INT. J. PROD. RES., 2001, VOL. 39, NO. 11, 2431–2451

Forming effective worker teams for cellular manufacturing

RONALD G. ASKIN†* and YUANSHU HUANG‡

Cellular manufacturing has been extensively adopted as a measure to reduce cycle time, increase productivity, and improve product quality. The past research in cellular manufacturing has focused on the methodology for identification of machine groups, part families, and determination of processing routes. The relocation of existing workers into cells and their training for a team-oriented, cellular manufacturing environment have largely been ignored. In this research, a mixed integer, goal programming model is formulated for guiding the worker assignment and training process to create worker teams with high team synergy and individual job fitness meeting cell requirements for technical and administrative skills. The model integrates psychological, organizational, and technical factors. Several solution methods including greedy heuristic, filtered beam search, and simulated annealing techniques are developed and tested. It appears that heuristics such as beam search are capable of obtaining good solutions with reasonable computational effort.

1. Introduction

Manufacturing companies have been actively transforming traditional jobshop manufacturing systems into manufacturing cells. Empirical evidence indicates reductions in throughput time, rework, scrap, labour, set-up time, and defects as a result of implementing cells (Wemmerlov and Hyer 1989). In cellular manufacturing (CM) a group of heterogeneous machines and a team of workers are dedicated to producing a family of similar parts. Previous research in CM has focused on the technological problems of forming appropriate part families and machine groups (see Burbidge 1975, King and Nakornchai 1982, Kusiak 1987, Askin and Vakharia 1990, Suresh 1991, 1992, and Singh 1993 for general reviews).

Burbidge (1975) listed a set of dedicated workers as a key principle of cell autonomy (or independence) which in turn is an essential aspect of successful cells in practice. In a survey of industry, Askin and Estrada (1999) found that training of workers was one of the top concerns when implementing cells. The conversion from traditional jobshop production to CM brings a new culture context to the worker team. In creating cells, workers with process oriented skills must be divided into part oriented teams and assigned to cells with heterogeneous processes. Worker training becomes an integral part of cellular team formation and success. In creating empowered teams, additional technical, teamwork, and administrative skills must be developed among the workforce. Workers may be cross-trained on several processes and asked to set-up their own machines and perform quality assurance checks. Together,

Revision received January 2001.
† Department of Systems and Industrial Engineering, The University of Arizona, Tucson, AZ 85721-0020, USA.
‡ Honeywell Inc, Tucson, AZ 85737, USA.
* To whom correspondence should be addressed. e-mail:@ron.sie.arizona.edu

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2001 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540110040466

workers must cooperate to identify and solve problems, schedule maintenance and production activities, order materials, and research potential process improvements. Cell productivity depends not only on the technical and administrative skills the workers possess but also the effective interaction among team members. Despite these critical human resource issues, the formation of worker teams in CM has received only scant attention from researchers.

In this research, the problem of forming an effective worker team for CM is studied. In the next section, we review past literature on the worker assignment problem and the approaches to predict and improve the effectiveness of team member interaction. We then formulate a goal programming model for forming an effective worker team in Section 3. Solution methods are developed in Section 4 for finding an optimal or near-optimal solution for the model. In Section 5, we test and evaluate these solution methods. Section 6 summarizes our work and discusses future directions.

2. Literature review and problem definition

Ebeling and Lee (1994) analysed the cost and benefit of employee cross-training on a mixed-model assembly line and formulated a mixed integer programming model to guide cross-training assignments for a specific number of assembly jobs and workers. Suer (1996) developed mixed integer and integer programming models to achieve optimal product and worker assignment to labor-intensive manufacturing cells. Min and Shin (1993) developed a multi-objective model to form cells with consideration of both machines and workers but training was not considered in their model. Warner *et al.* (1997) discussed relevant factors in assigning workers to cellular teams from both technological and human interaction perspectives. Askin and Huang (1997) formulated an integer programming model for an aggregate worker assignment and training problem for use in converting a functionally organized manufacturing environment into a CM arrangement. Each of these papers addresses specific issues related to our cell team development model. However, we believe this paper is the first to propose and solve a comprehensive quantitative model that incorporates worker skills, technical requirements and team dynamics.

Worker groups are typically formed in an ad hoc manner when CM is implemented. Despite the growing trend to knowledge based job design and the increasing role of the line worker in flexible manufacturing, workers have been treated as an afterthought, with emphasis being placed on technological equipment. Stevens and Campion (1994) studied the knowledge, skill, and ability (KSA) requirements for teamwork. The KSAs were classified as interpersonal type and self-management type and fourteen specific KSAs were summarized for teamwork in their research. These KSAs were summarized for teamwork in their research. These KSAs can assist human resource management in personnel selection and staffing, performance appraisal, career development, compensation, and job analysis. Barrick and Mount (1991) analysed five major personality dimensions and job performance and concluded that certain personality traits could be valid job performance predictors for some occupations and some criterion types. For example, extroversion was a valid predictor for managers and sales positions involving social interactions. Kembel (1996) characterizes personality types along the dimensions Rational, Organized, Loving, and Energized (ROLE). An individual's dominant dimension(s) can be roughly identified by a quiz. Each type of personality has its characteristic motivations and goals, learning process, and modes of communication, leadership and management. The ROLE characterization is used to help workers understand fellow team members.

McCaulley (1990) examined the history, characteristics, and use of the Myers–Briggs Type Indicator (MBTI). The MBTI classifies individuals with bi-polar preferences for four dimensions generating 16 types. Lyman and Richter (1995) used MBTI in forming a quality function deployment team. Team members who were tested by MBTI had a better understanding of each other and this helped the team members to work together more effectively to accomplish team goals. The least Preferred Co-Worker (LPC) measure is also a tool to evaluate the social distance between group members. Individuals with high LPCs are hypothesized to perceive less social distance between themselves and their least preferred co-worker (Hoffman 1984). Schriesheim *et al*. (1994) used lPC scores to further characterize leadership performance predictions.

Kolbe (1993) notes that other psychological profiles concentrate on personality. The Kolbe Conative Index (KCI) was developed to measure the conative or instinctive behaviour traits of individuals. The KCI measures an individual's basic instincts towards the action modes of Probe (Fact Finder), Pattern (Follow Through), Innovate (Quick Start), and Demonstrate (Hands-on Implementor). Values indicate tendencies on a scale from resistance to initiation of actions within each mode. In each dimension, values between 0 and 3 indicate a nature towards resistance to that action, and, values between 7 and 10 indicate a tendency towards initiating actions of that form. A score between 3 and 7 indicates accommodation or a 'responder' at that trait. Most people have a mix of the four traits and are 'insistent' in some traits and 'resistant' in others. Additional measures evaluate the match between an individual and a specific job. The combinations of individual types that produce the most effective teams have been identified. The ideal synergistic team mix consists of individuals that aggregate to a 25%, 50%, 25% mix of Initiator, Responder, and Preventor/Resistor across the four action modes.

Many qualitative studies exist detailing the roles of team members and commenting on how to form a high performance self-directed team. These studies share the shortcomings of not being directly related to cellular manufacturing nor do they provide a clear basis for mathematical modelling. Measures such as MBTI and the Kolbe Index provide a quantification that one can attempt to use for formally defining techniques for cell formation. In this paper, we use KCI measures to define the conative tendencies of workers and their fit for specific jobs and synergy in specific groups.

3. Worker assignment and training model

Our intent is to develop a model for guiding the formation of a worker team for each cell. We assume that part families have already been created based on geometric, usage, and/or processing similarities. Machines have been allocated for the production of each family. Given the technical requirements for each cell and the capabilities and conative measures of the existing labour pool, we intend to select the team of workers for each cell, the training schedule for each worker, and the assignment of tasks to workers within each cell. Technical requirements are derived from process plans and organizational procedures. We denote these as 'skills'. Skills include traditional technical duties such as machine set-up and operation, but also the administrative responsibilities assigned to empowered work teams such as scheduling, and purchasing. Capabilities can be deduced from work history and worker labour grades. KCI measures provide the data required for measuring individual aptitude and team synergy. We define the model as follows.

Decision variables

$$Y_{ik} = \begin{cases} 1 & \text{if worker } i \text{ is assigned to cell } k, \\ 0 & \text{otherwise;} \end{cases}$$

$$Z_{ik} = \begin{cases} 1 & \text{if worker } i \text{ acquires skill } j, \\ 0 & \text{otherwise;} \end{cases}$$

X_{ijk} is the proportion of time worker i does skill j in cell k.

Deviational variables

d_{train} is the amount by which the total training cost exceeds 0;
d^+_{mk} is the amount by which the team assigned to cell k exceeds the goal m;
d^-_{mk} is the amount by which the team assigned to cell k is under the goal m;
d_{fit} is the amount by which the total fitness exceeds 0.

Data coefficients

$$a_{itm} \begin{cases} 1 & \text{if worker } i\text{'s mode of operation is at level } m \text{ for trait } t, \\ 0 & \text{otherwise;} \end{cases}$$

c_{ij} is the cost to train worker i fully in skill j;
f_{ij} is the fitness score of worker i to skill j; $0 \le f_{ij} \le 1$; the closer f_{ij} is to 0, the fitter the worker i is for skill j;
S_k is the number of workers to be assigned to cell k;
S_{jk} is the amount of skill j needed in cell k;
w_{fit} is the Cardinal weight assigned to d_{fit};
w^+_{mk} is the Cardinal weight assigned to d^+_{mk};
w^-_{mk} is the Cardinal weight assigned to d^-_{mk};
w_{train} is the Cardinal weight assigned to d_{train};

The Worker Assignment and Training Model (WAT) becomes:

$$\text{Minimize } Z = w_{\text{train}} d_{\text{train}} + w_{\text{fit}} d_{\text{fit}} + \sum_k \sum_{m=1}^{3} (w^+_{mk} d^+_{mk} + w^-_{mk} d^-_{mk}) \tag{1}$$

Subject to:

$$\sum_i \sum_j c_{ij} Z_{ij} - d_{\text{train}} = 0 \tag{2}$$

$$\sum_k \sum_i \sum_j f_{ij} X_{ijk} - d_{\text{fit}} = 0 \tag{3}$$

$$\sum_i \sum_{t=1}^{4} a_{itm} Y_{ik} + d^+_{mk} - d^-_{mk} = S_k; \quad m = 1, 3; \quad \text{for all } k \tag{4}$$

$$\sum_{l}\sum_{t=1}^{4} a_{itm} Y_{ik} + d_{mk}^{+} - d_{mk}^{-} = 2.0 S_k; \quad m = 2; \quad \text{for all } k \tag{5}$$

$$\sum_{k} Y_{ik} = 1, \quad \text{for all } i. \tag{6}$$

$$\sum_{i} Y_{ik} = S_k, \quad \text{for all } k \tag{7}$$

$$\sum_{i} X_{ijk} \geq S_{jk}, \quad \text{for all } j \text{ and } k \tag{8}$$

$$\sum_{j} X_{ijk} \leq Y_{ik}, \quad \text{for all } i \text{ and } k \tag{9}$$

$$\sum_{k} X_{ijk} \leq Z_{ij}, \quad \text{for all } i \text{ and } j \tag{10}$$

$$Y_{ik} \ 0 \text{ or } 1; \quad Z_{ij} \ 0 \text{ or } 1; \quad 0 \leq X_{ijk} \leq 1; \quad 0 \leq d_{mk}^{+}; \quad 0 \leq d_{mk}^{-} \tag{11}$$

The objective function (1) presents a weighted average of the three objectives — minimizing training cost, minimizing misfit between worker conative traits and job requirements, and minimizing deviations from desired team synergy. Team synergy is measured across the three instinctive modes (m) of initiation, accommodation, and resistance. Equations (2)–(5) are goal constraints and equations (6)–(11) are feasibility constraints. Equation (2) is for minimizing the total training cost. We allow training costs to depend on the worker and skill. Factors such as experience, educational background, motivation, base expertise, and natural aptitude can be include in the model. Equation (3) is included to minimize the total task fitness score. Equations (4) and (5) are to minimize the deviation from the ideal team synergy level. A balance of 25% resistor, 25% initiator, and 50% accommodator across the four action modes is required to ensure that ideas are generated, adequately defined and analysed, and successfully implemented if and only if they are found to be justifiable. Equations (6) and (7) ensure that each worker is assigned to a single cell and each cell receives the necessary number of workers. Equation (8) ensures that the time spent by the set of workers assigned to perform each skill j in each cell k is at least as large as the requirements for that cell. Equation (9) prevents the model from assigning tasks to any worker unless they have been assigned to that cell. If we choose to assign tasks in cell k to worker i, then we must set Y_{ik} to 1. Otherwise the expression forces the time allocated by that worker to any task in the cell to be 0. Equation (9) also limits the amount of work that can be assigned to any individual. Workers cannot be assigned more than full time duties. We must also ensure that workers are only assigned tasks for which they are trained. This is ensured by equation (10). Finally, equation (11) enforces the binary and non-negative restrictions on the variables.

Management can set the weights for the goals. One choice may be that all the weights are set equal to one if management does not differentiate the priority on goals of training cost, individual fitness to their tasks, and team synergy level. Sensitivity and other standard multiobjective decision making explorations can be performed on the weights. Note also that additional constraints, such as limits on the maximum training allowed for any worker , can be easily added to the model if desired.

4. Solution methods for worker assignment and training model

Even simplified versions of the WAT model that do not involve the X_{ijk} task assignment variables can be shown to be NP-hard by reduction of the set covering problem (see Huang 1999). As such, we investigate several specialized heuristics for the problem. We investigate a simple greedy heuristic and several versions of beam search. Simulated annealing is also considered for comparison. These methods provide a range of computational complexity and ability to specialize to the WAT model.

4.1. *Greedy heuristic*

We first propose a computationally efficient, single-pass greedy heuristic that iteratively assigns workers to cells. The technique is greedy in the sense that each assignment is made to optimize myopically the objective define in equation (1). After each assignment, the remaining unfilled requirements for the cell are updated. At each iteration, all available workers are permuted against the available cells and the matches are evaluated. The best match is determined after the total impact on the three goals of training cost, fit, and synergy is evaluated. The detailed task assignment and time allocation to each task for the worker in a possible match is determined by a subroutine within the iterations of the greedy heuristic.

The procedure to select the best match at iteration l is expressed as follows. To conform with our subsequent description of the beam search procedures, we will refer to iterations as levels. The assignment process from the top level to level $l-1$ gives a partial solution to the problem of WAT. Let θ_{l-1} be the set of workers assigned in levels 1 to $l-1$. At level l, each possible match of a worker $w \notin \theta_{l-1}$ with a cell c represents a possible partial solution extended to level l. We term the impact on the three goals from this match as $Z^{l,w,c}$, where

$$Z^{l,w,c} = w_{\text{train}} d^{l,w,c}_{\text{train}} + w_{\text{fit}} d^{l,w,c}_{\text{fit}} + \sum_k \sum_{m=1}^{3} (w^{+}_{mk} d^{l,w,c}_{mk^+} + w^{-}_{mk} d^{l,w,c}_{mk^-}). \qquad (12)$$

Consider the impact on team synergy first. We look at a match of worker w to cell c in level l after we have completed $l-1$ levels of assignment. Y_{ik}s representing the assignments of the worker i to cell k from top level to level l in the partial solution are known. If the number of workers assigned to cell k, $k = 1, \ldots, K$, is M_k, we can calculate $d^{l,w,c}_{mk^+}$ and $d^{l,w,c}_{mk^-}$ by solving the following equation set (equations (13) and (14)).

$$\sum_{i \in \{\theta_{l-1}, w\}} \sum_{t=1}^{4} a_{itm} Y_{ik} + d^{l,w,c}_{mk^+} - d^{l,w,c}_{mk^-} = M_k; \quad m = 1, 3; \forall k \qquad (13)$$

$$\sum_{i \in \{\theta_{l-1}, w\}} \sum_{t=1}^{4} a_{itm} Y_{ik} + d^{l,w,c}_{mk^+} - d^{l,w,c}_{mk^-} = 2M_k; \quad m = 2; \forall k. \qquad (14)$$

The value of the term $\sum_k \sum_{m=1}^{3} (w_{mk^+} d^{l,w,c}_{mk^+} + w_{mk^-} d^{l,w,c}_{mk^-})$ in (12), can be determined given $d^{l,w,c}_{mk^+}$ and $d^{l,w,c}_{mk^-}$.

Consider the impact on the goals of training cost and individual job fitness. The impact of the partial solution at level l with assignment of worker w to cell c on these two goals can be evaluated by calculating $Z_2^{l,w,c}$, where

$$
\begin{aligned}
Z_2^{l,w,c} &= w_{\text{train}} d_{\text{train}}^{l,w,c} + w_{\text{fit}} d_{\text{fit}}^{1,w,c} \\
&= w_{\text{train}} (d_{\text{train}}^{l-1} + \delta d_{\text{train}}^{l,w,c}) + w_{\text{fit}} (d_{\text{fit}}^{l-1} + \delta d_{\text{fit}}^{l,w,c}) \\
&= w_{\text{train}} d_{\text{train}}^{l-1} + w_{\text{fit}} d_{\text{fit}}^{l-1} + w_{\text{train}} \delta d_{\text{train}}^{l,w,c} + w_{\text{fit}} \delta d_{\text{fit}}^{l,w,c},
\end{aligned}
$$

$\delta d_{\text{train}}^{l,w,c}$ is the training cost incurred for the assignment of worker w to cell c at level l, and

$\delta d_{\text{fit}}^{l,w,c}$ is the individual job fitness score when worker w is assigned to cell c at level l.

Since we do not know the exact task assignment and time allocation to each assigned task for worker w in cell c at level l, the value of $\delta d_{\text{train}}^{l,w,c}$ and $\delta d_{\text{fit}}^{l,w,c}$ cannot be easily determined. A suitable task assignment and time allocation for worker w in cell c can reduce the impact of the assignment on the goals of training cost and individual job fitness. The determination of task assignment and time allocation for worker w at level l can be modelled as an integer programming model, termed M, if we look at the assignment of worker w to cell c at level l locally. The model M is as follows.

$$\text{Minimize } Z = w_{\text{train}} \delta d_{\text{train}}^{l,w,c} + w_{\text{fit}} \delta d_{\text{fit}}^{l,w,c} \tag{15}$$

Subject to:

$$i = w \tag{16}$$

$$k = c \tag{17}$$

$$\sum_i \sum_j c_{ij} Z_{ij} - \delta d_{\text{train}}^{l,w,c} = 0 \tag{18}$$

$$\sum_k \sum_i \sum_j f_{ij} X_{ijk} - \delta d_{\text{fit}}^{l,w,c} = 0 \tag{19}$$

$$\sum_i X_{ijk} \le S'_{jk}, \quad \text{for all } j \tag{20}$$

$$\sum_j X_{ijk} = 1 \tag{21}$$

$$\sum_k X_{ijk} \le Z_{ij}, \quad \text{for all } j \tag{22}$$

$$Y_{ik} \text{ 0 or 1}; \quad Z_{ij} \text{ 0 or 1}; \quad 0 \le X_{ijk} \le 1; \quad 0 \le d_{\cos\text{t}}; \quad 0 \le d_{mk^+}; \quad 0 \le d_{mk^-}. \tag{23}$$

Model M is intentionally written in a format similar to model WAT to save length of explanation, but the cell index k is confined to c and worker index i is confined to w. The meaning of each variable and data coefficient stays the same as those in model WAT, except S'_{jk} is the amount of skill k still needed in cell k after the assignments of $l-1$ levels while S_{jk} in model WAT is the amount of skill j needed in cell k. The time allocated to skill j should be less than or equal to the unfulfilled skill requirements, as is explained in equation (20). Equation (21) makes sure that the worker's time is fully utilized in the cell. We assume that every worker's time has to be fully allocated to problem WAT.

The local optimal task assignment for worker w in cell c can be determined by solving model M. However, we develop a greedy heuristic with polynomial time behaviour to assign tasks to worker w. The heuristic is computationally faster than using branch-and-bound to solve the model optimally.

We repeat the following step to assign task(s) to the worker w. We assign one task and determine the time the worker should spend on it at each step until the worker's time is consumed. The time required for the tasks is S_j', $j = 1, \ldots, J$. At each step, we calculate a selection factor R_j for each task j with $S_j' > 0$. We only need to consider the skill in the cell that still requires a worker.

$$R_j = (w_{\text{train}} c_{wj} + w_{\text{fit}} f_{ij} Q_j)/Q_j, \quad \text{for } j \text{ with } S_j' > 0$$
$$Q_j = \min(S_j', X),$$

where X represents the available time remaining for worker w. The time the worker w can perform skill j will not exceed Q_j, the minimum of the available time remaining for worker w, X, and the time required for skill j, S_j'. With W_j as the denominator, R_j is the average cost per time unit if we assign worker w to task j. If $k = \arg\min\{R_j : 1 \le j \le J \text{ and } S_j' > 0\}$, we assign worker w to skill k with time Q_k and update the available time of this worker $(X = X - Q_k)$ and time requirements for each skill in this cell $(S_k' = S_k' - Q_k)$. We repeat the above step until the worker's time limit or the time requirement of the cell is reached and obtain the heuristic solution. The feasibility of the task assignment heuristic solution is also guaranteed in this way. With the above greedy heuristic, we can obtain a solution for the local optimal task assignment problem and know the objective function value Z of this greedy solution.

Therefore, the value of $Z^{l,w,c}$, the total impact on the three goals by assigning worker w to cell c at level l, is determined after we plug the greedy heuristic's solution for the local optimal assignment problem only at level l in the expression.

$$\begin{aligned}
Z^{l,w,c} &= w_{\text{train}} d_{\text{train}}^{l,w,c} + w_{\text{fit}} d_{\text{fit}}^{l,w,c} + \sum_k \sum_{m=1}^{3} (w_{mk^+} d_{mk^+}^{l,w,c} + w_{mk^-} d_{mk^-}^{l,w,c}) \\
&= w_{\text{train}} d_{\text{train}}^{l-1} + w_{\text{fit}} d_{\text{fit}}^{l-1} + w_{\text{train}} \delta d_{\text{train}}^{l,w,c} + w_{\text{fit}} \delta d_{\text{fit}}^{l,w,c} \\
&\quad + \sum_k \sum_{m=1}^{3} (w_{mk^+} d_{mk^+}^{l,w,c} + w_{mk^-} d_{mk^-}^{l,w,c}),
\end{aligned}$$

The best match at level l is the match with the least value of $Z^{l,w,c}$. The worker in the best match is assigned to the cell in the best match at level l. Along with the match of a worker to a cell, the task assignment and time allocation for the tasks are also determined for this worker. After the assignment at one level is done, the assigned worker is taken out of the available worker list and the remaining skill and labour requirements for the cell are updated. The cell is taken out of the available cell list if its labour requirements are met. After all the workers are assigned, the formation of each cellular manufacturing team is fixed.

We call this greedy heuristic the hierarchical worker-cell greedy heuristic. It can be described in the following procedure.

Hierarchical worker-cell greedy heuristic for model WAT

Step 0. Initialize the available worker list $U = \{1, 2, \ldots, I)$ and the available cell list $C = \{1, 2, \ldots, K)$; Set level number $l = 1$;

Step 1. For all c, $c \in C$ and w, $w \in U$, calculate value of $Z^{l,w,c}$ in the ascending order of w and then c with the task assignment heuristic.

Step 2. Let $(w^*, c^*) = \arg\min\{Z^{l,w,c} : w \in U; c \in C\}$. Assign worker w^* to cell c^*. If there is a tie, the first match of w^* and c^* is selected;

Step 3. $S_{jc^*} = S_{jc^*} - a_{w^*j}$, for all j and $S_{jc^*} \geq 1$; a_{w^*j} = time spent on task j by worker w^*

Step 4. $U = U - \{w^*\}$;

Step 5. $S_{c^*} = S_{c^*} - 1$;

Step 6. If $S_{c^*} = 0$.
$C = C - \{c^*\}$;

Step 7. If $U = \phi$,
Stop;
Else
$l = l + 1$; Go back to step 1.

4.2. *Greedy-heuristic based beam search*

Because the greedy heuristic is myopic, a more sophisticated beam-search based heuristic is also developed for model WAT. The beam search employs the upper bound on the optimal solution obtained by using the hierarchical worker-cell greedy heuristic at a node as the evaluation function value. Initially we also tried using the LP relaxation as a lower bound for comparing nodes. The LP solutions were slower computationally and failed to provide significant improvement. As the beam search is heuristic, we choose to use the quicker upper bound to prioritize nodes. However, we may lose the ability to prove optimality in those cases in which an optimal solution is obtained.

Similar to the greedy heuristic, the beam search heuristic proceeds from level to level. It performs a breadth-first search with no backtracking. Each node in the search tree contains a combination of a worker and a cell, which means the worker is assigned to the cell. The information of task assignment and time allocation of tasks for this worker in the cell is also included in the node. The node also knows the node at the next higher level from which it emanated so that we can trace back to the top level. A path from a node at the top level to any node in the tree represents a series of selection and assignment processes, i.e. a partial or complete solution for problem WAT.

Assume beam width w. Only the w best nodes at every level are expanded or sprouted into further nodes at the next level. The beam search method implemented for problem WAT employs a filter as well as the restricted beam. All expanded nodes representing the possible assignments are first evaluated with a filter function, which calculates the value of $Z^{l,w,c}$, the local impact on the three goals at this level only. Only the filter width f child nodes with the best results are kept for a father node. For a partial solution represented by any of the child nodes not filtered out, the hierarchical worker-cell greedy heuristic developed in last section is used to complete the solution. In this way, a global estimate at a node on the impact of three goals is obtained. This is also the evaluation function value at this node and is used to judge the promise of the node. Only w nodes are finally kept for this level and are sprouted into the next level. The filter does not consume much computational time but enables us to filter out the nodes with poor performance before we have to compute a global estimate for each node likely worth keeping.

Given the greedy-heuristic based beam search described above, the following proposition can be concluded in this section.

Proposition 1: For problem WAT, let the objective function value of the solution by a greedy-heuristic based beam search be Z_b and the objective function value of the solution by greedy heuristic be Z_g. We have $Z_b \leq Z_g$.

Proof: At the very first level of the hierarchical worker-cell greedy-heuristic based beam search, all possible matches (nodes) of worker and cell are enumerated and the filter function is applied to each one of these matches. The node A with the lowest local impact or filter function value is guaranteed to pass through the filter. This holds for every successive stage as well. The evaluation function value at node A is actually the objective function value Z_g of the solution by the greedy heuristic if node A is at the top level. If node A is not a node at the top level, the evaluation function value at it is the same with that at its retained parent node. Since the best w nodes are kept at each level, the lowest evaluation function value at a level is always less than or equal to that at the level right above. Therefore, we have $Z_b \leq Z_g$. This completes the proof.

We write the procedure of the greedy-heuristic based beam search algorithm as follows.

Greedy-heuristic based beam search algorithm for model WAT

Step 1. Set beam width $= w$, filter width $= f$, level $l = 1$, Available worker set $U = \{1, 2, \ldots, I\}$, available cell set $C = \{1, 2, \ldots, K\}$, Let $R_l =$ the retained node set at level l and R_0 contains only one null node.

Step 2. Form the initial node set S for level l; $S = \{(n, w, c) | n \in R_{l-1}, w \in U_n, c \in C_n\}$. $U_n =$ available worker set from node n. $C_n =$ available cell set from node n.

Step 3. Trace back from each node in S to a retained node at the top level and obtain a partial solution P.

Step 4. Compute the objective function value Z_p for the partial solution with the task assignment greedy heuristic for each $n \in S$.

Step 5. For each retained node $n \in R_{l-1}$, keep the f child nodes with lowest Z_p.

Step 5. For the f kept child nodes from each retained node n at level $l - 1$, obtain the evaluation function value Z_e at these nodes with the hierarchical worker-cell greedy heuristic. Keep the best w nodes with the lowest Z_e as the retained nodes for level l.

Step 8. If $l = I$,
    Go to step 9;
  Else
    $l = l + 1$; go back to step 2.

Step 9. Start from the retained nodes at the bottom level, trace back to level 1 and get the best w solutions.

5. Experiment results and comparison of the heuristics

We use a full factorial model with crossed factors to test the performance of the heuristics developed for model WAT. The number of workers per cell, number of cells, number of skills required at each cell, and initial skill level of the workforce are factors in our experiments. In addition, the individual job fitness coefficient, f_{ij}, is assumed to be uniformly distributed between 0 and 1. For each trait, an individual's mode of operation, (MO) as represented in the a_{itm} coefficients, is generated with a probability of 20% for Initiating, 20% for Preventing, and 60% for Responding. This distribution stems from large sample population results tabulated by the Kolbe Corporation (Kolve 1993).

Ten replications are generated for each test configuration. Using CPLEX optimization software running on a Sun Sparc workstation, each problem instance was solved with an upper time limit set to 3600 seconds. Solutions were also found for the greedy heuristic, and four versions of filtered beam search. The average over the ten replications of objective value deviation against lower bound and solution time are used to evaluate the performance of each algorithm. The test results for each problem size or test configuration are ordered in the tables by the ascending problem size. The number of workers to be assigned is the first ranking factor followed by the number of cells, number of skills, and skill probability level in that order.

Table 1 shows the solution quality of CPLEX and our heuristics. Values are average percentage deviations above the optimal or lower bound. We use the lower bound obtained by CPLEX using branch and bound with the 3600 second time limit. Figure 1 illustrates these results. Table 2 and Figure 2 display average solution times.

First, we discuss the experiment results separately for the three groups of solution methods: (1) CPLEX optimization software, (2) greedy heuristic, and (3) filtered beam search. Then the heuristics are compared and suggestions on how to use these solution methods are provided.

5.1. *Experience on solving model WAT by CPLEX*

From figure 1, we can see that CPLEX solved the model to optimality for relatively small problem sizes in test configurations 1 to 11. Evel for larger problem sizes with 8 workers per cell, 8 cells, 8 skills required by each cell, and all skill probability levels (test configuration 19, 20 and 21), the optimal solution was obtained by CPLEX for all test samples. Test configurations 19, 20 and 21 possess a special structure. The number of workers and skills per cell were equal. It can be shown in this case that the workload assignment subproblem simplifies to a unimodular assignment problem in this case. With 64 workers and 8 workers per cell, CPLEX could quickly identify an optimal solution having perfect job fit and team synergy without any required training. Smaller problems with 2 workers and skills per cell were more difficult to solve since the total acquired skill pool was smaller and it was more difficult to achieve team synergy. CPLEX could not solve problems with 8 cells and more skills than workers to optimality within the set time limit. In these problems workers must split their time between tasks. The gap between the feasible solution found by CPLEX and the lower bound identified ranged from 25% to 88% in deviation from the lower bound.

In general, we can say that CPLEX performs well for small problem sizes or even big problem sizes with special coefficient values. But it fails to find the optimal solution or a solution close to optimality within a reasonable time for general large problems.

Test configuration	No. of cells	No. of wo/cell	No. of skills	Skill probability	Lower bound by Cplex	Cplex	Greedy	$B=1$, $F=1$	$B=1$, $F=2$	$B=2$, $F=1$	$B=2$, $F=2$
1	2	2	2	0.8	17.20	0.00	8.60	0.91	0.01	0.01	0.01
2	2	2	2	0.5	23.49	0.00	7.70	2.47	0.01	0.01	0.01
3	2	2	2	0.2	25.43	0.00	6.56	1.71	0.01	0.01	0.01
4	2	2	4	0.8	22.67	0.00	19.21	8.34	2.83	2.83	2.41
5	2	2	4	0.5	31.02	0.00	11.89	9.43	3.28	0.98	0.77
6	2	2	4	0.2	44.99	0.00	12.18	4.84	2.51	2.40	2.40
7	2	8	8	0.8	31.44	0.00	24.34	8.15	1.51	3.73	1.52
8	2	8	8	0.5	43.82	0.00	21.86	7.09	2.77	3.54	1.99
9	2	8	8	0.2	68.43	0.00	15.89	5.30	2.34	3.39	1.36
10	2	8	16	0.8	35.28	0.00	42.37	14.27	7.51	9.48	6.25
11	2	8	16	0.5	43.81	0.00	51.55	19.36	12.43	16.38	10.67
12	2	8	16	0.2	67.69	6.20	39.12	18.70	15.79	19.18	14.19
13	8	2	2	0.8	67.05	0.47	11.79	5.43	3.91	5.27	3.77
14	8	2	2	0.5	82.06	1.78	11.57	7.52	4.94	7.00	4.79
15	8	2	2	0.2	111.31	0.41	8.42	3.91	3.09	3.92	3.33
16	8	2	4	0.8	55.69	25.09	47.64	34.68	28.20	34.49	27.81
17	8	2	4	0.5	60.58	75.74	69.89	62.10	59.54	63.12	57.42
18	8	2	4	0.2	86.72	88.03	80.14	75.84	71.98	73.73	72.10
19	8	8	8	0.8	122.10	0.00	14.47	6.37	4.41	5.98	4.05
20	8	8	8	0.5	158.74	0.00	14.76	6.51	4.14	5.94	3.73
21	8	8	8	0.2	250.60	0.00	9.41	4.03	2.68	4.16	3.10
22	8	8	16	0.8	105.27	34.05	46.77	30.32	27.88	28.99	27.80
23	8	8	16	0.5	116.38	52.60	63.87	45.35	40.81	43.68	41.79
24	8	8	16	0.2	188.45	64.50	60.20	49.54	46.92	49.35	46.80
		Average			77.51	14.54	29.17	18.01	14.56	16.15	14.09

Beam Search Algorithm is tested with two levels of beam width ($B=1$ and $B=2$) and two levels of filter width ($F=1$ and $F=2$).
$B=1$: Beam width $= 1 \times$ number of workers per cell. $B=2$: Beam width $= 2 \times$ number of worker sper cell. $F=1$: Filter width $= 2$. $F=2$: Filter width $= 8$.

Table 1. Average percentage deviation over lower bound of objective value by each algorithm.

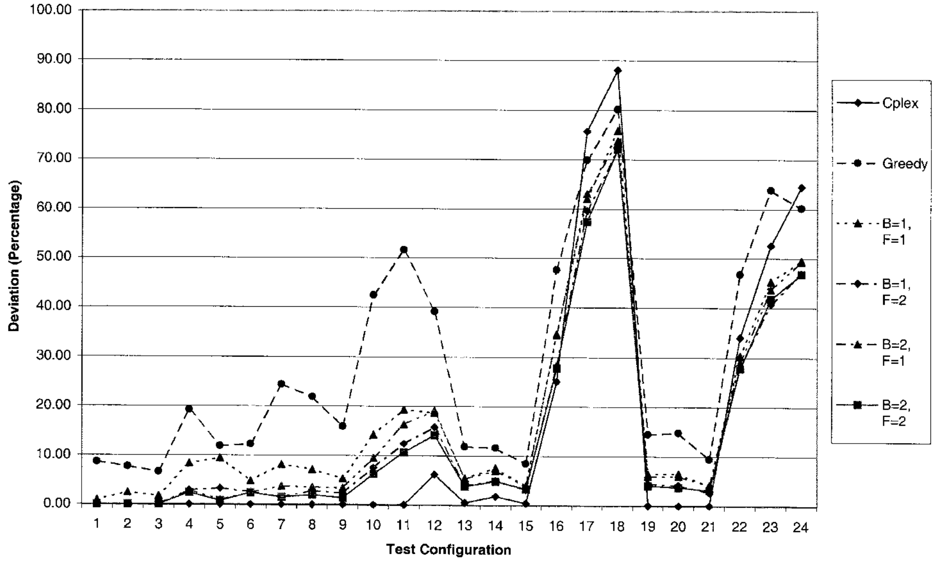

Figure 1. Performance of algorithms by deviation to the lower bound.

5.2. *Experience on solving model WAT by the greedy heuristic*

The greedy heuristic could find a feasible solution for the model in a fairly short computation time, as shown in table 2. The computational time was less than 4 seconds for all problems tested.

Table 1 shows that the greedy heuristic frequently performed poorly in identifying the optimal solution relative to the other methods. Optimal solutions were only found for some problems with 4 workers to be assigned into 2 cells. For relatively small problem sizes from test configurations 1 to 11, the solution found by the greedy heuristic deviated from the optimal solution obtained by CPLEX by 6.56% to 51.55% on average. For the problems with number of skills equal to number of cell workers, deviations from optimality never exceeded 15%. With multiple tasks per worker, deviations reached as large as 88% for CPLEX. Interestingly, for many of the difficult problems, such as cases 17, 18 and 24, the greedy heuristic found better solutions than CPLEX despite one hour of computational time for CPLEX and less than 4 seconds for the greedy heuristic.

5.3. *Experience on solving model WAT by filtered beam search*

The filtered beam search algorithms significantly outperform the greedy heuristic on the same test samples. When beam width and filter width both are at their high level (filter width = 8 and beam width = twice the number of workers per cell), we see that the largest deviation by the beam search algorithm from the optimal solution was 2.41% for the test configurations 1 to 9. The longest average solution time was 18 seconds. For larger problem sizes of 8 workers per cell, 2 cells, 2 skills required in each cell, and 8 workers per cell, 8 cells, 8 skills required in each cell, the test results from the beam search algorithm were still less than 5% away from the lower bound identified by CPLEX. However, solution times matched CPLEX for test configuration 21 with 8 workers per cell, 8 cells, and 8 skills required in each cell. For the hardest problems with 8 cells and twice as many skills as workers/cell, the filtered

Test configuration	No. of cells	No. of wo/cell	No. of skills	Skill probability	Cplex	Greedy	$B=1$, $F=1$	$B=1$, $F=2$	$B=2$, $F=1$	$B=2$, $F=2$
1	2	2	2	0.8	0.2	0	0	0	0	0.2
2	2	2	2	0.5	0.3	0	0	0	0.1	0
3	2	2	2	0.2	0	0	0	0.1	0	0
4	2	2	4	0.8	0.4	0	0	0.1	0.1	0
5	2	2	4	0.5	0.6	0	0	0	0.1	0
6	2	2	4	0.2	2.5	0	0.1	0	0.1	0
7	2	8	8	0.8	1	0	2.9	8.9	6.1	17
8	2	8	8	0.5	1.6	0.1	3.2	9.2	6.1	18
9	2	8	8	0.2	1.7	0.1	3	9.2	5.9	17.8
10	2	8	16	0.8	29	0	5.4	18.1	11.3	33.8
11	2	8	16	0.5	458	0	5.6	18.1	11.6	34.4
12	2	8	16	0.2	3672.1	0	5.7	18.1	11.5	34.2
13	8	2	2	0.8	1200.9	0.1	1.6	4.2	2.8	8.7
14	8	2	2	0.5	2632	0	1.6	4.4	2.9	8.9
15	8	2	2	0.2	780.6	0.1	1.5	4.5	3	9
16	8	2	4	0.8	3549.7	0.1	2.3	6.8	4.4	13.8
17	8	2	4	0.5	3623	0.1	2.6	6.7	4.8	13.8
18	8	2	4	0.2	3625.9	0.1	2.5	7	4.7	14.4
19	8	8	8	0.8	34.6	1.5	522.5	1854.7	1060.8	3767.3
20	8	8	8	0.5	38.6	1.7	533.9	1913.8	1088.6	3853.5
21	8	8	8	0.2	55.2	1.4	541.4	1928.9	1077.1	3873.7
22	8	8	16	0.8	3433.8	3.8	1249.3	4442.3	2439.8	8883.5
23	8	8	16	0.5	3632.1	3.5	1215.4	4465	2438.1	8860.6
24	8	8	16	0.2	3642.1	3.7	1227.8	4412.4	2427	8861.9
		Average			1267.3	0.7	222.0	797.2	442.0	1596.9

Beam Search Algorithm is tested with two levels of beam width ($B=1$ and $B=2$) and two levels of filter width ($F=1$ and $F=2$).
$B=1$: Beam width $= 1 \times$ number of workers per cell. $B=2$: Beam width $= 2 \times$ number of worker sper cell. $F=1$: Filter width $= 2$. $F=2$: Filter width $= 8$.

Table 2. Solution time by each algorithm for Model P2 (in seconds).

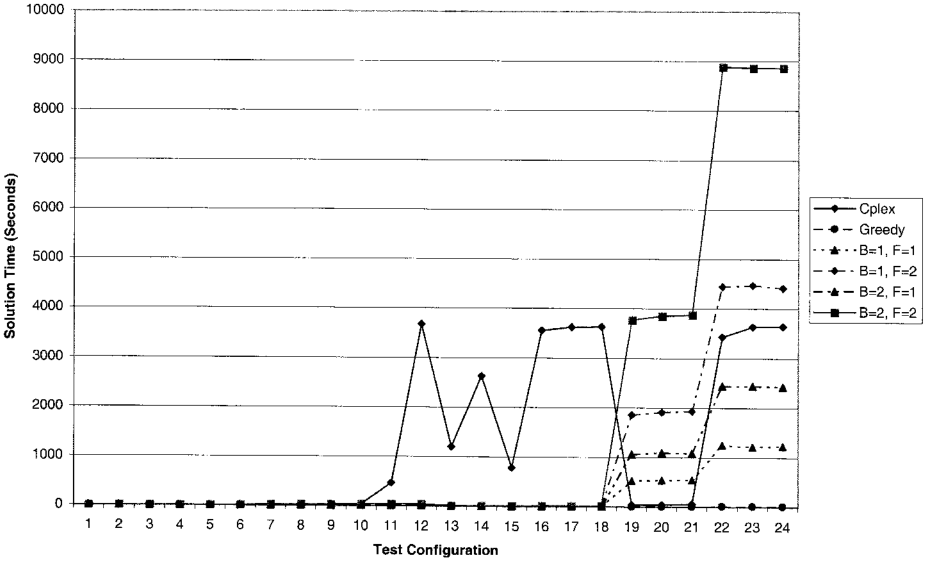

Figure 2. Solution time of each algorithm.

beam search at the high width values outperformed CPLEX in almost all cases. Over the entire experiment, the filtered beam search yielded values averaging 14.1% above the lower bound as compared with 14.5% for CPLEX. For problems with 64 workers and 16 skills in each cell, the deviations from the lower bound CPLEX achieved were 34.05%, 52.60% and 64.50% as compared with 27.80%, 41.79% and 46.80% for the filtered beam search at the three skill probability levels. CPLEX was stopped at the time limit of 3600 seconds for these test configurations. Further tests on a couple of test samples showed that the CPLEX would run out of computer memory when computational time got to about 7000 seconds and there was little gain in the lower bound and solution quality. The average solution time to complete the beam search with beam and filter width at level 2 was 8860.6 seconds to 8883.5 seconds. When we relaxed the beam width and filter width both to level 1, the solution time beam search algorithm required was about 1200 seconds. The deviations by beam search with low beam and filter widths were 30.32%, 45.35% and 49.54%.

Solution times increased almost linearly for versions of filtered beam search as the filter and beam widths were widened. Solution quality increased modestly. The overall average deviation was 18.0%, 14.6%, 16.2% and 14.1% for the four beam and filter width combinations tested. Corresponding average solution times were 222, 979, 442 and 1597 seconds. By examining the detailed solution output file, we found that the higher level of beam width obtained a better upper bound on the objective value than the lower beam width at the first few levels of beam search. Further exploration provided only minimal improvement. This suggests that a varying width may be advisable.

5.4. *Additional testing with simulated annealing*

Even though the beam search achieved a smaller deviation from the lower bound than CPLEX over the large instances in the test configurations 22, 23 and 24, the deviation by beam search was still significant. The high deviation may be due to the poor quality of the lower bound obtained by CPLEX within the time limit and computer memory. To investigate this issue, we attempted to solve several harder problems by simulated annealing using an inhomogeneous cooling schedule, which reduces temperature after each iteration. Simulated annealing has proven to work well on many combinatorial optimization problems and can be configured to guarantee convergence to the optimal solution (Kirkpatrick *et al.* 1983, Cheh *et al.* 1991). The notation we use for applying the simulated annealing is as follows.

- K number of cells;
- I number of workers per cell;
- J number of skills required in each cell;
- T_n temperature at each move. $n = 0, 1, 2, \ldots, N$;
- s_0 initial solution obtained by the greedy heuristic. It can be defined by a $K \times I$ matrix Y and $I \times J$ matrices $\boldsymbol{X}_k, k = 1, 2, \ldots, K$. Y_{ki} represents the binary identifier indicating if worker i is assigned to cell k and X_{ij} in $\boldsymbol{X}_k$ represents the amount of time worker number i performs skill j in cell k;
- s Current solution;
- s' new neighbour solution randomly generated from s;
- s^* current best solution;
- $f(s)$ objective function of s.

To generate a new neighbour solution from the current solution, we first randomly choose two cells $k1$ and $k2$. Then we generate two random numbers $w1$ and $w2$ in the range of 1 and number of workers per cell I, and switch worker $w1$ in cell $k1$ and worker $w2$ in cell $k2$. However, the values of X_{ij}s in the new $\boldsymbol{X}_{k1}$ and $\boldsymbol{X}_{k2}$ still need to be determined after we switch the two workers. Since we already know the workers assigned to cell $k1$ or $k2$, the determination of task assignment and time allocation can be formulated as an integer programming model, termed Q, as follows.

Decision variables

$$Z_{ij} = \begin{cases} 1 & \text{if worker } i \text{ acquires skill } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{ij} = \text{proportion of time worker } i \text{ performs skill } j.$$

Data coefficients

- c_{ij} is the cost to fully train worker i in skill j.
- f_{ij} is the fitness score of worker i to skill j; $0 \le f_{ij} \le 1$; the closer f_{ij} is to 0, the fitter the worker i is for skill j.
- S_j amount of skill j needed.

The model then becomes

$$Q: \text{Minimize } Z = \sum_i \sum_j c_{ij} Z_{ij} + \sum_i \sum_j f_{ij} X_{ij} \tag{24}$$

Subject to

$$\sum_i X_{ij} \ge S_j, \quad \text{for all } j \tag{25}$$

$$\sum_j X_{ij} \leq 1, \quad \text{for all } i \tag{26}$$

$$X_{ij} \leq Z_{ij}, \quad \text{for all } i \text{ and } j \tag{27}$$

$$Z_{ij} \ 0 \text{ or } 1; \quad 0 \leq X_{ij} \leq 1. \tag{28}$$

The objective function (24) computes the cost of training and the total amount of individual fitness. since the Kolbe team synergy levels are deterministic given that the workers assigned to the cell are known, the team synergy objective is ignored. Equation (25) ensures that the time spent by the set of workers assigned to perform each skill j is at least as large as the requirements for the cell. Equation (26) limits the amount of work that can be assigned to any individual. Workers cannot be assigned with more than full time duties. Equation (27) ensures that workers are only assigned tasks for which they are trained. Equation (28) enforces the binary and non-negative restrictions on the variables.

We then develop a greedy heuristic for model Q. It is very similar to the hierarchical worker-cell greedy heuristic in section 4.1 for model WAT. The greedy heuristic starts from the first level and ends at the level of number of workers I. It determines the time allocation for one worker at each level of assignment. At each level, the heuristic calculates the total of training cost and individual fitness, $Z^{l,w}$ for each worker available and allocates the time for the worker with the least amount of training and individual fitness cost. For calculating the value of $Z^{l,w}$ for a worker, we also apply the concept of average cost per time unit, R_j used in the greedy heuristic for model WAT. Against a worker, we keep assigning his or her time to the task with the lowest average cost per time unit until the individual's time is used up. Now that the values of X_{ij}s in the new X_{k1} and X_{k2} are determined, the training cost, total of individual fitness can be simply computed for cells $k1$ and $k2$. We keep the time allocation and task assignment in the cells other than $k1$ and $k2$ the same so that the objective function values for these cells stay the same. The value of function $f(s')$ is the summation of the objective function value for all the cells.

After the generation and evaluation of a near neighbour solution as discussed above, our simulated annealing algorithm is described in the following procedure.

Simulated annealing procedure model WAT

Step 0. Obtain the initial solution S_0 for a problem WAT instance with the greedy heuristic. Set $s = s^* = s_0; n = 0$.
Step 1. Generate a new neighbour solution s' from current solution s.
Step 2. If $f(s') \leqslant f(s)$, accept s' and replace s with s'; if $f(s') < f(s^*)$, replace the current best solution with $\mathbf{s}'$.
Step 3. If $f(s') > f(s)$, accept s' and replace s with s' with probability $\mathrm{EXP}((f(s) - f(s'))/T_n)$.
Step 4. $T_{n+1} = \alpha T_n$.
Step 5. If $n = N$,
    Stop.
  Else
    Go back to Step 1.

We applied this simulated annealing algorithm to obtain the solution for the 10 instances at the test configuration 23 with 8 cells, 8 workers per cell, 16 skills required

Replication	Lower bound by CPLEX	CPLEX	Greed, y	$B = 1$, $F = 1$	$B = 1$, $F = 2$	$B = 2$, $F = 1$	$B = 2$, $F = 2$	Simulated annealing
1	119.2	49.14	60.01	44.76	40.78	43.26	40.78	45.21
2	127.3	41.52	59.13	42.04	40.10	42.04	40.10	39.25
3	101.7	63.08	76.22	52.80	52.63	57.27	57.10	48.27
4	113.5	57.20	62.16	43.98	35.28	43.98	42.72	37.78
5	117.5	58.32	83.34	54.38	51.23	54.38	50.25	46.02
6	103.9	62.04	75.12	57.32	46.75	48.39	46.75	43.99
7	101.3	56.44	53.97	41.01	36.15	36.54	36.15	38.84
8	122.4	44.55	68.82	48.47	42.21	46.40	39.42	42.97
9	133.9	43.52	44.02	29.89	28.74	28.90	28.74	33.95
10	123.1	50.24	55.87	38.82	34.20	35.65	35.94	40.77
Average	116.4	52.6	63.9	45.4	40.8	43.7	41.8	41.7

$B = 1$: bam width = No. of workers per cell.
$B = 2$: Beam width = $2 \times$ No. of workers per cell.
$F = 1$: Filter width = 2.
$F = 2$: Filter width = 8.

Table 3. Percentage deviation above lower bound for solution methods.

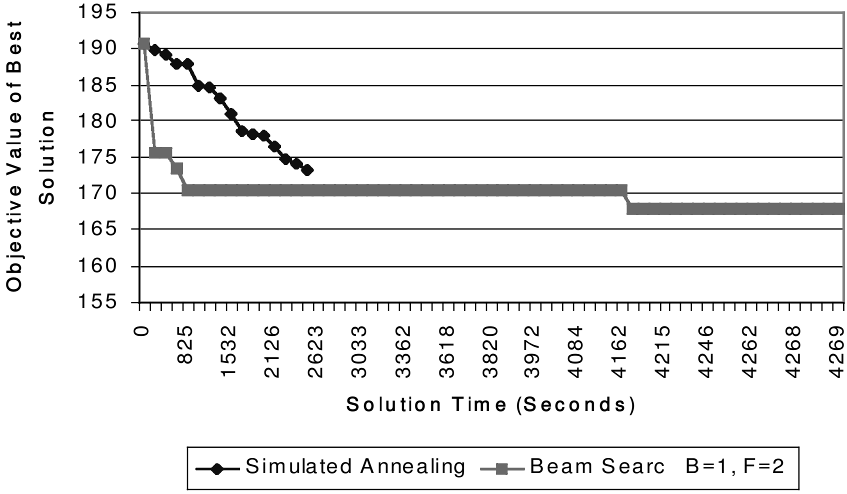

Figure 3. Best solution improvement over solution time.

per cell, and skill probability 0.5 in the experiments of section 5. the initial acceptance rate was set to around 60% and stopping temperature was set to have the acceptance rate less than 1%. The solution time is set to about one hour, which was in the range of the solution times by the beam search algorithm.

Table 3 shows the deviation from the lower bound achieved by the simulated annealing and the performance of the filtered beam search. The average solution time for the ten replications by simulated annealing was 2832 seconds on the same Sun workstation. We can see that the simulated annealing performed better than the greedy heuristic and was similar overall to the filtered beam search. Figure 3 further shows how fast the beam search and simulated annealing solutions improved during the search for a typical replication. We can see that the filtered beam search achieved a lower objective function value faster and thus may be a better choice when computational time is limited.

6. Conclusions

We formulated a detailed worker assignment and training (WAT) model for formation of worker teams in cellular manufacturing. The procedures allow an existing functional organization to be converted to cells so as nearly to maximize team synergy and the fit between worker abilities/instincts and task requirements while minimizing training cost. The model supports the design of manufacturing cells to meet the operational requirements created by external demand, while establishing a cohesive and cooperative working environment, and individual job satisfaction. The model output locates workers into cells and details the task assignments for each worker.

Owing to the computational complexity of model WAT, solving the model optimally is not feasible for large problems. We developed a greedy heuristic and greedy-heuristic based filtered beam search algorithm. The algorithms were tested over a range of parameter combinations. Additional tests with simulated annealing were performed for certain difficult cases. The results indicate that the filtered beam search may be an effective solution approach, preferable to straightforward optimization or possibly simulated annealing when computational time and memory are limited.

This research provides a usable methodology for manufacturing managers to allocate workers and tasks into cells and for human resource managers to improve their return on training investment. Future research plans include determining additional factors beyond synergy that can reliably predict team performance and inclusion of these in the model.

Acknowledgements

This article is based upon work supported by the National Science Foundation under Grant No. DMI 9900057.

References

Askin, R. G. and Estrada, S., 1999, A survey of cellular manufacturing practices. *Handbook of Cellular Manufacturing Systems*, edited by S. Irani (New York: Wiley).

Askin, R. G. and Huang, Y., 1997, Employee training and assignment for facility reconfiguration. *Proceedings of the IERC*, pp. 426–431.

Askin, R. G. and Vakharia, A., 1990, Group technology planning and operation. *The automated Factory Handbook*, edited by D. Cleland and B. Bopaya (Blue Ridge Summit, PA: TAB Books), pp. 317–366 .

Barrick, M. R. and Mount, M. K., 1991, The big five personality dimensions and job performance: a meta-analysis. *Personnel Psychology*, **44**(1), pp. 1–26.

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

Cheh, K. M., Goldberg, J. B. and Askin, R. A., 1991, A note on the effect of neighborhood structure in simulated annealing. *Computers and Operations Research*, **18**(6), 537–547.

Ebeling, A. C. and Lee, C.-Y., 1994, Cross training effectiveness and profitability. *International Journal of Production Research*, **32**(12), 2843–2859.

Hoffman, E., 1984, An internal dimensional analysis of the least preferred co-worker measure. *The Journal of social Psychology*, **123**, 35–42.

Huang, Y., 1999, *Employee training and assignment for team_based manufacturing systems*, PhD dissertation, University of Arizona.

Kembel, J., 1996, *ROLE Play – Personalities in Action* (Life Change Partners).

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, Optimization by simulated annealing. *Science*, **220**, 671–680.

King, J. R. and Nakornchai, V., 1982, Machine component group formation in group technology – review and extension. *International Journal of Production Research*, **20**, 117–133.

Kolbe, K., 1994, *Pure Instinct* (New York: Random House).

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

Lyman, D. and Richter, K., 1995, QFD and personality type: the key to team energy and effectiveness. *Industrial Engineering*, **27**(2), 57–61.

McCaulley, M. H., 1990, The Myers–Briggs type indicator: a measure for individuals and groups. *Measurement and Evaluation in Counseling and Development*, **33**(4), 181–195.

Min, H. and Shin, D., 1993, Simultaneous formation of machine and human cells in group technology: a multiple objective approach. *International Journal of production Research*, **31**(10), 2307–2318.

Schriesheim, C., Tepper, B. J., Tetrault, L. A., 1994, Least preferred co-worker score, situational control, and leadership effectiveness: a meta-analysis of contingency model performance predictions. *Journal of Applied Psychology*, **79**(4), 561–573.

Singh, N., 1993, Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, **69**(3), 284–291.

Stevens, M. J. and Campion, M. A., 1994, The knowledge, skill, and ability requirements for teamwork: implications for human resource management. *Journal of Management*, **20**(2), 503–530.

Suer, G. A., 1996, Optimal operator assignment and cell loading in labor-intensive manufacturing cells. *Computers and Industrial Engineering*, **26**(4), 155–158.

Suresh, N. C., 1991, Partitioning work centers for group technology: insights from an analytical model. *Decision Sciences*, **22**(4), 772–791.

Suresh, N. C., 1992, Partitioning work centers for group technology: analytical extension and shop-level simulation investigation. *Decision Sciences*, **23**(2), 267–290.

Warner, R. C., Needy, K. L. and Bidanda, B., 1997, Worker assignment in implementing manufacturing cells. *Proceedings of the IERC*, pp. 240–345.

Wemmerlov, U. and Hyer, N. L., 1989, Cellular manufacturing practices. *Manufacturing Engineering*, 79–82.