



HAL
open science

MANUFACTURING CELLS DESIGN WITH ALTERNATIVE ROUTINGS IN GENERALIZED GROUP TECHNOLOGY: REDUCING THE COMPLEXITY OF THE SOLUTION SPACE

Kimon Spiliopoulos, Stella Sofianopoulou

► **To cite this version:**

Kimon Spiliopoulos, Stella Sofianopoulou. MANUFACTURING CELLS DESIGN WITH ALTERNATIVE ROUTINGS IN GENERALIZED GROUP TECHNOLOGY: REDUCING THE COMPLEXITY OF THE SOLUTION SPACE. *International Journal of Production Research*, 2007, 45 (06), pp.1355-1367. 10.1080/00207540600719674 . hal-00512911

HAL Id: hal-00512911

<https://hal.science/hal-00512911>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



MANUFACTURING CELLS DESIGN WITH ALTERNATIVE ROUTINGS IN GENERALIZED GROUP TECHNOLOGY: REDUCING THE COMPLEXITY OF THE SOLUTION SPACE

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2006-IJPR-0212
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	16-Mar-2006
Complete List of Authors:	Spiliopoulos, Kimon Sofianopoulou, Stella; University of Piraeus, Department of Industrial Mgmt and Tech
Keywords:	CELLULAR MANUFACTURE, GROUP TECHNOLOGY
Keywords (user):	ALTERNATIVE PROCESS PLANS



1
2
3
4
5
6
7
8
9
10
11 **MANUFACTURING CELLS DESIGN WITH ALTERNATIVE ROUTINGS**
12
13 **IN GENERALIZED GROUP TECHNOLOGY:**
14
15 **REDUCING THE COMPLEXITY OF THE SOLUTION SPACE**
16
17
18
19
20
21
22
23

24 K. Spiliopoulos and S. Sofianopoulou

25
26 Department of Industrial Management, University of Piraeus
27
28
29

30 **Word count** 4,737 words
31
32
33

34
35 **Abstract**
36

37
38 In the presence of alternative routings, the manufacturing cells design
39 involves the solution of a large number of (simple) cell formation problems.
40
41 We present a bounding scheme which examines all combinations of
42 alternative routings and solves only a few cell formation problems, thereby
43 limiting the solution space which is searched heuristically. This leads to
44 increased reliability of the solutions obtained. The resulting algorithm is
45 shown to be applicable for problems arising in practice frequently, when the
46 initial design includes relatively few exceptional elements, and machine
47 duplication and/or genuine multiple process plans are used correctively, to
48 eliminate the remaining intercellular traffic.
49
50
51
52
53
54
55
56
57
58
59
60

1. Introduction

The manufacturing systems Cell (or Cluster) Formation Problem (CFP) addresses the problem of creating manufacturing cells with minimum interactions, a central issue in the area of Group Technology. The aim is to decentralise production by first grouping the machines into clusters and the various part types into part families, and then allocate the processing of each part family to a machine cluster. The benefits from such an arrangement include the reduction of transport, queuing and processing times, the elimination of the need for frequent set-ups and tool changes in the machines, the reduction of inventories and the simplification of the production plan. These benefits lead naturally to better delivery times, quality improvements, more efficient management and customer satisfaction. The application of cellular manufacturing is also an appropriate first step towards a JIT production scheme.

In practice, the creation of completely autonomous manufacturing cells is very seldom feasible (Wemmerlov and Hyer 1989) and one uses additional equipment, parts subcontracting and alternative process plans, in an attempt to decrease as much as possible the intercellular traffic, that is the traffic generated by parts visiting machines in different cells.

In the presence of alternative routings for some part families (due to genuine multiple process plans or duplicated machines), one has to solve a very large number of CFP problems, each corresponding to a specific

1
2
3 combination of process plans. But CFP alone is already difficult to solve, if
4 the problem is not over-simplified (for example with clustering methods).
5
6 Exact algorithms (see for example Chen and Heragu 1999, Heragu and
7
8 Chen 1998) are computationally expensive, therefore the usual approach for
9
10 solving multiple CFP problems is to use a heuristic technique to obtain near-
11
12 optimal solutions, see indicatively the algorithms given in Sofianopoulou
13
14 (1996, 1997), Lozano *et al.* (1999) and Spiliopoulos and Sofianopoulou
15
16 (2003).
17
18
19
20
21
22
23
24
25

26 Necessarily, in an alternative routings setting, the above have led to
27
28 algorithms which have a doubly stochastic nature, that is both the cell
29
30 formation and the selection of routings are examined in a heuristic
31
32 framework. For example, the method in Sofianopoulou (1999) first
33
34 generates random routings and then solves the resulting CFP problems with
35
36 a simulated annealing procedure. Adenso-Díaz *et al.* (2001) attempt to
37
38 minimize the heuristic component by moving only in the solution space of
39
40 the CFP problem in a tabu search fashion, and assigning the best routes to
41
42 each machine cell configuration. The algorithm in Wu *et al.* (2004) is
43
44 similar: tabu search is used to move in the solution space of alternative
45
46 routings and for each combination, a CFP solution is obtained with a
47
48 similarity coefficients technique. The second and third approaches are not of
49
50 course without problems, because in the space being searched one does not
51
52 have a priori information about the potential changes in the objective
53
54
55
56
57
58
59
60

1
2
3 function. This effectively reduces the power of the heuristic technique
4 employed.
5
6
7

8
9
10 In this work, we attempt to eliminate half of the stochastic component and
11 reduce the complexity of the solution space, by examining all combinations
12 of process plans, thereby limiting the heuristic computations to solving
13 distinct CFP sub-problems. The advantage of this approach is in the
14 increased reliability of the solutions obtained. We apply a bounding
15 technique and show that the number of CFP sub-problems that actually
16 need to be solved is very small compared to the total number of CFP
17 problems that arise (one for each combination of process plans). The
18 proposed approach is shown to be practical when the average intercellular
19 traffic (of the CFP sub-problems solved) is not excessive. This situation
20 appears frequently in practice, when the initial design is adequate, and
21 additional equipment and/or alternative process plans are used to eliminate
22 the remaining intercellular traffic.
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

The relation of the proposed procedure with the CFP is shown in the following figure 1. As shown in this figure, the actual search is done on a subset of the combinations of process plans, as many of them lead to equivalent CFP sub-problems.

[Figure 1]

The problem statement is presented in section 2, including the IP formulation for the CFP problem and the method for the calculation of machines "distances" used. Section 3 presents the description of the algorithm and computational results with test cases are given in section 4. Finally, conclusions are given in section 5.

2. Problem Statement

For each part type p , $p=1,\dots,P$, let $r(p) = 1,\dots, R_p$ denote the alternative routings for part type p . Then for each routing and each machine i , $i=1,\dots,n$ let:

$$a_{r(p)i} = \begin{cases} t, & \text{if part } p \text{ with routing } r(p) \text{ visits} \\ & \text{machine } i \text{ at step } t \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

and

$$\delta_{ijr(p)} = \begin{cases} 1, & \text{if } a_{r(p)i} a_{r(p)j} \neq 0 \text{ and } |a_{r(p)i} - a_{r(p)j}| = 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

Then, in (2.2), $\delta_{ijr(p)}$ shows whether part p with routing $r(p)$ visits machines i, j in two successive steps or not. The modifications for the case of parts re-visiting some machines are trivial: in (2.1), $a_{r(p)i}$ can take more than one non-zero value, all of which must be taken into account in (2.2). Collecting the contributions from all parts, to represent the traffic between machines i and j in any fixed combination of routings, we arrive at the following 0-1 IP formulation for the CFP with bounded cell size from above:

$$z = \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{p=1}^P \delta_{ijr(p)} (1 - x_{ij}) \quad (2.3)$$

s.t.

$$\sum_{i=1}^{k-1} x_{ik} + \sum_{j=k+1}^n x_{kj} \leq M - 1 \quad k = 1, \dots, n \quad (2.4)$$

$$\left. \begin{aligned} x_{ij} + x_{ik} - x_{jk} &\leq 1 & i = 1, \dots, n-2 \\ x_{ij} - x_{ik} + x_{jk} &\leq 1 & j = i+1, \dots, n-1 \\ -x_{ij} + x_{ik} + x_{jk} &\leq 1 & k = j+1, \dots, n \end{aligned} \right\} \quad (2.5)$$

$$x_{ij} = \begin{cases} 1, & \text{if machines } i, j \text{ are in the same cell} & i = 1, \dots, n-1 \\ 0, & \text{otherwise} & j = i+1, \dots, n \end{cases} \quad (2.6)$$

For any fixed assignment of parts to alternative routings, the objective function (2.3) minimizes the inter-cellular moves. Constraint (2.4) ensures that any machine is in the same cell with at most M-1 other machines, and constraints (2.5), the "triangular" constraints, enforce the integrity of cells.

Therefore, the overall problem is a series of CFP sub-problems, i.e. to find a combination of routings $r(1), r(2), \dots, r(P)$ so as to minimize the solution of (2.3) – (2.6).

This formulation for the CFP was first given in Sofianopoulou (1994, 1996) and takes into account the most important parameters, namely the sequencing order of the part families in the routings and the maximum

1
2
3 allowed size for the cells. The importance of considering the sequencing
4 order of the parts in a routing, a somewhat neglected factor, is discussed in
5 Harhalakis, Nagi and Proth (1990). This formulation combines simplicity and
6 adequate fit to realistic situations. Other considerations, such as having
7 volume weighting in the routings and separation and/or co-location
8 constraints for the machines, although not implemented in this work, can be
9 easily incorporated in the proposed algorithm, as discussed in section 3.
10
11
12
13
14
15
16
17
18
19
20
21
22

23 CFP is known to be NP-hard, see Garey and Johnson (1979). In fact, when
24 the cell sizes are fixed, it can be equivalently formulated as a Quadratic
25 Assignment (QAP), a Quadratic Transportation (QTP) or a Matrix Norm
26 Minimisation problem, see for example Spiliopoulos and Sofianopoulou
27 (1998). As it will be explained in section 3, we do transform the problem to
28 an equipartition one, by adding dummy machines without parts traffic, in
29 order to exploit some advantages of the QTP formulation.
30
31
32
33
34
35
36
37
38
39
40
41
42

43 **3. Description of the Algorithm**

44 The algorithm is based on a nested procedure. The outer loop examines the
45 cost matrices which arise from the different combinations of process plans.
46 The inner procedure solves cell formation problems with these cost
47 matrices.
48
49
50
51
52
53
54
55
56
57

58 As stated in section 1, we want to avoid stochastic components in the outer
59 loop, therefore the examination of the combinations of process plans is
60

1
2
3 exhaustive, only taking care to avoid repetitions of cost matrices which
4
5 have exactly the same elements. To achieve the latter, we notice that in
6
7 practice, cost matrices are repeated if in the sequence of parts routings
8
9 $r(1), r(2), \dots, r(P)$, there is a subset of parts P_s and two different selections
10
11 of routings for the parts in P_s :
12
13

$$14 \quad r(p) \neq r'(p), p \in P_s$$

15
16
17 that are equivalent, i.e.
18
19
20
21

$$22 \quad \sum_{p \in P_s} \delta_{ijr(p)} = \sum_{p \in P_s} \delta_{ijr'(p)} \quad \text{for every } i, j \quad i = 1, \dots, n-1 \quad j = i+1, n$$

23
24
25
26
27
28
29
30
31 Then, from any set of combinations with a "partial" selection of routings as
32
33 above and the same routings in the parts not in P_s , one can keep only one
34
35 member of the set. Furthermore, one does not need to store and search
36
37 actual routings selection information to locate such repetitions. Because of
38
39 the depth-tree like search of the combinations, we use an appropriate
40
41 indexing system, test the routings selections with "masks" to see if there is
42
43 a repetition, and then calculate the index of the equivalent combination.
44
45
46
47
48
49

50
51 In all test cases examined in section 4, it was sufficient to search for
52
53 subsets of parts of cardinality 2 and 3, to locate all cost repetitions. Larger
54
55 subsets should appear in rare cases with too many parts and are not of
56
57 interest.
58
59
60

1
2
3
4 After this pre-processing, a bounding technique is used to decide which of
5
6 the cell formation sub-problems need to be solved in the inner procedure.
7
8 In this, a solution is obtained with an Ant Colony Optimization (ACO)
9
10 algorithm. The description of this algorithm is out of scope in this work, first
11
12 because it can be replaced by any other similar procedure and second
13
14 because it contains several details that need to be elaborated in length. To
15
16 give a rough idea, as noted in section 2, the ACO procedure is applied on a
17
18 reformulation of the problem as an equipartition one, by adding dummy
19
20 machines without parts traffic. In this case, the problem is as a Quadratic
21
22 Transportation Problem (QTP) which means that methods from the
23
24 successful ACO applications for the more general Quadratic Assignment
25
26 Problem (QAP) (see for example Solimanpur *et al.* 2004, for an application
27
28 in the cells layout design) can be adjusted and applied. Furthermore,
29
30 contrary to the QAP case, there is a tight bound for QTP with equipartitions,
31
32 namely the Rendl-Wolkowicz one, see for example Spiliopoulos and
33
34 Sofianopoulou (1998), which can enhance the solution procedure.
35
36 Indicatively, the ACO procedure for the cell formation problem produced, in
37
38 lower computation times, the same (proven optimal) results as the Tabu
39
40 Search algorithm proposed in Spiliopoulos and Sofianopoulou (2003), and
41
42 even improved the results in some very large instances for which optimality
43
44 had not been proven.
45
46
47
48
49
50
51
52
53
54
55
56
57

58 The bounding technique is based on the observation that CFP problems with
59
60 small differences in their cost matrices cannot differ much in their optimal

solutions. More specifically, let $z(C^1)$ denote the optimal solution of a problem with cost matrix C^1 , and $C^2=C^1+\Delta C$ the cost matrix of any "neighbouring" CFP sub-problem, that is a sub-problem where there is only one change in a process plan, in one part. For $z(C^2)$, the optimal solution to this problem:

$$\begin{aligned} z(C^2) &= \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij}^2 (1-x_{ij}) = \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n (C_{ij}^1 + \Delta C_{ij}) (1-x_{ij}) \\ &\geq \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij}^1 (1-x_{ij}) + \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta C_{ij} (1-x_{ij}) \\ &= z(C^1) + \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta C_{ij} (1-x_{ij}) \end{aligned}$$

The difference matrix ΔC contains as only non-zero elements:

- a value of -1 in a position (i,j) when there is a link between machines i and j in the first process plan (with C^1) but not in the second (with C^2),
- a value of +1 in a position (i,j) when there is a link between machines i and j in the second process plan but not in the first.

The optimal solution to the CFP problem with this difference matrix ΔC is, in general, the number of elements equal to -1, because in (2.3) - (2.6) one can set $x_{ij}=0$ for these elements and $x_{ij}=1$ for the elements equal to +1.

This holds for the simple cases when the difference is in only one or two links, that is when two routings differ in a start or end process machine or in an intermediate process machine. In the general case, one must take into account the maximum cell size (which prevents too many +1 elements to be

1
2
3 put in the same cell) and the triangular constraints (which may force some
4
5
6 -1 elements to take a value of +1). The extension to cases where some
7
8 links appear twice or more in any matrix (because of repeated moves
9
10 between the same machines) is straightforward. The important point is that
11
12 this optimal solution, say k , can be pre-computed as above and stored for
13
14 each pair of alternative routings in the same part. Then, $z(C^2) \geq z(C^1) - k$, or
15
16 by symmetry, $z(C^1) \geq z(C^2) - k$.

17
18
19
20
21
22
23 If we store lower bounds z_{LB} for the unique cost matrices and C^1 is the
24
25 current cost matrix being examined:

$$z(C^1) \geq z(C^2) - k \geq z_{LB}(C^2) - k$$

26
27
28
29
30
31
32
33
34
35 Therefore, when a cost matrix C^1 is met, we test its neighbours with cost
36
37 matrices C^2 . If a lower bound is stored for C^2 , $z_{LB}(C^2) - k$ is a candidate
38
39 lower bound for C^1 . The maximum of these values is a lower bound for the
40
41 current cost matrix. If this lower bound is greater than or equal to the value
42
43 of the best solution found so far, the current cost matrix can be fathomed.
44
45
46
47
48
49

50
51 If a cost matrix is not fathomed, the ACO procedure is used to obtain a
52
53 solution z_{ACO} . To allow for greater propagation of bounds to cost matrices
54
55 not yet met, we assume that the ACO procedure produces optimal solutions.
56
57 This is not so restrictive, as it will be discussed later. Furthermore, this
58
59 assumption is not an unreasonable one, given the sizes of the problems we
60

1
2
3 are interested in and the performance of this procedure in practice. We note
4
5 here that the ACO procedure consistently produces optimal solutions for
6
7 sizes up to 30 machines.
8
9

10
11
12 Under this assumption, when the ACO procedure has produced a solution
13
14 $z_{ACO}(C^1)$ for the current cost matrix C^1 , for all its neighbouring matrices C^2 :
15
16

$$17 \quad z(C^2) \geq z(C^1) - k = z_{ACO}(C^1) - k$$

18
19
20
21
22
23
24
25

26 Therefore, the lower bounds for the neighbouring cost matrices can be set
27
28 or updated.
29
30
31
32

33 An important point is that, when the ACO procedure is used to obtain a
34
35 solution z_{ACO} , we do not use this value to update the global best solution,
36
37 but an even better one. Given the assignment of machines into cells in each
38
39 ACO solution, we select, for each part, the routing with the minimum
40
41 number of intercellular moves. This selection does not of course correspond
42
43 to the particular CFP sub-problem but results to an overall feasible value
44
45 which is obviously always better than z_{ACO} . This practice leads to better
46
47 fathoming and also allows to the final solution to appear early in the search,
48
49 an important feature when computations take a long time.
50
51
52
53
54
55
56
57

58 This is also why it is not crucial for the inner procedure to perform
59
60 optimally. Even if some combination that would lead to a better value is

1
2
3 incorrectly fathomed, it is the particular cost matrix that is lost, and not the
4
5
6 corresponding allocation of machines into cells. Obviously, there are many
7
8
9 different cost matrices that can lead to the same equivalent allocation (to
10
11 see this, one can start from an optimal solution and swap indices in
12
13 machines that belong to the same cell). Furthermore, different allocations of
14
15 machines into cells can have the same "best cost", i.e. cost resulting from
16
17 selecting the best routes, therefore the probability of actually losing a better
18
19 value is further reduced.
20
21
22
23
24
25

26 As mentioned in section 2, other considerations, such as having volume
27
28 weighting in the routings and separation and/or co-location constraints for
29
30 the machines, although not implemented in this work can be easily
31
32 incorporated in the proposed algorithm:
33
34
35
36
37

- 38 - In the case of volume weightings in the routings, each link in a cost
39
40 matrix is actually multiplied by a corresponding volume factor.
41
42
- 43 - Separation and co-location constraints can be taken into account in the
44
45 inner procedure, by preventing moves that separate some machines, or
46
47 by enforcing simultaneous movement of some machines as a group,
48
49 respectively.
50
51
52
53
54

55 The overall algorithm is shown in the following figure 2.
56
57
58
59
60

[Figure 2]

4. Computational Results

The proposed algorithm was tested with several problems from the literature. The machine used for the tests had a processor speed 1.8 GHz and 512 MB RAM. The programming language was Fortran 90 and the compiler was NAGWare f95 for Windows (which produces C code for the Cygwin gcc compiler).

The results of the tests are summarized in Table 1 below, sorted by increasing number of parts (P), which is more relevant than the number of machines. In this table, n and M are the number of machines and the maximum cell size, respectively. The table also shows, for each problem, the total number of CFP sub-problems, the number of those sub-problems that were actually solved, the best value for the intercellular moves obtained (z) and the computation time.

[Table 1]

Problems P1–P4 are problems with small numbers of CFP sub-problems and the bounding procedure does not have adequate time to show benefits, hence the high percentage of the CFP sub-problems actually solved (except from problem P2). Problem P1 is from Sofianopoulou (1999), and is in fact a version of a classic small problem in Kusiak (1990), modified to have sequence information. The solution produced is the same as in this work. Problem P2 is from Logendran *et al.* (1994). This problem has a small

1
2
3 number of parts and machines (5 and 4, respectively) but the operations in
4 the alternative process plans are not machine-specific and can be performed
5 in either of two machines. The optimal solution has no intercellular moves
6 and was found already from the first CFP problem solved. The 3 seconds
7 CPU time required was for the pre-processing. Problem P3 is the second
8 largest problem in Won and Kim (1997) - the first largest one is an artificial
9 one with far too many combinations because almost all parts from 28 ones
10 had alternative routings, and quite dissimilar ones. The original problem had
11 no sequence information and this was added randomly. The same solution
12 as in the work cited was found in 1 second, after solving only 111 CFP sub-
13 problems. Problem P4 is again from Sofianopoulou (1999) and the solution
14 produced is the same as in this work.
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

34
35 Problems P5-P7 are problems of medium size for parts and larger numbers
36 of CFP sub-problems. Problem P5 is from Sofianopoulou (1999), and is a
37 typical representative of the situations most suitable for the application of
38 the proposed algorithm. In the work cited, this problem was constructed
39 from problem P4, by first selecting only the first process plan for each part
40 and then adding two duplicate machines. Although the resulting number of
41 combinations (2,097,152) is relatively high, the actual CFP problems solved
42 were only 0.13%. Problem P6 is another problem representative of the
43 applicability of the proposed algorithm, a frequently quoted one from Nagi
44 *et al.* (1990) and clearly shows that the critical factor for the performance is
45 not the number of combinations but the quality of the bounds, i.e. the
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3 average number of intercellular moves in the CFP solutions. The original
4 problem had also production volumes information which was ignored here.
5
6 This problem has a large number of combinations (10,077,696), due to the
7
8 presence of several functionally equivalent machines. However, the
9
10 solutions of the CFP sub-problems have low values with result to tight
11
12 bounds. As a result, the CFP problems solved were only 1,042. The solution,
13
14 as shown in figure 3, has only one inter-cellular move. In this figure, the
15
16 index after each part number shows the routing selected.
17
18
19
20
21
22
23
24
25
26

27 **[Figure 3]**
28
29

30
31 We are not aware of that solution found in the literature (for example,
32 Adenso-Díaz *et al.* 2001 report a solution with 4 inter-cellular moves). Of
33 course, as the test with problem P5 shows, it is not necessary for the
34 average inter-cellular traffic to be that low for the algorithm to produce a
35 solution fast. Problem P7 is also from Sofianopoulou (1999). In this
36
37 problem, the proposed algorithm took excessive time but improved the
38 solution from 34 to 32 intercellular moves. Furthermore, the final solution
39
40 was obtained at the combination numbered 3,399 (out of 4,423,680) i.e. in
41
42 a few seconds. It should be pointed out here that, because of the method
43
44 used for the update of the global best solution, the final solution appears
45
46 early in the search. This means that the procedure can be interrupted at an
47
48 early stage, when the computation time has passed a certain threshold.
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3 Problems P8 and P9 are experiments with larger numbers of parts, taken
4 from simple CFP data. Problem P8 is the classic 43-parts, 16-machines
5 problem of Burbidge (1975) with the sequence of machine operations
6 introduced by Kern and Wei (1991). We modified this problem to include
7 two duplicate machines and set the maximum cell size to 6. To select the
8 machines to duplicate, we first produced a solution to this single CFP
9 problem, shown in figure 4 below.
10
11
12
13
14
15
16
17
18
19
20
21
22

23 **[Figure 4]**
24
25
26
27

28 This solution has 27 intercellular moves. Parts visiting machines 6, 8 and 10
29 create almost all of the intercellular traffic. Machines 6 and 8 are a very
30 special case, because too many parts (31) visit one or both of them (8 and
31 23 parts, respectively). Selecting to duplicate these two machines
32 concurrently would result to an enormous number of CFP combinations ($4^8 \times$
33 $2^{23} = 5.5 \times 10^{11}$) and memory problems would occur. Therefore, machines 6
34 and 10 were duplicated. Although the resulting number of combinations is
35 still large (67,108,864), only 1,584 CFP problems had to be solved, in 186
36 CPU seconds. The solution obtained reduced the intercellular moves to 19.
37
38 Finally, problem P9 is a similar experiment with a 50-parts, 30-machines
39 incidence matrix from Gonçalves and Resende (2004) (problem number 31
40 in that work), where we added processing sequence numbers and set the
41 maximum cell size to 6. The solution to the single CFP problem had 16
42 intercellular moves. Contrary to problem P8, each part is processed by only
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3 a few machines, therefore we could afford to duplicate the 3 machines that
4 create the most intercellular traffic (machines 5, 7 and 10). The solution
5 obtained reduced the intercellular moves to 14, in 189 CPU seconds. The
6 small number of CFP sub-problems clearly shows that the limiting factor is
7 not the number of parts per se, but the degree of overlap in the sequences.
8 Also, because the number of CFP sub-problems actually solved is small, the
9 number of machines does not play a significant role.
10
11
12
13
14
15
16
17
18
19
20
21
22

23 **5. Conclusions**

24 We presented a bounding scheme which allows to examine all combinations
25 of alternative routings and solve only a few cell formation problems, thereby
26 limiting the solution space which is searched heuristically. This is of obvious
27 importance for the reliability of the solutions obtained. Much of the
28 elimination of cell formation sub-problems is based on the assumption that
29 the inner procedure used to obtain individual solutions is exact. Although we
30 could not derive results in terms of probabilities, we provide arguments to
31 support that this assumption is not crucial. This inner procedure is not
32 described in detail in this work, one reason being that it can be considered
33 as a black-box one, and can be replaced by any other similar algorithm (in
34 fact, by any algorithm that incorporates additional parameters excluded in
35 the current implementation).
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

58 The computational tests indicate that this approach is viable for problems
59 where the average intercellular traffic for the CFP sub-problems solved is
60

1
2
3 not excessive. Examples of applications include the cases when machine
4 duplication and genuine multiple process plans are used to eliminate
5 remaining exceptional elements in an already adequate design, a situation
6 which arises frequently in practice. Counter-examples are complicated cases
7 where even in the optimal solution there is too much intercellular traffic.
8
9

10
11 The existence of hundreds of millions of combinations of routings is only a
12 constraint with the current implementation of the algorithm, due to memory
13 limitations in information stored for book-keeping purposes. We believe that
14 there are many potential improvements, namely in the pre-processing of a
15 problem, to locate reduction tests and eliminate combinations that are
16 inferior to others. This is an area of further investigation, together with the
17 most proper way to add other parameters for the CFP problem.
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

38 **8. References**

39
40 Adenso-Díaz, B., Lozano, S., Racero, J., and Guerrero, F., 2001, Machine
41 cell formation in generalized group technology. *Computers and Industrial*
42 *Engineering*, **41**, 227-240.
43
44
45
46
47

48 Burbidge, J. L., 1975, *The Introduction of Group Technology* (NY: Wiley).

49
50 Chen, J.-S., and Heragu, S. S., 1999, Stepwise decomposition approaches
51 for large scale formation problems. *European Journal of Operational*
52 *Research*, **113**, 64 –79.
53
54
55
56

57
58 Garey, M. R., and Johnson, D. S., 1979, *Computers and Intractability: A*
59 *Guide to the Theory of NP-Completeness* (NY: Freeman).
60

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
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Goncalves, J. F., and Resende, M. G. C., 2004, An evolutionary algorithm for manufacturing cell formation. *Computers and Industrial Engineering*, **47**, 247-273.

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**, 185-198.

Heragu, S. S., and Chen, J.-S., 1998, Optimal solution of cellular manufacturing system design: Bender 's decomposition approach. *European Journal of Operational Research*, **107**, 175 –192.

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**, 1535-1547.

Kusiak, A., 1990, *Intelligent Manufacturing Systems* (Englewood Cliffs, NJ : Prentice Hall).

Logendran, R., Ramakrishna, P., and Sriskandarajah, C., 1994, Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research*, **32**, 273-397.

Lozano, S., Adenso-Díaz, B., Eguia, I., and Onieva, L., 1999, A one-step tabu search algorithm for manufacturing cell design. *Journal of the Operational Research Society*, **50**, 509-516.

Nagi, R., Harhalakis, G., and Proth, J. M., 1990, Multiple routings and capacity considerations in group technology applications. *International Journal of Production Research*, **28**, 2243-2257.

1
2
3 Sofianopoulou, S., 1994, A mathematical programming approach to the
4 manufacturing systems cell formation problem. Proceedings, 2nd Balkan
5 Conference on OR, October 1993, Salonica, Greece.
6
7

8
9 Sofianopoulou, S., 1996, Part and machine partitioning problem in cellular
10 manufacturing: multiple machine environment. *Yugoslav Journal of*
11 *Operations Research*, **6**, 183-191.
12
13

14
15 Sofianopoulou, S., 1997, Application of simulated annealing to a linear
16 model for the formation of machine cells in group technology. *International*
17 *Journal of Production Research*, **35**, 501-511.
18
19

20
21 Sofianopoulou, S., 1999, Manufacturing cells design with alternative process
22 plans and/or replicate machines. *International Journal of Production*
23 *Research*, **37**(3), 707-720.
24
25

26
27 Solimanpur, M., Vrat, P., and Shankar, R., 2004, Ant colony optimization
28 algorithm to the inter-cell layout problem in cellular manufacturing.
29 *European Journal of Operational Research*, **157**, 592-606.
30
31

32
33 Spiliopoulos, K., and Sofianopoulou, S., 1998, An optimal tree search
34 method for the manufacturing systems cell formation problem. *European*
35 *Journal of Operational Research*, **105**, 537-551.
36
37

38
39 Spiliopoulos, K., and Sofianopoulou, S., 2003, Designing manufacturing
40 cells: a staged approach and a tabu search algorithm. *International Journal*
41 *of Production Research*, **41**(11), 2531 –2546.
42
43

44
45 Wemmerlov, U., and Hyer, N. L., 1989, Cellular manufacturing in the U.S.
46 industry: A survey of users. *International Journal of Production Research*,
47 **27**(9), 1511-1530.
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3 Won., Y., and Kim, S., 1997, Multiple criteria clustering algorithm for solving
4 the group technology problem with multiple process routings. *Computers*
5 *and Ind. Engineering*, **32**(1), 207-220.
6
7
8
9

10
11 Wu, T.-H., Chen, J.-F., and Yeh, J.-Y., 2004, A decomposition approach to
12 the cell formation problem with alternative process plans. *International*
13 *Journal of Advanced Manufacturing Technology*, **24**, 834–840.
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

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
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

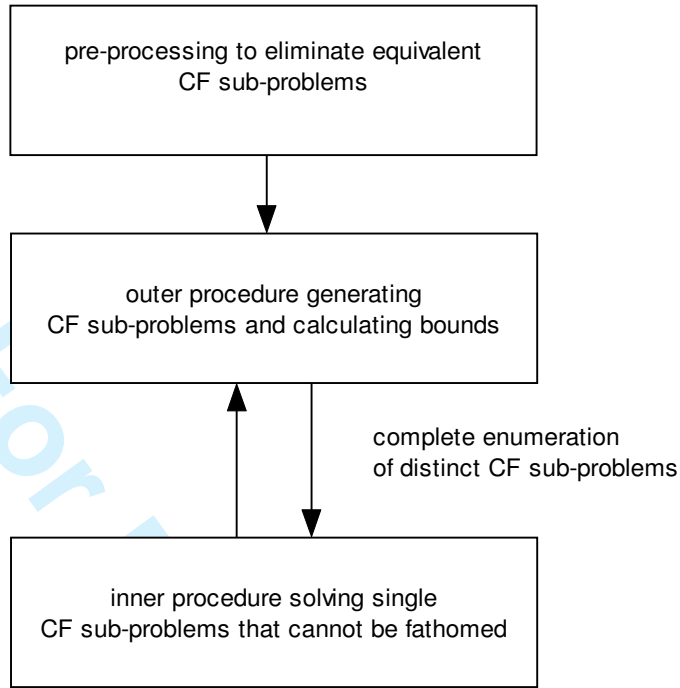


Figure 1.

The relation of the proposed algorithm with the CFP.

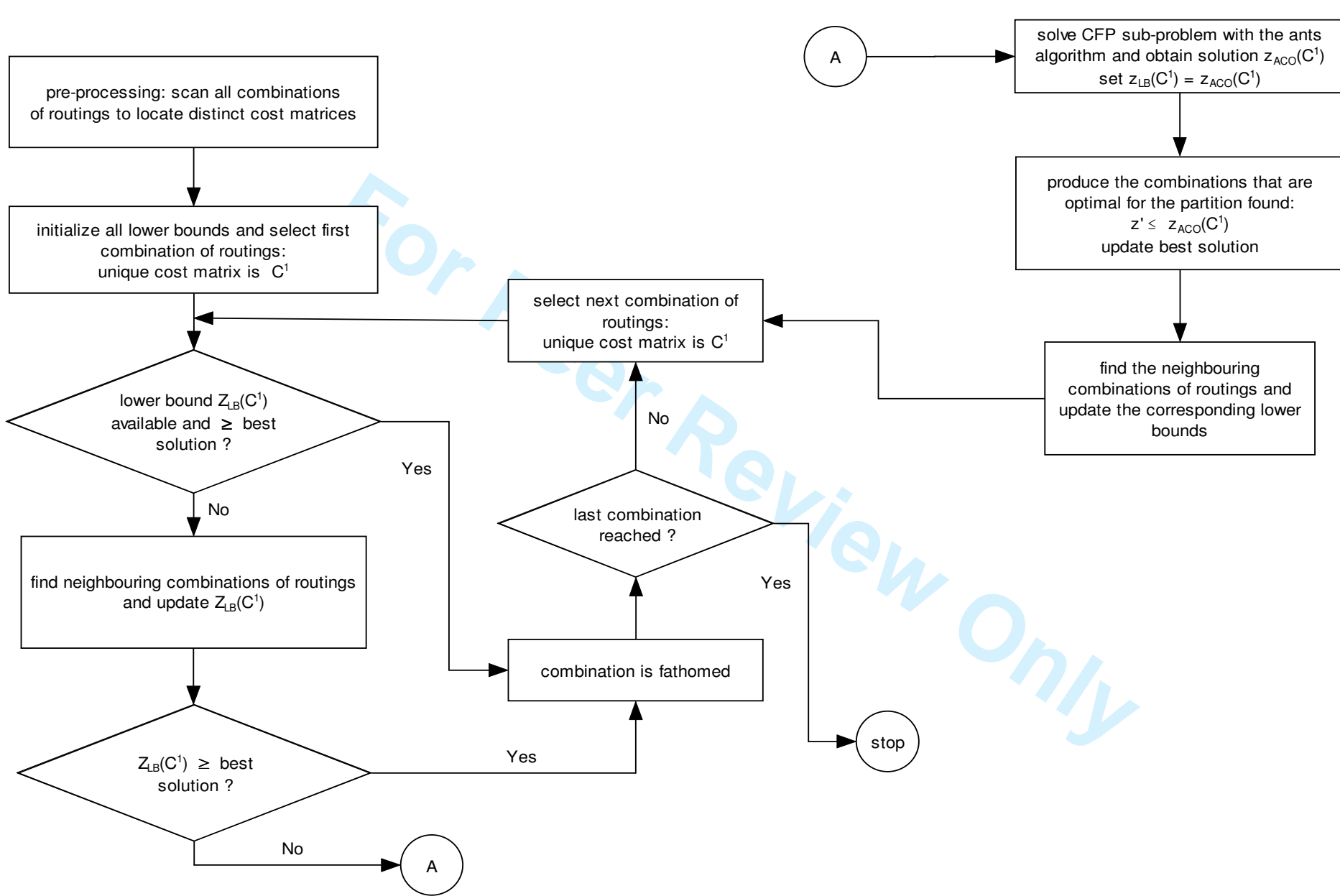


Figure 2. Outline of the proposed algorithm.

parts	machines																			
	10	14	17	18	4	13	15	1	6	9	12	2	5	7	16	19	3	8	11	20
5 (1)	3		2	1																
9 (1)	3	2	1																	
11 (1)	1	3	4	2																
19 (1)	2	3		1																
3 (1)					2		1													
13 (1)					2	1	3													
20 (1)					1	2	3													
1 (1)								3	2	1										
6 (1)								3	1		2									
10 (1)								1	4	2	3									
12 (1)								1	3		2									
14 (1)								2		1	3					4				
2 (1)												2		4	1	3				
4 (2)												3	2	1						
7 (1)													1	4	2	3				
16 (1)												3		2	1					
17 (2)												1	3	4	2					
8 (3)																	3	2	1	4
15 (3)																	2	1	3	4
18 (3)																	1	2		3

Figure 3. The solution to the 20-parts, 20-machines problem from Nagi *et al.* (1990).

parts	machines												inter-cellular moves				
	2	3	6	9	14	16	4	5	8	11	12	15		1	7	10	13
2	3		2	1	6	5			4								2
4				1													
6				1	2												
7		1	2														
10	2			1		3											
12				1				3						2			2
17			1	2		3											
18					1		2										
28	3				1				2								2
32	1		3	2		4											
34		2	1														
35			1			2											
36			1														
37	6		2	1		5			4				3				3
38	2			3		1			4								1
39				2										1			1
40	1		2	3													
42	3		4	2		5							1				1
3								3	2					1			1
5							3	2									
8			3					1	2								1
9							4	1	3	2							
11								1			2						
14			3				2	1									2
15								2	1								
16								1									
19				5			1	4	2								1
20								2	1								
21							1	2	4								3
22											1						
23			4				1	2	3								1
24								3	2	1				4			1
27								1	3	2							
29							2	1									
30										1	2						
31									1						2		1
33			2					3									2
41								1	3								
43			1					2	3								1
1			2						1					3	4		2
13			2											3	1		2
25														1	2		
26															1		

Figure 4. Selection of machines to duplicate in Burbidge's 43-parts, 16-machines problem.

Problem	P	n	M	Total CFP problems	CFP problems solved	%	z	CPU seconds
P1	5	4	2	48	8	16.67	0	< 1
P2	5	4	2	36,864	1	< 0.01	0	3
P3	10	11	3	2,304	111	4.82	4	1
P4	20	12	5	48	29	60.42	29	1
P5	20	14	5	2,097,152	2,656	0.13	25	37
P6	20	20	5	10,077,696	1,042	< 0.01	1	82
P7	30	18	7	4,423,680	19,514	0.44	32	687
P8	43	18	6	67,108,864	1,584	< 0.01	19	186
P9	50	33	6	131,072	632	0.48	14	189

Table 1. The performance of the proposed algorithm with problems from the literature.

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
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

**MANUFACTURING CELLS DESIGN WITH ALTERNATIVE
ROUTINGS IN GENERALIZED GROUP TECHNOLOGY:
REDUCING THE COMPLEXITY OF THE SOLUTION SPACE**

Corresponding author : Associate Professor S. Sofianopoulou

Department of Industrial Management,

University of Piraeus

80 Karaoli & Demetriou Str., 185 34 Piraeus,

Greece.

Phone: 210-4142147 Fax: 210-4142328

e-mail:sofianop@unipi.gr