \quad (\forall\ p, m, c, r\ \ \&\ \ a^{r}_{pm} = 1) \tag{5.6}$$

$$\delta^{-}_{pmcr} \geq x^{r}_{pc} + y_{mc} - 1 \quad (\forall\ p, m, c, r\ \ \&\ \ a^{r}_{pm} = 0) \tag{5.7}$$

$$\delta^{+}_{pmcr},\ \delta^{-}_{pmcr}\ \geq\ 0 \quad \forall\ p, m, c, r \tag{5.8}$$

It should be noted that although δ^{+}_{pmcr} and δ^{-}_{pmcr} are defined as continuous non-negative variables, these variables will take only binary values due to the structure of the constraints (5.6) and (5.7).

5.3 Solution Procedures

For small problems an optimal solution can be obtained by solving the LM. However, large problems require a prohibitive amount of computer time. Hence, there is a need to develop an efficient heuristic procedure which can provide a good solution in an acceptable amount of time. In section 5.3.1, we extend the iterative procedure (AAA) which provides good results when alternate routings are not considered. In the presence of the alternate plans, this iterative procedure has a tendency to converge to a local optimum (Adil et al., 1993 and see section 5.5.1). Therefore, in section 5.3.2, we develop a Simulated Annealing Algorithm that gives an objective value closer to the global optimum.

5.3.1 An Iterative Algorithm (AAA)

One possible approach could be similar to AAA presented in Chapter 3. Kasilingam and Lashkari (1991) also used a similar approach in the presence of alternate process plans. In the NLM, two sets of variables y_{mc} and x^r_{pc} were defined. The first set relates to the machines and other to the parts. The nonlinearity of the model arises because of the product terms of these two variable sets in the objective function. The model becomes linear if one of the variable sets has known values. For instance, if y_{mc} are fixed. This will give the following submodel:

SUBMODEL – M2.1 (Allocation submodel)

$$Min \quad z_1 = \sum_{pmcr} B_{pmcr} x^r_{pc} \tag{5.9}$$

where, $B_{pmcr} = w_{pm}.a^r_{pm}(1 - Y_{mc}) + (1 - w_{pm}).(1 - a^r_{pm})Y_{mc}$

subject to:

$$\sum_{cr} x^r_{pc} = 1 \quad \forall\, p \tag{5.10}$$

$$x_{pc}^{r} \quad \in \quad \{0,1\} \tag{5.11}$$

This submodel is separable by parts. This means the variable x_{pc}^{r}, can be solved for each $p = 1, 2, .., P$, independently. The resulting model for each part p, can be solved optimally by inspection in the following way. Set $x_{pc^*}^{r^*} = 1$ such that,

$$B_{pc^*r^*} = \underset{cr}{Min} \; \sum_{m} B_{pmcr}$$

The remaining x_{pc}^{r} $(c \neq c^*, \ r \neq r^*)$, should be set equal to 0.

Similarly, if the values of x_{pc}^{r} are fixed, y_{mc} can be obtained by solving the following submodel:

$$\text{SUBMODEL} - \text{M2.2} \ \ (\text{Assignment} \ \ \text{submodel})$$

$$Min \quad z_2 = \sum_{pmcr} w_{pm} \ a_{pm}^{r} X_{pc}^{r} + \sum_{pmcr} D_{pmcr} y_{mc}$$

where, $D_{pmcr} = -w_{pm}.a_{pm}^{r} X_{pc}^{r} + (1 - w).(1 - a_{pm}^{r}) X_{pc}^{r}$

subject to:

$$\sum_{c} y_{mc} = N_m \qquad \forall \ m \tag{5.12}$$

$$y_{mc} \quad \in \quad \{0,1\} \tag{5.13}$$

This submodel can be solved optimally for each y_{mc} variable as follows. Set $y_{mc^*} = 1$ such that,

$$D_{mc^*} = \sum_{pr} w_{pm} \ a_{pm}^{r} X_{pc}^{r} + \underset{cr}{Min} \; \sum_{p} D_{pmcr}$$

The additional units of each machine type can be assigned by arranging the non-positive D_{mc} $(c \neq c^*)$ in ascending order and selecting them in that order.

The assignment and allocation submodels can be solved iteratively until a convergence is achieved. Adil et al. (1993) implemented this procedure for cell formation considering alternate process plans. However, when alternate process plans were considered (with no additional units of same machines, i.e., N_m=1) it was observed that the initial input matrix greatly affects the quality of the solution. This warranted a further improvement in the solution procedure. In the next section we develop a Simulated Annealing Algorithm. The part allocation procedure proposed above will be used with the Simulated Annealing Algorithm as well.

5.3.2 Simulated Annealing Algorithm (SAA)

The Simulated Annealing Algorithm developed in Chapter 4 is extended to consider alternate process plans and additional units of same machines. The initial machine grouping in this case is generated as follows. The number of cells, C is set at $M1+1$, where $M1$ is the number of machines considering the additional units of the same machine separately. Initially each machine is assigned to a separate cell and the last cell is left empty to allow for the parts to be external. For this machine assignment an initial part allocation is obtained by solving the allocation submodel. Thus, an initial solution (PFs and MGs) and the objective function value are obtained. At each subsequent iteration one machine is moved from the current cell to another cell in order to get a new machine assignment. The machine to be moved and the cell for this machine are selected randomly. This is done in such a way that the new cell does not already have a unit of the same machine type. Part allocation is made for this new machine assignment and the objective value is computed.

The detailed steps of the proposed implementation of SAA considering alternate routings are presented below:

Simulated Annealing Algorithm (SAA)

[0] Initialize

0.1 Define the annealing parameters: initial temperature T_0, minimum accepted transition at each temperature AT_{min}, decrementing factor α, maximum number of iterations i_{max}, and final acceptance ratio R_f.

0.2 Initialize iteration counter: i = 0

0.3 Generate initial machine assignment and allocate parts by solving the allocation submodel (get SOL^0, OBJ^0).

[1] Execute outer loop, i.e., steps (1.1 - 1.7) until conditions in step 1.7 are met

1.1 Initialize inner loop counter $l = 0$, and accepted number of transitions AT = 0

1.2 Initialize solution for inner loop, $SOL_0^i = SOL^i$, $OBJ_0^i = OBJ^i$

1.3 Execute inner loop, i.e., steps (1.3.1 - 1.3.5) until conditions in step 1.3.5 are met

1.3.1 Update: $l = l + 1$

1.3.2 Generate a neighboring solution by perturbing machine assignment and obtaining part allocation for new machine assignment (get SOL_l^i, OBJ_l^i.)

1.3.3 $\delta = OBJ_l^i - OBJ_{l-1}^i$

1.3.4 If $\delta \leq 0$ or random(0,1) $\leq e^{-\frac{\delta}{T_i}}$ then

- SOL_l^i and OBJ_l^i are accepted
- Update AT = AT + 1

else

- solution is rejected, $\text{SOL}_l^i = \text{SOL}_{l-1}^i$,$\text{OBJ}_l^i = \text{OBJ}_{l-1}^i$

1.3.5 If one of the following conditions holds true: $\text{AT} \geq \text{AT}_{min}$ or $l \geq$ M1xM1 (M1 = number of machines), then assign L_i (length of Markov chain) = l, terminate the inner loop and go to 1.4, else continue the inner loop and go to 1.3.1.

1.4 Update: i = i + 1

1.5 Update: $\text{SOL}^i = \text{SOL}_{L_{i-1}}^{i-1}$, $\text{OBJ}^i = \text{OBJ}_{L_{i-1}}^{i-1}$

1.6 Reduce the cooling temperature: $\text{T}_i = \alpha\, T_{i-1}$

1.7 If one of the following conditions holds true: $\text{i} \geq \text{i}_{max}$; or the acceptance ratio (defined as $\text{AT}/L_i) \leq \text{R}_f$; or the objective function value for the last 20 iterations remains the same, then terminate the outer loop and go to 2, else continue the outer loop and go to 1.1.

[2] Print the best solution and terminate the procedure.

5.4 Numerical Examples

In this section we consider four problems to illustrate the application of the model developed. The weights for exceptional elements for all part machine combinations are assumed to be the same in these problems, i.e., $w_{pm} = w \;\forall\; pm$. The first problem is from Kusiak (1987). The second problem shows how this model considers the tradeoff between exceptional elements and voids. Problem 3 illustrates the ability of the model to form loose or tight cells by changing the value of weights for exceptional elements and voids. In the above problems it is assumed that only one unit of each machine is available. The last problem shows that the groupability can be improved by considering the additional units of same machines as available. All these problems were formulated as linear models (LM) and solved optimally using Hyperlindo on PC486/33Hz.

PROBLEM 1: The first problem is given in Table 5.2. This problem was solved for weight $w = 0.5$. Thus, an equal weight is given to the exceptional elements and the voids. The following solution was obtained.

PF1 = {1(2), 3(2)}, PF2 = {2(2), 4(2), 5(2)}.

Objective function value = 0.5.

Number of voids = 1, Number of exceptional elements = 0.

This is the same solution as obtained by Kusiak (1987).

PROBLEM 2: To illustrate the tradeoff between voids and exceptional elements, the problem shown in Table 5.4 is solved for weights, $w = 0.5$ (case 1), 0.3 (case 2), and 0.7 (case 3). In case 1 both are given equal weights, in case 2 the exceptional elements are given less weights than the voids, and in case 3 the exceptional elements are given more weights than the voids. The solutions obtained for the three cases are as follows.

Case-1

PF1 = {1(2), 3(2)}, PF2 = {2(2), 4(2), 5(2)}.

MG1 = {2, 4}, MG2 = {1, 3}.

Objective function value = 0.5.

Number of voids = 1, Number of exceptional elements = 0.

Case-2

PF1 = {1(2), 3(2)}, PF2 = {2(2), 4(2), 5(3)}.

MG1 = {2, 4}, MG2 = {1, 3}.

Objective function value = 0.3.

Number of voids = 0, Number of exceptional elements = 1.

Case-3

PF1 = {1(2), 3(2)}, PF2 = {2(2), 4(2), 5(2)}.

MG1 = {2, 4}, MG2 = {1, 3}.

Objective function value = 0.3.

Number of voids = 1, Number of exceptional elements = 0.

Cases 1 and 3, give 1 void and no exceptional elements, while case 2 gives 1 exceptional element and no voids. This can be explained as follows. In case 1, an equal weight is given to the voids and the exceptional elements; so, any one of the above result will give the same objective function value of 0.5. The model selects the first solution. In case 2, the exceptional elements are given less importance (weight of 0.3) than the voids. Hence, the model finds a solution which gives no void. In case 3, the situation is reversed.

PROBLEM 3: This problem (see Table 5.6) is considered to illustrate how changing weights allow the formation of loose or tight cells. The problem was solved for two weights, 0.3 and 0.7. For weight equal to 0.3, i.e., when exceptional elements have less weight relative to voids, the solution obtained is shown in Table 5.7. Three cells are formed. The process plans for parts were such that machine 6 was not selected. No voids and 4 exceptional elements resulted. Thus the cells here are tight. If we look at the solution corresponding to the weight, $w = 0.7$ (shown in Table 5.8), we have 3 voids, only 1 exceptional element and all the machines are assigned to one of the cells. Here the cells are loose compared to the former case.

PROBLEM 4: Finally, we show the improvement in decomposition by considering additional units of machines as available. To illustrate this we modified the following plan in PROBLEM 3:

part	process plan	machines required
5	2	1, 5, 9, 10

The problem was solved for weight = 0.3 and with only 1 unit of each machine type being available. The result is shown in Table 5.9. The partition yields 4 exceptional elements and no voids. If two units of machine type 1 are available the resulting partition for the same weight, i.e., w = 0.3, is shown in Table 5.10. In this case the number of exceptional elements has decreased from 4 to 3.

5.5 Computational Experience

In this section we will discuss the computational experience with AAA and SAA. First we compare the quality of the AAA solutions with the optimal solutions in section 5.5.1. Then we compare the quality of the SAA solutions with the optimal solutions for an example problem in section 5.5.2. Finally, computational experience with SAA for larger problems is presented in section 5.5.3. The following values were selected for the parameters in the SAA.

T_0=5, AT_{min}=30, i_{max}=300, R_f=0.01 and α =0.98

5.5.1 Comparison of AAA with optimal model

We considered PROBLEM 3 (two cases) and PROBLEM 4 (case 1 which considers 1 unit of machine) to compare the performance of AAA with optimal solutions. The following table shows the comparison.

PROBLEM	AAA			Optimal from LM		
	Voids	Exceptional elements	Objective function value	Voids	Exceptional elements	Objective function value
3 (case 1)	2	5	2.9	0	4	1.2
3 (case 2)	3	1	1.6	3	1	1.6
4 (case 1)	2	5	2.9	0	4	1.2

The AAA identified the optimal solution for 1 problem and nonoptimal solution for 2 problems. In our experience, even for small problems, this iterative procedure was observed to converge to a local minimum for most initial solutions provided.

5.5.2 Comparison of SAA with optimal model and iterative procedure

All the eight problems: PROBLEM 1, PROBLEM 2 (three cases), PROBLEM 3 (two cases) and PROBLEM 4 (two cases) were solved using SAA. The SAA gave an optimal or an alternate optimal solution for all these problems. Next we considered a 15 part, 10 machine problem solved by Kasilingam and Lashkari (1991). From the information given about tools required by parts and tools available on the machines, the data are translated into a part machine matrix as shown in Table 5.11. The rearranged part machine matrix from their solution is shown in Table 5.12. This solution yields 24 voids and 6 exceptional elements. We solved this problem using SAA for a value of w equal to 0.8. SAA identified a partition with fewer voids and exceptional elements (20 voids and 5 exceptional elements). The solution is shown in Table 5.13.

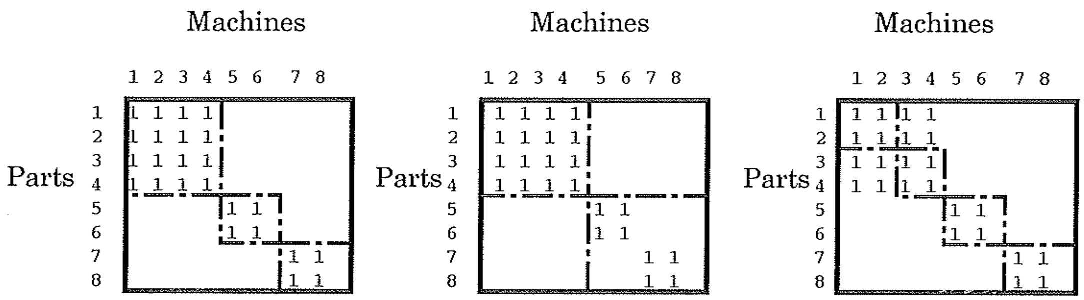
5.5.3 Performance of SAA for larger problems

In this section, SAA is applied to large problems. Five large problems are generated from PROBLEM 3 (Table 5.6) by replicating rows and columns of the matrix. For example, a 100 part, 50 machine problem can be generated from PROBLEM 3 (10 parts and 10 machines) by replicating each row 10 times and each column 5 times. We can compute the optimum objective function value for the generated problems using the solution of PROBLEM 3. This can be done as follows. Table 5.8 shows the optimal partition for a value of w=0.7. The objective function value for this solution is 0.7x1+0.3x3=1.6. The optimum objective value of a generated problem which is obtained by copying each row a times and each column b times will be 1.6xaxb. Problems up to 300 parts (with 720 process plans) and 50 machines were solved on a Sun Sparc 2 station. A summary of the computational results is provided in Table 5.14. SAA identified the optimal solutions for all the problems.

5.6 Summary

In this chapter, a nonlinear integer programming model was developed for cell formation. The model accounts for the possibility of having alternate process plans for the parts and additional units of same machines. This improves the possibility of obtaining a good block diagonal matrix. The objective of the model is to minimize the weighted sum of voids and exceptional elements. Small problems can be solved using the linear model presented. For solving large problems a Simulated Annealing Algorithm is developed. The solutions obtained using the Simulated Annealing Algorithm were found to be optimal for all the problems tested.

Table 5.1: Effects of forming different number of cells.



(a) C = C* (b) C < C* (c) C > C*

Table 5.2: Original part machine matrix from Kusiak (1987).

	Machines →	1	2	3	4
Part(process plan)	1 (1)			1	1
	(2)		1		1
	(3)	1	1		
	2 (1)		1	1	
	(2)	1		1	
	3 (1)	1			1
	(2)		1		1
	4 (1)	1			1
	(2)	1		1	
	5 (1)			1	1
	(2)	1			

PROBLEM 1

Table 5.3: Rearranged matrix (SOLUTION 1).

	Machines →	2	4	1	3
Part (process plan)	1 (2)	1	1		
	3 (2)	1	1		
	2 (2)			1	1
	4 (2)			1	1
	5 (2)			1	

Table 5.4: Modified part machine matrix.

	Machines →	1	2	3	4
Part (process plan)	1 (1)			1	1
	(2)		1		1
	(3)	1	1		
	2 (1)		1	1	
	(2)	1		1	
	3 (1)	1			1
	(2)		1		1
	4 (1)	1			1
	(2)	1		1	
	5 (1)			1	1
	(2)	1			
	(3)	1	1	1	

PROBLEM 2

Table 5.5: Rearranged matrix (SOLUTION 2).

	Machines →	2	4	1	3
Part (process plan)	1 (2)	1	1		
	3 (2)	1	1		
	2 (2)			1	1
	4 (2)			1	1
	5 (3)	1		1	1

Table 5.6: A 10 parts and 10 machines problem.

	Machines →	1	2	3	4	5	6	7	8	9	10
	1 (1)	1	1		1						
	(2)				1		1				
	(3)			1	1						1
	2 (1)							1	1		1
	(2)	1				1				1	1
	3 (1)			1				1	1		
	(2)	1				1		1			
	(3)	1	1								1
	4 (1)			1				1		1	
	(2)			1				1	1		
	5 (1)	1				1					1
	(2)		1			1				1	1
Part (process plan)	6 (1)	1	1	1							
	(2)			1				1	1		
	(3)					1	1	1			
	7 (1)		1		1		1				
	(2)	1	1		1						
	(3)				1			1	1		
	8 (1)	1				1		1		1	1
	(2)	1								1	1
	9 (1)			1				1	1		1
	(2)			1				1		1	
	10 (1)	1		1					1		
	(2)			1				1	1		

PROBLEM 3

Table 5.7: Rearranged matrix for weight = 0.3.

Part (process plan)	Machines 1	2	4	5	9	10	3	7	8	6
1 (1)	1	1	1							
7 (2)	1	1	1							
2 (2)	1			1	1	1				
5 (2)		1		1	1	1				
8 (1)				1	1	1		1		
3 (1)							1	1	1	
4 (2)							1	1	1	
6 (2)							1	1	1	
9 (1)						1	1	1	1	
10 (2)							1	1	1	

Table 5.8: Rearranged matrix for weight = 0.7.

Part (process plan)	Machines 1	5	9	10	2	4	6	3	7	8
2 (2)	1	1	1	1						
5 (1)	1	1		1						
8 (2)	1		1	1						
1 (2)						1	1			
7 (1)					1	1	1			
3 (1)								1	1	1
4 (2)								1	1	1
6 (2)								1	1	1
9 (1)				1				1	1	1
10 (2)								1	1	1

Table 5.9: Rearranged matrix for weight = 0.3 and 1 unit of each machine.

Part (process plan)	Machines: 1	2	4	5	9	10	3	7	8	6
1 (1)	1	1	1							
7 (2)	1	1	1							
2 (2)	1			1	1	1				
5 (2)	1			1	1	1				
8 (1)				1	1	1		1		
3 (1)							1	1	1	
4 (2)							1	1	1	
6 (2)							1	1	1	
9 (1)						1	1	1	1	
10 (2)							1	1	1	

Table 5.10: Rearranged matrix for weight = 0.3 and 2 units of machine type 1.

Part (process plan)	Machines: 1	2	4	1	9	10	3	7	8	5	6
1 (1)	1	1	1								
7 (2)	1	1	1								
2 (2)				1	1	1				1	
5 (2)				1	1	1				1	
8 (2)				1	1	1					
3 (1)							1	1	1		
4 (2)							1	1	1		
6 (2)							1	1	1		
9 (1)						1	1	1	1		
10 (2)							1	1	1		

Table 5.11: Input matrix from Kasilingam and Lashkari (1991).

	Machine type →	1	2	3	4	5	6	7	8	9	10
	Total units →	2	1	1	1	1	1	1	1	2	2
	1 (1)	1		1							1
	2 (1)					1				1	1
	(2)	1		1		1				1	
	(3)	1		1		1				1	
	3 (1)	1				1		1			
	(2)					1		1			
	4 (1)			1		1					1
	(2)		1			1					1
	(3)					1				1	1
	5 (1)			1	1					1	1
	6 (1)	1	1				1		1		
	(2)	1	1				1			1	
Part	7 (1)			1	1					1	
(process plan)	8 (1)	1		1	1						
	(2)	1			1						
	9 (1)						1		1		1
	10 (1)	1		1					1	1	
	11 (1)	1	1				1	1			
	(2)		1				1			1	
	(3)		1				1	1	1		
	12 (1)					1		1			
	(2)					1		1			1
	13 (1)			1					1	1	
	14 (1)	1	1				1				
	(2)	1	1						1		
	15 (1)			1		1		1		1	
	(2)					1		1		1	1
	(3)					1		1	1		

<u>**PROBLEM 5**</u>

Table 5.12: Solution from Kasilingam and Lashkari (1991).

Part (process plan)	Machines												
	1	3	4	10	1	2	6	8	9	5	7	9	10
1 (1)	1	1		1									
5 (1)		1	1	1					1				
7 (1)		1	1						1				
8 (2)	1		1										
10 (1)	1	1						1	1				
6 (2)					1	1	1		1				
9 (1)				1			1	1					
13 (1)		1						1	1				
14 (1)					1	1	1						
11 (2)						1	1		1				
2 (1)										1		1	1
3 (2)										1	1		
4 (3)										1		1	1
12 (2)										1	1		1
15 (2)										1	1	1	1

Table 5.13: SAA solution for weight = 0.80.

		Machines												
		6	2	8	1	10	3	9	4	1	10	9	5	7
Part (process plan)	6 (1)	1	1	1	1									
	9 (1)	1		1		1								
	11 (1)	1	1		1									1
	14 (1)	1	1		1									
	1 (1)						1			1	1			
	5 (1)						1	1	1		1			
	7 (1)						1	1	1					
	8 (1)						1		1	1				
	10 (1)			1			1	1		1				
	13 (1)			1			1	1						
	2 (1)										1	1	1	
	3 (2)												1	1
	4 (3)										1	1	1	
	12 (2)										1		1	1
	15 (2)										1	1	1	1

Table 5.14: Computational experience with large problems.

Prob. No.	Number of		Objective value		Computation time (in minutes)
	Parts (process plans)	Machines	SAA	Optimum	
1	50 (120)	30	24	24	15.03
2	100 (240)	20	32	32	15.81
3	100 (240)	30	48	48	53.76
4	200 (480)	40	128	128	83.26
5	300 (720)	50	240	240	244.97

Chapter 6

Cell Formation Considering Material Handling

Cellular manufacturing aims to achieve the decomposition of a manufacturing system into smaller subsystems or cells. One motivating factor for introducing a cell system is reduction in material handling. In this chapter we address the cell formation problem for an existing shop. The objective considered is to minimize the material handling costs. Sequence of operations, cell size, options of assigning an operation to alternate machines, and additional units of same machine are considered. The problem becomes intractable when the above factors are considered simultaneously. So we develop a two stage procedure. In stage 1, machines are partitioned into machine groups minimizing the total material handling cost. In the calculation of material handling costs: production quantity, effect of cell size on intra cell handling cost, effect of sequence of operations and multiple visits to the same machine by parts, are considered. All the similar machines are lumped into 1 machine type. Although there may be other machines which can perform the same operation, in stage 1 we consider only one principal machine for each operation. Generating options to perform each operation on alternate machines are neither easy nor required. In stage 2, we consider generating other options for only those operations identified as exceptional elements in stage 1 solution. Also, additional units of machines available from stage 1 solution are considered for reassignment. Thus, in stage 2, by re-allocating operations that resulted in exceptional elements and reassigning all the extra units of machines available in the cells, the solution is further improved.

The remainder of the chapter is organized as follows. Section 6.1, presents the stage 1 procedure. The mathematical model and Simulated Annealing Algorithm are developed in this section. The tradeoff between intra and inter cell material handling cost trade is also illustrated followed by a comparison with previously published results. In section 6.2, we develop an integer programming model for stage 2. We illustrate the two stage procedure developed in this chapter by considering an example problem in section 6.3. Sections 6.4 and 6.5 provide discussion and summary respectively.

6.1 Stage 1: Initial Grouping

At stage 1, initial machine and part groupings are obtained. We make the following assumptions:

1. All the similar machines are lumped into 1 machine type.
2. A particular operation of a part can be performed by only one type of machine.
3. The unit intra cell move cost is a function of the number of machines assigned to the cell. We assume the following form of variation of the intra cell cost per move per unit of material (HCW_p):

$$\text{HCW}_p(N^c) = \lambda_0^p + \lambda_1^p.N^c, \qquad \text{for } N^c \geq 2 \tag{6.1}$$

   where, N^c is the number of machines assigned to the cell; and λ_0^p & λ_1^p are constants. If there is only one machine in the cell the intra cell handling cost is 0 and the function is defined only for more than 1 machine in a cell. Sankaran and Kasilingam (1993) made a similar assumption about the intra cell material handling cost and represented it by a stepwise linear function. It should be noted that the simulated annealing algorithm presented in this chapter to solve model M3.1 is not restricted to a linear form.

If assignment of machines to cells is known, then from assumptions (1) and (2), part routing is unique and material handling costs can be calculated without actually associating the parts to any cell. This means that the material handling cost for a given machine assignment will be the same regardless of the part allocation. Model M3.1 identifies machine groups. After machine groups are identified, each part is allocated to a cell which results in a minimum number of exceptional elements. This allocation is necessary only to proceed to stage 2.

6.1.1 Mathematical model

In this section a nonlinear integer programming model is developed for identification of machine groups to minimize total material handling costs. The model is as follows:

MODEL - M3.1

$$\text{Min } z = \sum_c \sum_p \sum_m \sum_{m'} \sum_o HCW_p(\sum_m N_m y_{mc})\, c^o_{pmm'}\, y_{mc}\, y_{m'c}\, Q_p \; + $$

$$\sum_c \sum_p \sum_m \sum_{m'} \sum_o HCB_p\, c^o_{pmm'}\, y_{mc}\, (1 - y_{m'c})\, Q_p \tag{6.2}$$

subject to:

$$\sum_c y_{mc} = 1 \qquad \forall\ m \tag{6.3}$$

$$y_{mc} \in \{0,1\} \quad \forall\ m, c \tag{6.4}$$

The objective function given by equation (6.2) has two nonlinear terms. The first term captures the contribution of the intra cell handling costs and the second term , the inter cell handling costs. Constraints (6.3) guarantee that each machine type is allocated to a cell. Binary restrictions on y variables are imposed by constraints (6.4).

C, the upper limit on the number of cells is given as one more than the total number of machines and the model will determine the optimal number of cells to be formed. We use a simulated annealing procedure to solve the model.

6.1.2 Simulated Annealing Algorithm (SAA)

In this section an implementation of the SAA to obtain the machine grouping is presented. This implementation is similar to that in Chapters 4 and 5. For generating a starting solution, the number of cells is set equal to $M+1$, where M is the number of machine types. Each machine type is assigned to a separate cell. The last cell, i.e., $M+1$, is left empty to allow for the parts to be external. For this machine assignment, an initial objective function value is calculated from equation (6.2). Thus, an initial solution (MGs) and the objective function value are obtained. At each subsequent iteration one machine is moved from the current cell to another cell in order to get a new machine assignment. The machine to be moved and the cell for this machine are selected randomly. The objective value is then computed for this new machine assignment.

The detailed steps of the proposed implementation of SAA are presented below.

Simulated Annealing Algorithm (SAA)

[0] Initialize

0.1 Define the annealing parameters: initial temperature T_0, minimum accepted transition at each temperature AT_{min}, decrementing factor α, maximum number of iterations i_{max}, and final acceptance ratio R_f.

0.2 Initialize iteration counter: i = 0

0.3 Generate initial machine assignment and compute the objective value (get SOL^0, OBJ^0).

[1] Execute outer loop, i.e., steps (1.1 - 1.7) until conditions in step 1.7 are met

1.1 Initialize inner loop counter $l = 0$, and accepted number of transitions AT = 0

1.2 Initialize solution for inner loop, $SOL^i_0 = SOL^i$, $OBJ^i_0 = OBJ^i$

1.3 Execute inner loop, i.e., steps (1.3.1 - 1.3.5) until conditions in step 1.3.5 are met

1.3.1 Update: $l = l + 1$

1.3.2 Generate a neighboring solution by perturbing machine assignment and compute the objective value for new machine assignment (get SOL^i_l, OBJ^i_l.)

1.3.3 $\delta = OBJ^i_l - OBJ^i_{l-1}$

1.3.4 If $\delta \leq 0$ or random(0,1) $\leq e^{-\frac{\delta}{T_i}}$ then

- SOL^i_l and OBJ^i_l are accepted
- Update AT = AT + 1

else

- solution is rejected, $SOL^i_l = SOL^i_{l-1}$,$OBJ^i_l = OBJ^i_{l-1}$

1.3.5 If one of the following conditions holds true: $AT \geq AT_{min}$ or $l \geq 5M^2$ (M = number of machine types), then assign length of Markov chain,

$L_i = l$, terminate the inner loop and go to 1.4, else continue the inner loop and go to 1.3.1.

1.4 Update: i = i + 1

1.5 Update: $\text{SOL}^i = \text{SOL}^{i-1}_{L_{i-1}}$, $\text{OBJ}^i = \text{OBJ}^{i-1}_{L_{i-1}}$

1.6 Reduce the cooling temperature: $T_i = \alpha\ T_{i-1}$

1.7 If one of the following conditions holds true: $i \geq i_{max}$; or the acceptance ratio (defined as AT/L_i) $\leq R_f$; or the objective function value for the last 20 iterations remains the same, then terminate the outer loop and go to 2, else continue the outer loop and go to 1.1.

[2] Print the best solution obtained and terminate the procedure.

In this chapter, the value of simulated annealing parameters were selected as follows: The initial value of cooling temperature T_0 is selected in such a way that the acceptance ratio (ratio of number of accepted solutions to number of generated solutions) is close to unity. We used $T_o = 5$ when demand for parts was 1 and $T_o =$ 100xP (where, P = number of parts) when part demand was of the order 100. The following values were selected for the other parameters:

$AT_{min} = 100$; $\alpha = 0.99$; $i_{max} = 300$; and $R_f = 0.01$.

6.1.3 Illustration of inter and intra cell tradeoffs

The tradeoff between intra cell and inter cell handling costs is illustrated in this section. A six part, five machine problem is considered and the part machine matrix is shown in Table 6.1. Costs of intra cell material handling vary as given by equation 6.1. The following values are assumed for the constants:

$\lambda_0^p = 0.5$; $\lambda_1^p = 0.5$ $\forall p$.

We considered three cases. All the problems were solved using SAA. In case 1, inter cell handling cost is taken as \$10 per operation for each part. This solution gives 1 cell with the objective value of \$27 (Table 6.1). The high cost of inter cell handling forces it to avoid any inter cell movement. In case 2, the cost of inter cell handling is reduced to \$5. This forms 2 cells with the objective value of \$23 (\$13 for intra cell and \$10 for two inter cell moves). The resulting partition is shown in Table 6.2. In case 3, the cost of inter cell handling is kept as \$2. The solution gives three cells as shown in Table 6.3. The objective function value for the solution is \$16 in this case (\$8 for intra cell and \$8 for inter cell moves). Also, in this case machine 5 which is required by parts from two cells, is assigned to the remainder cell, cell 3. As the inter cell move cost reduces, more cells are formed.

6.1.4 Comparison of results

In this section we provide a comparison of SAA with published results. The inter cell flow reduction heuristic due to Okogbaa et al. (1992) and the heuristic procedure by Harhalakis et al. (1990) consider the sequence of operations of parts in calculating the inter cell moves and they capture the material handling moves exactly. However, they have not considered the intra cell move. For a predefined maximum number of machines in a given cell, these procedures obtain a grouping which minimize the total inter cell material handling. To make the comparison with the above two procedures consistent with the procedure developed in this chapter, we define intra cell move cost as follows. When the number of machines in a cell is less than or equal to the maximum specified number of machines in a cell the intra cell cost is taken as 0 otherwise a very high cost is given to the intra cell move cost. This does not allow formation of a cell which has more machines than the limit specified. The inter cell handling cost per move is taken as \$1 per part per move, because the two heuristics mentioned above consider minimizing the total flow of parts.

Okogbaa et al. (1992) considered three problems with part machine matrices of size: (i) 14x7; (ii) 7x5; and (iii) 43x16. Each of the problems was solved for two cases. For problem (i) and (ii), two demand patterns were used. For problem (iii) two different cell size restrictions were considered. The data and solutions obtained by SAA for each of the problems are shown in Tables 6.4 to 6.9. The solution obtained by SAA for these problems were the same as in Okogbaa et al. (1992). Harhalakis et al. (1990) considered a 20 part, 20 machine problem shown in Table 6.10. Two cases were considered. The demand for parts are kept uniform in both cases while the maximum number of machines in a cell are 5 and 7 for case 1 and 2 respectively. SAA solution for case 1 is shown in Table 6.11. Harhalakis et al. (1990) obtained the same solution with the number of inter cell moves as 17. The case 2 solutions from Harhalakis et al. (1990) and SAA are shown in Tables 6.12 and 6.13 respectively. SAA gave a better solution that resulted in 13 inter cell moves as compared to 14 from Harhalakis et al. (1990). We can also observe in these solutions that although the number of exceptional elements in a solution by SAA is more than that obtained by Harhalakis et al. (1990) the total inter cell moves is less. This illustrates the importance of considering the sequence of operations. SAA thus compares favorably with the existing procedures. In addition, SAA can consider the effect of cell size on intra cell material handling. An illustrative example considering the effect of cell size is provided in section 6.4.

6.2 Stage 2: Improvement in Grouping

Stage 1 gives the initial part machine groups. This solution may contain a number of exceptional elements. The exceptional elements can be eliminated if the following options are available: if an extra unit of the machine that is required to process the exceptional element can be moved to the cell where the part is allocated; or the operation can be re-scheduled on a different machine type that is available in the part's parent cell. If the first option is chosen, the inter cell move of the part is eliminated but at the same time, since, other parts in a cell do not need processing on this machine the intra cell move costs for these parts will increase. If the second option is selected, the processing cost for that operation will increase in addition to an increase in intra cell move cost corresponding to that operation. In this section, we consider these two options to improve the grouping if this reduces material handling further. Other options, such as further dividing these operations into two or more sub-operations could also be considered at this stage.

Before, we start stage 2 we modify the stage 1 solution as follows. All the exceptional elements are removed. Loads due to the remaining operations on each type of machine is calculated. Based on this load the required number of units of each machine type is determined. Extra units of machines are removed and considered for reassignment. The new objective function value after removing the exceptional elements and the extra units of machines is calculated.

Stage 2 considers reducing the exceptional elements by assigning a unit of the machine to the cell or by re-allocating the operation to alternate machine types within cells. Cell size increases due to the reassignment of the extra units of machines.

Because of the cell size increase, cost per intra cell move also increases. We compute this increase in cost for all the operation assignment which were made in the part's parent cell in stage 1. However, for the operations that resulted in exceptional elements in stage 1, this increase is small and can be ignored. If an exceptional element is re-allocated to the parent cell the intra cell handling cost increases; and if it allocated to a different cell inter cell handling cost increases. We assume that if the operation is the first or the last operation, it contributes to one intra/inter cell handling and if it is an intermediate operation it contributes to two intra/inter cell handling upon re-allocation. The sequence of operations of parts is exactly captured when the exceptional elements do not correspond to two consecutive operations in one cell.

6.2.1 Mathematical model

We develop an integer programming model for stage 2. The model is as follows:

MODEL - M3.2

$$\text{Min z} = \sum_{c}\sum_{m}\sum_{p}\sum_{e} C'^{e}_{pm} x^{e}_{pmc} Q_e +$$

$$\left(\sum_{c}\sum_{p} \Delta HCW^{c}_{p} \sum_{m}\sum_{f} y^{f}_{mc} + \sum_{c}\sum_{p}\sum_{e}\sum_{m} \alpha^{e}_{p}\, HCW^{c}_{p}\;\; x^{e}_{pmc} Q_p\right)$$

$$+ \quad \sum_{c}\sum_{p}\sum_{e}\sum_{m} \alpha^{e}_{p}\, HCB^{c}_{p}\;\; x^{e}_{pmc}\, Q_p \qquad (6.5)$$

subject to:

$$\sum_{m}\sum_{c} x^{e}_{pmc} = 1 \qquad \forall\, p, e \qquad (6.6)$$

$$\sum_{c} y^{f}_{mc} = 1 \qquad \forall\, m, f \qquad (6.7)$$

$$\sum_p \sum_e Q_p t^e_{pm} x^e_{pmc} \leq b^0_{mc} + \sum_f b_m \ y^f_{mc} \qquad \forall \ m, c \tag{6.8}$$

$$x^e_{pmc}, y^f_{mc} \in \{0, 1\} \qquad \forall \ e, p, f, m, c \tag{6.9}$$

The objective function represented by equation (6.5) has 4 terms. The first term captures the incremental processing cost. This is incurred when an operation is not assigned to the principal machine. The second term captures the increase in intra cell handling of assigned parts due to assignment of extra units of machines in the cells. The third term shows the contribution of intra cell handling of all the exceptional elements. The fourth term captures the total inter cell handling costs. Constraints (6.6) state that all the exceptional elements should be re-allocated to a cell. Constraints (6.7) ensure that each additional unit of a machine is assigned to some cell. Machine capacity availability is represented by constraints (6.8). Finally, binary restrictions on x and y variables are imposed by constraints (6.9). If the splitting of part demands to different machines is allowed then variable x represents the fraction of demand produced and thus can be specified as a continuous variable.

6.3 An Example Problem

In this section we illustrate the two stage procedure with an example problem. We consider the part machine matrix from Harhalakis et al. (1990) shown in Table 6.10. We generate other required data for this problem as shown in Table 6.14. Processing times of operations are generated randomly between 2 and 5. It is assumed that machines have different capacities to take into account the variability in down time, number of shifts the machine can be operated, availability of special tools or operator etc. Capacities of different machine types are selected between 1,000 and 3,000 time units. Demands for parts are generated between 100 and 250. Based on this information, the load on each type of machine is calculated. The required number of units L_m, of machine type m is the ratio of total load on this machine type to its capacity b_m. L_m is rounded to the next integer, say $\overline{L_m}$. The number of units of each type of machine available N_m was assumed to be equal to $\overline{L_m}$ for this example.

The following parameter values were used in equation (6.1) to calculate intra cell move cost per unit of material moved $\mathrm{HCW}_p(\mathrm{N}^c)$.

$\lambda_0^p = 0.8, \qquad \lambda_1^p = 0.1 \qquad \forall p$

The cost of inter cell move per unit per move HCB_p, was assumed to be \$5 for all part.

Stage 1: We used data from Table 6.14 on intra cell and inter cell handling costs, part demand, number of units of each machine type and operation sequence of parts to solve model M3.1 to obtain the machine groups. Although we use the number of units of same machine type in cell size calculation, we assign all the units to the same cell. Parts are assigned to the cell where it results in a minimum number of exceptional elements. The part machine groupings thus obtained is shown in Table 6.15. There are 15 exceptional elements in this solution. The total handling cost for the solution is \$25,270 (\$12,270[1] + \$13,000[2]). It is also worth pointing out that machine 5 is assigned to cell 4 and it processes 3 parts (parts 6, 7 and 19) from cell 2 and only one part (part 12) from cell 4. It appears that for machine 5, cell 2 would be the better cell assignment than cell 4. However, in doing so it will increase the net handling cost. The operations of parts 6, 7 and 19 on machine 5 are either the first or the last operation while the operation of part 12 corresponds to an intermediate operation. In the present assignment a total of 350 units are handled once, while if machine 5 is assigned to cell 2, 250 units of part 12 will have to be moved twice and the inter cell handling cost will increase by \$250. Although the intra cell

[1]intra cell handling cost; [2]inter cell handling cost

handling cost will reduce by \$139 there will be a net increase of \$111 in handling cost.

Stage 2: We now improve the stage 1 solution. We do not change the allocation of any part operation which is carried out in the part's parent cell. Thus, only exceptional elements resulting from the stage 1 solution are considered for re-allocation. Similarly, we reassign only additional units of machines available in the cells which are not required for completing the allocated part operations. In Table 6.16, units of machine which can be reassigned are determined in the following way. The load on each type of machine due to all the allocated operations in the parent cell, i.e., all operations excluding the exceptional elements, is calculated. The sum ν_r, is rounded to the next integer ν_a. After assigning ν_a units of machines to the parent cell, any additional machine units available are considered for the reassignment. This ensures that the capacity requirement is met for all the allocated part operations in the parent cell. From the table we observe that machine types, 17, 20, 10 and 4 have 1 extra unit each. After removing the extra units from the cell we compute the remaining time available b^0_{mc}, on each machine type. The total number of machines remaining in each cell, N^0_c, and the cost of intra cell per move $\mathrm{HCW}_p(\mathrm{N}^0_c)$ for the corresponding cell are given in Table 6.17. The objective function value after removing extra units of machines and the exceptional elements is recomputed to be \$11,170.

Table 6.17 shows the calculation of change in the intra cell move cost of each part allocated in a cell ΔHCW^c_p, that would result if an additional machine is assigned to the cell. To re-allocate exceptional elements we consider alternate machines where these can be processed. Table 6.18 provides information about the alternate machine and cell combination, processing time and an incremental processing cost to perform each exceptional element. If the principal machine is chosen then the incremental processing cost is 0. Table 6.18 also shows the value of α_p^e. Model M3.2 was formulated using the information on Tables 6.16 to 6.18 and solved using Hyperlindo software. Two solutions were obtained. In the first case splitting of part demand into machines was not allowed by restricting the x variables to take a binary value. The solution is shown in Table 6.19. The number of exceptional elements is reduced from 15 in stage 1 solution to 9. The objective function value is \$12,650. Model M3.2 gave the cost of re-allocating exceptional elements and extra units of machines. We have the cost of material handling from stage 1 after removing exceptional elements and extra units of machines as \$11,170. Adding these two costs gives the total cost as \$23,820 (\$11,170 + \$12,650). The cost thus computed may differ from the actual cost directly obtained from the final part machine groupings shown in Table 6.19. This is because in the formulation of model M3.2 we ignore the effect of increase in cell size on intra cell move cost of exceptional elements. Also, if two consecutive operations are performed in the same cell it results in overestimating the number of moves.

The actual cost can be computed from the solution (part machine groupings) shown in Table 6.19 as follows. For each part, the number of inter cell moves and intra cell moves in different cells can be found. Depending upon the number of machines assigned to a cell, the cost of unit intra cell move can be computed. The unit inter cell move cost is taken as \$5. The product of the number of intra/inter cell moves, cost per intra/inter cell move and the part demand will give the total handling cost for the part. The processing cost is calculated by multiplying the incremental cost with part demand for those operations which are performed on the machine other than the principal machine as indicated in Table 6.19.

The actual cost thus obtained is \$23,190 (\$13,190[1] + \$9000[2] + \$600[3]). The two costs differ only by 2.72%.

In the second case splitting of part demand was allowed by defining the x variables as continuous variables) the solution was obtained as shown in Table 6.20. In this solution 6 exceptional elements are completely eliminated and 7 elements are partially transferred to their parent cells. The actual cost for this solution is \$22,650 (\$15,089[1] + \$4857[2] + \$2704[3]) and that obtained from the models is \$22,663 (\$11,170 + \$11,493). Allowing splitting of demands thus gives a lower total handling plus processing cost.

6.4 Discussion

The importance of considering the tradeoff between intra cell and inter cell handling costs is often stated in the literature (McAuley, 1972, Logendran, 1990 & 1991). As the system is decomposed into more cells the total inter cell handling cost increases while the total intra cell handling cost decreases. With an increase in number of cells formed it is likely that the cells are located far apart and physical travel of part between cells increases. Also, the part visits more cells which requires additional coordinating effort. The number of inter cell moves and the cost per inter cell move thus increase. Therefore, the total inter cell handling cost increases. In contrast, as the number of cells increases the cell size in general decreases which results in a decrease in the cost per intra cell move and the number of intra cell moves. Hence the total intra cell handling cost decreases. The material handling cost (MHC) can be expressed in the following form:

$$\mathrm{MHC} = \theta_1(.)\ n_1 + \theta_2(.)\ n_2$$

[1]intra cell handling cost; [2]inter cell handling cost; [3]incremental processing cost

where, $\theta_1(.)$ is a weight reflecting cost per inter cell move;

$\theta_2(.)$ is a weight reflecting cost per intra cell move;

n_1 is the number of inter cell moves; and

n_2 is the number of intra cell moves.

In the above equation $\theta_1(.)$ is a function of number of cells/cell layout; and $\theta_2(.)$ is a function of size of a cell (number of machines or parts) /cell configuration. In the literature, the material handling tradeoff is often expressed by assigning constant values of weights: $\theta_1(.) = \theta_1$ and $\theta_2(.) = \theta_2$ in such a way that $\theta_1 > \theta_2$. The purpose of considering the tradeoff between inter cell and intra cell handling costs is defeated in this case and the objective simply leads to the minimization of number of inter cell moves (or maximization of number of intra cell moves). This can be explained as follows. Let a part require n operations. The sum of the number of intra and inter cell moves is always one less than the number of operations, i.e., $n_1 + n_2 = n - 1$. The objective function is in this case is minimization of:

$$\theta_1 n_1 + \theta_2 n_2 = (\theta_1 - \theta_2)n_1 + \theta_2(n-1) \qquad \text{(by substituting } n_2 = n - 1 - n_1\text{)}$$

In the above expression $\theta_1 > \theta_2$, and all the other parameters except n_1 are constant, the objective function then becomes minimization of n_1 or the number of inter cell moves.

Logendran (1991) considers the variation of weight $\theta_1(.)$ assuming that the layout is known. However, the values of the weights are such that for a given part $\theta_1(.)$ is always greater than θ_2. Unless $\theta_1(.)$ becomes less that θ_2 for some configuration this is equivalent to the minimization of the number of inter cell moves only. The tradeoff between intra and inter cell move can not be exploited and will lead to the formation of one cell unless other restrictions such as minimum number of cells to be formed or maximum number of machines in a cell, are imposed.

McAuley (1972) computes material handling cost for different cell configuration by considering the weight $\theta_2(.)$ as a linear function of number of machines. Weight $\theta_1(.)$ takes a constant value. Depending upon the type of layout three different linear equations are defined for $\theta_2(.)$.

$$\theta_2(.) = \frac{1}{3}(1 + M)\ A \qquad \text{for a straight line layout}$$

$$\theta_2(.) = \frac{1}{3}(\mathrm{R} + M)\ A \qquad \text{for a rectangular layout with R rows}$$

$$\theta_2(.) = \frac{\sqrt{2}}{3}M\ A \qquad \text{for square layout}$$

where, A is a constant; and M is the number of machines in a cell. This allows exploitation of the tradeoff between the inter and intra cell handling costs because as the number of cells decreases the value of $\theta_2(.)$ increases and beyond some point it becomes greater than θ_1.

The above expressions (for assumed layout) were used by McAuley (1972) to evaluate different solutions and it was not a part of the cell formation procedure. In this chapter, the proposed model identifies the part-machine grouping considering the intra cell cost $\theta_2(.)$ as a linear function of the number of machines. The linear function can represent any of the situations considered by McAuley (1972). The model and solution procedure developed in this chapter can be suitably modified to incorporate $\theta_1(.)$ as a function of number of cells and layout.

6.5 Summary

One major advantage of the cell system of production is the reduction in material handling. In this chapter, we proposed a two stage procedure for cell formation. In stage 1, we developed a nonlinear mathematical model to minimize the total intra cell and inter cell material handling costs. In the calculation of material handling costs: production quantity, effect of cell size on intra cell handling, and effect of sequence of operations have been considered. A solution procedure based on simulated annealing was presented. The results compare well with existing procedures. In stage 2, an integer programming model was developed which considers allocating an operation to alternate machines and extra units of machines available to further improve the solution obtained in stage 1. The two stage procedure was illustrated by solving an example problem.

Table 6.1: An un-decomposed system.

		Machines				
		1	2	3	4	5
P	1	1		2^{+}		
a	2		3		1	2
r	3		1		2	
t	4	1		2		3
s	5		2	1		3
	6		1		2	

$^{+}$ sequence of operation of part

Table 6.2: Decomposition into 2 cells.

		Machines				
		1	3	2	4	5
P	1	1	2^{+}			
a	4	1	2			3
r	2			3	1	2
t	3			1	2	
s	5		1	2		3
	6			1	2	

$^{+}$ sequence of operation of part

Table 6.3: Decomposition into 3 cells.

		Machines				
		1	3	4	2	5
P	1	1	2^+			
a	4	1	2			3
r	5		1		2	3
t	2			1	3	2
s	3			2	1	
	6			2	1	

$^+$ sequence of operation of part

Table 6.4: 14x7 matrix from Okogbaa et al. (1992).

		Machines						
		1	2	3	4	5	6	7
	1		1				2	3
	2	2				1		
	3				1			
	4		1					2
P	5	1	3					2
a	6		3				2	1
r	7		1		2			
t	8	1				2	3	
s	9	3			1			2
	10			1				2
	11			1	2			
	12			1	2			
	13			1				
	14	2				1		

Table 6.5: SAA solution for 14x7 matrix in Table 6.4.

Case 1: (a) Data: (i) Maximum cell size = 3 machines
(ii)Uniform part demand

(b) Solution: Machine groups

Cell number	Machine assigned
1	7,2,6
2	5,1
3	4,3

Case 2: (a)Data: (i) Maximum cell size = 3 machines
(ii)Part demand

Part	1	2	3	4	5	6	7
Demand	100	250	200	400	300	100	150
Part	8	9	10	11	12	13	14
Demand	600	300	200	250	300	500	300

(b)Solution: Machine groups

Cell number	Machine assigned
1	2,7
2	1,5,6
3	3,4

Table 6.6: 7x5 matrix from Okogbaa et al. (1992).

		Machines				
		1	2	3	4	5
	1			1	2	
P	2	1		2		
a	3		2		1	3
r	4	1		2		3
t	5		2			1
s	6				1	2
	7	2		1		

Table 6.7: SAA solution for 7x5 matrix in Table 6.6.

(a)Data: (i) maximum cell size = 3 machines
(ii)Distribution of part demand

	Distribution 1 (Case 1)						
Part	1	2	3	4	5	6	7
Demand	100	100	100	100	100	100	100
	Distribution 2 (Case 2)						
Part	1	2	3	4	5	6	7
Demand	500	150	100	100	300	200	150

(b)Solution: Machine groupings

Cell	Machine assignment	
	Distribution 1 (Case 1)	Distribution 2 (Case 2)
1	1,3	3,4,1
2	2,5,4	5,2

Table 6.8: 43x16 matrix from Okogbaa et al. (1992).

Parts	Machines 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
1							1			2						
2		1							2					4		3
3											1		2			
4									1							
5				1	3										2	
6			2											1		
7			2											2		
8					1											
9				1	2										3	
10		1							3							2
11												1				
12										1						
13							1			2						
14				1	2										3	
15					1										2	
16					1										2	
17												1		2		
18					1											
19				1	3										2	
20												1				
21				2	1										3	
22												1				
23				1	2											
24											3	2	1			
25							1			2						
26										1						
27											1	2				
28		2							1							
29				2	1											
30											1	2				

(continued in the next page.....)

(Table 6.8: continued.....)

		Machines															
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
	31										1						
	32		2							2							1
	33					1											
	34			1													
P	35			2											1		
a	36			1													
r	37	1	2							3							4
t	38		2							3							1
s	39										1						
	40		1							2							
	41					1										2	
	42	2	1							3							4
	43				3	2										1	

Typographical error identified in the paper was suitably corrected.

Table 6.9: SAA solution for 43x16 matrix in Table 6.8.

Cell	Machine assignments	
	Case 1 (cell size = 6)	Case 2 (cell size = 4)
1	4,5,11-13,15	4,5,15
2	1,2,3,9,14,16	1,2,9,16
3	7,10	7,10
4		11-13
5		3,14

Part demands were uniform in both cases.

Table 6.10: A 20x20 matrix from Harhalakis et al. (1990).

		Machines																			
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
	1	2								3			1						4		5
	2		3	2								1									
	3								1											3	2
	4		3	1							4	2									
	5				1		3	4								2					
	6					5						1			2		3	4			
	7					1											2	3			
P	8				5			3		4				2		1					
a	9	4								2		3	5						1		
r	10								3											1	2
t	11			3								1			2						
s	12	5				3				1			4						2		
	13						1	2								3		4			
	14	3	4						1		2										
	15													1	2		3	4			
	16						3	2								1				4	
	17	2								1			3								
	18								1		4									2	3
	19		2	1		4						3									
	20	3									2		4						1		

Table 6.11: SAA 5-cell solution of 20x20 matrix in Table 6.10.

(i)Maximum cell size = 5 machines
(ii)Uniform part demand of 1

Machines

Parts	01	09	10	12	18	02	03	11	14	04	06	07	13	15	05	16	17	08	19	20
1	2	3		1	4															5
9	4	2		5	1			3												
12	5	1		4	2										3					
14	3		2			4												1		
17	2	1		3																
20	3		2	4	1															
2						3	2	1												
4			4			3	1	2												
11							3	1	2											
19						2	1	3							4					
5										1	3	4		2						
8		4								5		3	2	1						
13											1	2		3			4			
16											3	2		1					4	
6								1	2						5	3	4			
7															1	2	3			
15									2				1			3	4			
3																		1	3	2
10																		3	1	2
18			4															1	2	3

Total number of inter cell move = 17

Table 6.12: Harhalakis et al.'s (1990) 4-cell solution of 20x20 matrix in Table 6.10.

(i)Maximum cell size = 7 machines
(ii)Uniform part demand of 1

Machines

Parts	01	09	10	12	18	02	03	05	11	14	16	17	04	06	07	13	15	08	19	20
1	2	3		1	4															5
9	4	2		5	1				3											
12	5	1		4	2			3												
14	3		2			4												1		
17	2	1		3																
20	3		2	4	1															
2						3	2		1											
4			4			3	1		2											
6								5	1	2	3	4								
7								1			2	3								
11							3		1	2										
15										2	3	4				1				
19						2	1	4	3											
5													1	3	4		2			
8		4											5		3	2	1			
13												4		1	2		3			
16														3	2		1		4	
3																		1	3	2
10																		3	1	2
18			4															1	2	3

Total number of inter cell move = 14

Table 6.13: SAA 4-cell solution of 20x20 matrix in Table 6.10.

(i)Maximum cell size = 7 machines
(ii)Uniform part demand of 1

Machines

Parts		12	01	19	18	10	04	03	02	11	14	16	17	05	20	09	06	07	15	08	13
	1	1	2	3	4										5						
	9	5	4	2	1					3											
	12	4	5	1	2									3							
	14		3			2			4											1	
	17	3	2	1																	
P	20	4	3		1	2															
a	2							2	3	1											
r	4					4		1	3	2											
t	6									1	2	3	4	5							
s	7											2	3	1							
	11							3		1	2										
	15										2	3	4								1
	19							1	2	3				4							
	3														2	3				1	
	5						1										3	4	2		
	8			4			5											3	1		2
	10														2	1				3	
	13												4				1	2	3		
	16															4	3	2	1		
	18					4									3	2				1	

Total number of inter cell move = 13

Table 6.14: Data for example problem.

Machine type, m →	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	Part demand Q_p
	Processing time of part p on machine type m, t_{pm}																				
Part, p ↓ 1	4								4			3						3		5	200
2		3	4								5										100
3								3											4	3	200
4		3	5							3	3										200
5				3		3	5								3						100
6					2						3			5		3	4				100
7					4											4	5				150
8				3			2		5				4		2						200
9	5								2		3	5						2			250
10								3											2	3	100
11			3								5			3							150
12	3				4				5			2						4			250
13						2	5								3		4				200
14	4	3						2		4											100
15													2	4		5	2				150
16						4	3								3				2		200
17	3								5			5									200
18								3		2									2	2	250
19		4	3		5						2										100
20	3									3		3						4			150
Machine units, N_m^*	3	1	2	1	1	2	1	1	2	2	2	2	1	1	1	1	2	1	2	2	

m →	1	2	3	4	5	6	7	8	9	10
$b_m^{\dagger}$ →	1500	2000	1200	1000	2500	1000	3000	2000	2500	1000
$L_m^{\ddagger}$ →	2.83	0.80	1.79	0.90	0.92	1.50	0.83	0.93	1.82	1.95
m →	11	12	13	14	15	16	17	18	19	20
b_m →	1800	2000	1500	2000	2000	2000	1500	3000	1000	1500
L_m →	1.72	1.90	0.73	0.72	0.95	0.83	1.50	0.90	1.90	1.60

* N_m is obtained by rounding L_m to the next integer

† b_m is capacity of each machine of type m

‡ $L_m = \frac{\sum_p Q_p t_{pm}}{b_m}$, units of machine type m required

Table 6.15: Part machine grouping solution for stage 1.

Machines

		1	0	0	1	0	1	0	1	1	1	0	2	1	1	0	1	0	1	0	0
		5	7	6	3	2	1	3	4	6	7	8	0	9	0	1	2	9	8	5	4
	5	2^{+}	3	4																	$\boxed{1}$
	8	1	3		2													$\boxed{4}$			$\boxed{5}$
	13	3	2	1							$\boxed{4}$										
	16	1	2	3										$\boxed{4}$							
P	2					3	1	2													
a	4					3	2	1							$\boxed{4}$						
r	6						1		2	3	4									$\boxed{5}$	
t	7									2	3									$\boxed{1}$	
s	11						1	3	2												
	15				$\boxed{1}$				2	3	4										
	19					2	3	1												$\boxed{4}$	
	3											1	2	3							
	10											3	2	1							
	14					$\boxed{4}$						1			2	$\boxed{3}$					
	18											1	3	2	4						
	1												$\boxed{5}$			2	1	3	4		
	9						$\boxed{3}$									4	5	2	1		
	12															5	4	1	2	3	
	17															2	3	1			
	20														$\boxed{2}$	3	4		1		

+ sequence of operation of part

$\boxed{1}$ operation 1 results in an exceptional element

Table 6.16: Load on machines for stage 1 solution.

Cell c	Machine type, m assigned	Fraction of machines used by the allocated part(s), p							Number of machines: required -ed ν_r	assign -ed ν_a^*	avail -able ν_t $=N_m$	extra$^+$ $\nu_e=$ $(\nu_t-\nu_a)$	Unused time†, b^0_{mc}
1	$p \rightarrow$ $m \downarrow$	5	8	13	16								
	15	0.15	0.20	0.30	0.30				0.95	1	1	0	100
	7	0.17	0.13	0.33	0.20				0.83	1	1	0	510
	6	0.30	-	0.40	0.80				1.50	2	2	0	500
	13	-	0.53	-	-				0.53	1	1	0	705
2	$p \rightarrow$ $m \downarrow$	2	4	6	7	11	15	19					
	2	0.15	0.30	-	-	-	-	0.20	0.65	1	1	0	700
	11	0.28	0.33	0.17	-	0.42	-	0.11	1.31	2	2	0	1242
	3	0.33	0.83	-	-	0.38	-	0.25	1.79	2	2	0	252
	14	-	-	0.25	-	0.22	0.30	-	0.77	1	1	0	460
	16	-	-	0.15	0.30	-	0.38	-	0.83	1	1	0	340
	17	-	-	0.27	0.50	-	0.20	-	0.97	1	2	1$^+$	45
3	$p \rightarrow$ $m \downarrow$	3	10	14	18								
	8	0.30	0.15	0.10	0.38				0.93	1	1	0	140
	20	0.40	0.20	-	0.33				0.93	1	2	1$^+$	105
	19	0.80	0.20	-	0.50				1.50	2	2	0	500
	10	-	-	0.40	0.50				0.90	1	2	1$^+$	100
4	$p \rightarrow$ $m \downarrow$	1	9	12	17	20							
	1	0.53	0.83	0.50	0.40	0.30			2.56	3	3	0	660
	12	0.30	0.62	0.25	0.50	0.22			1.89	2	2	0	220
	9	0.32	0.20	0.50	0.40	-			1.42	2	2	0	1450
	18	0.20	0.17	0.33	-	0.20			0.90	1	1	0	300
	5	-	-	0.40	-	-			0.40	1	1	0	1500
	4	-	-	-	-	-			0.00	0	1	1$^+$	0

* ν_r is rounded to next integer to get ν_a
+these machines will be reassigned in the stage-2
$\dagger b^0_{mc} = b_m(\nu_a - \nu_r)$

Table 6.17: Assigned operations of parts.

cell c	number of machines assigned, N_c^0	part p allocated	demand Q_p	intra cell move cost $HCW_p^c(N_c^0)$	number of intra moves, μ_p	total intra cell move cost	$\Delta HCW_p^c = \lambda_1^p.\mu_p.Q_p$
1	5			1.3			
		5	100		2	260	20
		8	200		2	520	40
		13	200		2	520	40
		16	200		2	520	40
total						1820	140
2	8			1.6			
		2	100		2	320	20
		4	200		2	640	40
		6	100		3	480	30
		7	150		1	240	15
		11	150		2	480	30
		15	150		2	480	30
		19	100		2	320	20
total						2960	185
3	5			1.3			
		3	200		2	520	40
		10	100		2	260	20
		14	100		1	130	10
		18	250		3	975	75
total						1885	145
4	9			1.7			
		1	200		3	1020	60
		9	250		2	850	50
		12	250		4	1700	100
		17	200		2	680	40
		20	150		1	255	15
total						4505	265
5	0			0.0		0.0	0.0
total						0.0	0.0

Table 6.18: Alternate plans for processing exceptional elements.

c°	$p^{\triangleright}$	$e^{\diamond}$	$o^{\triangleleft}$	Current assignment		Options available for processing						α_p^e
						$1^{\diamondsuit}$		2		3		
				$m^{\dagger}$	$c^{\ddagger}$	$m^{\dagger}$	$c^{\ddagger}$	$m^{\dagger}$	$c^{\ddagger}$	$m^{\dagger}$	$c^{\ddagger}$	
1	5	1	1	4	4	$4(3,0)^{+}$	$1,2,3,4,5^{\times}$	6(4,4)	1	13(6,5)	1	1
	8	1	4	9	4	$9^{*}(5,0)$	4	6(8,4)	1	7(9,3)	1	2
		2	5	4	4	$4^{\star}(3,0)$	1,2,3,4,5	15(5,3)	1	13(6,4)	1	1
	13	1	4	17	2	17(4,0)	1,2,3,4,5	6(7,4)	1		1	1
	16	1	4	19	3	19(2,0)	3	6(4,5)	1	13(5,4)	1	1
2	4	1	4	10	3	10(3,0)	1,2,3,4,5	14(7,3)	2			1
	6	1	5	5	4	5(2,0)	4	2(4,4)	2	3(5,3)	2	1
	7	1	1	5	4	5(4,0)	4	14(7,4)	2			1
	15	1	1	13	1	13(2,0)	1	3(5,4)	2	2(5,3)	2	1
	19	1	4	5	4	5(5,0)	4	14(7,3)	2			1
3	14	1	4	2	2	2(3,0)	2	19(5,3)	3			1
		2	3	1	4	1(4,0)	4	19(6,4)	3			2
4	1	1	5	20	3	20(5,0)	1,2,3,4,5	5(8,4)	4			1
	9	1	3	11	2	11(3,0)	2	5(5,3)	4			2
	20	1	2	10	3	10(3,0)	1,2,3,4,5	9(3,4)	4	5(5,3)	4	2

$\circ$ assigned cell of the part
$\triangleright$ part
$\diamond$ exceptional element
$\triangleleft$ operation
$\dagger$ machine
$\ddagger$ cell where the machine is available/ or may be reassigned
$\diamondsuit$ includes the current assignment which can also be selected
$+$ within bracket are shown processing time(t_{pm}) and incremental processing cost C_{pm}^{e}
$\times$ cell 5 is an option given to allow to form a remainder cell
$*$ machine 9 does not have an extra unit so it is available only in cell 4
$\star$ machine 4 has an extra unit so it can be assigned to any of the cells, 1, 2, 3, 4 or 5

Table 6.19: Part machine grouping solution for stage 2: job splitting not allowed.

Machines

Parts	15	07	06	13	04	17	02	11	03	14	16	17	10	08	20	19	10	01	12	09	18	05	20
Units →	1	1	2	1	1	1	1	2	2	1	1	1	1	1	1	2	1	3	2	2	1	1	1
5	2^+	3	4		$\underline{1}$																		
8	1	3		2	$\underline{5}$															$\boxed{4}$			
13	3	2	1			$\underline{4}$																	
16	1	2	3													$\boxed{4}$							
2							3	1	2														
4							3	2	1				$\underline{4}$										
6								1		2	3	4										$\boxed{5}$	
7											2	3										$\boxed{1}$	
11								1	3	2													
15				$\boxed{1}$						2	3	4											
19							2	3	1													$\boxed{4}$	
3														1	2	3							
10														3	2	1							
14							$\boxed{4}$							1			2	$\boxed{3}$					
18														1	3	2	4						
1																		2	1	3	4		$\underline{5}$
9								$\boxed{3}$										4	5	2	1		
12																		5	4	1	2	3	
17																		2	3	1			
20																		3	4	$\overline{\mathbf{2}}$	1		

+ sequence of operation of part

$\boxed{1}$ operation 1 results in an exceptional element

$\underline{\mathbf{2}}$ operation 2 is assigned on the same machine type

$\overline{\mathbf{3}}$ operation 3 is assigned on a different machine type

Table 6.20: Part machine grouping solution for stage 2: job splitting allowed.

	Machines																						
	15	07	06	13	04	17	02	11	03	14	16	17	10	08	20	19	10	01	12	09	18	05	20
Units→	1	1	2	1	1	1	1	2	2	1	1	1	1	1	1	2	1	3	2	2	1	1	1
Parts 5	2^{+}	3	4		$\underline{\mathbf{1}}$																		
8	1	3,$\overline{\mathbf{4}}$	$\overline{\mathbf{4}}$	2	$\underline{\mathbf{5}}$															$\boxed{4}$			
13	3	2	1			$\underline{\mathbf{4}}$																	
16	1	2	3	$\overline{\mathbf{4}}$												$\boxed{4}$							
2							3	1	2														
4							3	2	1				$\underline{\mathbf{4}}$										
6							$\overline{\mathbf{5}}$	1	$\overline{\mathbf{5}}$	2	3	4											
7											2	3										$\boxed{1}$	
11								1	3	2													
15				$\boxed{1}$			$\overline{\mathbf{1}}$			2	3	4											
19							2	3	1	$\overline{\mathbf{4}}$												$\boxed{4}$	
3														1	2	3							
10														3	2	1							
14							$\boxed{4}$							1		$\overline{\mathbf{3}}$	2	$\boxed{3}$					
18														1	3	2	4						
1																		2	1	3	4		$\underline{\mathbf{5}}$
9								$\boxed{3}$										4	5	2	1	$\overline{\mathbf{3}}$	
12																		5	4	1	2	3	
17																		2	3	1			
20																		3	4	$\overline{\mathbf{2}}$	1		

+ sequence of operation of part; $\boxed{1}$ operation 1 results in an exceptional element; $\underline{\mathbf{2}}$ operation 2 is assigned on the same machine type; $\overline{\mathbf{3}}$ operation 3 is assigned on a different machine type.

Fraction of demand (x^{e}_{pmc}) assigned to different machine(m) and cell(c) combinations

Part	EE	Operation	x^{e}_{pmc} (m,c)		
p	e	o	1	2	3
8	1	4	0.28(7,1)	0.31(6,1)	0.41(9,4)
16	1	4	0.49(13,1)	0.51(19,3)	-
6	1	5	0.50(2,2)	0.50(3,2)	-
15	1	1	0.73(13,1)	0.27(2,2)	-
19	1	4	0.66(14,2)	0.34(5,4)	-
14	2	3	0.51(1,4)	0.49(19,3)	-
9	1	3	0.42(11,2)	0.58(5,4)	-

Chapter 7

Cell Formation Considering Investment and Operational Costs

The majority of the cell formation models consider grouping of parts and machines, based on clustering techniques. The performance of cells thus formed indicates that the cellular systems perform more poorly in terms of work-in-process inventory, average job waiting time and job flow time than the improved job shops. However, they have superior performance in terms of average move times and setup times. The main reason for such a poor performance is that the current cell design procedures do not consider the operational aspects during the cell formation. Therefore, the objective of this chapter is to consider the investment and operational costs simultaneously during the design of a cellular manufacturing system. For this purpose we develop a mixed integer programming model and illustrate the tradeoff relationships between the investment and operational variables, such as sequence dependence setup, machine idle time, part inventory, part early and late finish (compared with due date), by considering examples. Computational experience is provided for randomly generated test problems.

7.1 Problem Background

In this chapter we consider cell formation in flow line manufacturing situations similar to those in repetitive manufacturing. The parts produced usually require the same set of machines in the same order, i.e., they go through the same processing stages. This situation arises in many chemical and processing industries. Typical examples include manufacture of detergents, paints, etc. The setups incurred during changeovers are usually sequence dependent. For example, in the manufacture of paints the equipment must be cleaned when there is a change from one color to another. The thoroughness of the cleaning is heavily dependent on the color being removed and the color for which the machine is being prepared.

In a sequence dependent manufacturing environment, where the demand for parts is repetitive in nature and the production requirements are similar, one can select the sequence in which to produce the parts, such that the total cost and time spent on setup is minimized. The sequence thus determined may give a schedule in which parts finish early or late as compared to their due dates. In addition one may have part waiting between machines, or machine idle time. Alternately, one could have a separate line for producing each part and avoid cost and time lost due to sequence dependence. Also the inventory can be reduced by synchronizing the production rate of cells with the demand rates. However, in this case the investment cost is high. Clearly, investment options between these two extremes are also available. For example, the late finishing of parts can be avoided by increasing capacity of bottleneck stages or by re-sequencing them after adding a new cell. The sequence of parts also affects the WIP and utilization of machines. Achieving minimum inventory and minimum machine idle time are two conflicting objectives as reduction in one often leads to an increase in other. Depending upon the scenario, the appropriate parameters should be considered and weighted accordingly.

Analyzing a few of the interactions stated above may be straightforward. However, when considered simultaneously, these interactions can be quite complex. As a result machine investment and operating (scheduling) strategies are rarely obvious and be counterintuitive. Cell formation under repetitive manufacturing as described in this chapter, thus, considers the following:

1. Cell design aspect- how many cells should be formed and what should be the capacity of each processing stage in a cell?

2. Allocation and sequencing aspect- how the parts are to be allocated, sequenced and scheduled in these cells?

The main contribution of this work is in providing a mathematical framework which simultaneously considers the tradeoffs between investment and operational costs (scheduling and sequencing) to address cell design in manufacturing environments with sequence dependence. The tradeoffs discussed in this chapter are illustrated by considering examples. Complexity of the model is studied by first considering a model which has the least number of variables, and then sequentially adding variables to it. Computational time for these models is summarized.

The remainder of this chapter is organized as follows. In section 7.2, a mathematical model considering tradeoffs between investment and operational variables is formulated. The tradeoffs are discussed by solving examples in section 7.3. Computational experience for the model is provided in section 7.4. Finally, summary is presented in section 7.5.

7.2 Mathematical Model

An mixed integer programming (MIP) model is formulated which forms cells by grouping the machines and parts simultaneously while considering the sequencing and scheduling of the parts. The following assumptions are made for formulating the model:

1. Each cell has the same processing stages, but the capacity of a stage may vary from cell to cell. We assume that the capacity can be increased continuously (rather than in steps), in the prescribed range (explained later).
2. Each part requires processing at all stages.
3. Part processing (throughput) time depends upon the cell capacity at each stage.
4. Setup times and costs are sequence dependent.
5. Triangular inequality of setup costs and times is assumed. This means that setup time (cost) of any part i, from part j, is always less than or equal to setup time (cost) required for part i, when first part j is produced and then part k is produced before part i.
6. Only one part type can be processed at each stage at any given time.
7. An operation once started on a machine cannot be interrupted before completion.
8. Only permutation scheduling is allowed.
9. No inter cell movement of parts is allowed.
10. Holding costs are directly proportional to the inventory levels.

It is assumed (assumption 1) that each cell has the same number of processing stages. The different stages are distinguished by their capacity. The number of copies of each machine type in different cells can vary. Thus, by increasing or decreasing the number of machines of any type the production capacity can be increased or decreased. Capacity can also be increased by increasing the processor speed by employing some auxiliary devices, modifying jigs and fixtures, etc. We assume that the capacity can be increased continuously with a linear cost increase. The increase in capacity results in a lower processing time for a part. We use the following type of relationship to model this:

$$p = u - vN \tag{7.1}$$

where p is processing time, u and v are constants, and N is an index showing capacity level of processor. N takes on the value 0 at minimum capacity and equals 1 at maximum capacity.

Model - M4

Objective function: Minimize the sum of the costs due to sequence dependence changeover, work in process inventory, early finish, late finish, machine idle time and investment on cells. It is assumed that if a cell is formed, there is a minimum fixed capacity in each stage; therefore, a fixed cost is associated with each cell formed. The cost of additional capacity, if needed, in different stages is captured separately by the last term.

$$\begin{aligned} Min \quad z = & \sum_{s=1}^{k}\sum_{j=1}^{n}\sum_{j'=1}^{n} C_{jj'}^{s} X_{jj'} + \sum_{j=1}^{n}\sum_{s=1}^{k-1} h_j^s W_j^s + \sum_{j=1}^{n} \alpha_j \Delta_j^+ + \sum_{j=1}^{n} \beta_j \Delta_j^- + \\ & \sum_{s=1}^{k}\sum_{j=1}^{n}\sum_{j'=1}^{n} \gamma^s I_{jj'}^s + \sum_{c=1}^{n} C^c X_{0c} + \sum_{s=1}^{k}\sum_{c=1}^{n} U^s L_c^s \end{aligned} \tag{7.2}$$

Constraints

1. **Assignment constraint:** Each part should be produced once, i.e., it will have

one follower and one predecessor:

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} X_{ij} = 1 \quad for \quad j \; = \; 1, 2, ..., n \qquad (7.3.a)$$

$$\sum_{\substack{l=1 \\ j \neq l}}^{(n+1)} X_{jl} = 1 \quad for \quad j \; = \; 1, 2, ..., n \qquad (7.3.b)$$

These two constraints are assignment constraints. Solution to these may result in many subtours. Part 0 and (n+1) are dummy parts. By providing a dummy at the beginning and end we identify each subtour (cell) by a unique number. This number depends on the first part which follows a dummy 0 in each cell. For example, if we have 5 part types (i.e., n = 5) and the solution to these constraints results in the following two subtours: 0-3-4-1-6 and 0-2-5-6, then we have two cells uniquely numbered as 3 and 2. The parts 0 and 6 at the beginning and end of each subtour are dummy parts. The unique numbering of cells as 3 and 2 allows us in the later stages to assign machines and parts to these cells.

2. **First stage completion time constraint:** Completion time of any part at the first stage should be at least equal to its arrival time plus processing time:

$$T_j^1 \geq a_j + p_j^1 \quad for \quad j \; = \; 1, 2, ..., n \qquad (7.4)$$

3. **Stage link constraint:** For any part, the completion time at stage $(s+1)$ is the sum of completion time at stage s, the processing time at stage $(s+1)$ and the waiting time after processing at stage s and before starting at stage $(s+1)$:

$$T_j^{s+1} = T_j^s + p_j^{s+1} + W_j^s \quad for \quad j \; = \; 1, 2, ..., n; s \; = \; 1, 2, ..., k-1 \qquad (7.5)$$

4. **Sequencing constraint:** At any stage s in a cell, the completion time of part j which succeeds part j' in the same cell is the sum of the completion time of part j', the setup time of the machine for part j from j', the idle time of the machine at stage s in that cell after processing part j' and before starting on part j, and the processing time of part j:

$$T_j^s \geq T_{j'}^s + S_{j'j}^s + I_{j'j}^s + p_j^s + B(X_{j'j} - 1) \qquad (7.6.a)$$

$$T_j^s \leq T_{j'}^s + S_{j'j}^s + I_{j'j}^s + p_j^s - B(X_{j'j} - 1) \qquad (7.6.b)$$

for $(j' \neq j)$; j, j' = 1,2,..., n; s= 1,2,..., k, and B is a large positive number.

These constraints work in pairs. If a sequence j'-j exists, then the two constraints will determine the sequencing of the job on all the stages. In this case,

$$T_j^s = T_{j'}^s + S_{j'j}^s + I_{j'j}^s + p_j^s$$

However, for any pair j-k for which a sequence does not exist, the two constraints will be relaxed, allowing $T_j^s \geq 0$.

5. **Capacity change constraint:** The processing time of a part at any stage in a cell is a function of the capacity level of the processor:

$$p_j^s = u_j^s - v_j^s N_j^s \quad for \quad j \;=\; 1,2,3,....,n; \quad s \;=\; 1,2,3,....,k \qquad (7.7)$$

Here, u_j^s and v_j^s are constants, which are determined from equation (7.1).

6. **Part machine link constraint:** If two parts are assigned to the same cell, then the capacity of the processor available to these parts at each stage is the same:

$$N_j^s \leq N_{j'}^s + B(1 - X_{jj'}) \qquad (7.8.a)$$

$$N_j^s \geq N_{j'}^s - B(1 - X_{jj'}) \qquad (7.8.b)$$

for $(j' \neq j)$; j, $j' = 1$, n; s= 1, k.

7. **Machine cell link constraint:** The level of capacity at stage s in a cell c (where cell is numbered as c if sequence 0-c exist) is given by

$$L_c^s \geq N_c^s - B(1 - X_{0c}) \quad for \quad c = 1,2,3...,n; \quad s = 1,2,3...,k \qquad (7.9)$$

The level of capacity at any stage is attached to the part type. If we use variable N_c^s in the objective function to capture the cost of additional capacity, then there will be a multiplicative effect, which depends on the number of parts in a cell. To avoid this overestimate in the objective we define a variable L_c^s which is attached to a cell. This in essence captures the actual additional capacity of each cell.

8. **Modeling constraint:** The maximum value of capacity level should be specified as 1:

$$N_j^s \leq 1 \quad for \quad s = 1,2,3,...,k; \quad j = 1,2,3,...,n. \qquad (7.10)$$

N_j^s has the value 0 at minimum capacity and the value 1 at maximum capacity.

9. **Due date constraint:** Early completion or late completion of a part as compared to its due date is given by

$$\Delta_j^+ - \Delta_j^- = d_j - T_j^k \quad for \quad j = 1,2,3,...,n \qquad (7.11)$$

In this model, there are $n^2 + n$ binary integer variables, $n^2k + 4nk + n$ continuous variables and $4n^2k + 3n$ constraints, where n is the number of part types and k is the number of processing stages. In the next section, we solve a few examples using the model developed and illustrate the importance of considering the investment and operational aspects simultaneously.

7.3 Examples

In this section, tradeoffs between various costs are illustrated by considering a few examples. Five part types which require processing in three stages are considered. Processing times required for these parts decrease with increase in cell capacity. Slope (u_j^s) and intercept (v_j^s) of lines representing these relationships ($p_j^s = u_j^s - v_j^s N_j^s$) are shown in Table 7.1. Processing times calculated from these expressions are in hours. Sequence dependence setup costs ($\sum_s C_{ij}^s$) and times (S_{ij}^s) are shown in Table 7.2. The setup time matrix is taken to be the same at each processing stage. Information regarding part inventory costs, early and late penalty, investment cost to acquire additional cell capacity, and cost of machine idle time are shown in Table 7.3. Machine idle time cost in reality should depend upon the number of idle machines. However, in the present analysis a constant value is taken. Part arrival time was taken as 0 for all part types. Data given in Tables 7.1-7.3 is the same in all the examples. The parameters which we vary to illustrate the tradeoffs are due dates, cost of cell and machine idle time. The values taken for different cases are shown in Table 7.4. The problems were formulated and solved on a personal computer 486 (33 MHz) using the Hyperlindo software package. The results are summarized in Tables 7.5-7.10. We will discuss the results next.

Case 1: Low cell investment cost- In this case the cell cost is considered to be lower than the sequence dependence cost. The cost of a cell was taken as \$2, which gives 5 as the optimum number of cells with an objective function value of \$10 (10+0+0+0+0+0+0).[1] Each part is assigned a separate cell (Table 7.5). This

[1] break up of costs into line (cell) + setup + WIP inventory + machine idle time + part early finish + part late finish + additional capacity acquisition costs. The costs obtained for each of the example problems are shown in Table 7.10.

result can be explained as follows. Minimum sequence dependence setup cost for any pair of parts is \$2 (Table 7.3). Also, the cost of new line (cell) is the same. If part 5 or 1 is produced in the same cell as with part 2, then a total of 4 cells instead of 5 will be required which will save on cell cost by \$2, but this will increase the cost due to sequence dependence by the same amount. From this it may appear that both solutions will give the same cost. This is in fact the solution the model developed by Rajamani et al. (1992) would give. However, producing two parts in one cell may result in either machine idle time between the parts or part wait time between the stages, hence will result in an increased overall cost relative to having a separate cell for each part.

Case 2: High cell investment cost- If the cost of the cell is very high as compared to the sum of the costs due to sequence dependence setup, machine idle time and part waiting time, early and late finish, then it is expected that the number of cells will be reduced. For instance take the cost of line as \$20 keeping all other data same as in case 1. This results in forming one cell (Table 7.6). The objective function value is \$59.8 (20+31+2.8+5.6+0.4+0+0) and part sequence is 1-2-5-4-3. Obviously, if we have more than one part family or smaller links of part, the scheduling cost (defined as sum of sequence dependence cost, machine idle time cost or part wait time cost, part early and late finish costs) may be reduced. But the cost of forming an additional cell adds \$20 to the total cost, which in this case is more than the saving in scheduling costs that would have resulted in making two or more part families.

Case 3: Intermediate cell investment cost- Now we will explore a few scenarios which exist for manufacturing companies in which the investment options lie between the two extremes. For example, the cost of the cell is taken as \$5 keeping all other data the same as in case 1. This results in 4 cells, by producing part 2 and part 5 in one cell and the remaining parts in separate cells (Table 7.7). The objective function value in this case is \$23.11 (20+2+0.1+0.7+0+0+0.31). The above three examples show tradeoffs mainly between cost of line and sequence dependence setup costs. Interactions of other costs in these examples could not be avoided completely but they had minor influences.

The following examples will show the interactions of other variables.

Case 4: Due date interaction- To illustrate the tradeoff between due date and other variables, we took the basic data from Case 2 and changed the due dates (Table 7.4). The solution obtained is shown in Table 7.8. The total cost in this case is \$52.61 (20+24+0.87+7.71+0+0+0.03) and part sequence is 2-3-4-5-1. If we compare this with the result of case 2, we observed that a new sequence is obtained to reduce the high cost of late finishing. Also, in this case additional cell capacity is required.

Case 5: WIP and machine idle time interactions- Case 5 shows the tradeoff between WIP inventory and machine idle time. All the data from Case 3 were taken except machine idle time cost. Machine idle time cost was increased from \$ 0.02 to \$ 0.1 per hour. The results are shown in Table 7.9. The total cost in this case is \$ 24.16 (20+2+1.85+0+0+0+0.31). This gives no machine idle time, but 40 hours of waiting time for part 2 and 35 hours for part 5. In Case 3 there was no part waiting time, instead there was a machine idle time of 40 hours. Thus, the increase in weight for machine idle has resulted in an increase in WIP and a reduction in machine idle time.

Thus, we see that different manufacturing situations depending on the costs and interaction of costs considered by investment and operational (scheduling) variables at design stage can lead to grouping of parts and machines, which are quite different. This suggests that grouping of parts and machines should simultaneously consider the interactions explored in this chapter for a good cell design.

7.4 Computational Experience

The computational difficulty in solving the generalized model developed in section 7.2, depends on the number of interactions considered in the objective function, the data set and the size of the problem. To study the impact of interactions we develop three sub models, which are obtained by deleting the variables and constraints not pertinent in the generalized model. Submodel 1 considers only the investment and sequence dependence costs. Submodel 2 is obtained by including due date and related variables and constraints to sub model 1. Similarly, submodel 3 is obtained by adding WIP, machine idle time and corresponding variables and constraints to submodel 1. The three submodels are briefly stated next.

Submodel 1: This considers the investment cost on cell and sequence dependence cost.

Objective function:

$$Min \;\; z_1 = \sum_{s=1}^{k}\sum_{j=1}^{n}\sum_{j'=1}^{n} C^s_{jj'}X_{jj'} + \sum_{c=1}^{n} C^c X_{0c} + \sum_{s=1}^{k}\sum_{c=1}^{n} U^s L^s_c \qquad (7.12)$$

Constraints: Constraint sets, (7.3), (7.4), (7.7) - (7.10) (refer to section 7.2), and the following:

$$T^{s+1}_j \geq T^s_j + p^{s+1}_j \quad for \;\; j \; = \; 1, \; n \;\; and \;\; s \; = \; 1, \; k-1 \qquad (7.13)$$

$$T^s_j \geq T^s_{j'} + S^s_{j'j} + p^s_j + B(X_{j'j} - 1) \qquad (7.14)$$

for $(j' \neq j)$, j, j' = 1, n, s= 1, k and B is a large positive number.

Submodel 2: This considers investment cost of the cell, sequence dependence cost and early and late finish of the parts as compared to their due dates.

Objective function:

$$Min \;\; z_2 = z_1 + \sum_{j=1}^{n} \alpha_j \Delta_j^+ + \sum_{j=1}^{n} \beta_j \Delta_j^- \tag{7.15}$$

Constraints: Constraint sets, (7.3), (7.4), (7.7) - (7.11) (refer to section 7.2), and the following:

$$T_j^{s+1} \geq T_j^s + p_j^{s+1} \quad for \;\; j \; = \; 1, \; n \;\; and \;\; s \; = \; 1, \; k-1 \tag{7.16}$$

$$T_j^s \geq T_{j'}^s + S_{j'j}^s + p_j^s + B(X_{j'j} - 1) \tag{7.17}$$

for $(j' \neq j)$, j, j' = 1, n, s= 1, k and B is a large positive number.

Submodel 3: This model considers investment cost on cell, sequence dependence cost, WIP and machine idle time.

Objective function:

$$Min \;\; z_3 = z_1 + \sum_{j=1}^{n} \sum_{s=1}^{k-1} h_j^s W_j^s + \sum_{s=1}^{k} \sum_{j=1}^{n} \sum_{j'=1}^{n} \gamma^s I_{jj'}^s \tag{7.18}$$

Constraints: Constraint sets, (7.3) - (7.10) (refer to section 7.2).

The number of variables and constraints for these submodels and the generalized model are given in Table 7.11.

In order to study the impact of interactions of the variables considered in the objective function and the effect of the data set we solved the three submodels and the generalized model for the 5 examples discussed in section 7.3. The computational times for these models are summarized in Table 7.11. The results indicate that the models can be ranked in terms of computational complexity as submodel 1, submodel 2, submodel 3 and the generalized model. Considering WIP and machine idle time explicitly in the model increases the computational burden to a great extent. Also, we see that the computational time is dependent on the data set. All the problems are five-part and three-stage problems, but there is a large variation in the computation time.

In addition, we also solve larger problems of up to 10 parts. For this purpose we randomly generated sequence dependence times and costs between 3 to 33, and 2 to 20, respectively. Due dates were randomly selected between 600 to 1000. Processing time data u_j^s were generated randomly from 30 to 180. v_j^s were kept as 50% of the u_j^s. The following information was assumed the same for all problems: k = 3, h_j^s = \$0.01, γ^s = \$0.01, U^s = \$2, α_j = \$0.01, β_j = \$0.05, C^c = \$4. In randomly generating the sequence dependence times and costs we have ignored the triangular inequality. This exists usually in a practical problem and hence an assumption in the problem statement. It does not however, limit the applicability of the model developed. Table 7.12 contains a summary of the results. The table shows the number of variables and constraints, number of nodes solved in branch and bound, the node at which optimum solution was obtained and the computation time, for each problem.

It is worth mentioning that in most cases the number of parts produced in repetitive manufacturing is not large. However, the problem size becomes large with an increase in part types. In such cases the proposed model can be effectively used by aggregating the part types having similar setups into fewer families. Also, in real life a number of sequences are not feasible due to process and product limitations. This further decreases the computational burden (Rajamani et al., 1992).

7.5 Summary

The majority of the cell formation models consider grouping of parts and machines based on diagonalization of part machine matrix. This approach is not applicable in flow shops, where all the parts require the same set of machines. Also, cellular systems designed without considering the operational variables can lead to poor performance. A mixed integer programming model was developed which considers investment cost (cost of cell and machines) and operational cost (sequence dependence setup, machine idle time, part WIP inventory, part early and late finish) simultaneously to form cells. The model determines the economic number of cells, capacities of processing stages in each cell formed, part allocation, sequencing and scheduling in these cells. Examples were considered which illustrate the model and cost tradeoffs considered in the model. Computational experience on up to 10 parts was reported. It was observed that the consideration of WIP and machine idle time considerably increased the computation times.

Table 7.1: Constants of processing time relationships (equation 7.7).

Part, j ↓	Processing stage, s →					
	1		2		3	
	u_j^1	v_j^1	u_j^2	v_j^2	u_j^3	v_j^3
1	100	20	80	16	70	14
2	40	8	60	12	80	16
3	70	14	90	18	60	12
4	210	42	90	18	30	6
5	100	20	40	8	160	32

Table 7.2: Sequence dependence setup costs, $\sum_s C^s_{ij}$ ($) and times, S^s_{ij} (hours) for parts.

From \ To	1	2	3	4	5
1	-	18* (16+)	10 (17)	10 (4)	10 (24)
2	2 (20)	-	4 (22)	3 (3)	2 (24)
3	5 (17)	18 (20)	-	8 (3)	10 (22)
4	6 (34)	17 (26)	7 (30)	-	8 (32)
5	4 (21)	12 (17)	5 (20)	4 (3)	-

$\star$ unbracketed numbers are setup costs
$+$ bracketed numbers are setup times (same for each stage, s)

Table 7.3: Information on costs related to inventory, early and late finish, investment cost on additional processor capacity.

Name of variable		Costs
Part Inventory*	(same for all part (j))	
	After first stage (h_j^1)	\$0.02 /hour
	After second stage (h_j^2)	\$0.03/hour
Part early finish	α_j(same for all part(j))	\$0.04/hour
Part late finish	β_j (same for all part (j))	\$ 0.8 /hour
Cost to increase capacity to maximum, U^s	(same for all stage, s)	\$1

⋆ As part goes through stages its value becomes more and the inventory cost increases. Hence, increased value of inventory holding cost is taken for successive stage.

Table 7.4: Data for example problems.

Part, j → / Case ↓	Due date for parts, d_j: 1	2	3	4	5	Cost of cell, C^c (\$)	Cost of machine idle time, γ^s (\$/hour)
1	250	366	763	613	540	2	0.02
2	250	366	763	613	540	20	0.02
3	250	366	763	613	540	5	0.02
4	766	180	282	475	675	20	0.02
5	250	366	763	613	540	5	0.10

Table 7.5: Result of Case 1.

Processing stages	Variable	Value				
	Cell, c	1	2	3	4	5
	+Part, j-j′	1	2	3	4	5
Stage 1	p_j^1	100	40	70	210	100
	T_j^1	100	226	613	493	340
	W_j^1	0	0	0	0	0
	$I_{jj'}^1$	0	0	0	0	0
	L_c^1	0				
Stage 2	p_j^2	80	60	90	90	40
	T_j^2	180	286	703	583	380
	W_j^2	0	0	0	0	0
	$I_{jj'}^2$	0	0	0	0	0
	L_c^2	0				
Stage 3	p_j^3	70	80	60	30	160
	T_j^3	250	366	763	613	540
	W_j^3	0	0	0	0	0
	$I_{jj'}^3$	0	0	0	0	0
	L_c^3	0				
Completion	Δ_j^+	0	0	0	0	0
	Δ_j^-	0	0	0	0	0

Note: $L_c^1 = 0$, indicates that no increase (from minimum) in capacity is required.
+j-j′ represents the sequencing of parts within the allocated cell. Here one part is allocated to each cell.

Table 7.6: Result of Case 2.

Processing stages	Variable	Value				
	Cell, c	1				
	+Part, j-j′	1	2	5	4	3
Stage 1	p_j^1	100	40	100	210	70
	T_j^1	100	156	280	493	593
	W_j^1	0	60	60	0	20
	$I_{jj'}^1$	0	0	0	0	0
	L_c^1	0				
Stage 2	p_j^2	80	60	40	90	90
	T_j^2	180	276	380	583	703
	W_j^2	0	0	0	0	0
	$I_{jj'}^2$	20	40*	110	0	0
	L_c^2	0				
Stage 3	p_j^3	70	80	160	30	60
	T_j^3	250	356	540	613	763
	W_j^3	0	0	0	0	0
	$I_{jj'}^3$	10	0	40	60	0
	L_c^3	0				
Completion	Δ_j^+	0	10	0	0	0
	Δ_j^-	0	0	0	0	0

* $I_{25}^2 = 40$, indicates that machine idle time between parts 2 and 5 is 40 hours

+j-j′ represents the sequencing of parts within the allocated cell. Here it is 1-2-5-4-3 in cell 1.

Table 7.7: Result of Case 3.

Processing stages	Variable	Value				
	Cell, c	1	2		3	4
	+Part, j-j′	1	2	5	3	4
Stage 1	p_j^1	100	40	100	70	210
	T_j^1	100	226	350	613	493
	W_j^1	0	5	0	0	0
	$I_{jj'}^1$	0	0	0	0	0
	L_c^1	0	0		0	0
Stage 2	p_j^2	80	60	40	90	90
	T_j^2	180	291	390	703	583
	W_j^2	0	0	0	0	0
	$I_{jj'}^2$	0	35	0	0	0
	L_c^2	0	0		0	0
Stage 3	p_j^3	70	75	150	60	30
	T_j^3	250	366	540	763	613
	W_j^3	0	0	0	0	0
	$I_{jj'}^3$	0	0	0	0	0
	L_c^3	0	0.312		0	0
Completion	Δ_j^+	0	0	0	0	0
	Δ_j^-	0	0	0	0	0

+j-j′ represents the sequencing of parts within the allocated cell. For example, sequence 2-5 exists in cell 2.

Table 7.8: Result of Case 4.

Processing stages	Variable	Value				
	Cell, c	2				
	+Part, j-j′	2	3	4	5	1
Stage 1	p_j^1	39.8	69.6	208.9	99.5	99.5
	T_j^1	40	131.6	343.5	475	595.5
	W_j^1	0	0.4	9.5	0	0
	$I_{jj'}^1$	0	0	0	0	0
	L_c^1	0.025				
Stage 2	p_j^2	60	90	90	40	80
	T_j^2	100	222	443	515	675.5
	W_j^2	0	0	2	0	20.5
	$I_{jj'}^2$	10	128	0	59.5	0
	L_c^2	0				
Stage 3	p_j^3	80	60	30	160	70
	T_j^3	180	282	475	675	776
	W_j^2	0	0	0	0	0
	$I_{jj'}^3$	20	160	8	0	0
	L_c^3	0				
Completion	Δ_j^+	0	0	0	0	0
	Δ_j^-	0	0	0	0	0

+j-j′ represents the sequencing of parts within the allocated cell. Here it is 2-3-4-5-1 in cell 2.

Table 7.9: Result of Case 5.

Processing stages	Variable	Value				
	Cell, c	1	2		3	4
	+Part, j-j′	1	2	5	3	4
Stage 1	p_j^1	100	40	100	70	210
	T_j^1	100	191	315	613	493
	W_j^1	0	40	0	0	0
	$I_{jj'}^1$	0	0	0	0	0
	L_c^1	0	0		0	0
Stage 2	p_j^2	80	60	40	90	90
	T_j^2	180	291	355	703	583
	W_j^2	0	0	35	0	0
	$I_{jj'}^2$	0	0	0	0	0
	L_c^2	0	0		0	0
Stage 3	p_j^3	70	75	150	60	30
	T_j^3	250	366	540	763	613
	W_j^3	0	0	0	0	0
	$I_{jj'}^3$	0	0	0	0	0
	L_c^3	0	0.312		0	0
Completion	Δ_j^+	0	0	0	0	0
	Δ_j^-	0	0	0	0	0

+j-j′ represents the sequencing of parts within the allocated cell. For example, sequence 2-5 exists in cell 2.

Table 7.10: Comparison of results.

Case	Different costs ($)							Total cost ($)
	Cell	Setup	WIP	M/c idle	Early finish	Late finish	Add. machine	
1	10	0	0	0	0	0	0	10
2	20	31	2.8	5.6	0.4	0	0	59.8
3	20	2	0.1	0.7	0	0	0.31	23.11
4	20	24	0.87	7.71	0	0	0.03	52.61
5	20	2	1.85	0	0	0	0.31	24.16

Table 7.11: Computational complexity of models.

Model →	Submodel 1	Submodel 2	Submodel 3	Generalized Model
Model Size[1]				
Number of variables				
(i) continuous	$4nk$	$4nk + 2n$	$n^2k + 4nk - n$	$n^2k + 4nk + n$
(ii) binary	$n^2 + n$	$n^2 + n$	$n^2 + n$	$n^2 + n$
Number of constraints	$3n^2k+nk+2n$	$3n^2k+nk+3n$	$4n^2k+2n$	$4n^2k+3n$
Computational time[2]				
Case 1	00:03	00:04	00:06	00:06
Case 2	01:13	04:55	06:59	08:51
Case 3	00:07	00:08	00:32	00:32
Case 4	01:13	03:13	07:00	07:45
Case 5	00:07	00:07	00:50	00:56

[1] n-number of parts and k is number of processing stages

[2] computational time (in minute:second) for problem size $n = 5$, $k = 3$

Table 7.12: Computational results for 6, 7, 8, 9 and 10 part problems.

Number of parts	Number of variables[1]	Number of constraints	Total nodes (pivot) solved	Optimum found at node (pivot)	Time (min:sec)
6	186(42)	450	8(431)	6(230)	00:32
7	238(56)	609	20(1237)	17(1064)	02:02
8	296(72)	792	59(6341)	34(3508)	03:00
9	360(90)	999	54(9204)	31(2780)	25:16
10	430(110)	1230	90(13713)	39(3687)	40:00

[1]unbracketed number is for continuous variables and bracketed is for binary variables.

Chapter 8

Conclusions

In this chapter, contributions of the research to cell formation are presented in section 8.1. Also, directions for future research are briefly discussed in section 8.2.

8.1 Contributions of the Research

The objective of the cell system of production or cellular manufacturing (CM) is to achieve efficiencies in production by suitably decomposing a larger system into smaller subsystems. An important issue in designing CM system is cell formation (CF). CF consists of identifying part families (PFs) and machine groups (MGs) such that the part families are processed within a machine group with minimum interaction with other cells. The objectives of this partition are two folds (Miltenburg and Zhang, 1991):

1. Within a part machine group or cell, each machine is visited by many parts, that is, there is a high usage of the machines by parts; and

2. Few parts require processing on machines in other cells.

The above objectives can be interpreted as minimization of voids and exceptional elements in a part machine matrix which represents the processing requirements of parts. The CF problem has attracted a great deal of research effort and numerous approaches have been proposed for partitioning the matrix. The common characteristics of most of the algorithms are the following.

- They do not provide a simultaneous approach to CF.
- Rearrangement of matrix is often sought by maximizing indirect measures such as similarity index or heuristic procedures.
- They do not have the flexibility to create large loose cells or small tight cells to provide the designer with alternate solutions in a structured manner.
- They require manual/subjective intervention to identify part families and machine groups. For large matrices this becomes very difficult.
- Performance of these algorithms are often reported for only small structured problems. Computation on large matrices is usually not reported.
- Consideration of the presence of alternate process plans and additional units of same machines are not adequately addressed.
- They do not consider the effect of cell size on material handling during cell formation, (except Sankaran and Kasilingam, 1993).
- No procedure is available to consider operational variables such as WIP inventories, machine utilization, part early and late finish, and sequence dependence; and considering investment in solving CF problem in a repetitive manufacturing environment.

In this research we provide cell formation (CF) procedures which overcome the above drawbacks. In Chapter 3, a nonlinear mathematical programming model is developed for CF which identifies part families (PFs) and machine groups (MGs) simultaneously. The model considers minimization of a weighted sum of voids and exceptional elements as the objective. Then, an iterative procedure called assignment allocation algorithm (AAA) is proposed to solve the model. In AAA changing weights for voids and exceptional elements gives the designer alternatives to form large loose cells or small tight cells. AAA does not require any manual intervention for identifying PFs and MGs. Performance of AAA has been compared with many well known procedures for the problems selected from literature. The problems tested consist of well structured as well as ill structured matrices. AAA has been found to be as good or better on this comparison. AAA is very simple and less computer intensive. Large problems with 400 parts and 240 machines were solved using this algorithm in less than a minute on Sun Sparc 2 station. In Chapter 4, a more robust Simulated Annealing Algorithm (SAA) is developed to solve the nonlinear model developed in Chapter 3. This procedure gives more consistent results than the iterative procedure, AAA. However, SAA takes significantly higher computation time. For larger problems it has been found preferable to use AAA a few times with different initial solution rather than using SAA.

In Chapter 5, the nonlinear model developed is further extended to consider alternate process plans for parts and additional units of same machines. The solution obtained using AAA on the extended model indicates that in the presence of alternate process plans the algorithm has a greater tendency to converge to a local minimum. To overcome this, simulated annealing (SAA) is again used to solve the model. SAA solutions were found to be optimal for all the problems tested up to a size of 300 parts with 720 process plans, and 50 machines.

A good decomposition of the part machine matrix could be arrived at by considering the tradeoff between intra cell material handling inter cell material handling. In Chapter 6, we developed a two stage procedure to handle this situation. In stage 1, a nonlinear model and Simulated Annealing Algorithm are developed to minimize the total intra cell and inter cell material handling costs. In the calculation of the material handling costs, production quantity, effect of cell size on unit intra cell handling costs, effect of sequence of operations and multiple non-consecutive visits to the same machine are considered. In stage 2, an integer programming model is developed to further improve the solution obtained in stage 1. The model considers the options to re-assign the operations which resulted in exceptional elements in stage 1, and extra units of machines as available.

In Chapter 7, we have addressed cell formation in a repetitive manufacturing system where the parts produced usually require the same set of machines and in the same order. A mixed integer programming model was developed with the objective of minimizing the investment and operational costs. The operational variables considered are sequence dependence setup, machine idle time, part inventory, part early and late finish compared to due date. Computational experience of the model for problems that consider up to 10 parts, which is usually the case in repetitive manufacturing environment, is provided.

8.2 Directions for Future Research

The following are proposed as directions for further research:

- Develop a 'hybrid' procedure which uses SAA at the beginning and switching over to AAA after a few iteration to provide AAA with a good initial solution. This should provide a good solution with less computational time.
- A more detailed analysis is needed on developing simulated annealing schedules for different problem sizes and ranges of values of input data. This will provide

a guideline for setting values of parameters for a particular problem. Also, for SAA different strategies for generating a neighborhood solution may be experimented with, instead of the use of random switching of machine strategy as adopted in this thesis.

- The presence of some parts (rows) and machines (columns) in the part machine matrix, adversely affects the the block diagonalization of the matrix. It would be useful to develop procedures/guidelines for eliminating parts and machines which affect the groupability of the matrix.

- At present, the objectives considered for obtaining a block diagonal matrix is the weighted sum of the voids and exceptional elements. A procedure which can directly consider the 'grouping measure' as the objective should be developed.

- All the models developed in this thesis assume that the data available are crisp. But in actual practice this may not be true. Procedures considering 'fuzziness in data' should be developed.

References

1. Adil, G.K., Rajamani, D., and Strong, D., (1993), "An algorithm for cell formation considering alternate process plans", Proceedings of the IASTED International Conference on Modelling and Simulation, Pittsburgh, PA, USA, May 10-12, 285-288.

2. Ahmed, M.U., Ahmed, N.U., and Nandkeolyar, U., (1991), "A volume and material handling cost based heuristic for designing cellular manufacturing cells", Journal of Operations Management, 10 (4), 489-511.

3. Askin, R.G., Cresswell, S.H., Goldberg, J.B., and Vakharia, A.J., (1991), "A hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing", International Journal of Production Research 29, (6), 1081-1100.

4. Ballakur, A. and Steudel, H.J., (1987), "A within cell utilization based heuristic for designing cellular manufacturing systems", International Journal of Production Research, 25 (5), 639-665.

5. Boctor, F.F., (1991), "A linear formulation of the machine-part cell formation problem", International Journal of Production Research 29, (2), 343-356.

6. Burbidge, J.L., (1993), "Comments on clustering methods for finding GT groups and families," Journal of Manufacturing Systems, 12 (5), 428-429.

7. Burbidge, J.L, (1989), "Production flow analysis for planning group technology," Clarendon Press, Oxford.

8. Chan, H.M. and Milner, D.A., (1982), "Direct clustering algorithm for group formation in cellular manufacture", Journal of Manufacturing Systems, 1 (1), 64-76.

9. Chandrasekharan, M.P. and Rajagopalan, R., (1986), "MODROC: an extension of rank order clustering for group technology," International Journal of Production Research, 24 (5), 1221-1233.

10. Chandrasekharan, M.P. and Rajagopalan, R., (1987), "ZODIAC- an algorithm for concurrent formation of part-families and machine-cells," International Journal of Production Research, 25 (6), 835-850.

11. Chandrasekharan, M.P. and Rajagopalan, R., (1989), "GROUPABILITY: an analysis of the properties of binary data matrices for group technology", International Journal of Production Research, 27 (7), 1035-1052.

12. Cheng, C., (1992), "Algorithms for grouping machine groups in group technology", OMEGA International Journal of Management Science, 20 (4), 493-501.

13. Choobineh, F., (1988), "A framework for the design of cellular manufacturing systems", International Journal of Production Research, 26 (7), 1161-1172.

14. Flynn, B.B. and Jacobs, F.R., (1986), "A simulation comparison of group technology with traditional job shop manufacturing", International Journal of Production Research, 24 (5), 1171-1192.

15. Gupta, J.N.D. and Dudek, R.A., (1971),"Optimality criteria for flowshop schedules", AIIE Transactions, 3 (3), 199-205.

16. Ham, I., Hitomi, K., and Yoshiba, T., (1985), "Group technology: applications to production management", Kluwer-Nijhoff Publishing, Boston-Dordrecht-Lancaster.

17. Harhalakis, G., Nagi, R., and Proth, J.M., (1990), "An efficient heuristic in manufacturing cell formation for group technology applications", International Journal of Production Research, 28 (1), 185-198.

18. Heragu, S.S., (1994), "Group technology and cellular manufacturing", IEEE Transactions in Systems, Man and Cybernetics, 24 (2), 203-215.

19. Kasilingam, R.G. and Lashkari, R.S., (1991), "Cell formation in the presence of alternate process plans in flexible manufacturing systems", Production Planning and Control, 2 (2), 135-141.

20. Kasilingam, R.G., (1989), "Mathematical programming approach to cell formation problems in flexible manufacturing systems," Unpublished Doctoral Dissertation, Department of Industrial Engineering, University of Windsor, Canada.

21. Kern, G.M. and Wei, J.C., (1991), "The cost of eliminating exceptional elements in group technology cell formation", International Journal of Production Research, 29 (8), 1535-1547.

22. King, J.R., (1980), "Machine-component group formation in group technology," OMEGA International Journal of Management Science, 8 (2), 193-199.

23. King, J.R. and Nakornchai, V., (1982), "Machine part group formation in group technology-review and extension", International Journal of Production Research, 20 (2), 117-133.

24. Kusiak, A., (1987), "The generalized group technology concept", International Journal of Production Research, 25 (4), 561-569.

25. Kusiak, A. and Cho, M., (1992), "Similarity coefficient algorithms for solving the group technology problem", International Journal of Production Research, 30 (11), 2633-2646.

26. Laarhoven, P.J.M. van and Aarts, E.H.L., (1987), "Simulated annealing: theory and applications", Kluwer Academic Publications, Dordrecht, Netherland.

27. Logendran, R., (1992), "A model for duplicating machines in the presence of budgetary limitations in cellular manufacturing", International Journal of Production Research, 30 (3), 683-694.

28. Logendran, R., (1991), "Impact of sequence of operations and layout of cells in cellular manufacturing", International Journal of Production Research, 29 (2), 375-390.

29. Logendran, R., (1990), "A workload based model for minimizing total inter cell and intracell moves in cellular manufacturing", International Journal of Production Research, 28 (5), 913-925.

30. McAuley, J., (1972), "Machine grouping for efficient production", The Production Engineers 51 (2), 53-57.

31. McCormick, W.T., Schweitzer, P.J., and White, T.W., (1972), "Problem decomposition and data recognization by a clustering technique", Operations Research, 20, 993-1009.

32. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., (1953), "Equation of state calculations by fast computing machines", Journal

of Chem. Physics, 21, 1087-1092.

33. Miltenburg, J. and Zhang, W., (1991), "A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology", Journal of Operations Management, 10 (1), 44-72.

34. Morris, J.S. and Tersine, R.J., (1990), "A simulation analysis of factors influencing the attractiveness of group technology cellular layouts", Management Science, 36 (12), 1567-1578.

35. Mosier, C.T., (1989), "An experiment investigating the application of clustering procedures and similarity coefficient to the GT machine cell formation problem", International Journal of Production Research, 27 (10), 1811-1835.

36. Okogbaa, O.G., Chen, M., Changchit, C., and Shell, R.L, (1992), "Manufacturing system cell formation and evaluation using new inter-cell flow reduction heuristic" International Journal of Production Research, 30 (5), 1101-1118.

37. Rajamani, D., Singh, N., and Aneja, Y.P., (1993), "A generalized framework for the design of cellular manufacturing system", A working paper, Department of Mechanical and Industrial Engineering, University of Manitoba, Winnipeg, Canada.

38. Rajamani, D., Singh, N., and Aneja, Y.P., (1992), "A model for cell formation in manufacturing systems with sequence dependence", International Journal of Production Research, 30 (6), 1227-1235.

39. Rajamani, D., (1990), "Design of cellular manufacturing systems", Unpublished Doctoral Dissertation, Department of Industrial Engineering, University of Windsor, Canada.

40. Ribeiro, J.F.F. and Pradin, B., (1993), "A methodology for cellular manufacturing design", International Journal of Production Research, 31 (1), 235-250.

41. Sankaran, S. and Kasilingam, R.G., (1993), "On cell size and machine requirement planning in group technology systems", European Journal of Operational Research, 69 (3), 373-383.

42. Seifoddini, H. and Wolfe, P.M., (1986), " Application of the similarity coefficient method in group technology", IIE Transactions, 18 (3), 271-277.

43. Shafer, S.M., Kern, G.M., and Wei, J.C., (1992), "A mathematical programming approach for dealing with exceptional elements in cellular manufacturing", International Journal of Production Research, 30 (5), 1029-1036.

44. Shafer, S.M. and Meredith, J.R., (1990), "A comparison of selected manufacturing cell formation techniques", International Journal of Production Research, 28 (4), 661-673.

45. Snyder, W.S. and Irani, S.A., (1986), "Is group-technology for you?", Manufacturing Systems, IE News, XX (4), spring, 1-3.

46. Song, S. and Hitomi, K., (1992), "GT cell formation for minimizing intercell parts flow", International Journal of Production Research, 30 (12), 2737-2753.

47. Srinivasan, G. and Narendran, T.T., (1991), "GRAFICS-a nonhierarical clustering algorithm for group technology", International Journal of Production Research, 29 (3), 463-478.

48. Stecke, K.E., (1981), "Production planning problems for flexible manufacturing systems", Unpublished Doctoral Dissertation, Purdue University, West Lafayette, Indiana, U.S.A.

49. Sule, D.R., (1991), "Machine capacity planning in group technology", International Journal of Production Research, 29 (9), 1909-1922.

50. Waghodekar, P.H. and Sahu, S., (1984), "Machine component cell formation in GT: MACE", International Journal of Production Research, 22 (6), 937-948.

51. Wemmerlov, U. and Hyer, N.L., (1987), "Procedures for the part family/machine group identification problem in cellular manufacturing", Journal of Operations Management, 6 (2), 125-147.

52. Wemmerlov, U. and Hyer, N.L., (1987), "Research issues in cellular manufacturing", International Journal of Production Research, 25 (3), 413-431.

53. Wemmerlov, U. and Hyer, N.L., (1986), "Procedures for the part family/machine group identification problem in cellular manufacturing", Journal of Operations management, 6 (2), 125-147.

Appendix

Let,

$|A|$ = number of elements in set A

c = index of cell

d = number of '1's in the diagonal blocks

e = number of exceptional elements in the solution

M = number of machines

M_c = set of machines assigned to cell c

m = index of machine

O = number of '1's in matrix a_{pm}

P = number of parts

P_c = set of parts allocated to cell c

p = index of part

v = number of voids in the solution

Then,

$$O = \sum_p \sum_m a_{pm}$$

$$d = \sum_c \sum_{p \in P_c} \sum_{m \in M_c} a_{pm}$$

$$v = \sum_c |M_c|\,|P_c| - d$$

$$e = O - d$$

Grouping measure (η_g)

$$\eta_g = \eta_u - \eta_m$$

where,

$$\eta_u = \frac{d}{d+v}$$
$$\eta_m = 1 - \frac{d}{O}$$

Grouping efficiency (η)

$$\eta = (1-w)\frac{O-e}{O-e+v} + w\frac{M.P-O-v}{M.P-O+e-v}$$

A value of 0.5 is recommended in ZODIAC (Chandrasekharan and Rajagopalan, 1987) for w.

Grouping efficacy (τ)

$$\tau = \frac{O-e}{O+v}$$