Int J Flex Manuf Syst
DOI 10.1007/s10696-006-9011-3

Solving group technology problems via clique partitioning

Haibo Wang · Bahram Alidaee · Fred Glover · Gary Kochenberger

Abstract This paper presents a new clique partitioning (CP) model for the Group Technology (GT) problem. The new model, based on a novel 0/1 quadratic programming formulation, addresses multiple objectives in GT problems by drawing on production relationships to assign differing weights to machine/part pairs. The use of this model, which is readily solved by a basic tabu search heuristic, is illustrated by solving 36 standard test problems from the literature. The efficiency of our new CP model is further illustrated by solving three large scale problems whose linear programming relaxations are much too large to be solved by CPLEX. An analysis of the quality of the solutions produced along with comparisons made with other models and methods highlight both the attractiveness and robustness of the proposed method.

Keywords Clique partitioning · Group technology · Tabu search · Metaheuristics

---

H. Wang
College of Business Administration, Texas A&M International University,
Laredo, USA
e-mail: hwang@tamiu.edu

B. Alidaee
School of Business, University of Mississippi, Oxford, USA
e-mail: balidaee@bus.olemiss.edu

F. Glover
University of Colorado, Boulder, USA
e-mail: Fred.Glover@colorado.edu

G. Kochenberger (✉)
School of Business, University of Colorado at Denver,
Denver, USA
e-mail: Gary.Kochenberger@cudenver.edu

1 Introduction

Group Technology (GT) is an attractive strategy employed to achieve economic efficiency in flexible manufacturing systems. The basic idea is to group machines and parts together in a manner that facilitates economies in time and cost. In flexible manufacturing settings, a machine/part pair is called an *exception* if and only if the part has to visit the machine in order to complete the processing when the part and the machine are not assigned to the same cell in the cellular formation. A machine/part pair is called a *void* if the part does not have to visit the machine but they are assigned to the same cell. In general, the objectives for the GT problem are: to reduce the number of duplicated machines, to reduce the number of exceptional elements and to increase the machine utilization rates. In particular, increase in the utilization rate of a machine can be achieved by reducing the number of *voids*. Increase in manufacturing productivity can be achieved by reducing the number of *exceptions* to shorten the traveling distance for materials used to produce parts. GT has many advantages over the traditional process organization such as shortening throughput times, providing better quality, reducing material handling cost, keeping loads balanced, increasing capacity due to shorter setup times, and even bringing better job satisfaction due to increased team work. In the past four decades, many models and methods have been proposed for addressing GT problems. Many of the key approaches are highlighted in Table 1 below.

The perspective on solving group technology problems advanced here is to adopt a graph theoretical point of view where nodes in the graph, representing machines and parts, are connected by edges denoting the association of each pair of nodes in the network. This basic approach to GT problems was first proposed by (Rajagopalan and Batra 1975) and similar approaches have been reported by (Chu 1995; Ham et al. 1985; King and Nakornchal 1982; Shafer and Rogers 1993). The partitioning problem, formally defined below in Sect. 2, is to cluster the nodes into cliques with similar characteristics. Despite the conceptual "fit", the clique partitioning model failed to emerge as a viable approach in practice due to the difficulty of solving the standard 0/1 programming model for CP. Even for modest sized GT problems, the standard optimization model for CP explodes in size making it difficult if not impossible to solve by standard methods. This computational difficulty has served to preclude the broader use of the clique partitioning model as a useful tool in the area of group technology.

The alternative model we present here for clique partitioning removes the size and computational issues mentioned above. Our purpose in this paper is to present this new model for clique partitioning and to show its potential application to solving GT problems. In the sections below we first present the classic model for clique partitioning followed by our new model. We then present a small example illustrating the use of the new model as a tool for GT. The model is further illustrated by applying it to 36 test problems from the literature. This is followed by our summary and conclusions.

Table 1 Literature review for existing methods on solving GT

Methodology	Features	References
Classification and Coding: Production Flow Analysis	It is based on the shape or function similarity among the parts.	(Burbidge 1963)
Binary Array Clustering Methods: Rank Order Clustering, Modified Rank Order Clustering, Similarity Coefficient with or without Seed	While two machines or parts are grouped together at some stages, there is no way to retrace the steps even if it leads to suboptimal clustering at the end. It precludes formation of better machine groups at later stages	King (1980a, b), Seifoddini and Wolfc (1986)
Multivariate Clustering Methods: Ideal Seed Non-hierarchical Clustering, Single Linkage clustering or Average Linkage Clustering, Bivariate Clustering	The machine cells and part families are also not formed simultaneously. The performance of these algorithms is associated to the data structure of a binary machine/part incidence matrix, which has a limitation of incorporating many production variables such as production volume, material handling cost and others.	Malakooti and Yang (2002), Rogers and Kulkarni (2005), Seifoddini (1988)
Graphic Theoretical, Mathematical Programming and Heuristic Approaches: Artificial Neural Network, Simulated Annealing, Genetic Algorithms, 0/1 integer programming formulation, Multiple Criteria Decision Making, Column Generation, and Cutting Plane algorithm for standard clique partitioning formulation.	The graph- theoretic approach tends to require a more complex implementation and longer computational time, it may produce a well-structured cell formulation	Gunasingh and Lashkari (1990), Jaumard et al. (1999), Joines et al. (1996), Malakooti and Yang (2002), Malakooti and Zhou (1998), Oosten et al. (2001), Rajagopalan and Batra (1975)

2 Clique partitioning

Consider a complete graph G = (V, E) with n vertices and unrestricted edge weights. The clique partitioning problem (CP) consists of partitioning the graph into cliques such that the sum of the edges weights over all cliques formed is as large as possible. This is an NP-hard problem with applications reported in many diverse areas. The standard optimization model for CP (see for instance, Grotschel and Wakabayashi 1989; Grotschel and Wakabayashi 1990; Chopra 1993; Oosten et al. 2001) is given by:

$$\begin{aligned}
\text{CP(Edge)} : \text{Max}\, x_0 \ &= \sum_{(i,j)\in E} w_{ij}x_{ij} \qquad (1)\\
\text{st}\quad & \\
& x_{ij} + x_{ir} - x_{jr} \leq 1 \ \ \forall \text{ distinct } \ i,j,r \ \in V \\
& x_{ij} \in \{0,1\}
\end{aligned}$$

where the w_{ij} are unrestricted edge weights and x_{ij} is defined to be 1 if edge (*i,j*) is in the partition, and equal to 0 otherwise. Note that this is an *edge-based* formulation and even for modest sized graphs, this model explodes is size having $n(n\text{–}1)/2$ variables and $3C_3^n$ constraints. Despite these size characteristics, the dominate methods presented in the literature for solving CP (edge) are exact approaches based on LP methods as illustrated by the cutting plane approaches of Grotschel and Wakabayashi (1989) and Oosten et al. (2001), and the column generation approach of Mehrotra and Trick (1998). These approaches have proven to be successful on small to moderate size problems. For larger instances, however, their application is severely limited due the challenge presented by the large size of CP (edge). For such cases, metaheuristic methods, coupled with a new formulation, prove to be very effective as illustrated below.

2.1 New formulation

The computational challenge posed by CP (edge) for large problem instances motivates the development of a new formulation that can be readily solved by basic metaheuristic methodologies. We first present the new model and then describe our solution approach.

As before, n is the number of nodes (vertices) and the w_{ij} are unrestricted edge weights. Without loss of generality we assume here that G is a complete graph. If necessary, artificial edges with negative (penalty) edge weights can be introduced as needed to produce a complete graph in those cases where G is not initially complete. In addition, define

$$\begin{aligned} k\_\text{max} = {} & \text{maximum number of cliques allowed} \\ & \text{(estimated based on domain knowledge)} \end{aligned}$$

and

$$x_{ik} = 1 \text{ if node i is assigned to clique } k; 0 \text{ otherwise}$$

Then our model is:

$$\text{CP (Node)} : \text{Max } x_0 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \sum_{k=1}^{k\_\max} x_{ik} x_{jk}$$

$$\text{st} \tag{2}$$

$$\sum_{k=1}^{k\_\max} x_{ik} = 1 \quad i = 1, n$$

Note that the quadratic terms in the objective function imply that the weight w_{ij} becomes part of the partition weight only when nodes i and j are assigned to the same clique. The constraints of (4) require that each node is assigned to one of the cliques formed.

Several remarks about this model are in order: First of all, note that this is a node-oriented model with many fewer variables than CP (edge) since $n(k\_\max)$ is typically much less than $n(n-1)/2$. Furthermore, the number of constraints here (n) is much smaller than the corresponding number $(3C_3^n)$ for the edge-oriented model of CP (edge). While CP (edge) is a linear model and CP (node) is quadratic, the size difference enables this quadratic alternative to be used for large instances of clique partitioning problems where the computational burden of CP (edge) precludes its practical use. As we'll demonstrate later in this paper, CP (node) can be effectively solved, even for large instances, by modern meta-heuristic methods such as tabu search.

2.2 Solving CP (node):

CP (node) could in principle be solved by any of a variety of methods designed for nonlinear integer programmes (See for example the papers by Hansen 1979; Hansen et al. 1993). In our work we adopt an approach that employs a slight reformulation that enables rapid solution via modern meta-heuristic methods we have implemented. We note that CP (node) is of the form

$$\text{Max } x'Qx$$

subject to assignment constraints requiring that each node is assigned to one of the $K\_\max$ cliques formed. Our approach to solving this model is to first re-cast it into the form of cardinality constrained binary quadratic program (CBQP) which we can readily solve by the tabu search method given in (Glover et al. 1998). This reformulated version of CP(node) takes the form

$$\text{Max } x'\hat{Q}x \tag{3}$$

subject to the single cardinality constraint

$$\sum_{i=1}^{n} \sum_{k=1}^{k\_\max} x_{ik} = n$$

The Q matrix is modified (to yield $\hat{Q}$) via the inclusion of penalties ensuring that a given node is assigned to at most one clique. The single cardinality constraint requires that exactly n assignments will be made. Working in concert, the penalties together with the cardinality constraint require that each node will be assigned to exactly one clique.

Such reformulation has proven to be very fruitful in a variety of other settings as we have reported in the survey paper (Kochenberger et al. 2004). Our motivation here is to leverage the advances we have reported elsewhere in the recent literature for solving unconstrained and cardinality constrained quadratic binary programmes.

This slightly reformulated model, CBQP, can be readily solved by a basic Tabu Search methodology designed for the generic cardinality constrained binary quadratic programme. An overview of the heuristic is given in the appendix of this paper. Complete details are given in (Glover et al. 1998).

2.3 Clique partitioning and the GT problem

Throughout the paper we assume that we have M machines and P parts. Clique partitioning can be used to group parts and machines by first representing the problem as a complete graph $G(V,E)$ where the vertex set contains a node for each part and for each machine (i.e., $| V | = M + P$). Edge weights are determined as follows. If the part is associated with the machine, the edge weight between the part node and the machine node is 1 and –1 otherwise. The weight of an edge between pair of parts or between pairs of machines is 0.

This approach is illustrated by the following example taken from (Kumar et al. 1986) with 9 machines and 15 parts (denoted as GT21 in Table 2). The standard binary part/machine incident matrix for this example is given in Fig. 1. The GT graph for this example has 24 nodes and allows a maximum of 9 possible cliques. Thus, with K_max taken to be 9 we have our CP (node) model:

$$\begin{aligned}
\text{Max}\quad f(x) = {} & w_{12}(x_{11}x_{21} + x_{12}x_{22} + x_{13}x_{23} + x_{14}x_{24} + x_{15}x_{25} + x_{16}x_{26} + x_{17}x_{27} \\
& + x_{18}x_{28} + x_{19}x_{29}) + \cdots\cdots + w_{23,24}(x_{23,1}x_{24,1} + x_{23,2}x_{24,2} \\
& + x_{23,3}x_{24,3} + x_{23,4}x_{24,4} + x_{23,5}x_{24,5} + x_{23,6}x_{24,6} + x_{23,7}x_{24,7} \\
& + x_{23,8}x_{24,8} + x_{23,9}x_{24,9}) \\
& \text{st} \\
& x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} = 1 \\
& \cdots\cdots \\
& x_{24,1} + x_{24,2} + x_{24,3} + x_{24,4} + x_{24,5} + x_{24,6} + x_{24,7} + x_{24,8} + x_{24,9} = 1 \\
& x_{ij} \in \{0,1\} \text{ for } i = 1,\ldots 24 \text{ and } j = 1,\ldots 9
\end{aligned} \tag{4}$$

where the edge weights, w_{ij} are 1, –1, or 0 based on the simple procedure described at the beginning of this section. This model, which has 216 binary

Table 2 List of the GT test problems and literature references

Name of test problems	References of test problems
GT1, GT2, GT3, GT4, GT5, GT6, GT7, GT8, GT9, GT10	Boctor (1991), Sofianopoulou (1997)
GT11	Boe and Cheng (1991), Chandrasekharan and Rajagopalan (1986), Li and Parkin (1997)
GT12	Boctor (1989), Burbidge (1963), Burbidge (1991), Chan and Milner (1982), Kattan (1997), Seifoddini and Wolfc (1986)
GT13	Burbidge (1963), Joines et al. (1996), Rogers and Kulkarni (2005)
GT14	Cantamessa and Turroni (1997)
GT24, GT25	Malakooti and Yang (2002)
GT15	Chandrasekharan and Rajagopalan (1987), Joines et al. (1996)
GT16	Burbidge (1963), Burbidge (1991), Kumar et al. (1986), Oosten et al. (2001)
GT17	Kattan (1997)
GT18	King (1980a, b)
GT28	Boe and Cheng (1991), Miltenburg and Zhang (1991)
GT19	Joines et al. (1996), Kattan (1997), King and Nakornchal (1982)
GT20	King and Nakornchal (1982)
GT21, GT22	Kumar et al. (1986)
GT23, GT35	Leskowski et al. (1987)
GT26	Masnata and Settineri (1997)
GT27	Mccormick et al. (1972)
GT29, GT30, GT31, GT32	Nair and Narendran (1996)
GT33	Seifoddini (1988)
GT34	Sule (1991)
GT36	Vannelli and Kumar (1986)

variables and 24 constraints, is too large to present in its entirety here. Complete details are available from the authors.

This model, recast into the form of CBQP, is readily solved by our tabu search heuristic to yield the solution displayed in Fig. 2. To compare with

part

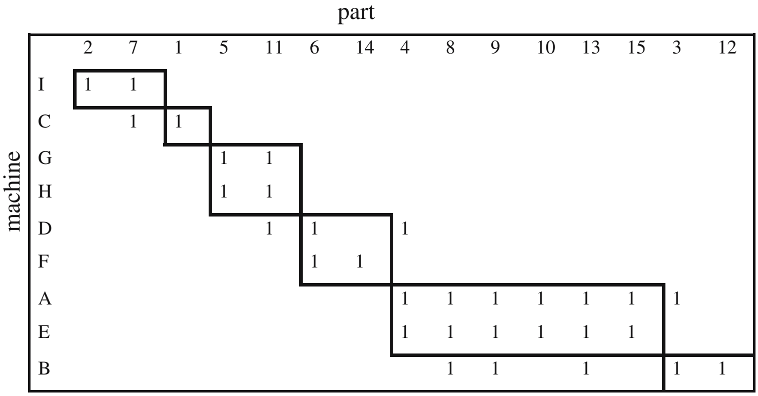
machine

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A			1	1				1	1	1			1		1
B			1					1	1			1	1		
C	1						1								
D				1		1					1				
E		1		1				1	1	1			1		1
F						1								1	
G					1						1				
H					1						1				
I		1					1								

Fig. 1 Binary part/machine incident matrix for 15 parts and 9 machines problem

conventional CP (edge) model, we solved the same problem with CP (edge) model using CPLEX 6.5 with MIP solver.

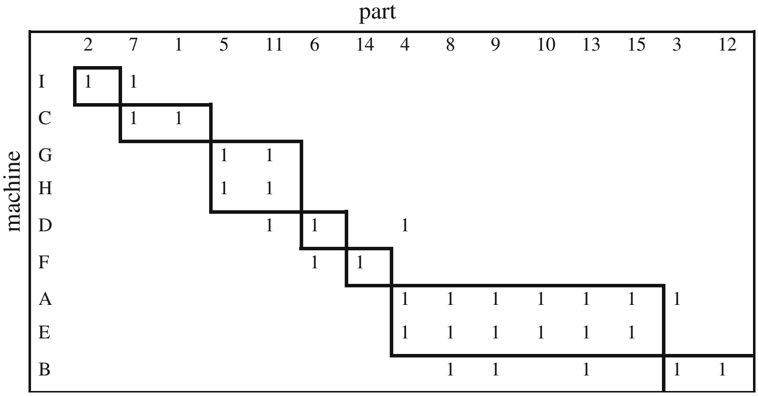
It is interesting to note that the aggregate weight of the groups formed in this solution (i.e., the objective function value) is 23 which is the same value given by the solution obtained by the conventional CP (edge) model as shown in Fig. 3 are quite different. That is, we have alternative optimal solutions with respect to this objective function criterion. Despite having the same aggregate group weight, these solutions differ along several key dimensions such as within-group compactness, number of exception cells, and the number of void cells. The solution from CP (node) model has less number of exceptional cell than the solution obtained by CP (edge) model. Both solutions have a larger objective function value than the solution produced by the K-Decomposition method in the literature (Kumar et al. 1986) as shown in Fig. 4. In general, the comparison of alternative solutions for grouping must extend beyond a single measure like aggregate group weight. Accordingly, in the section below on computational experience, we employ additional measures of solution quality to facilitate a more robust comparison of alternative solutions.



part

machine	2	7	1	5	11	6	14	4	8	9	10	13	15	3	12
I	1	1													
C		1	1												
G				1	1										
H				1	1										
D					1	1		1							
F						1	1								
A								1	1	1	1	1	1	1	
E								1	1	1	1	1	1		
B									1	1		1		1	1

Fig. 2 Group formation solution via CP (node) model



part

machine	2	7	1	5	11	6	14	4	8	9	10	13	15	3	12
I	1	1													
C		1	1												
G				1	1										
H				1	1										
D					1	1		1							
F						1	1								
A								1	1	1	1	1	1	1	
E								1	1	1	1	1	1		
B									1	1		1		1	1

Fig. 3 Group formation solution via CP (edge) model

3 Computational results

To provide a comparative assessment of the performance of CP (node) relative to other methods, we solved 36 standard test problems from the literature. For each problem, the best solution available from the literature was used as a benchmark of comparison for our solutions. Table 2 lists these problems along with the appropriate references.

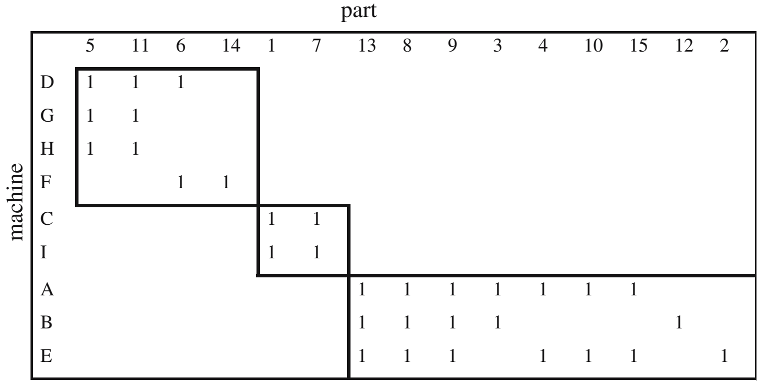
In an effort to provide a comprehensive comparison with other methods, solutions were evaluated along the following three dimensions: Aggregate Grouping Weight, Grouping Measure, and Grouping Efficiency. These measures, especially the last two, are widely used in the literature and collectively enable objective performance comparisons to be made. In what follows we report summary results obtained from our model and from the literature. Detailed results of group formations for all 36 problems are available from the authors upon request.

3.1 Aggregate grouping weight

For this assessment, the total weight of the groups formed according our solution and the solution obtained from the literature was compared for each problems. The resulting values are listed in Table 3. For this measure of solution quality, our method clearly produced attractive results. For most of the problems, the results from our model are strictly preferred to those obtained from other models. In no case did another method produce a better result although there were ten ties out of the 36 problems.

3.2 Grouping measure

In general, the aim of employing group technology is to approximate self contained production cells with few parts requiring processing on the machines in other groups. Some methods for solving GT problems perform



	5	11	6	14	1	7	13	8	9	3	4	10	15	12	2
D	1	1	1												
G	1	1													
H	1	1													
F			1	1											
C					1	1									
I					1	1									
A							1	1	1	1	1	1	1		
B							1	1	1	1				1	
E							1	1	1		1	1	1		1

Fig. 4 Group formation by K-Decomposition method from literature (Kumar et al. 1986)

Table 3 Total grouping weights

Problem ID	Result from CP (node)	Best result from other models
GT1	**58**	23
GT2	**61**	26
GT3	**60**	23
GT4	**50**	21
GT5	**72**	37
GT6	**76**	35
GT7	**78**	46
GT8	**61**	30
GT9	**87**	52
GT10	**70**	45
GT11	**80**	68
GT12	**72**	49
GT13	**103**	100
GT14	**157**	–24
GT15	348	348
GT16	55	55
GT17	**177**	–541
GT18	**41**	32
GT19	**66**	64
GT20	**118**	–947
GT21	23	23
GT22	**53**	8
GT23	**30**	29
GT24	40	40
GT25	42	42
GT26	**41**	37
GT27	43	43
GT28	**46**	–120
GT29	117	117
GT30	**91**	90
GT31	**74**	71
GT32	93	93
GT33	54	54
GT34	46	46
GT35	**78**	14
GT36	**109**	–16

reasonably well with respect to a given objective but fall short on other dimensions of performance. Two widely proposed metrics for assessing grouping results are *within-group (cell) compactness* and the *number of exceptional cells*. It is generally accepted that one group formation is preferred to another if it has greater within-group compactness and a smaller number of exceptional cells. To compare different solution along these dimensions, we employ the grouping metric (GM) used by Islamt and Sarker (2000) (Joglekar et al. 2001; Miltenburg and Zhang 1991), and group efficiency metric (GE) used by Chandrasekharan and Rajagopalan (1987). Our assessment involving GM is given here while that involving GE follows in section c) below.

The grouping measure (GM), first introduced by Miltenburg and Zhang, is designed to calculate the difference between machine utilization rates and parts movement rates. This metric is defined as follows:

$$\begin{aligned}
&\xi_g = \xi_u - \xi_m \\
&\xi_u = \left(\sum_r \sum_{\substack{i \in M_r \\ j \in C_{r'}}} a_{ij} \right) \Big/ \sum_r (|M_r|\ |C_r|) \\
&\xi_m = 1 - \left(\sum_r \sum_{\substack{i \in M_r \\ j \in C_{r'}}} a_{ij} \right) \Big/ \left(\sum_{i,j} a_{ij} \right)
\end{aligned} \tag{5}$$

where ξ_g is the grouping measure, ξ_u is the machine utilization rate, which measures within-group compactness, ξ_m is the parts movement rate. a_{ij} is equal to 1 if the part is processed by the machine, r is the rth machine/part group (cell) in the final group formation, M_r is the number of machines in rth group (cell) and C_r is the number of parts in rth group (cell). Larger ξ_g values indicate better grouping solutions.

As an illustration, consider the two solutions to the small example from Sect. 2.3. For the CP (node) solution (Fig. 2) we have ξ_u = 24/25 = 0.96, ξ_m = 1–24/32 = 0.25, ξ_g = ξ_u – ξ_m = 0.96 – 0.25 = 0.71 and for the solution from Fig. 4 we have ξ_u = 32/47 = 0.681, ξ_m = 0.0 then ξ_g = ξ_u – ξ_m = 0.681 – 0.0 = 0.681. Since the former ξ_g value is greater than the later, we conclude that grouping result obtained via CP (node) is preferred to that obtained from the K-Decomposition method with respect to this metric even though the aggregate grouping weights obtained by both methods are the same.

Table 4 reports the GM results for the 36 problems used in this study. A comparison problem by problem shows that the performance of CP (node) with respect to this metric is strictly preferred to that of the other solutions in 31 of the 36 cases and tied in the remaining 5 cases. In none of the 36 cases did CP (node) produce an inferior solution based on this metric.

3.3 Grouping efficiency

The Grouping Efficiency (GE) metric, η, due to Chandrasekharan and Rajagopalan (1987), is designed to measure the difference between intra-cell utilization and inter-cell movement. This metric utilizes a weighting factor q which can reflect specific requirements of a problem but is commonly set to 0.5 if the density of 1’s in parts/machine matrix is normal. The GE metric for M machines and N parts is defined as follows:

$$
\begin{aligned}
&\eta = q\eta_1 + (1-q)\eta_2 \\
&\eta_1 = \left(\sum_r \sum_{\substack{i \in M_r \\ j \in C_r{}^r}} a_{ij} \right) \Bigg/ \sum_r (|M_r|\,|C_r|) \\
&\eta_2 = 1 - \frac{\left(\sum_{\substack{i \in M \\ j \in N}} a_{ij} - \sum_r \sum_{\substack{i \in M_r \\ j \in C_r}} a_{ij} \right)}{\left(MN - \sum_r |M_r||C_r| \right)} \\
&0 \le q \le 1
\end{aligned}
\tag{6}
$$

Larger values of η denote better grouping results. Taking q to be 0.5 and referring once again to our example of Sect. 2.3 we have for our CP (node) solution η_1 = 24/25 = 0.96, η_2 = 1–(32–24)/(135–25) = 0.9273, and η = 0.9437. For the solution of Fig. 4 we get η_1 = 32/47 = 0.681, η_2 = 1–0/88 = 1, and η = 0.8405. Since the η value for the former is greater that that of the later, we conclude once again that the grouping produced by CP (node) is preferred to the result obtained from the K-Decomposition approach.

Table 5 reports the GE results for the 36 problems considered here. Once again, the performance of CP (node) relative to the other methods is very attractive across the entire line up of test problems with strictly preferred results coming on 31 cases and ties on the remaining 5 cases.

It is clear from the results displayed in Tables 3, 4, and 5 that our solutions, across all three metrics, are very attractive compared to the solutions previously reported in the literature for these test problems. For all problems our approach quickly finds high quality solutions. A more detailed comparative analysis of the solutions indicates that our method strikes a nice balance between intra-call compactness and inter-cell movement.

4 Computational efficiency

In Sect. 2 we presented the standard model for clique partitioning, CP (edge), and we commented that this model, while conceptually sound for application to group technology problems, is in fact flawed due to its excessive computational requirements. In this section we present computational experience illustrating this computational burden by comparing the computational times for the standard model, CP (edge), with those of our new model, CP (node) on the 36 test problems. For each problem, the results from CP (edge) were

Table 4 Grouping measure (GM) comparison on 36 GT problems

Problem ID	Result from CP (node)	Best result from other models
GT1	**0.4753**	0.4675
GT2	**0.5521**	0.5181
GT3	**0.6359**	0.5317
GT4	**0.4508**	0.4429
GT5	**0.6929**	0.5401
GT6	**0.7565**	0.5705
GT7	**0.7163**	0.5923
GT8	**0.5034**	0.4933
GT9	**0.7463**	0.587
GT10	**0.6304**	0.5711
GT11	**0.4859**	0.4135
GT12	**0.5183**	0.4870
GT13	**0.7646**	0.7585
GT14	**0.5427**	–0.0241
GT15	0.8286	0.8286
GT16	**0.4503**	0.005
GT17	**0.4064**	0.1629
GT18	**0.6886**	0.6289
GT19	**0.5015**	0.2345
GT20	**0.3318**	0.1756
GT21	**0.71**	0.681
GT22	**0.4447**	0.2726
GT23	**0.4491**	0.2690
GT24	0.7654	0.7654
GT25	0.92	0.92
GT26	**0.6442**	0.5952
GT27	**0.4954**	0.4832
GT28	**0.53**	0.0075
GT29	0.81	0.81
GT30	**0.6332**	0.6318
GT31	**0.5878**	0.5107
GT32	0.7961	0.7961
GT33	**0.7076**	0.6259
GT34	**0.6298**	0.559
GT35	**0.5941**	0.2883
GT36	**0.3134**	0.1357

obtained by using CPLEX 6.5 and results from CP (node) were obtained from our Tabu Search heuristic. All runs were made on a SUN Enterprise 450 server.

The results of these runs are shown in Table 6. The times listed in Table 6 for our tabu search heuristic are the times required to execute 100 SPAN cycles. (SPAN cycles are defined in the appendix). The times shown for the CPLEX runs are the times required to complete the tree search process. Note that while both models and solution methods were able to successfully solve all 36 problems, the time performance of CP (edge) is erratic and, in most cases, excessive. In contrast to this, the performance of the CP (node) and our tabu search heuristic is very uniform across all problems. These results are displayed graphically in Fig. 5. In most cases, the CP (node)

Table 5 Grouping efficiency comparison on 36 GT problems

Problem ID	Result from CP (node)	Best result from other models
GT1	**0.9044**	0.7598
GT2	**0.8748**	0.7776
GT3	**0.91**	0.7815
GT4	**0.9149**	0.7587
GT5	**0.8860**	0.7949
GT6	**0.9181**	0.7988
GT7	**0.8918**	0.8104
GT8	**0.8662**	0.7739
GT9	**0.9071**	0.8242
GT10	**0.9054**	0.8103
GT11	**0.8911**	0.8482
GT12	**0.9082**	0.7717
GT13	**0.9126**	0.9041
GT14	**0.9226**	0.6429
GT15	0.9521	0.9521
GT16	**0.8891**	0.5628
GT17	**0.8698**	0.6209
GT18	**0.9594**	0.8274
GT19	**0.9274**	0.7251
GT20	**0.9124**	0.5946
GT21	**0.9437**	0.8405
GT22	**0.9035**	0.6616
GT23	**0.9173**	0.7017
GT24	0.915	0.915
GT25	0.96	0.96
GT26	**0.9167**	0.8566
GT27	**0.9254**	0.923
GT28	**0.9660**	0.5933
GT29	0.9635	0.9635
GT30	**0.9178**	0.9050
GT31	**0.9254**	0.8720
GT32	0.9233	0.9233
GT33	**0.8879**	0.8614
GT34	**0.8854**	0.7795
GT35	**0.9575**	0.6777
GT36	**0.9464**	0.5679

approach produced the optimal solution in a fraction of the time required by CPLEX and CP (edge). While CP (edge) often took several days to solve a problem, the largest of problems was solved via CP (node) in little over 1 min.

We note that comparisons of the type made in Table 6 must be made with appropriate caution as our tabu search approach is a heuristic and CPLEX is an exact method. That is, one would expect a heuristic to generally have a time advantage over an exact method. Our purpose here of using CPLEX as a benchmark is to demonstrate that CP (edge) is very difficult for standard commercial methods. In comparison, our approach is very efficient, effective and robust.

Table 6 Comparison of computational times for CP (node) and CP(edge)

Problem ID	# of node	# of edge	CP (node) via tabu search			CP (edge) via CPLEX65			
			# of vars.	Soln	Time(s)	# of vars.	# of constraints	Soln.	Time(s)
GT1	46	1035	736	58	12	1035	45540	58	610255.79
GT2	46	1035	736	61	12	1035	45540	61	11354.76
GT3	46	1035	736	60	12	1035	45540	60	2770.44
GT4	46	1035	736	50	12	1035	45540	50	476203.2
GT5	46	1035	736	72	13	1035	45540	72	783.04
GT6	46	1035	736	76	13	1035	45540	76	3.19
GT7	46	1035	736	78	13	1035	45540	78	10.09
GT8	46	1035	736	61	12	1035	45540	61	38782.30
GT9	46	1035	736	87	13	1035	45540	87	3.67
GT10	46	1035	736	70	12	1035	45540	70	3272.41
GT11	55	1485	1100	80	25	1485	78705	80	567875.57
GT12	59	1711	944	72	18	1711	97527	72	504144.55
GT13	55	1485	550	103	5.3	1485	78705	103	8.11
GT14	68	2278	680	157	5.8	2278	150348	157	395441.53
GT16	43	903	860	55	15	903	37023	55	291346.09
GT18	38	703	532	41	7.4	703	25308	41	11.24
GT19	59	1711	708	66	7.6	1711	97527	66	219550.01
GT21	24	276	216	23	2.5	276	6072	23	0.17
GT22	43	903	860	53	15	903	37023	53	269368.22
GT23	38	703	494	30	6.8	703	25308	30	3269.32
GT24	25	300	250	40	3.3	300	6900	40	0.21
GT25	25	300	250	42	2.9	300	6900	42	0.34
GT26	35	595	350	41	4.4	595	19635	41	262.97
GT27	39	741	624	43	9.4	741	27417	43	15558.70
GT28	60	1770	1500	46	43	1770	102660	46	14106.64
GT29	64	2016	1536	117	50	2016	124992	117	6.95
GT30	64	2016	1536	91	46	2016	124992	91	269160.2
GT31	64	2016	1536	74	45	2016	124992	74	45650.82
GT32	64	2016	512	93	5.2	2016	124992	93	10.34
GT33	33	528	363	54	4.6	528	16368	54	141.85
GT34	31	465	341	46	4.1	465	13485	46	413.77
GT35	71	2485	2130	78	82	2485	171465	78	10142.38

To provide insight into the computational performance of our approach on even larger instances of GT problems, 3 new test problems were generated and solved. These problems, which range in size from 50 machines and 200 parts to 150 machines and 1,000 parts, were modeled ala CP (node) and solved by our Tabu Search heuristic. We note that “real” test problems of the size considered here are not available from the literature for research purposes. As such, we randomly generated these new test problems which are available from the authors for others to try.

The results from these new problems are shown in Table 7. As shown there, even the largest of the problems is readily solved. We comment that these problems are too large to approached via the alternative CP(edge) model as even the initial LP relaxation is too large to solve.

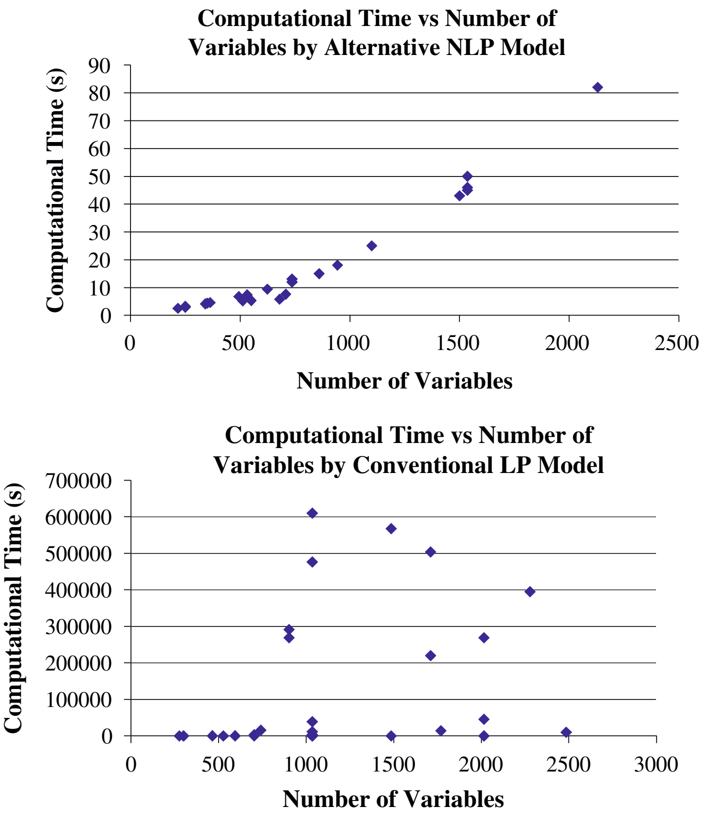


Fig. 5 Computational time versus number of variables

5 Summary and conclusions

In this paper we presented a new modeling and solution methodology, based on clique partitioning and tabu search, for solving group technology problems. This new approach was applied to 36 standard test problems and assessments were made comparing our solutions with the best solutions available from the literature. In making the comparisons, three metrics gauging solution quality were applied. Across all 36 problems, our solutions were uniformly attractive, surpassing the other solutions in quality in most cases and tying them in the

Table 7 Computational results for large-sized test problems via CP (node). Note that these problems are available from the author upon request

Problem	Size ($M \times P$)	# of Cells	# Nodes	# Vars	Solution	Time(s)
GT_50_200	50 × 200	7	250	2500	205	123
GT_100_700	100 × 700	8	800	8,000	704	441
GT_150_1000	150 × 1000	10	1,150	23,000	1,753	1,658

remaining cases. In no case was an alternative solution preferred to ours on any of the metrics.

While the computational testing reported here was carried out on binary matrix test problems, we note that our approach is not restricted to such cases and can be readily applied to the non-binary case. We also note that the model given here can be easily modified to accommodate additional domain knowledge that may be important in a given GT setting. For example, in a parts-oriented setting, a positive edge weight could be added to the pair of part nodes to encourage all part nodes to be grouped into a cell in the final solution. In a similar fashion, a positive edge weight could be assigned to the pair of machine nodes if the machines are required to be grouped. Other special cases can be accommodated by similar constructs.

Based on the results we have presented, we conclude that the model and solution approach advanced here represent an attractive methodology for solving group technology problems. On on-going research addressing larger and more difficult GT problems will be reported in future papers.

Appendix: Overview of tabu search method for CBQP

Our TS method for CBQP is centred around the use of strategic oscillation, which constitutes one of the primary strategies of tabu search. The variant of strategic oscillation we employ may be sketched in overview as follows.

The method alternates between constructive phases that progressively set variables to 1 (whose steps we call “add moves”) and destructive phases that progressively set variables to 0 (whose steps we call “drops moves”). To control the underlying search process, we use a memory structure that is updated at *critical events*, identified by conditions that generate a subclass of locally optimal solutions. Solutions corresponding to critical events are called *critical solutions*. For CBQP a critical event occurs during the solution process when exactly n variables are equal to 1.

A parameter *span* is used to indicate the amplitude of oscillation about a critical event. We begin with *span* equal to 1 and gradually increase it to some limiting value. For each value of *span*, a series of alternating constructive and destructive phases is executed before progressing to the next value. At the limiting point, *span* is gradually decreased, allowing again for a series of alternating constructive and destructive phases. When *span* reaches a value of 1, a *complete span cycle* has been completed and the next cycle is launched.

Information stored at critical events is used to influence the search process by penalizing potentially attractive add moves (during a constructive phase) and inducing drop moves (during a destructive phase) associated with assignments of values to variables in recent critical solutions. Cumulative critical event information is used to introduce a subtle long term bias into the search process by means of additional penalties and inducements similar to those discussed above.

References

Boctor F (1989) Alternative formulations of the machine-part cell formation problem., presented at Proc. 3rd ORSA/TIMS Conf. On Flexible Manufacturing Systems, Amsterdam

Boctor F (1991) A linear formulation of the machine-part cell formation problem. Int J Prod Res 29:343–356

Boe WJ, Cheng CH (1991) A close neighbor algorithm for designing cellular manufacturing systems. Int J Prod Res 29:2097–2116

Burbidge JL (1963) Production flow analysis. Prod Eng 42:742–752

Burbidge JL (1991) Production flow analysis for planning group technology. J Operat Manage, Special Issue Group Technol Cell Manuf 10:5–27

Cantamessa M, Turroni A (1997) A pragmatic approach to machine and part grouping in cellular manufacturing system design. Int J Prod Res 35(4):1031–1050

Chan HM, Milner DA (1982) Direct clustering algorithm for group formation in cellular manufacturing. J Manuf Syst 1:65–74

Chandrasekharan MP, Rajagopalan R (1986) MODROC: an extension of rank order clustering for group technology. Int J Prod Res 24:1221–1233

Chandrasekharan MP, Rajagopalan R (1987) ZODIAC—an algorithm for concurrent formation of part-families and machine-cells. Int J Prod Res 25:835–850

Chopra S, Rao MR (1993) The partition problem. Mathemat Progr 59:87–115

Chu CH (1995) Recent advances in math programming for cell formation. In: Kamrani AK, Parsaei HR, Liles DH (eds) Manufacturing research and technology, Vol 24. Elsevier Science Publishing, Amsterdam, pp 3–46

Glover F, Kochenberger GA, Alidaee B (1998) Adaptive memory tabu search for binary quadratic programs. Manage Sci 44:336

Grotschel M, Wakabayashi Y (1989) A cutting plane algorithm for a clustering problem. Mathemat Progr 45:59–96

Grotschel M, Wakabayashi Y (1990) Facets of the clique partitioning polytope. Mathemat Progr 47:367–387

Gunasingh K, Lashkari R (1990) Simultaneous Grouping of Parts and Machines in Cellular Manufacturing Systems—an Integer programming approach. Comput Indust Eng 20:111–117

Ham I, Hitomi K, Yoshida T (1985) Group technology: applications to production management. Kluwer publishing, Boston

Hansen PB (1979) Methods of nonlinear 0–1 programming. Ann Dis Math 5:53–70

Hansen PB, Jaumard B, Mathon V (1993) constrained nonlinear 0–1 programming. J Comput 5:97

Islamt KMS, Sarker BR (2000) A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems. Int J Prod Res 38:699–720

Jaumard B, Labit P, Ribeiro C (1999) A Column Generation Approach to Cell Formation Problems in Cellular Manufacturing. In: 3rd International Industrial Engineering Conference. Montreal, Canada

Joglekar P, Chund Q, Tavana M (2001) Note on a comparative evaluation of nine well-know algorithm for solving the cell formation problem in group technology. J Appl Mathemat Decision Sci 5:253–267

Joines JA, Culbreth CT, King RE (1996) Manufacturing cell design: an integer programming model employing genetic algorithms. IIE Trans 28:69–85

Kattan I (1997) Design and scheduling of hybrid multi-cell flexible manufacturing systems. Int J Prod Res 35:1239–1257

King JR (1980a) Machine component group formation in group technology. Omega 8:193–199

King JR (1980b) Machine component grouping in production flow analysis: an approach using rank order clustering. Int J Prod Res 18:213–232

King JR, Nakornchal V (1982) Machine component group formation in group technology, review and extension. Int J Prod Res 20:117–133

Kochenberger G, Glover F, Alidaee B, Rego C (2004) A Unified Modeling and Solution Framework for Combinatorial Optimization Problems. Operat Res Spectrum 26:237–250

Kumar RK, Kusiak A, Vannelli A (1986) Grouping of parts and components in flexible manufacturing systems. Eur J Operat Res 24:387–397

Leskowski Z, Logan L, Vanelli A (1987) Group technology decision aids in an expert system for plaint layout, presented at Modern production management system: Proc IFIP TC 5/WG 5-7 Working Conf. On Advances in Production Management Systems, Amsterdam

Li M, Parkin R (1997) Group technology revisited: a simple and robust algorithm with enhanced capability. Int J Prod Res 35:1969–1992

Malakooti B, Yang Z (2002) Multiple criteria approach and generation of efficient alternatives for machine-part family formation in group technology. IIE Trans 34:837–846

Malakooti B, Zhou Y (1998) Approximating polynomial functions by feedforward artificial neural networks: capacity, analysis and design. Appl Mathemat Comput 90:27–52

Masnata A, Settineri L (1997) An application of fuzzy clustering to cellular manufacturing. Int J Prod Res 35:1077–1094

Mccormick WT, Schweitzer PJ, White TW (1972) Problem decomposition and data reorganization by a clustering technique. Operat Res 20:993–1009

Mehrotra A, Trick M (1998) Clique and clustering: a combinatorial approach. Operat Res Lett 22:1–12

Miltenburg J, Zhang W (1991) A comparative evaluation of nine well-known algorithms for solving the cell formation in group technology. J Operat Manage 10:44–72

Nair GJ, Narendran TT (1996) Grouping index: a new quantitative criterion for goodness of block-diagonal forms in group technology. Int J Prod Res 34:2767–2782

Oosten M, Rutten J, Spieksma F (2001) The Clique partitioning problem: facets and patching facets. Networks 38:209–226

Rajagopalan R, Batra JL (1975) Design of cellular production systems. Int J Prod Res 13:567–579

Rogers DF, Kulkarni SS (2005) Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing. Eur J Operat Res 160:423–444

Seifoddini H (1988) Machine grouping—expert systems: Comparison between single linkage and average linkage clustering techniques in forming machine cells. Comput Indust Eng 15:210–216

Seifoddini H, Wolfc PM (1986) Application of the similarity coefficient method in group technology. IIE Trans 18:271–277

Shafer SM, Rogers DF (1993) Similarity and distance measures for cellular manufacturing: Part 1: a survey. Int J Prod Res 31:1133–1142

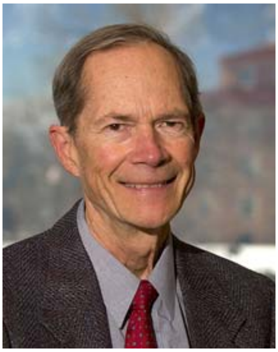
Sofianopoulou S (1997) Application of simulated annealing to a linear model forthe formulation of machine cells in group technology. Int J Prod Res 35:501–511

Sule D (1991) Machine capacity planning in group technology. Int J Prod Res 29:1902–1922

Vannelli A, Kumar KR (1986) A method for finding minimal bottleneck cells for grouping part-machine families. Int J Prod Res 24:387–400

Dr. Haibo Wang is assistant professor of decision science in the Division of International Business and Technology Studies, College of Business Administration at Texas A&M International University. In addition, he is certified doctoral faculty in the Ph. D. program of International Business at Texas A&M International University. He earned a B.S in Biochemical Engineering (1991), a M.S in Chemistry (1996), a M.E. in Computer and Information Science (1997) and a Ph. D. in Production Operations Management (2004). Prior to joining Texas A&M International University, he was a faculty member of Hearin Center for Enterprise Science at University of Mississippi and consultant at Mississippi Center for Supercomputing Research. He has consulting experiences in the areas of web-based application development, software development, supercomputing, business simulation, optimization, international marketing, operations and supply chain management. He has provided consulting, training and application development service to a number of organizations in both private and public sectors. Dr. Wang has publications in such outlets as the European Journal of Operational Research, Journal of Combinatorial Optimization and Journal of Optimization Letters.

Dr. Bahram Alidaee received the B.S. degree from the University of Tehran, Iran, the M.B.A. degree from the University of North Texas and the Ph.D. degree from the University of Texas at Arlington, 1988. He is currently a professor of operations management at the School of Business Administration, the University of Mississippi. His research interests include applied graph theory, heuristic programming, complex systems, game theory and cost allocations. He has published in journals such as Management Science, Transportation Science, Production and Operations Management, European Journal of Operational Research, Journal of Operational Research Society, IEEE Transactions on Systems, Man and Cybernetics, Operations Research Letter, Annals of Operations Research, Information Processing Letters, Applied Mathematics Letters, Journal of Combinatorial Optimization, Annals of Operations Research, Transportation Research, Computers & OR, Computers & IE and other journals. He is member of INFORMS, DSI, POMS, APICS, ISM, and IEEE Computer Society.



Dr. Fred Glover is the MediaOne Chaired Professor in Systems Science at the University of Colorado, Boulder, and Distinguished Professor of the University of Colorado System. He has authored or co-authored more than three hundred fifty published articles and eight books in the fields of mathematical optimization, computer science and artificial intelligence, with particular emphasis on practical applications in industry and government. In addition to holding editorial and advisory posts for journals in the U.S. and abroad, Dr. Glover has been featured as a National Visiting Lecturer by the Institute of Management Science and the Operations Research Society of America and has served as a host and lecturer in the U.S. National Academy of Sciences Program of Scientific Exchange. Professor Glover is the recipient of the distinguished von Neumann Theory Prize, an elected member of the National Academy of Engineering, and has received numerous other awards and honorary fellowships, including those from the American Association for the Advancement of Science (AAAS), the NATO Division of Scientific Affairs, the Institute of Operations Research and Management Science (INFORMS), the Decision Sciences Institute (DSI), the U.S. Defense Communications Agency (DCA), the Energy Research Institute (ERI), the American Assembly of Collegiate Schools of Business (AACSB), Alpha Iota Delta, the Miller Institute for Basic Research in Science, and an honorary doctorate from the Institute of Cybernetics of the Ukrainian Academy of Science. He serves on the advisory boards of several organizations and is co-founder of OptTek Systems, Inc.

Dr. Gary A. Kochenberger is Professor of Operations Management, University of Colorado at Denver (since 1989). Professor Kochenberger earned a Ph.D. in Management Science from the University of Colorado at Boulder (1970) where he also holds a degree in Electrical Engineering. Prior to his years at the University of Colorado, he was a professor of management science at Penn State University from 1969 to 1986 and the Martec Professor of Management at the University of Southern Colorado from 1987 to 1989. He spent the 2000 and 2001 academic years as a visiting Hearin Professor and co-director of the Hearin Center for Enterprise Science at the University of Mississippi. Dr. Kochenberger's academic specialty centers around building, testing, and implementing algorithms for solving resource allocation problems that rise in both the private and public sectors. In recent years, his focus has been on problems of a combinatorial nature as commonly found in logistical management, operations management, and related areas. He has published three books and numerous research papers in top journals in his field including *Management Science, Mathematical Programming, Journal of Optimization Theory and Applications, Operation Research, Computers and Operations research, Naval Research Logistics Quarterly, Decision Sciences, Journal of the Operational Research Society, Interfaces, Operations Research Letters, Omega, and the Journal of the Production and Operations Management Society.* Moreover, he is actively engaged in several major journals including positions as area editors for both INTERFACES and the Journal of the Production and Operations Management Society (POMS). In addition to his academic work, Dr. Kochenberger has been actively engaged in executive education and as a consultant to firms in a variety of industries.