FORMING EFFECTIVE TEAMS IN A WORKPLACE ENVIRONMENT

by

Erin Lillian Fitzpatrick

A Thesis Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS & INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN INDUSTRIAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2 0 0 0

UMI Number: 1403181

UMI®

UMI Microform 1403181

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment of the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: ____________________

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

____________________ 12/4/00

Ronald G. Askin
Professor of Engineering

Date

ACKNOWLEDGEMENTS

I would sincerely like to thank those who have made my foray into the world of graduate studies possible. I certainly would not have had this opportunity or enjoyed it as much if it were not for these people. First, I would like to recognize the Natural Science and Engineering Research Council of Canada (NSERC) for providing me with a PGSA scholarship that was tenable abroad. I would also like to extend my deepest thanks to Dr. Ronald Askin, who welcomed me into this program, acted as my advisor and who has been a true mentor to me. The knowledge he has provided me with extends beyond what can be found in any textbook. As well, I would like to thank the faculty and staff of the Department of Systems and Industrial Engineering for always being friends and for being quick with a smile. Special thanks to Dr. J. Pemberton Cyrus who was the first person, and most persistent, to suggest graduate school as an option. I extend a warm thank you to all my family and friends who supported me and who were a great source of encouragement throughout it all. Finally, I would like to thank my fiancé, Ray, for always believing in me and for helping me believe in myself.

DEDICATION

look ma… no hands!!!

TABLE OF CONTENTS

TABLE OF CONTENTS - Continued

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Throughout much of the past century, manufacturing efficiencies were gained by constructing systems from independently designed and optimized tasks. Recent theories and practice have extolled the virtues of team-based practices that rely on human flexibility and empowerment to improve integrated system performance. The formation of teams requires consideration of innate tendencies and interpersonal skills as well as technical skills. In this project we develop and test mathematical models for formation of effective human teams. Team membership is selected to ensure sufficient breadth and depth of technical skills. In addition, measures of worker conative tendencies are used along with empirical results on desirable team mix to form maximally effective teams.

1 INTRODUCTION

1.1 Background

Facing intensified competition in a growing global market, manufacturing companies are reengineering their integrated production systems to achieve lean manufacturing (Askin and Huang [2000]). In recent years there appears to be a trend showing increasing popularity of cellular manufacturing, CM, and other team-related approaches in the workplace (Bailey [1997], Hut and Molleman [1998]) to achieve this goal. Substantial research has been performed to improve the grouping of machines and parts into cells as a result of this trend towards CM (see Burbidge [1975], King and Nakornchai [1982], Kusiak [1987], Askin and Vakharia [1990], Suresh [1991, 1992], and Singh [1993] for general reviews). However, until recently the human element in this area has been mostly ignored. Minimal research has been conducted regarding the selection of team members and subsequent training requirements. As is typical in the field of engineering, it is failure that has led to change. Manufacturing cells that have been formed solely on machine-part interaction have frequently shown limited benefits (see Carr, Groves [1998] for a list of examples). This failure has led researchers to search for other factors that impact the performance of the work cell, culminating in an increasing interest in the effects of personal skills and traits in the performance of teams.

The elements of effective team formation are not limited to personal skills and traits. Burbidge [1975] listed a set of dedicated workers as a key principle of cell autonomy (or independence) that in turn is an essential aspect of successful cells in practice. In a survey of industry, Askin and Estrada [1998] 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. Cell productivity depends not only on the technical and administrative skills the workers possess but also the effective interaction among team members. This interaction and the related personality aspects are difficult to include in the aforementioned models due to the problems associated with quantifying their measures. Many systems exist that attempt to do so and we will evaluate the potential of several of these to be measured quantitatively as well as their demonstrated impact on productivity.

1.2 Problem Statement

Based on this need for effective interaction among team members, the purpose of this thesis is outlined as follows. Given an existing labor pool, it is desired to extract (a) one team or (b) multiple teams. It would be necessary to form a single team in a case such as creating a new manufacturing cell, undertaking a design project, creating a management or quality team, etc. Multiple team formation would be required if we were to shift from a non-cellular manufacturing environment to a cellular manufacturing environment. In this case we would need to determine which skilled individuals to place together in which cell. Multiple teams may be composed of the entire labor pool or just some of the labor pool. Skill requirements for the team may be identical or not. For example, if an entire segment of an organization were shifting to cellular manufacturing, the entire labor pool would need to be redistributed. However, if only a portion of the organization was being formed into a small number of teams, the entire labor pool would be considered but only a portion of it allocated. Depending on the nature of the work, cells could have the same makeup or vary from cell to cell.

We assume the labor pool itself is segregated into skill categories. Each member of the labor pool is assigned to one and only one skill category. The categories are defined according to the jobs or roles that need to be fulfilled on the team(s). For example, a team may require a milling machine operator, a turning machine operator, an inspector and an assembler. Each of these would become a skill category and any individuals belonging to the labor pool would have to be classified according to one of these skills.

There are several assumptions made for this problem scenario. It is first assumed those skill categories and team skill requirements have been clearly defined to the satisfaction of management. This should be developed with due care as lack of appropriate skills will prevent a team from completing its job. The assignment of individuals to skill groups will have been performed as well. The case where an individual possesses more that one skill and so may choose which skill to fulfill is not considered here. It is also assumed that there exist sufficient individuals in the various skill categories to meet the requirements for the teams. That is, all team skill requirements will be met. It is expected that any deficiencies in requirement availability have been removed through training and/or hiring as appropriate. Several papers (Ebeling and Lee [1994], Suer [1996], Min and Shin [1993], Askin and Huang [1997]) exist that have formulations for solving the training and related aspects of this problem. It is further assumed that all members of a skill category possess equal skill. While this may not be a realistic assumption in many situations, it is satisfactory for our purposes. Again, an approximation to this may be achieved through suitable training.

Finally, it is assumed that we have knowledge of the interpersonal mix required within a cell to promote effective team interaction. In order to utilize this knowledge, we assume that we have personality profiles of all potential team members being considered with which to measure the interpersonal mix against desired levels. The source of these tools for measuring effective team interaction is described in Section 3. This interpersonal mix will be the determining factor in deciding the construction of the team.

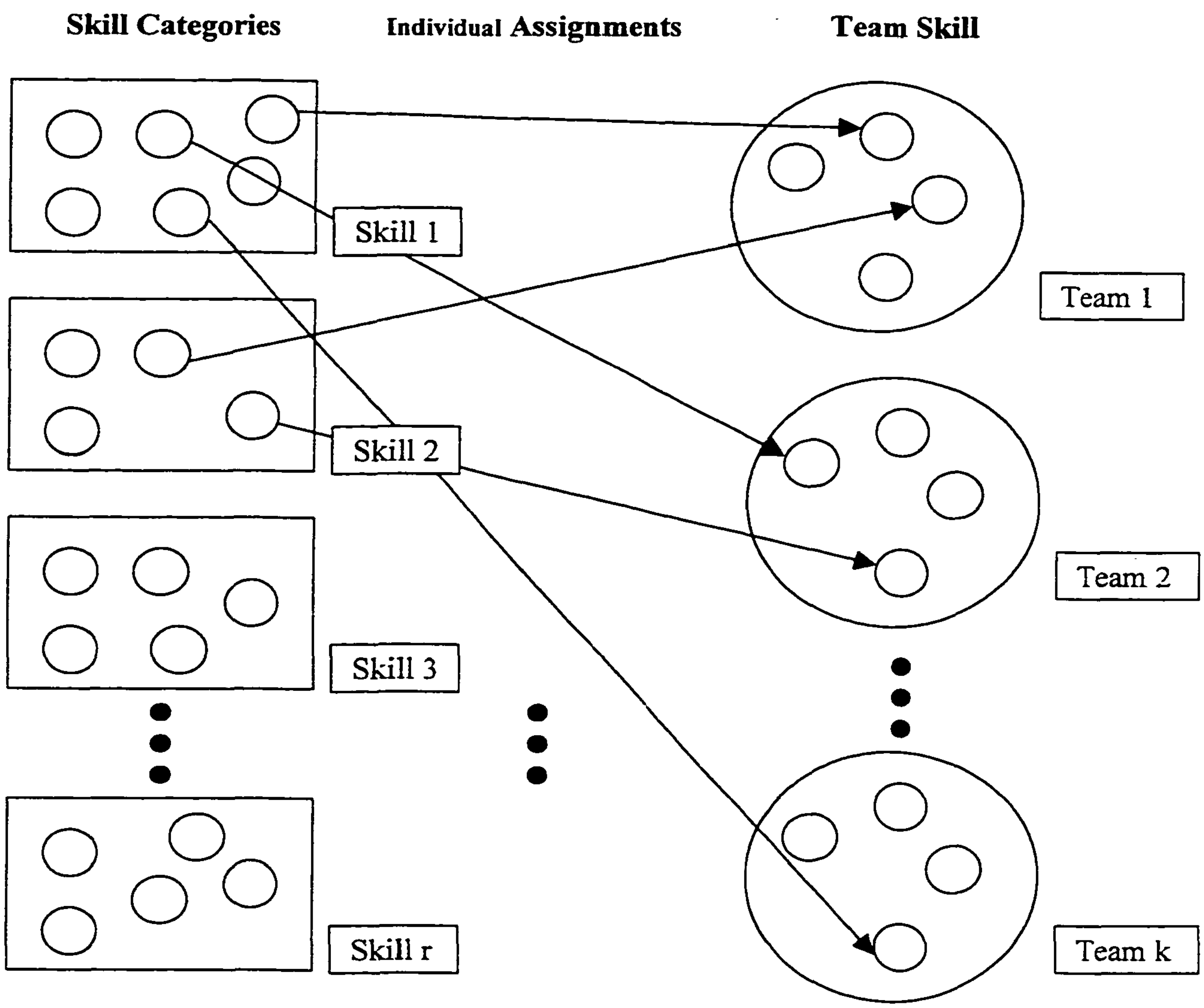


Figure 1 - Illustration of Problem Statement

Figure 1 illustrates the multiple team, partial labor pool, varying construction problem scenario. All individuals are classified into appropriate skill categories. The problem is then to assign individuals to the disparate teams in such a way that each team has a 'good' interpersonal construction. 'Good' interpersonal construction will be defined later in Section 3.

2 LITERATURE REVIEW

2.1 Teams, Skills, Training

It is stated in this paper that several factors including interpersonal relationships are important in forming teams. For evidence of this see Warner, Needy, and Bidanda [1996]. Here, relevant factors in assigning workers to cellular teams from both technological and human interaction perspectives are discussed. In addition, the problem statement made several assumptions. Mainly it was assumed that skill requirements were identified and necessary training had been determined and carried out. Several papers show these factors in the absence of interpersonal relations. In Suer [1996] mixed integer and integer programming models to achieve optimal product and worker assignment to labor-intensive manufacturing cells are developed. However, Min and Shin [1993] developed a multi-objective model to form cells with consideration of both machines and workers. Training was not considered in these models. The following papers give insight into how this has been addressed. Ebeling and Lee [1994] analyzed the cost and benefit of employee cross-training process 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. 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.

2.2 Psychology

In addition to the personal skills and training requirements addressed in the previous section, we propose a comprehensive evaluation of personality factors is necessary for an effective picture of team performance. D. Wechsler [1997], as president of the Division of Clinical and Abnormal Psychology, 'holds that intellectual ability [including skills] cannot be measured independently of drive, temperament, or emotion', hereafter referred to as personality traits. There are an increasing variety of suggested methodologies for the definition and use of these traits in the formation of successful teams. They all address the same basic questions: can we improve the performance of teams through the inclusion of such traits and if so, how?

Although psychology literature does not provide a universal consensus on personality elements, we find that most commonly, as D. Wechsler [1997] puts forth, there are three main classifications of these elements: cognitive, conative, and temperament. Cognitive refers to intellectual ability. This is represented in the individual's ability to learn the skills required, thus we do not consider this here. Conative refers to a person's will, drive or instinct; that approach which they will feel drawn to take in any given situation. Temperament delves into the emotional state and behavior of the individual.

Several sources demonstrate the validity of various types of personality measurement in models of team performance (Barrick and Mount [1991], Bailey [1999], Henry and Stevens [1998], Lingard and Berry [2000]). Juran, et. al. [1997] proposes that any variables representing psychological or behavioral characteristics must meet certain criteria before they are used to model a system:

(1) Variables must be stable enough over time so as to justify efforts to plan production systems around them.

(2) Variables must be easily and reliably measured.

(3) Variables must be demonstrably linked to the productivity and/or quality of the production system.

These criteria are used in support of the personality dimension measurement system commonly referred to as 'The Big Five'. It focuses on the temperament element of personality. The variables used in this metric are Neuroticism, Extraversion, Openness, Agreeableness and Conscientiousness. The study also included additional demographic factors. The results of the experiment demonstrated that most of the metrics could be used to explain a large proportion of the variability in productivity. Another frequently employed metric is the social-style implement, Myers-Briggs Type Indicator (MBTI). Hammer and Huszczo [1996] review its use in improving and predicting team performance. Major results in this regard appear limited to a negative correlation between 'similarity' according to the MBTI and performance. R. M. Belbin [1981] evaluates the formation of teams in a management setting using eight roles of team members: company worker, chairman, shaper, plant, resource investigator, monitor-evaluator, team worker, and completer-finisher. Studies and anecdotal evidence support this method of team formation (Henry and Stevens [1998], Belbin, Aston and Mottram [1976], Belbin, Watson and West [1997]). However, Henry and Stevens [1998] comment on the need for subjective expertise for the analysis to be effective and Furnham, Steele and Pendleton [1993] question the validity of its psychometric value.

The Kolbe Conative Index (KCI) gives a psychometric system that can be used to form successful teams. This system is presented in Kolbe [1989, 1993]. The index measures conation, which refers to the part of the mind that controls conscious effort and strives to carry out volitional acts. In other words, it measures the way an individual instinctively approaches problem solving, arranges ideas or objects, and uses time and energy. Due to the heavy use of this measure in this research, we explain it more fully in the next section.

3 KOLBE CONCEPT® MEASURES

Due to many of the positive attributes of the Kolbe Concept® outlined below it was decided this measurement tool warranted further consideration. Investigation was initiated to determine the suitability of these measures as a factor in the formation of work teams.

There exists several years of data and analysis showing empirical proof of this system's effectiveness in predicting an individual's 'fit' into various roles. While this method's application to the prediction of team dynamics has been shorter in duration, the empirical results look promising. Several statistical studies have been carried out on the results of Kolbe Concept® analysis. These consist of both contrived experiments and historical analysis of data. Independent experiments carried out at academic institutions show a positive correlation between Kolbe team performance predictions and actual team performance. These experiments were performed in the Business School at the University of Chicago on teams of MBA students. The study concluded that the Kolbe Concept® was accurate in predicting the performance of teams in this environment.

Analysis of historical results show a positive impact was achieved through the use of selecting both individuals for specific roles and for forming teams in the workplace. This evidence comes from industry clients of Kolbe Corp. Analysis of historical data, specifically the testing and retesting of individuals, showed the consistent nature of Kolbe measures. When retested at a later date a significantly high percentage of individuals stay within ten percent of their original scores [Kolbe, 1989]. This constancy is an asset for any measurement tool used to form permanent work teams.

Finally, a claimed attribute of the Kolbe Concept® that carries great importance when considering people for placement in employment or for a position within an organization is its unbiased nature. Unbiased refers to the measures being blind to personal factors not related to performance. These factors include gender, race, and socio-economic background amongst others. When statistical analysis is carried out on the very large pool of individuals who have completed a Kolbe Index measurement tool, there is no correlation with a specific score to any of these factors [Kolbe, 1989]. Thus we cautiously state that based on analysis of historical data the measures are independent of these factors.

3.1 Individual Kolbe Measures - The Kolbe Conative Index

The Kolbe Conative Index (KCI) measures an individual's instinctive behavior or drive. Instinct refers to an innate, action-oriented, sub-conscious drive to approach and perform tasks in a specific manner. It is not a learned behavior but an inherent part of who we are. The KCI defined instincts are neither 'good' nor 'bad' instincts, merely a means to classify and understand the reasons we do things the way we do or react to things in certain ways. With this understanding we can pick and choose the way we approach problems with the power of knowledge on our side.

The Kolbe Concept® classifies instinctive behaviors into four categories: probing, patterning, innovating and demonstrating. Each category has its own corresponding 'Action Mode'. The probing instinct creates a need to investigate in depth. Its corresponding action mode is termed Fact Finder. Fact Finder behaviors are generally linked in some way to the gathering of information. Someone who possesses this instinctive trait would be concerned with details, strategies, research, etc. How this concern occurs depends on the task presented, as it does for the other action modes. On the opposite end of the spectrum for this mode, someone who was not comfortable with fact finder may find themselves unable to cope with large volumes of details and avoid them altogether. The patterning instinct causes us to seek a sense of order. The corresponding action mode here is called Follow Through. This mode is characterized by how a person relates to structure. An individual strong in this action mode would tend to make lists and behave in a sequential fashion, comfortable with systems, procedures, design and order. On the contrary, someone who finds this type of instinctive behavior difficult would feel as if they were getting boxed in or overly structured if faced with this kind of environment. The third category of instinct is innovating, which is the force behind experimentation. Its action mode, Quick Start, is distinguished by perspective on risk. Someone governed by this mode would initiate change, deadlines and relish uniqueness. The opposite person would feel like they were surrounded by chaos and in a constant crisis atmosphere where a Quick Start would feel at home. Finally, the demonstrating instinct converts ideas into tangible form. The related action mode of

Implementor determines how we relate to objects and physical space. A large amount of this type of instinct would lead a person to construct, transport, manipulate and/or protect tangible goods. A lack of this type of instinct would likely feel the abstract and abstract notions to be as important or more important than the tangible.

These four categories, or action modes, are used to define the instinctive behavior of an individual. The Kolbe Concept® considers that each person has a finite amount of instinctive energy that is allocated over the four modes. How much or how little instinct an individual has in each mode defines an instinctual makeup. An individual's instinctive energy is assumed to sum to twenty. These twenty units are then allocated among the four action modes, resulting in a score of one to ten in each mode.

Once a score has been obtained for an action mode, it is classified as to the type of behavior it will represent. These classifications are referred to as Operating Zones. The first operating zone consists of all scores from one to three and is referred to as Prevention. It is characterized by the instinct to resist a form of action mode. Fundamentally, a person who is in the prevention operating zone for an action mode will tend to avoid behaving in that mode and attempt to reign in someone who is behaving in that mode. For example, if an individual is in the prevention zone for follow through, they will tend to prevent over-regulating or getting boxed in by someone who has a significant amount of follow through instinct. The next operating zone covers all scores from four to six and is called Response. It is characterized by the ability to accommodate behavior in an action mode. While someone who is in the response zone may not initiate a type of behavior they can certainly work in this way for a limited amount of time; either a small to moderate portion of each day or all day for a short time. For example, a person who is a responder to follow through will be able to work within a structure or procedures created by others. The last operating zone encompasses the scores of seven or greater and is known as Initiation. An initiator will often insist performing in the corresponding action mode given any choice. They will tend to initiate that type of behavior and are most comfortable working in that way. For example, a follow through initiator will create plans or procedures to follow and be happy defining a structure to act within.

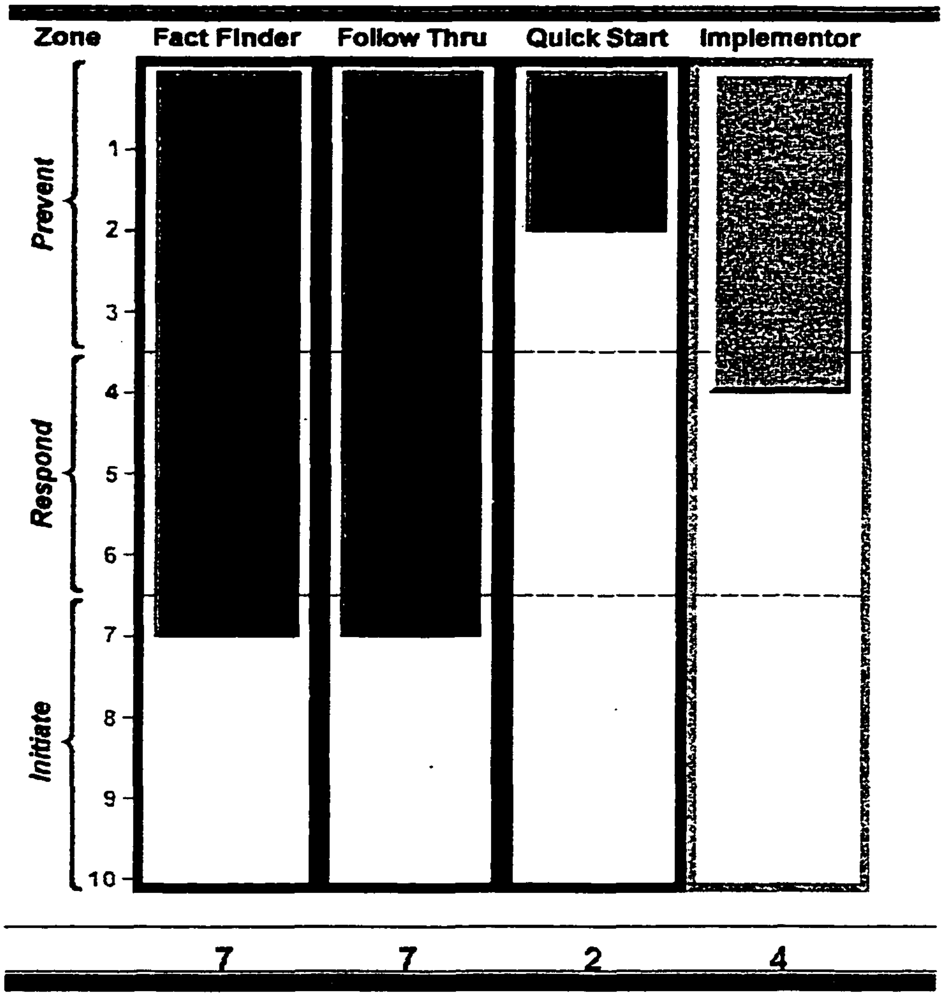
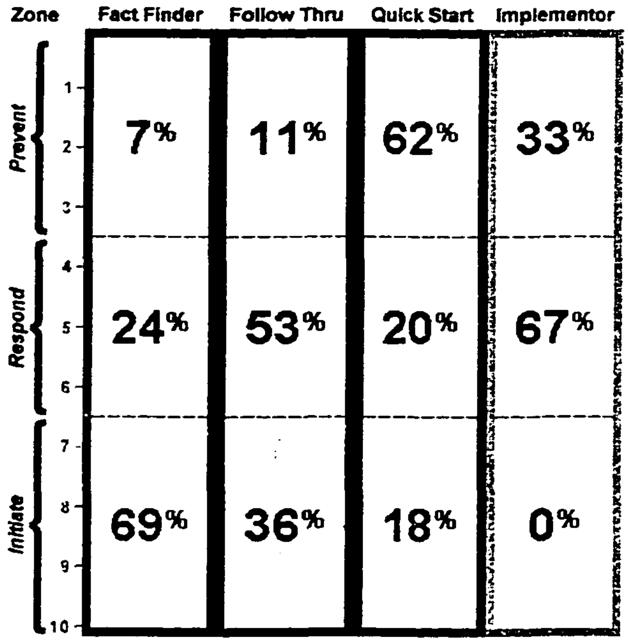


Figure 2 - Individual Kolbe Index Score

Figure 2 gives a visual interpretation of an individual's Kolbe Index score. The individual shown would be an initiator in fact finder and follow through, an accommodator in implementer and resist quick start behavior. When communicating this result an MO or modus operandi would be used. An MO consists of the individual's scores in the order shown above. In this case it is 7724.

3.2 Kolbe Team Measures

Now that we have discussed measures of an individual's instinctive talent, we shift our focus to understanding how this will help us form effective and productive teams. The

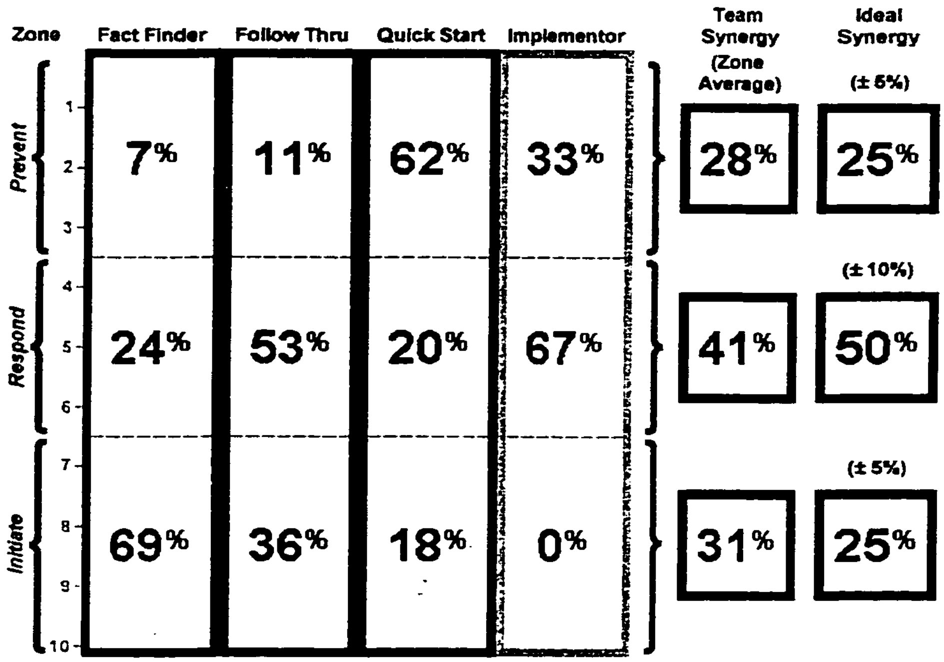


This analysis is based on a team of 52 with 45 respondents.

Figure 3 - Raw Kolbe Team Score

primary goal in all of the measures we will present is balance. A team's success or failure hinges strongly on the balance of conative energies present within it. A raw team score is constructed by calculating the percent of the team's members who fall in each operating zone for each action mode. This is shown in Figure 3 above.

Synergy is the first measure we will discuss. It refers to the distribution of team energy across the operating zones, m, without respect to the action modes, t. That is, the percent



This analysis is based on a team of 52 with 45 respondents.

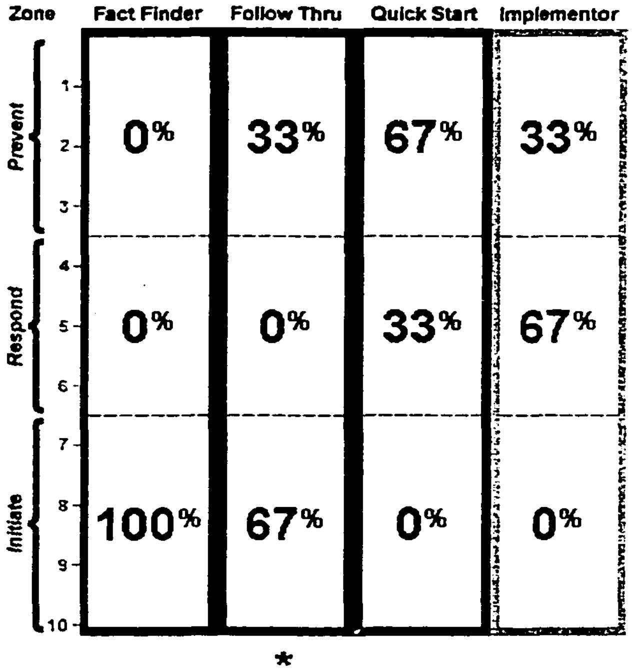
Figure 4 - Kolbe Team Synergy

of team members in each operating zone are added across action modes and then divided by four to determine the synergy rating for that operating mode. E.g. Let p_{mt}=percentage of individuals in zone m for mode t. Then for mode t, synergy is $S_i=\sum_{t=1}^{4} p_{mt} \big/ 4$. Kolbe Corp has determined through experimental evidence that the optimum distribution of synergy is 25% prevent, 50% respond, and 25% initiate. Allowances are given of ±5%, ±10% and ±5% respectively. The synergy of the team shown is shown in Figure 4.

It can be seen in Figure 4 above that the team measured only deviates from the optimum range in the initiate operating zone, and by only one percent. This is highlighted in red in the team synergy column. The deviation from ideal synergy, positive (d_m^-) or negative (d_m^+), is described by another measure called profitability. Profitability is a score out of one hundred that represents how close the team is to ideal synergy. High Profitability indicates the team uses an appropriate amount of energy initiating solutions to problems, accommodating actions required, and preventing problems from occurring. In this case, the profitability score is 95% as determined by Kolbe Corp's proprietary software.

A second measure we will look at is called Inertia. It is more specific in its approach. Instead of looking at all action modes at once it reviews them individually. Inertia occurs when there is an excess of conative energy in a combination of action mode and operating zone. An excess (d_{mt}^-) is considered anything above 30% in action mode in the prevent or initiate zone and above 60% in the respond zone. E.g. Let p_{1t}=percentage of individuals in zone 1 for mode t. Then for zone 1, mode t, the inertia is any positive value of e_{mt}= $p_{1t} - 0.30$ and the total inertia is $E_{mt}=\sum_{m=1}^{3}\sum_{t=1}^{4} p_{mt}$. Synergistic teams are generally more successful because there is enough "push and pull" from members' differing natural strengths and ways of acting (prevention and initiation zones) to create a balanced opposition of efforts. Second, an ideal team has enough members that can bridge the difference, evaluate and accommodate actions of both sides (response zone), and help to create a mutually acceptable solution in the creative problem-solving process.

The level of inertia present on a team is quantified by a measure out of one hundred called goal attainment. The goal attainment for the team presented in Figures 3 and 4 was determined by Kolbe Corp's proprietary software to be 64%. This low score indicates there is a stagnation of energy on the team. Passive behavior, sluggishness, and procrastination characterize the team.



This analysis is based on a team of 3 with 3 respondents.

Figure 5 - Illustration of Team Polarization

Viability is also a score out of one hundred and is a weighted average of profitability and goal attainment. When combined with a measure known as polarization, it gives a comprehensive prediction of team performance. Polarization identifies areas of potential conflict in a team by highlighting action modes where there are opposing levels of high energy in conjunction with a low level of energy in the corresponding respond zone. The team score given in Figure 5 shows polarization in the follow through action mode. As a result, the group may struggle unnecessarily trying to develop efficient, but not rigid, plans for achieving goals. By nature, the organized steps laid out by the people who initiate in Follow Through may be perceived as a stranglehold on those who don't. Conversely, those who operate without a sequential plan may bewilder others who need a guide for how the group will get from A to Z.

4 EXPERIMENTAL EVIDENCE

Prior to the inclusion of the Kolbe Concept® measures in a team selection model, it was deemed prudent to conduct an independent validation of the measures' effectiveness. Despite the quantity of data and analysis supporting its performance, our validation was necessary for completeness.

With the cooperation of Dr. Jeff Goldberg and his students we aimed to form synergistic teams in the SIE250/260 class and measure their effectiveness. This class was considered particularly suitable as it consists of over forty students and the class performs two projects during the course of the semester in groups of four. After approval was obtained from the University of Arizona's Human Subjects Committee (see Appendix A), the students were asked to complete the Kolbe Conative Index questionnaire. The completion of this and further participation were strictly voluntary. A sample of the consent form can be found in Appendix B. The results, in the form of individual MOs were recorded and used later to predict team performance. The students were only informed that I would form their teams as part of my research but were not given any information as to the nature of Kolbe Concept®. It should be noted that the ability to balance the teams synergistically is limited by the make up of the class itself. For example, there are no implementer initiators in the class and a large number of fact finder initiators. Thus the best solution is constrained by these factors.

In the problem definition we assume distinct and defined skill groups. In approximation of these skill groups we segregated the class according to known grade point average (GPA). Where no GPA was available, transfer credits or other suitable measures were used to estimate their skill. The top 25% of GPAs in the class is considered skill group one, the second quarter is skill group two, the third quarter is skill group three and the last quarter is skill group four. To remove the bias of academic ability from the teams we placed one person from each skill group in each team. This constitutes choosing candidate team members from various skill groups into teams as in our problem description. Half of the class was formed with the intent to maximize effectiveness according to Kolbe team measures goal attainment, profitability, and viability. The other half of the class was placed into teams randomly with the same skill group requirements. The teams used in the final analysis are given below in Table 1. Teams 1 and 9 were formed randomly.

Team 1		Team 4		Team 9	
Student	**MO**	**Student**	**MO**	**Student**	**MO**
1A	8814	4A	7634	9A	7816
1B	7535	4B	7346	9B	7634
1C	8633	4C	7473	9C	7625
1D	5636	4D	5835	9D	7454
Team 10		**Team 11**		**Team 13**	
Student	**MO**	**Student**	**MO**	**Student**	**MO**
10A	3483	11A	6483	13A	6445
10B	5834	11B	7534	13B	7534
10C	7436	11C	7653	13C	8723
10D	8634	11D	7724	13D	8831

Table 1 - Project 1 Team Structures

Once the teams were formed, Warewithal® software, kindly provided by Kolbe Corp, was used to predict the performance of the teams along the measures described above; viability, profitability and goal attainment. Any individual whose results indicated they were 'in transition' was excluded from the analysis. In order to prevent any unintentional bias, the instructor for the course, Jeff Goldberg, was neither informed which teams were formed by which method nor what the Kolbe Concept® predicted their performance would be. The projects were graded by Dr. Goldberg and the results, in the form of project scores, provided for analysis.

On the advice of Kolbe Corp, teams whose scores were based on fewer than four members were excluded from analysis. The results of the remaining six teams are given below. Standard correlation was used.

Team	Actual Score	Viability	Profitability	Goal Attainment
1	90	71	98	35
2	95	82	98	60
3	92	66	98	23
4	90	85	100	65
5	85	53	33	79
6	90	67	88	40
	Correlation	0.7117	0.8155	0.3571

Table 2 - Initial Validation Correlation Results

It is clear from table 2 that there is a positive correlation between the actual results and the Kolbe Concept®'s profitability measure and the joint measure, viability. It is also clear that there is a lesser correlation between the actual team scores and goal attainment. We see several possible explanations for this. First, as the term progressed it became clear that many of the previous GPAs were not representative of students' capabilities. This resulted in a number of teams not having equal skill levels, some had greater skill, some less. This affected the actual team results.

To correct this bias, with the aid of the instructor, the students were reassigned to the four skill groups. The class was again divided in two halves. This time, the Kolbe-based teams were formed using the heuristic provided in Section 6. The teams formed using this are given in Table 3.

Team 1				Goal	Team 4				Goal
Student	**MO**	**Viability**	**Profit.**	**Attain.**	**Student**	**MO**	**Viability**	**Profit.**	**Attain.**
1A	3484	85	100	65	4A	7266	82	98	60
1B	8723				4B	6483			
1C	5636				4C	7724			
1D	7454				4D	7535			
Team 2				**Goal**	**Team 5**				**Goal**
Student	**MO**	**Viability**	**Profit.**	**Attain.**	**Student**	**MO**	**Viability**	**Profit.**	**Attain.**
2A	7816	82	98	60	5A	7733	51	59	40
2B	8652				5B	5834			
2C	4375				5C	7534			
2D	7534				5D	8633			
Team 3				**Goal**	**Team 6**				**Goal**
Student	**MO**	**Viability**	**Profit.**	**Attain.**	**Student**	**MO**	**Viability**	**Profit.**	**Attain.**
3A	8831	80	98	55	6A	7733	57	59	55
3B	4736				6B	7473			
3C	8623				6C	5835			
3D	6445				6D	7346			

Table 3 - Project 2 Teams and Predictive Scores

Of the teams shown in Table 3, team 4 and team 6 were eliminated from the study due to the loss of one or more team members. The remaining four, along with randomly formed teams 7 and 8, performed as given under ‘Actual Score’ in Table 4.

Team	Actual Score	Viability	Profitability	Goal Attainment
1	92	85	100	65
2	95	82	98	60
3	90	80	98	55
5	85	51	59	40
7	88	55	78	25
8	95	82	98	60
	Correlation	0.8181	0.8624	0.6033

Table 4 – Project Two Correlation Results

It can be seen by comparing the correlations from Table 2 and Table 4 that the predictions made using the Kolbe measures for project two were better on every criteria. The greatest increase in correlation occurs for Goal Attainment. This may be attributable to the additional input from the instructor on the skill levels of participants. This reflects the importance of the assumption that skills have previously been carefully assigned to individuals and shows the additional importance of management input in doing so.

To gain a further understanding of the Kolbe Concept®, the behavior of individuals within teams was subjectively evaluated. From this evaluation we saw evidence of the predictive abilities of the Kolbe Concept® with respect to particular individual and team behaviors, not simply overall team performance. For example, team 2's high-GPA person was determined to be an insistent Follow Through according to her MO. From this, we would expect to see her demonstrate patterning behavior, e.g. scheduling, planning, etc. In fact, after consulting with the professor, Dr. Goldberg, it was clear that she had initiated this type of behavior from the start. She had developed a plan and executed it according to schedule, leading the overall well-balanced group to a successful project submission. Two individuals on other teams were noted by Dr. Goldberg to have demonstrated particularly evident behaviors. When their MOs were referred to they proved to be insistent in these action modes. On a team level, team 4 was heavily laden with Fact Finder with all team members in the prevent operating zone for Quick Start. The anticipated behavior for a group with this construction is that they would carry out excessive research and produce large volumes of work with very little ability to reach clear conclusions or make a decisive finish. According to the professor, this group indeed performed a very large amount of work on the project but only drew a single conclusion from the analysis they had performed.

Based on the positive correlations between the Kolbe Concept® measures and actual team performance we submit that it would be an asset to include these in a team formation model.

5 MIP FORMULATION

Labor Pool Extraction Model – Single Team: The first situation we will model requires one to form a single team from a pool of candidates with the necessary skills. This situation commonly occurs in dynamic operations. The method itself takes the form of a selection model. In the model it is assumed that the part-machine cell has been predetermined. Subsequently, the set of employee skills for the cell is evaluated. It is further assumed that these skills are available in the pool of employees and that each employee has been categorized as to the skill role they fill. The problem then becomes the selection of the appropriate employees for each role in the cell in such a way that the Kolbe team performance indices are maximized.

The model can be described via:

Decision Variables:

Y_i = 1 if worker i is assigned to team and 0 otherwise

Data Coefficients:

Π_j = set of workers in skill group j;

a_{itm} = 1 if worker i exhibits operating zone m for action mode t and 0 otherwise;

S_j = number of workers of skill group j required;

W_m, W_{mt} = relative weights for zone and mode-zone combinations;

Z_m = desired level of zone m across all modes t;

Z_{mt} = maximum desired level of mode t at zone m.

The zones $m = 1, 2$ and 3 correspond to preventer, responder, and initiator tendencies of individuals respectively. The model becomes:

Labor Pool Extraction Model - Single Team (LPEST):

Minimize D

$$= \sum_m W_m (d_m^+ + d_m^-) + \sum_m \sum_t W_{mt} d_{mt}^- \qquad (1)$$

subject to:

$$\sum_{i \in \Pi_j} Y_i = S_j \ , \quad \forall j \qquad (2)$$

$$\sum_t \sum_i a_{itm} Y_i + d_m^+ - d_m^- = Z_m \cdot \sum_j S_j \ / 4, \quad \forall m \quad (3)$$

$$\sum_i a_{itm} Y_i + d_{mt}^+ - d_{mt}^- = Z_{mt} \cdot \sum_j S_j \ , \quad \forall m,t \qquad (4)$$

$$Y_i \in \{0,1\}, \text{ All vars} \geq 0$$

The objective function (1) computes the deviation from optimal Kolbe indices. Only the deviations in excess of the maximum values are included for the zone/mode combination as we are unconcerned with how low the values reach. The factors are weighted to facilitate model flexibility. The weights can be set to standard levels or set to reflect management input. Additional indices from the Kolbe system can be included as desired. Equation (2) ensures the desired number of team members is correct in each category.

Equations (3) and (4) define the deviations from the optimal Kolbe profitability and goal attainment indices used in the objective function. The ±5% and ±10% allowances are not provided for here. The optimal values of the indices are set as recommended by the Kolbe system. Finally, we enforce the binary and nonnegativity restrictions on the variables.

Labor Pool Extraction Model – Multiple Teams: The next situation we will model requires one to form multiple teams from a pool of candidates with the necessary skills. The teams may consist of all or some of the available employees. This situation often occurs when converting to a new cellular manufacturing layout. This method also takes the form of a selection model. As mentioned previously, the model assumes that the part-machine cells have been predetermined. Subsequently, and also prior to the running of the model, the set of employee skills for the cells is evaluated. It is further assumed that these skills are available in the pool of employees and that each employee has been categorized as to the skill role they fill. The problem then becomes the selection of the appropriate employees for each role in the various cells in such a way that the Kolbe team performance indices are maximized. The model can be formulated by incorporating minor changes to the original model. It is shown in its entirety below.

<u>Decision Variables:</u>

$Y_{ik} = 1$ if worker i is assigned to team k and 0 otherwise

<u>Data Coefficients:</u>

Π_j = set of workers in skill group j;

a_{itm} = 1 if worker i exhibits operating zone m for action mode t and 0 otherwise;

S_{jk} = number of workers of skill group j required for team k;

W_m, W_{mt} = relative weights for zone and mode-zone combinations;

Z_m = desired level of zone m across all modes t;

Z_{mt} = maximum desired level of mode t at zone m.

Labor Pool Extraction Model - Multiple Teams (LPEMT):

Minimize $D2$

$$= \sum_k \sum_m W_m (d^+_{mk} + d^-_{mk}) + \sum_k \sum_m \sum_t W_{mt} d^-_{mtk} \qquad (5)$$

subject to:

$$\sum_{i \in \Pi_j} Y_{ik} = S_{jk} \ , \quad \forall j,k \qquad (6)$$

$$\sum_k Y_{ik} \le 1, \ \forall i \qquad (7)$$

$$\sum_t \sum_i a_{itm} Y_{ik} + d^+_{mk} - d^-_{mk} = Z_m \cdot \sum_j S_{jk} \ /4, \ \ \forall m,k \qquad (8)$$

$$\sum_i a_{itm} Y_{ik} + d^+_{mtk} - d^-_{mtk} = Z_{mt} \cdot \sum_j S_{jk} \ , \ \ \forall m,t,k \qquad (9)$$

$$Y_{ik} \in \{0,1\} \ , \ \text{All vars} \ge 0$$

The objective function (5) computes the deviation from optimal Kolbe indices across all teams. Only the deviations in excess of the maximum values are included for the zone/mode combination as we are unconcerned with how low the values reach. The factors are weighted to facilitate model flexibility. The weights can be set as described above. Equation (6) ensures that each team is formed with the appropriate number of members from each skill group. Equation (7) ensures that a worker is assigned to at most one team. Equations (8) and (9) define the deviations for each team from the optimal Kolbe profitability and goal attainment indices used in the objective function. The optimal values of the indices are set as recommended by the Kolbe system. Finally, we enforce the binary and nonnegativity restrictions on the variables.

These simple adjustments give the model the flexibility to form more than one team at a time. If required, the second model can be run with only one team. This would result in the original single team model.

The LPEMT was programmed for a moderately small instance, five teams of four people and one team of five. Using CPLEX v.6.0 this program was run to optimality. The computation time to solve this was 36908.69 seconds, or approximately 10 hours and 15 minutes.

6 HEURISTIC APPROACH

Initial computational experience (using CPLEX) indicated that even moderately small instances of LPEMT are difficult to solve optimally by standard integer programming techniques. In this section we propose heuristics for solving the group formation problem. The notation used is given below.

<u>*Notation:*</u>

Π_j Set of employees in skill group j	$j \in [1, J]$ Skill groups
Φ_l Set of unfulfilled teams	$i \in [1, I]$ Employees
T_k Members of team k.	$k \in [1, K]$ Teams
Θ_s Set of unfulfilled skill groups	$t \in [1, 4]$ Action modes; FF,FT,QS,I respectively
$a_{itm} = 1$ if employee i operates in zone m for action mode t, $= 0$ otherwise.	$m \in [1, 3]$ Operating zones; P,R,I respectively
$n_j =$ number of individuals available from skill group j.	$n_k =$ number of members required for team k.

6.1 Simple Greedy

We will first address the single team problem. An easy approach to this problem would be the simplest greedy heuristic. That is to arbitrarily select the first member of the team and progress through the remaining skill requirements for the team choosing the new member at each stage that minimizes the value of the objective function (1). In order to find an improved solution, the first member selected could be varied through the skill group it was chosen from and the best resulting solution chosen. The algorithm is as follows:

1. Let $i \in [1, I]$ be the set of potential team members. Arbitrarily select an individual from Π_j, where j is the first skill group with a requirement. Place this individual in T_k, the set of team members, and remove him/her from further consideration.
2. If there is no further requirement for individuals from the skill group whose member was most recently placed, remove all other members of that skill group from further consideration.
3. Beginning with the first individual still available for placement, place them one at a time temporarily into T_k. For each evaluate:

$$\sum_t \sum_i a_{itm} + d_m^+ - d_m^- = Z_m \cdot \sum_j S_j \,/4, \;\; \forall m \qquad (9)$$

$$\sum_i a_{itm} + d_{mt}^+ - d_{mt}^- = Z_{mt} \cdot \sum_j S_j \;, \;\; \forall m,t \qquad (10)$$

and finally,

$$D = \sum_m W_m (d_m^+ + d_m^-) + \sum_m \sum_t W_{mt} d_{mt}^- \qquad (11)$$

4. Select the potential team member with the smallest value of D and make it a permanent member of T_k. Remove this individual from further consideration.

5. If team requirements remain, return to step 2, else the team has been formed.

Computationally, as we start with each of the I individuals and, in each case, evaluate the linear functions (9), (10) and (11) for each of the remaining available individuals (at most I-1), this heuristic has computations of $O(I^2)$.

6.2 Balanced Placement

6.2.1 Description

When multiple teams are required, we propose a Balanced Placement heuristic. This algorithm is designed to combine operating zones of high density in available employees with operating zones of low density in partially formed teams. This idea is combined with careful selection and matching procedures to form the Balanced Placement heuristic described below.

Balanced Placement Heuristic

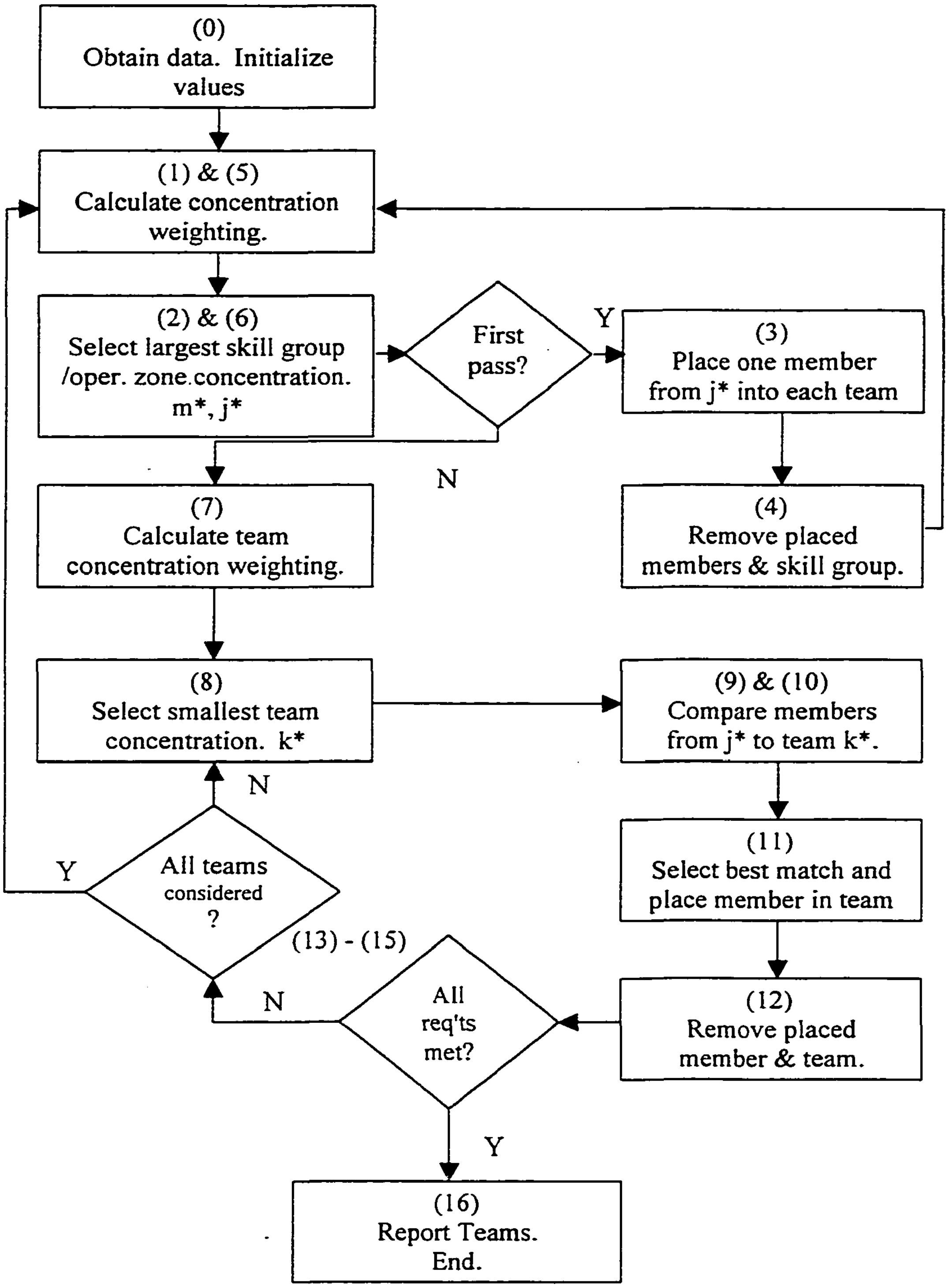


Figure 6 - Balanced Placement Heuristic

Step (0) initialized the heuristic, obtaining all a_{itm} data and creating the original sets of skills and teams as described in Figure 1. Each individual belongs to a skill group j and the teams k are currently empty. The a_{itm} for an individual can be thought of as a two-dimensional matrix with four columns representing action modes, t, and three rows representing operating zones, m, e.g. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

Step (1) computes weights, W_{mj}, for each operating zone/skill group combination. These are averaged over the number of individuals in the skill group. For example, W_{11} sums all the first row entries for individuals in skill group one and then divides by the number of individuals in skill group one. Weights for the respond operating zone are multiplied by one half as the Kolbe Concept® deems the optimum level of this zone to be twice the others. The multiplication allows the weights to be compared directly.

Step (2) selects the skill group that has the greatest concentration in any operating zone and marks it for further use. High concentration refers to a skill group whose members largely fall into an operating zone. If a tie occurs, the heuristic looks for the greatest concentration across all operating zones for those skill groups in the tie. Simply taking the tied skill group with the smallest index breaks persistent ties.

Step (3) arbitrarily places one individual from the selected dense skill group into each team that has a requirement for that skill. This team was chosen first to avoid trying to find good matches of high-concentration skill groups to partially formed teams. It is easier to find individuals from low-density skill groups to fill gaps in teams with high densities than it is to match individuals from high-density skill groups to teams with most members already placed. As the teams are currently empty it doesn't matter which team obtains which individual.

Step (4) updates the set of available individuals by removing those placed from the set of employees in the skill group. The entire skill group is taken out of consideration until every other skill group has placed individuals. If there is a requirement for additional members of the skill group, they will be placed in a subsequent pass.

Step (5) is skipped for now, but is used to recalculate new weights, W_{mj}, should multiple passes be necessary.

Step (6) selects from the remaining skill groups the one that has the greatest concentration in any operating zone and marks it and the operating zone where the concentration was found for further use. Here we also check the set of teams to determine which teams' requirements have not yet been fully met. Only these teams will be considered further.

Step (7) computes the concentrations in the partially formed teams for the operating zone where high concentration was found in the skill group selected. These are averaged over the total number of team members required.

Step (8) selects the team with the lowest operating zone of those examined. By doing this, we will be matching potential team members' high concentrations with partially formed teams' low concentrations, thus *balancing* team concentrations. By averaging over the total number of team members required instead of the only the team members placed preference is given to teams with more unfilled spots. The only tie-breaking mechanism employed is the smallest index as there is no apparent way to break the tie that would improve the solution.

Step (9) creates a matrix of values for a team similar to those representing a_{itm} for an individual by summing along the operating zones and action modes. For example, if team k* had two team members with matrices $\begin{bmatrix} 0\,1\,1\,0 \\ 1\,0\,0\,1 \\ 0\,0\,0\,0 \end{bmatrix}$ and $\begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,1 \\ 0\,1\,0\,0 \end{bmatrix}$, the values of V_{tm} could be represented as $\begin{bmatrix} 1\,1\,1\,0 \\ 1\,0\,1\,2 \\ 0\,1\,0\,0 \end{bmatrix}$. In step (10) the absolute value deviations of potential team members from the values in this new matrix are measured and summed. As the maximum total deviation will be selected in the next step, the deviations for the respond operating zone are doubled here to reflect the desire to have twice as much in this zone as the others. The maximum of the deviations for each potential team member is also measured.

Step (11) selects the potential team member whose difference from the team is highest, again *balancing* the team makeup. If there is a tie, the potential team member whose highest single deviation was smallest is chosen. This represents the individual who varied in more values as opposed to an individual who may have varied greatly in one value. If the tie persists, there is no apparent difference between the tied potential team members so the one with the smallest index is chosen.

Step (12) updates the set of available individuals by removing the one placed from the set of employees in the skill group. The entire team is then taken out of consideration until every other team has acquired individuals from this skill group if required.

Steps (13) through (15) evaluate and direct the heuristic to repeat steps as necessary. Steps (8) through (12), placing individuals from the current selected skill group to teams, need to be repeated for every team that requires that skill until all teams have been considered once. Then we can move on to the next skill group by going to step (6). Once we have placed individuals from all the skill groups once, we can repeat the process for any skill requirements still remaining. If the largest number of a single skill required for any team is four then there will be three passes of the heuristic. This results as two skill groups are placed in the first pass and one in each pass after. Once all team requirements are met, the heuristic ends and the teams are determined by $T_k \quad \forall k$.

6.2.2 Formulation

The Balanced Placement Heuristic described above is shown in notational detail below. The notation used is restated for readability.

Notation:

Π_j Set of employees in skill group j	$j \in [1, J]$ Skill groups
Φ_l Set of unfulfilled teams	$i \in [1, I]$ Employees
T_k Members of team k.	$k \in [1, K]$ Teams
Θ_s Set of unfulfilled skill groups	$t \in [1, 4]$ Action modes; FF,FT,QS,I respectively
$a_{itm} = 1$ if employee i operates in zone m for action mode t, $= 0$ otherwise.	$m \in [1, 3]$ Operating zones; P,R,I respectively
$n_j =$ number of individuals available from skill group j.	$n_k =$ number of members required for team k.

Balanced Placement Heuristic

(0) Obtain or read-in all employee Kolbe indices. Let Θ_s be the set of skills that are required to form the teams. Let Φ_l be the set of required teams.

(1) Calculate

$$W_{mj} = \sum_{i \in \Pi_j} \sum_{t=1}^{4} a_{itm} \Big/ n_j \quad \forall m = 1,3 \quad \forall j \in \Theta_s; \qquad W_{mj} = \frac{1}{2} \sum_{i \in \Pi_j} \sum_{t=1}^{4} a_{itm} \Big/ n_j \quad \forall m = 2 \quad \forall j \in \Theta_s.$$

(2) Set (m*, j*)=argmax$_{m,j}$ $\left(W_{mj}\right)$.

Break tie by selecting max $\left(\sum_{m=1}^{3} W_{mj}\right)$, if stiil tie choose smallest index.

(3) Place one $i \in \Pi_{j^*}$ into each team T_k where $k \in \Phi_l$ with a requirement for same.

(4) Remove all placed personnel from Π_{j^*} and remove skill group j* from Θ_s. Go to (6).

(5) Calculate

$$W_{mj} = \sum_{i \in \Pi_j} \sum_{t=1}^{4} a_{itm} \Big/ n_j \quad \forall m = 1,3 \quad \forall j \in \Theta_s; \qquad W_{mj} = \frac{1}{2} \sum_{i \in \Pi_j} \sum_{t=1}^{4} a_{itm} \Big/ n_j \quad \forall m = 2 \quad \forall j \in \Theta_s.$$

(6) Set (m*, j*)=argmax$_{m,j}$ $\left(W_{mj}\right)$, where $j \in \Theta_s$. Let Φ_l be the set of teams whose requirements are not yet met. Break ties as in (2).

(7) Calculate $U_k = \sum_{i \in T_k} \sum_{t=1}^{4} a_{itm^*} \Big/ n_k \quad \forall k \in \Phi_l$.

(8) Set (k*)=argmin$_k$ $\left(U_k\right)$, where $k \in \Phi_l$. Ties go to the smallest index.

(9) Calculate $V_{tm} = \sum_{i \in T_{k^*}} a_{itm} \quad \forall t, m$.

(10) Calculate

$$d_{itm} = \left|V_{tm} - a_{itm}\right| \quad \forall i \in \Pi_{j^*}, t, m = 1,3; \quad d_{itm} = 2\left|V_{tm} - a_{itm}\right| \quad \forall i \in \Pi_{j^*}, t, m = 2.$$

$$D_i = \sum_{t=1}^{4} \sum_{m=1}^{3} d_{itm} \quad \forall i \in \Pi_{j^*}; \quad D_i' = \left(\max\left(d_{itm}\right)\right) \quad \forall i \in \Pi_{j^*}.$$

(11) Set (i*)=argmin$_i$ $\left(D_i\right)$, place i* in T_{k^*}.

If tie, select employee i* corresponding to (i*)=argmin$_i$ $\left(D_i^{'}\right)$. If tie persists, choose smallest index.

(12) Remove employee i* from Π_{j^*} and remove group k* from Φ_l.

(13) If all teams' requirements have been met, go to (16).

(14) If $\Phi_l \not\in 0$, go to (8) else, remove skill group j* from Θ_s.

(15) If $\Theta_s \not\in 0$, go to (6) else, reset Θ_s to include all skill groups still required by teams and go to (5).

(16) Report teams $T_k \quad \forall k$, END.

6.2.3 Example

The following example is used to illustrate the use of the Balanced Placement heuristic.

Problem:

Two teams need to be formed. The teams are to consist of one individual from skill group 1, j_1, and one individual from skill group 2, j_2. We have a pool of six members to choose from. Three members are from j_1 and three are from j_2. The Kolbe Indexes of these individuals will be determined in step (0).

Heuristic:

Step (0): Normally we would obtain the individual Kolbe Index scores from an existing database or have them tested if no scores already exist. In this case they were generated randomly using the F program "Members". The output as generated from the program is contained in Appendix G. The profiles are:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	id	j	k	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	a_{43}
1	01	1	0	1	0	0	0	0	1	1	0	0	0	0	1
2	02	2	0	0	0	1	1	0	0	0	0	1	1	0	0
3	03	1	0	0	1	0	1	0	0	0	0	1	1	0	0
4	04	2	0	0	0	1	1	0	0	0	0	1	1	0	0
5	05	1	0	0	0	1	0	1	0	0	1	0	0	1	0
6	06	2	0	1	0	0	1	0	0	0	1	0	0	0	1

The leftmost column indicates row numbers. The topmost row indicates column numbers. The second row gives column labels. Column 1 lists member identification numbers, column 2 shows skill group, column 3 shows the assigned team (zero if none). Columns 4, 7, 10 and 13 show whether an individual is in the prevent zone for any of the action modes. A "1" in the body of the table means the individual is in the prevent zone. The prevent zone can be determined by the second subscript 1. The first subscript represents the action mode: 1 for fact finder, 2 for follow through, 3 for quick start and 4 for implementer. Columns 5, 8, 11 and 14 represent the respond operating zone and columns 6, 9, 12 and 15 the initiate operating zone.

Step (1): The weights, w_{mj}, are determined by summing across the operating zone, m, and skill group j. Thus w_{11} can be determined by summing columns 4, 7, 10 and 13 for rows 1, 3, and 5. This would represent the amount of prevent energy present in the skill group. To equalize between the groups, we divide by the number of people in the skill group, 3. We also divide operating zone 2 by two. Thus, w_{11}=4/3=1.333. Likewise, w_{12}=2, w_{21}=0.500, w_{22}=0.167, w_{31}=1.667, w_{32}=1.667.

Step (2): Clearly the maximum weight from (1) is w_{12}. As a result we let operating zone m*=1 and skill group j*=2.

Step (3): As this is the first pass we simply place the first member of skill group j*=2, 02, into the first team that has a requirement for it, team 1. Then we place the next member of skill group 2, 04, into team 2. As there are no more groups we go on to the next step.

Step (4): Members 02 and 04 are removed from further consideration. As is skill group 2. These can be reinstated later if needed. Our tableau now looks like this:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	id	j	k	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	a_{43}
1	01	1	0	1	0	0	0	0	1	1	0	0	0	0	1
3	03	1	0	0	1	0	1	0	0	0	0	1	1	0	0
5	05	1	0	0	0	1	0	1	0	0	1	0	0	0	1
6	06	2	0	1	0	0	1	0	0	0	1	0	0	0	1
2	02	2	1	0	0	1	1	0	0	0	0	1	1	0	0
4	04	2	2	0	0	1	1	0	0	0	0	1	1	0	0

Step (5): Now we need to determine the next skill group to be placed. We do this by calculating the weights again. As no weights have changed, only the group to be considered we can reuse the weights from step (2). We simply exclude the weights relating to skill group 2. Thus, w_{11}=1.333, w_{21}=0.500, w_{31}=1.667.

Step (6): The maximum weight is 1.667. Therefore we select w_{31}. As a result we let operating zone m*=3 and skill group j*=1.

Step (7): To select the team in which the first of skill group 1 will be placed, we must look for low concentration in m*=3 in the placed teams. Team 1 currently consists of one member, 02. Thus, the weight for this team is (0+0+1+1)/2=1. The weight for team 2 is also (0+0+1+1)/2=1.

Step (8): As the weights are identical, we must choose arbitrarily between them. Set team k*=1.

Step (9): To compare the members of skill group 1 with team 1 we first sum each a_{mt} across the members placed in team 1 (only one in this case). We then take the difference between the sum and each a_{mt} for each potential member of skill group 1, 03 and 05. Differences are doubled for operating zone 2. The differences for member 01 are 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, and the differences for member 03 are 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0. Similarly, the differences for member 05 are 0, 0, 0, 1, 2, 0, 0, 2, 1, 1, 0, 1. Finally, we sum these differences D_{01}=8, D_{03}=3 and D_{05}=8.

Step (11): To make our selection we take the maximum of the D's determined in step (9). Because this is tied, we choose the member with the smallest maximum difference. This corresponds to member 01. Therefore, we place member 01 in team 1.

Step (12): We remove member 05 from further consideration and temporarily remove team 1 from consideration as well. The tableau now looks like:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	id	j	k	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	a_{43}
3	03	1	0	0	1	0	1	0	0	0	0	1	1	0	0
5	05	1	1	0	0	1	0	1	0	0	1	0	0	0	1
6	06	2	0	1	0	0	1	0	0	0	1	0	0	0	1
2	02	2	1	0	0	1	1	0	0	0	0	1	1	0	0
1	01	1	0	1	0	0	0	0	1	1	0	0	0	0	1
4	04	2	2	0	0	1	1	0	0	0	0	1	1	0	0

Step (13): We have not met all our requirements

Step (14): We have not considered all the teams on this pass yet so we return to step (8).

Step (8): We select the one team still available for consideration, team 2.

Step (9): We compare the members from team 2 to the remaining members of skill group 1. D_{03}=0+2+1+0+ 0+0+ 0+0+0+0+0+0=3 and D_{05}=0+0+0+1+2+0+0+2+1+1+0+1=8.

Step (11): We select the maximum D, which corresponds to member 05. Them member 05 is placed in team 2. This tableau results:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	id	j	k	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	a_{43}
3	03	1	0	0	1	0	1	0	0	0	0	1	1	0	0
6	06	2	0	1	0	0	1	0	0	0	1	0	0	0	1
2	02	2	1	0	0	1	1	0	0	0	0	1	1	0	0
1	01	1	1	1	0	0	0	0	1	1	0	0	0	0	1
4	04	2	2	0	0	1	1	0	0	0	0	1	1	0	0
5	05	1	2	0	0	1	0	1	0	0	1	0	0	0	1

Step (12): We remove member 05 from further consideration and temporarily remove team 2.

Step (13): We have met all our requirements

Step (16): The teams are: Team 1 (01, 02), Team 2 (04, 05).

6.2.4 Code Verification

The Balanced Placement Heuristic was coded to enable it to be tested on a series of problems. The results of the test problems are given in Section 7. The code was written in the F programming language, which was courteously provided along with manuals by Unicomp of Tucson, Arizona. This code for the main program, TeamBuilder is included in Appendix C. The related module, TB, is in Appendix D. The Kolbe Index scores for the test problems were also generated using this language. The code for the program, Members, used to generate these random values is included in Appendix E. Its related module, Generate, is in Appendix F.

In order to verify the code itself, the various outputs generated by the code were analyzed for the example problem given in Section 6.2.3. These outputs include the final team placements, Final, the scores for the teams, Score, and a partial trace of the program itself, Trace. These outputs can be found in Appendices H, I, and J, respectively.

The first step in the verification was to check the sequence of subroutines calls from the trace to make sure it was behaving as expected. The order was:

```
GetData
CalcWeights1
SelectMax1
Place1
RemoveSkGroup
SelectMax1
RemoveSkGroup
CalcTeamWeights
SelectTeam
ChooseMember
Iterate
SelectTeam
ChooseMember
Iterate
Score
```

This clearly follows the steps outlined in the example problem.

Next, we looked at each subroutine individually to assess whether it was performing its function properly. The first subroutine, GetData, trace output was as follows:

```
GetData
  1  1  0  0
  1  1  0  0
  4
```

The first two lines shows that two teams are required and that they need one individual from skill groups one and two. The four zeroes indicate there are no requirements for skill groups three and four. Line three indicates a total of four team members (over all teams) is required.

Subroutine CalcWeights1 below shows the weights were calculated as given in step (1) of the example. The maximum weight is correctly chosen in SelectMax1 and the resulting individuals are placed appropriately in Place1.

```
CalcWeights1
The weights are:
   1.33   2.00   0.00   0.00   0.50   0.17   0.00   0.00
1.67   1.67   0.00   0.00
SelectMax1
The selected weight is:
   2.00
The operating zone, skill group:
  1  2
Place1
The team for  2 is
  1
The team for  4 is
  2
```

The previously considered skill group 2 is removed from consideration by setting its associated weights to zero in RemoveSkGroup. The next weight is selected as described in the example by SelectMax1 in the second visit to this subroutine. Subsequently, skill group 1 is removed from consideration by recalling subroutine RemoveSkGroup. All the weights are now equal to zero.

```
RemoveSkGroup
The current weights are:
  1.33  0.00  0.00  0.00  0.50  0.00  0.00  0.00  1.67
0.00  0.00  0.00
SelectMax1
The selected weight is:
   1.67
The operating zone, skill group:
  3  1
RemoveSkGroup
The current weights are:
  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00
```

As we are past the first pass, we must calculate team weights to determine which team to consider. This calculation and selection are carried out accurately by subroutines CalcTeamWeights and SelectTeam shown here.

```
CalcTeamWeights
The team weights are:
   1.00
   1.00
SelectTeam
The lowest team weight is
   1.00
The team is
  1
```

Finally, the first iteration is ended choosing the individual to place on team one. The trace file shows us that the values calculated by the subroutine ChooseMember match those determined in the example problem and that the correct individual is chosen and placed in team 1.

```
ChooseMember
The team sums are
   0   0   1   1   0   0   0   0   1   1   0   0
The total deviation for   1 is   8
The single max deviation is
   1
The total deviation for   3 is   3
The single max deviation is
   2
The total deviation for   5 is   8
The single max deviation is
   2
maxdev=    8
The maximum total deviation is
   8
q=   1p=   1
q=   2p=   5
The member chosen is
  1
and placed in team
   1
```

The end of this subroutine invokes the Iterate subroutine, which chooses the appropriate next course of action. This correctly passes the program to SelectTeam, which is followed by ChooseMember. These place the last required individual, 5, in team 2. As this completes our requirements, the subroutine Iterate terminates and the Score subroutine is called to calculate the necessary values.

```
Iterate
```

```
SelectTeam
The lowest team weight is
   1.00
The team is
  2
ChooseMember
The team sums are
   0    0    1    1    0    0    0    0    1    1    0    0
The total deviation for   3 is   3
The single max deviation is
   2
The total deviation for   5 is   8
The single max deviation is
   2
maxdev=   8
The maximum total deviation is
   8
q=   1p=   5
The member chosen is
  5
and placed in team
   2
Iterate
Score
```

7 PERFORMANCE

The Balanced Placement Heuristic was coded to enable it to be tested on a series of problems. The intention of these tests is to assess the performance of the heuristic with respect to a set of lower bounds.

The following scenarios in Table 4 were run on the program implementing the Balanced Placement Heuristic:

Trial Set	Number of Teams	Members per Team	Member Pool
1	4	4	16
2	4	8	32
3	4	12	48
4	8	4	32
5	8	8	64
6	8	12	96
7	12	4	48
8	12	8	96
9	12	12	144

Table 4 - Test Problems

From the randomly generated sets, lower bounds on the performance of the heuristic were established. The lower bounds for Profitability were determined by summing the operating zones for the entire pool of members and dividing by the number of teams. The result was then normalized to a value out of one. This represents the best possible distribution of possible team members to teams for Synergy. The lower bounds for Goal Achievement were constructed similarly. Instead of summing over just operating zones, they were calculated over each combination of operating zone and action mode. This was also divided by the number of teams. The result was also normalized to a value out of one. This represents the best possible distribution of possible team members to teams for the Inertia measure. The weighted average of these two measures is the lower bound on Viability.

Each test problem was required to have four skill groups and an even distribution of the required number of each skill group in each team. Ten trials of each problem were performed providing Profitability, Goal Attainment and Viability scores out of one. The results were then compared to the lower bounds and a difference obtained. The average differences from lower bounds of these trials are shown in Table 5.

	Maximum	Team	Deviation	Average	Teams'	Deviations
Trial Set	Profitability	Goal Attainment	Viability	Profitability	Goal Attainment	Viability
1	0.27187	0.13252	0.20915	0.14064	0.07970	0.11319
2	0.26562	0.11044	0.19581	0.12032	0.06641	0.09606
3	0.24584	0.10483	0.18241	0.11357	0.06113	0.08997
4	0.26718	0.14974	0.21431	0.11354	0.07954	0.09823
5	0.27811	0.14390	0.21772	0.12081	0.07980	0.10237
6	0.26015	0.13383	0.20331	0.10935	0.07916	0.09577
7	0.28228	0.16166	0.22800	0.11166	0.07883	0.09688
8	0.34531	0.19578	0.27804	0.12496	0.08646	0.10764
9	0.27272	0.17352	0.22809	0.10587	0.08544	0.09669

Table 5 - Trial Results

The maximum deviation columns refer to the average over a trial set of the largest deviation of any team for particular problem solution. Based on these results, the 10% confidence intervals for the mean maximum and average lower bound deviation are:

Measure	Mean	Upper Confidence Limit	Lower Confidence Limit
Maximum Team Deviation for Trial			
Profitability	0.281419	0.286697	0.276141
Goal Attainment	0.150363	0.155919	0.144807
Viability	0.222451	0.227666	0.217236
Average of Team Deviations for Trial			
Profitability	0.117386	0.119116	0.115657
Attainment	0.079557	0.081364	0.077750
Viability	0.100366	0.101580	0.099151

Table 6 - Analysis of Results

The lower bounds were also calculated for the pool of individuals whose team assignment was solved to optimality. These lower bounds and the corresponding performance measures for the optimal solution are given in Table 7.

Measure	Bounds	Maximum Deviation Among Teams in the Trial	Average Deviation over Trial
Profitability	0.00000	0.12500	0.05830
Goal Attainment	0.14500	0.09250	0.02370
Viability	0.06520	0.11040	0.04280

Table 7 – Performance of Optimally Solved Problem

8 CONCLUSIONS

Through the literature survey we identified the Kolbe Concept® as a potential measurement tool for personality traits. It specifically measures the conative, or instinctual, tendencies of individuals. We found this to be a good candidate because it professed to be stable over time, easily measured and demonstrably linked to the productivity of the system. By carrying out experiments, we gathered evidence of this link and thus justified its inclusion in our model. However, we are not limiting this approach to one measurement system. There are other potential candidates to be investigated. For example, in our literature survey we identified 'The Big Five' personality dimensions. It's focus is the temperament element of individuals. This too may prove similarly successful in the prediction of team behavior upon further study. The basic premise of our research is not to promote a particular measurement tool but to demonstrate the possible effectiveness of forming teams with the consideration of personality traits. We have simply developed the heuristic using one measurement tool, the Kolbe Concept® that appeared particularly promising.

The heuristic itself had both positive and negative performance characteristics. The average deviation from the lower bound was always less than 12% in our experiments. In itself this is a positive result. It has the potential to be improved with the addition of stronger lower bounds. As we would expect, the maximum deviations from the selected measures are higher but we still, on average, stay below 22%. It is also a positive that the

10% confidence intervals were narrow, indicating that the heuristic was consistent over the range of problems we considered.

There are some situations, however, where the heuristic has the possibility of performing poorly. For instance, if the pool of potential members is significantly greater than the required number of members, the calculated weights might not be a good indicator of potential problem areas. Concentrations in each operating zone/action mode combination may provide a better route. Another potential shortcoming is the method of team member selection. They are selected only considering one measure, Goal Attainment. This could lead to arbitrarily bad solutions with respect to Profitability. Consideration of a combined measure, such as Viability, for team member selection may prove a more robust choice.

For limited applications, the Balanced Placement Heuristic provides a means of incorporating interpersonal mix into the team formation process. It does this by combining a tested personality trait measurement tool, Kolbe Concept®, with assignment techniques. This heuristic can be modified to expand its base of applications and to incorporate other performance-linked factors.

APPENDIX A - Human Subjects Committee Approval

Human Subjects Committee

THE UNIVERSITY OF
ARIZONA
HEALTH SCIENCES CENTER

1622 E. Mabel Street
PO Box 245137
Tucson AZ 85724-5137
520 626-6721

6 September 2000

Erin L. Fitzpatrick, MSIE
Advisor: Ronald G. Askin, Ph.D.
Department of Systems/Industrial Engineering
Engineering Building, Room 123
PO BOX 210020

RE: DEVELOPMENT AND EVALUATION OF A TOOL FOR FORMING EFFECTIVE WORKER-TEAMS

Dear Ms. Fitzpatrick:

We received documents concerning your above cited project. Regulations published by the U.S. Department of Health and Human Services [45 CFR Part 46.101(b)(2)] exempt this type of research from review by our Committee.

Thank you for informing us of your work. If you have any questions concerning the above, please contact this office.

Sincerely,

David G. Johnson, M.D.

David G. Johnson, M.D.
Chairman
Human Subjects Committee

DGJ/js
cc: Departmental/College Review Committee

APPENDIX B – Sample Consent Form

SUBJECT'S CONSENT FORM

Development & Evaluation of a Tool for Forming Effective Worker-Teams

I AM BEING ASKED TO READ THE FOLLOWING MATERIAL TO ENSURE THAT I AM INFORMED OF THE NATURE OF THIS RESEARCH STUDY AND OF HOW I WILL PARTICIPATE IN IT, IF I CONSENT TO DO SO. SIGNING THIS FORM WILL INDICATE THAT I HAVE BEEN SO INFORMED AND THAT I GIVE MY CONSENT. FEDERAL REGULATIONS REQUIRE WRITTEN INFORMED CONSENT PRIOR TO PARTICIPATION IN THIS RESEARCH STUDY SO THAT I CAN KNOW THE NATURE AND RISKS OF MY PARTICIPATION AND CAN DECIDE TO PARTICIPATE OR NOT PARTICIPATE IN A FREE AND INFORMED MANNER.

PURPOSE
I am being invited to participate voluntarily in the above-titled research project. The purpose of this project is to test the validity of an instinctive-behavior measuring survey in the development of a tool for forming effective worker-teams.

SELECTION CRITERIA
I am being invited to participate because this class utilizes predetermined groups for projects involving team work. Approximately 100 subjects will be enrolled in this study.

PROCEDURE(S)
If I agree to participate, I will be asked to consent to the following: (1) the completion of a questionnaire requiring 20 minutes of class time; (2) the use of my answers to the questionnaire to form groups for projects (if the questionnaire is not used the teams will still be assigned by the professor); (3) and the use of my team's project score in a statistical study. Student names or other identification will not be included in the study in any way.

RISKS
There are no known risks.

BENEFITS
There is no direct benefit from my participation. However, improvement of the way teams are formed in the workplace may be improved through the results of this study.

CONFIDENTIALITY
All data will be kept confidential. Subject names and data will be kept separate. Access to the data will be limited to the following people:
Erin L. Fitzpatrick MSIE Principal Investigator; Dr. R. G. Askin
Dept. Head Project Advisor

PARTICIPATION COSTS AND SUBJECT COMPENSATION
There are no costs or compensation involved.

CONTACTS
I can obtain further information from the principal investigator Erin L. Fitzpatrick, MSIE at (520)621-6551 or efitzpat@u.arizona.edu. If I have questions concerning my rights as a research subject, I may call the Human Subjects Committee office at (520) 626-6721.

AUTHORIZATION

Before giving my consent by signing this form, the methods, inconveniences, risks, and benefits have been explained to me and my questions have been answered. I may ask questions at any time and I am free to withdraw from the project at any time without causing bad feelings. My participation in this project may be ended by the investigator or by the sponsor for reasons that would be explained. New information developed during the course of this study which may affect my willingness to continue in this research project will be given to me as it becomes available. This consent form will be filed in an area designated by the human subjects committee with access restricted to the principal investigator, Erin L. Fitzpatrick, MSIE or authorized representative of the systems & industrial engineering department. I do not give up any of my legal rights by signing this form. A copy of this signed consent form will be given to me.

______________________________ _________________________

Subject's Signature Date

INVESTIGATOR'S AFFIDAVIT

I have carefully explained to the subject the nature of the above project. I hereby certify that to the best of my knowledge the person who is signing this consent form understands clearly the nature, demands, benefits, and risks involved in his/her participation and his/her signature is legally valid. A medical problem or language or educational barrier has not precluded this understanding.

______________________________ _________________________

Signature of Investigator Date

1/2000

APPENDIX C – Program TeamBuilder

Main Program:

```
program TeamBuilder

use TB

call Welcome ()
call GetData ()
call CalcWeights1 ()
call SelectMax1 ()
call Place1 ()
call RemoveSkGroup ()
call SelectMax1 ()
call RemoveSkGroup ()
call CalcTeamWeights ()
call SelectTeam ()
call ChooseMember ()
call Iterate ()
call Score ()

end program TeamBuilder
```

APPENDIX D – Module TB

```
module TB

 public :: Welcome, GetData, CalcWeights1, SelectMax1, Place1,RemoveSkGroup,&
        CalcTeamWeights,SelectTeam,ChooseMember,Iterate,Score

   integer, public :: emid,sgrp,tmid,b11,b12,b13,b21,b22,b23,b31,b32,b33,b41,b42,b43,&
         iostat_var,i,Nemp,m,j,k,reqteam,iter,team,totalmember
   integer, public, dimension(1:200) :: EmpID,SkGrp,TeamID,a11,a12,a13,a21,a22,a23,&
        a31,a32,a33,a41,a42,a43,SkG1,SkG2,SkG3,SkG4,Hold,D,Dp
   integer, public, dimension(1:1) ::  v11,v12,v13,v21,v22,v23,v31,v32,v33,v41,v42,v43
   real, public :: w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34,maxweight,&
              minteam,s1,s2,s3
   real, public, dimension(1:200) :: u

 contains

   subroutine Welcome ()
    open(unit=1, file="welcome.txt", status="replace", action="write", form="formatted")
    open(unit=2, file="m12x12_10.txt", status="old", action="readwrite",
form="formatted")
    print*, "Welcome to the TeamBuilder Program"
    print*, "Here are the current available personnel:"

    iter=0
    totalmember=0
    i=0
    do
     read(unit=2, fmt="(i5,14i3)", iostat=iostat_var)
emid,sgrp,tmid,b11,b12,b13,b21,b22,b23,&
      b31,b32,b33,b41,b42,b43

       if (iostat_var < 0) then
        exit
       endif
      i=i+1

      EmpID(i)=emid
      SkGrp(i)=sgrp
      TeamID(i)=tmid
      a11(i)=b11
```

```
    a12(i)=b12
    a13(i)=b13
    a21(i)=b21
    a22(i)=b22
    a23(i)=b23
    a31(i)=b31
    a32(i)=b32
    a33(i)=b33
    a41(i)=b41
    a42(i)=b42
    a43(i)=b43
    print*, EmpID(i),SkGrp(i),TeamID(i),a11(i),a12(i),a13(i),a21(i),&
         a22(i),a23(i),a31(i),a32(i),a33(i),a41(i),a42(i),a43(i)
   end do
   close(unit=1)
   close(unit=2)
  end subroutine Welcome
!*********************************************************************
  subroutine GetData()

   integer :: d1,z
   open(unit=2, file="teaminfo.txt", status="old", action="readwrite",
form="formatted",&
      position="append")
   open(unit=3, file="teamreqs.txt", status="replace", action="write", form="formatted")

write(unit=3, fmt="(a)") "GetData"

   print*, "Please enter the number of required teams:"
   read*, reqteam

  k=0
  do
   k=k+1
   if (k>reqteam) then
     exit
   endif

   print*, "For team ", k
   print*, "Please enter the number of skill group 1 required:"
   read*, SkG1(k)
   print*, "Please enter the number of skill group 2 required:"
   read*, SkG2(k)
   print*, "Please enter the number of skill group 3 required:"
```

```
   read*, SkG3(k)
   print*, "Please enter the number of skill group 4 required:"
   read*, SkG4(k)
   write (unit=3, fmt="(4i3)") SkG1(k),SkG2(k),SkG3(k),SkG4(k)
  end do

do z=1,reqteam
 totalmember=totalmember+SkG1(z)+SkG2(z)+SkG3(z)+SkG4(z)
end do

   write (unit=3, fmt="(i3)") totalmember

  do
   print*, "Would you like to create additional personnel? (y=1,n=0)"
   read*, d1
   if (d1/=1) then
    print*, "Thank-you"
    exit
   endif
    print*, "Enter employee id:"
    read*,  emid
    print*, "     skill group:"
    read*,  sgrp
    print*, " team (0 if none):"
    read*,  tmid
    print*, "           a11:"
    read*,  b11
    print*, "           a12:"
    read*,  b12
    print*, "           a13:"
    read*,  b13
    print*, "           a21:"
    read*,  b21
    print*, "           a22:"
    read*,  b22
    print*, "           a23:"
    read*,  b23
    print*, "           a31:"
    read*,  b31
    print*, "           a32:"
    read*,  b32
    print*, "           a33:"
    read*,  b33
    print*, "           a41:"
```

```
    read*,  b41
    print*, "          a42:"
    read*,  b42
    print*, "          a43:"
    read*,  b43

    i=i+1

     EmpID(i)=emid
     SkGrp(i)=sgrp
     TeamID(i)=tmid
     a11(i)=b11
     a12(i)=b12
     a13(i)=b13
     a21(i)=b21
     a22(i)=b22
     a23(i)=b23
     a31(i)=b31
     a32(i)=b32
     a33(i)=b33
     a41(i)=b41
     a42(i)=b42
     a43(i)=b43

     print*, i,EmpID(i),SkGrp(i),TeamID(i),a11(i),a12(i),a13(i),a21(i),&
          a22(i),a23(i),a31(i),a32(i),a33(i),a41(i),a42(i),a43(i)
     write (unit=2, fmt="(i5,14i3)") EmpID(i),SkGrp(i),TeamID(i),a11(i),&
          a12(i),a13(i),a21(i),&
          a22(i),a23(i),a31(i),a32(i),a33(i),a41(i),a42(i),a43(i)
  end do
  close (unit=2)
 Nemp=i
  end subroutine GetData
!*****************************************************************
subroutine CalcWeights1 ()

integer :: count1, count2, count3, count4

write(unit=3, fmt="(a)") "CalcWeights1"

 count1=0
 count2=0
 count3=0
 count4=0
```

```
 i=0
 do
  i=i+1
  if(i>Nemp) then
     exit
  endif
   if(TeamID(i)==0)then
       if(SkGrp(i)==1) then
         w11=w11+a11(i)+a21(i)+a31(i)+a41(i)
         w21=w21+a12(i)+a22(i)+a32(i)+a42(i)
         w31=w31+a13(i)+a23(i)+a33(i)+a43(i)
         count1=count1+1
         else if (SkGrp(i)==2) then
         w12=w12+a11(i)+a21(i)+a31(i)+a41(i)
         w22=w22+a12(i)+a22(i)+a32(i)+a42(i)
         w32=w32+a13(i)+a23(i)+a33(i)+a43(i)
         count2=count2+1
         else if (SkGrp(i)==3) then
         w13=w13+a11(i)+a21(i)+a31(i)+a41(i)
         w23=w23+a12(i)+a22(i)+a32(i)+a42(i)
         w33=w33+a13(i)+a23(i)+a33(i)+a43(i)
         count3=count3+1
         else
         w14=w14+a11(i)+a21(i)+a31(i)+a41(i)
         w24=w24+a12(i)+a22(i)+a32(i)+a42(i)
         w34=w34+a13(i)+a23(i)+a33(i)+a43(i)
         count4=count4+1
    endif
   endif
  end do
i=i-1

if ((count1+count2+count3+count4)==Nemp) then
s1=0
s2=0
s3=0
s1=w11+w12+w13+w14
s2=w21+w22+w23+w24
s3=w31+w32+w33+w34
end if

 if(count1==0) then
      w11=0
```

```
      w21=0
      w31=0
 else
      w11=w11/count1
      w21=0.5*w21/count1
      w31=w31/count1
endif
if(count2==0) then
      w12=0
      w22=0
      w32=0
 else
      w12=w12/count2
      w22=0.5*w22/count2
      w32=w32/count2
endif
if(count3==0) then
      w13=0
      w23=0
      w33=0
 else
      w13=w13/count3
      w23=0.5*w23/count3
      w33=w33/count3
endif
if(count4==0) then
      w14=0
      w24=0
      w34=0
 else
      w14=w14/count4
      w24=0.5*w24/count4
      w34=w34/count4
endif

print*, "The weights are: ", w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34
write(unit=3, fmt="(a)") "The weights are:"
write(unit=3, fmt="(12f7.2)") w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34

end subroutine CalcWeights1
!*********************************************************************
subroutine SelectMax1 ()

write(unit=3, fmt="(a)") "SelectMax1"
```

```
maxweight=max(w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34)
print*, "The selected weight is:",maxweight
write(unit=3, fmt="(a)") "The selected weight is:"
write(unit=3, fmt="(f7.2)") maxweight

if (maxweight==w11) then
   m=1
   j=1
  else if (maxweight==w12) then
   m=1
   j=2
  else if (maxweight==w13) then
   m=1
   j=3
  else if(maxweight==w14) then
   m=1
   j=4
  else if (maxweight==w21) then
   m=2
   j=1
  else if (maxweight==w22) then
   m=2
   j=2
  else if (maxweight==w23) then
   m=2
   j=3
  else if (maxweight==w24) then
   m=2
   j=4
  else if (maxweight==w31) then
   m=3
   j=1
  else if (maxweight==w32) then
   m=3
   j=2
  else if (maxweight==w33) then
   m=3
   j=3
  else
   m=3
   j=4
end if
```

```
print*, "It belongs to operating zone, skill group:",m,j
write(unit=3, fmt="(a)") "The operating zone, skill group:"
write(unit=3, fmt="(2i3)") m,j
iter=iter+1
Hold(iter)=j

end subroutine SelectMax1
!*******************************************************************
subroutine Place1 ()

integer :: q,p
write(unit=3, fmt="(a)") "Place1"

if (j==1) then
 q=0
 do
 q=q+1
  if (q>reqteam) then
     exit
  end if
  p=0
  do
  p=p+1
    if (p>Nemp) then
      print*, "There is a shortage of skill 1."
      exit
    end if
    if (SkG1(q)>0) then
      if (SkGrp(p)==1) then
        if (TeamID(p)==0) then
          TeamID(p)=q
write(unit=3, fmt="(a, i3,a)") "The team for", p, " is"
write(unit=3, fmt="(i3)") TeamID(p)
          exit
        end if
      end if
    else
      exit
    end if
  end do
 end do

else if (j==2) then
 q=0
```

```
 do
 q=q+1
  if (q>reqteam) then
     exit
  end if
  p=0
  do
  p=p+1
    if (p>Nemp) then
       print*, "There is a shortage of skill 2."
       exit
    end if
    if (SkG2(q)>0) then
      if (SkGrp(p)==2) then
        if (TeamID(p)==0) then
          TeamID(p)=q
write(unit=3, fmt="(a, i3,a)") "The team for", p, " is"
write(unit=3, fmt="(i3)") TeamID(p)
          exit
        end if
      end if
    else
      exit
    end if
  end do
 end do

else if (j==3) then
 q=0
 do
 q=q+1
  if (q>reqteam) then
     exit
  end if
  p=0
  do
  p=p+1
    if (p>Nemp) then
       print*, "There is a shortage of skill 3."
       exit
    end if
    if (SkG3(q)>0) then
      if (SkGrp(p)==3) then
        if (TeamID(p)==0) then
```

```
          TeamID(p)=q
write(unit=3, fmt="(a, i3,a)") "The team for", p, " is"
write(unit=3, fmt="(i3)") TeamID(p)
          exit
        end if
      end if
    else
      exit
    end if
  end do
 end do

else
 q=0
 do
 q=q+1
  if (q>reqteam) then
    exit
  end if
  p=0
  do
  p=p+1
   if (p>Nemp) then
     print*, "There is a shortage of skill 4."
     exit
   end if
   if (SkG4(q)>0) then
     if (SkGrp(p)==4) then
       if (TeamID(p)==0) then
         TeamID(p)=q
write(unit=3, fmt="(a, i3,a)") "The team for", p, " is"
write(unit=3, fmt="(i3)") TeamID(p)
         exit
       end if
     end if
   else
     exit
   end if
  end do
 end do
end if
p=0
do
p=p+1
```

```
  if (p>Nemp) then
    exit
  end if
print*,"Placed in",TeamID(p)
end do

end subroutine Place1
!*****************************************************************
```

subroutine RemoveSkGroup ()

```
integer :: p
write(unit=3, fmt="(a)") "RemoveSkGroup"

p=0
do
p=p+1
 if (p>iter) then
   exit
 end if
 if (Hold(p)==1) then
   w11=0
   w21=0
   w31=0
  else if (Hold(p)==2) then
   w12=0
   w22=0
   w32=0
  else if (Hold(p)==3) then
   w13=0
   w23=0
   w33=0
  else if (Hold(p)==4) then
   w14=0
   w24=0
   w34=0
  else
end if
end do

print*, w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34
write(unit=3, fmt="(a)") "The current weights are:"
write(unit=3, fmt="(12f6.2)") w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34

end subroutine RemoveSkGroup
```

```
!*******************************************************************
subroutine CalcTeamWeights ()

integer :: k,p
integer, dimension(1:200) :: counter
write(unit=3, fmt="(a)") "CalcTeamWeights"

k=0
do
k=k+1
  if (k>reqteam) then
     exit
  end if
  counter(k)=0
end do
p=0
do
p=p+1
  if (p>Nemp) then
    exit
  end if
  k=0
  do
  k=k+1
    if (k>reqteam) then
      exit
    end if
    if (TeamID(p)/=0) then
      if(TeamID(p)==k) then
        counter(k)=counter(k)+1
        if(m==1) then
          u(k)=u(k)+a11(p)+a21(p)+a31(p)+a41(p)
        else if (m==2) then
          u(k)=u(k)+a12(p)+a22(p)+a32(p)+a42(p)
        else
          u(k)=u(k)+a13(p)+a23(p)+a33(p)+a43(p)
        end if
      else
      end if
    end if
  end do
end do

k=0
```

```
do
k=k+1
 if (k>reqteam) then
  exit
 endif
 if (counter(k)==0) then
  u(k)=0
 else
  u(k)=u(k)/(SkG1(k)+SkG2(k)+SkG3(k)+SkG4(k))
 end if
end do

write(unit=3, fmt="(a)") "The team weights are:"

do k=1, reqteam
 print*, "The team weights are:", u(k)
 write(unit=3, fmt="(f7.2)") u(k)
end do

end subroutine CalcTeamWeights
!*********************************************************************
subroutine SelectTeam ()

integer :: z
write(unit=3, fmt="(a)") "SelectTeam"

minteam=minval(u(1:reqteam))
print*, "The lowest team weight is: ",minteam
write(unit=3, fmt="(a)") "The lowest team weight is"
write(unit=3, fmt="(f7.2)") minteam

do z=1,reqteam
 if (minteam==u(z)) then
   team=z
   print*, "team is", team
write(unit=3, fmt="(a)") "The team is"
write(unit=3, fmt="(i3)") team
   exit
 end if
end do

end subroutine SelectTeam
!*********************************************************************
subroutine ChooseMember ()
```

```
integer :: p,d11,d21,d31,d41,d12,d22,d32,d42,d13,d23,d33,d43,&
        maxdev,member,memberval,q,z,temp1,temp2
integer, dimension(1:36) :: Match
write(unit=3, fmt="(a)") "ChooseMember"

  v11(1)=0
  v12(1)=0
  v13(1)=0
  v21(1)=0
  v22(1)=0
  v23(1)=0
  v31(1)=0
  v32(1)=0
  v33(1)=0
  v41(1)=0
  v42(1)=0
  v43(1)=0
p=1
do p=1,Nemp
 if (TeamID(p)==team) then
  v11(1)=v11(1)+a11(p)
  v12(1)=v12(1)+a12(p)
  v13(1)=v13(1)+a13(p)
  v21(1)=v21(1)+a21(p)
  v22(1)=v22(1)+a22(p)
  v23(1)=v23(1)+a23(p)
  v31(1)=v31(1)+a31(p)
  v32(1)=v32(1)+a32(p)
  v33(1)=v33(1)+a33(p)
  v41(1)=v41(1)+a41(p)
  v42(1)=v42(1)+a42(p)
  v43(1)=v43(1)+a43(p)
 else
 end if
end do

write(unit=3, fmt="(a)") "The team sums are"
write(unit=3, fmt="(12i4)") v11,v12,v13,v21,v22,v23,v31,v32,v33,v41,v42,v43

p=1
do p=1,Nemp
if (TeamID(p)==0) then
 if (SkGrp(p)==j) then
```

```
   d11=abs(v11(1)-a11(p))
   d21=abs(v21(1)-a21(p))
   d31=abs(v31(1)-a31(p))
   d41=abs(v41(1)-a41(p))
   d12=2*abs(v12(1)-a12(p))
   d22=2*abs(v22(1)-a22(p))
   d32=2*abs(v32(1)-a32(p))
   d42=2*abs(v42(1)-a42(p))
   d13=abs(v13(1)-a13(p))
   d23=abs(v23(1)-a23(p))
   d33=abs(v33(1)-a33(p))
   d43=abs(v43(1)-a43(p))
   D(p)=d11+d21+d31+d41+d12+d22+d32+d42+d13+d23+d33+d43
   Dp(p)=max(d11,d21,d31,d41,d12,d22,d32,d42,d13,d23,d33,d43)
print*, "The total deviation for",p,"is",D(p)
print*, "The max deviation is",Dp(p)
write(unit=3, fmt="(a,i3,a,i4)") "The total deviation for ",p," is",D(p)

write(unit=3, fmt="(a)") "The single max deviation is"
write(unit=3, fmt="(i4)") Dp(p)

 else
  D(p)=0
  Dp(p)=0
 end if
else
 D(p)=0
 Dp(p)=0
end if
end do
maxdev=maxval(D)
write(unit=3, fmt="(a, i4)") "maxdev=", maxdev
write(unit=3, fmt="(a)") "The maximum total deviation is"
write(unit=3, fmt="(i4)") maxdev

Match=0
q=0
p=1
do p=1,Nemp
 if (SkGrp(p)==j) then
  if (D(p)==maxdev) then
    q=q+1
    Match(q)=p
write(unit=3, fmt="(a,i4,a,i4)") "q=",q, "p=",p
```

```
  end if
 end if
end do
print*, Match(1:q)

member=0
print*, q

 if (q==1) then
   member=Match(q)
print*, "q=1"
 else
 temp2=999

do p=1,Nemp
  if (member/=0) then
     exit
  end if
   if (SkGrp(p)==j) then
    do z=1,q
     if (p==Match(z)) then
       temp1=Dp(p)
     if (temp1<temp2) then
       temp2=Dp(p)
     end if
     end if
    end do
   end if
end do

memberval=temp2
print*, "memberval=", memberval
   do p=1,Nemp
     if (member/=0) then
      exit
     end if
      do z=1,q
      if (p==Match(z)) then
         if (Dp(p)==memberval) then
            member=p
            exit
         end if
      end if
      end do
```

```
  end do
 end if

write(unit=3, fmt="(a)") "The member chosen is"
write(unit=3, fmt="(i3)") member
TeamID(member)=team
write(unit=3, fmt="(a)") "and placed in team"
write(unit=3, fmt="(12i4)") team
print*, "The selected new member is", member,"for team",team
u(team)=9999

end subroutine ChooseMember
!*******************************************************************
```

subroutine Iterate ()

```
integer :: placed,z,p

open(unit=4, file="final.txt", status="replace", action="write", form="formatted")

do z=1,300
write(unit=3, fmt="(a)") "Iterate"
placed=count(TeamID/=0)
write(unit=3, fmt="(i3)") placed
if (placed/=totalmember) then
  if (minval(u(1:reqteam))==9999) then
   u(1:reqteam)=0
   if (max(w11,w12,w13,w14,w21,w22,w23,w24,w31,w32,w33,w34)/=0) then
     call SelectMax1 ()
     call RemoveSkGroup ()
     call CalcTeamWeights ()
     call SelectTeam ()
     call ChooseMember ()
   else
     iter=0
     Hold=-1
     call CalcWeights1 ()
     call SelectMax1 ()
     call RemoveSkGroup ()
     call CalcTeamWeights ()
     call SelectTeam ()
     call ChooseMember ()
   end if
  else
   call SelectTeam ()
```

```
    call ChooseMember ()
   end if
 end if
 end do

 do p=1,Nemp
  if (TeamID(p)/=0) then
 write(unit=4, fmt="(a)") "EmpID, SkGrp, TeamID"
 write(unit=4, fmt="(3i4)") EmpID(p),SkGrp(p),TeamID(p)
  end if
 print*, EmpID(p),SkGrp(p),TeamID(p)
 end do

 close(unit=4)


 end subroutine Iterate
!*******************************************************************
subroutine Score ()

  integer :: p,q
  real :: x11,x21,x31,x41,x12,x22,x32,x42,x13,x23,x33,x43,bound,bound1,bound2,&
   maxbnd1,maxbnd2,maxbnd,avgbnd1,avgbnd2,avgbnd
  integer, dimension(1:200) :: counter
  real, dimension(1:200) :: v11,v21,v31,v41,v12,v22,v32,v42,v13,v23,v33,v43,&
        bnd1,bnd2,syn1,syn2,syn3,std2
write(unit=3, fmt="(a)") "Score"

open(unit=6, file="score.txt", status="old", action="readwrite", form="formatted",&
      position="append")


  s1=s1/(Nemp*4)
  s2=s2/(Nemp*4)
  s3=s3/(Nemp*4)
  x11=0
  x12=0
  x13=0
  x21=0
  x22=0
  x23=0
  x31=0
  x32=0
  x33=0
```

```
  x41=0
  x42=0
  x43=0
p=1
do p=1,Nemp
  x11=x11+a11(p)
  x12=x12+a12(p)
  x13=x13+a13(p)
  x21=x21+a21(p)
  x22=x22+a22(p)
  x23=x23+a23(p)
  x31=x31+a31(p)
  x32=x32+a32(p)
  x33=x33+a33(p)
  x41=x41+a41(p)
  x42=x42+a42(p)
  x43=x43+a43(p)
end do

  x11=x11/Nemp-0.3
 if (x11<0) then
  x11=0
 end if
  x12=x12/Nemp-0.6
 if (x12<0) then
  x12=0
 end if
  x13=x13/Nemp-0.3
 if (x13<0) then
  x13=0
 end if
  x21=x21/Nemp-0.3
 if (x21<0) then
  x21=0
 end if
  x22=x22/Nemp-0.6
 if (x22<0) then
  x22=0
 end if
  x23=x23/Nemp-0.3
 if (x23<0) then
  x23=0
 end if
  x31=x31/Nemp-0.3
```

```
 if (x31<0) then
  x31=0
 end if
  x32=x32/Nemp-0.6
 if (x32<0) then
  x32=0
 end if
  x33=x33/Nemp-0.3
 if (x33<0) then
  x33=0
 end if
  x41=x41/Nemp-0.3
 if (x41<0) then
  x41=0
 end if
  x42=x42/Nemp-0.6
 if (x42<0) then
  x42=0
 end if
  x43=x43/Nemp-0.3
 if (x43<0) then
  x43=0
 end if

bound1=abs(0.25-s1)+(0.5-s2)+(0.25-s3)
bound2=x11/4+x21/4+x31/4+x41/4+x12/4+x22/4+x32/4+x42/4+x13/4+x23/4+x33/4+x4
3/4
bound=0.55*bound1+0.45*bound2

syn1=0
syn2=0
syn3=0
counter=0

do p=1,Nemp
 do q=1,reqteam
  if(TeamID(p)==q)then
    syn1(q)=syn1(q)+a11(p)+a21(p)+a31(p)+a41(p)
    syn2(q)=syn2(q)+a12(p)+a22(p)+a32(p)+a42(p)
    syn3(q)=syn3(q)+a13(p)+a23(p)+a33(p)+a43(p)
    counter(q)=counter(q)+1
    exit
  end if
 end do
```

```
end do

do q=1,reqteam
  syn1(q)=syn1(q)/(counter(q)*4)
  syn2(q)=syn2(q)/(counter(q)*4)
  syn3(q)=syn3(q)/(counter(q)*4)
  bnd1(q)=abs(0.25-syn1(q))+abs(0.5-syn2(q))+abs(0.25-syn3(q))
  maxbnd1=maxval(bnd1(1:reqteam))
end do
  avgbnd1=sum(bnd1(1:reqteam))/reqteam

v11=0
v12=0
v13=0
v21=0
v22=0
v23=0
v31=0
v32=0
v33=0
v41=0
v42=0
v43=0
counter=0

p=1
do p=1,Nemp
 do q=1,reqteam
 if (TeamID(p)==q) then
  v11(q)=v11(q)+a11(p)
  v12(q)=v12(q)+a12(p)
  v13(q)=v13(q)+a13(p)
  v21(q)=v21(q)+a21(p)
  v22(q)=v22(q)+a22(p)
  v23(q)=v23(q)+a23(p)
  v31(q)=v31(q)+a31(p)
  v32(q)=v32(q)+a32(p)
  v33(q)=v33(q)+a33(p)
  v41(q)=v41(q)+a41(p)
  v42(q)=v42(q)+a42(p)
  v43(q)=v43(q)+a43(p)
  counter(q)=counter(q)+1
  exit
 end if
```

```
  end do
end do

do q=1,reqteam
   v11(q)=v11(q)/counter(q)-0.3
  if (v11(q)<0) then
   v11(q)=0
  end if
   v12(q)=v12(q)/counter(q)-0.6
  if (v12(q)<0) then
   v12(q)=0
  end if
   v13(q)=v13(q)/counter(q)-0.3
  if (v13(q)<0) then
   v13(q)=0
  end if
   v21(q)=v21(q)/counter(q)-0.3
  if (v21(q)<0) then
   v21(q)=0
  end if
   v22(q)=v22(q)/counter(q)-0.6
  if (v22(q)<0) then
   v22(q)=0
  end if
   v23(q)=v23(q)/counter(q)-0.3
  if (v23(q)<0) then
   v23(q)=0
  end if
   v31(q)=v31(q)/counter(q)-0.3
  if (v31(q)<0) then
   v31(q)=0
  end if
   v32(q)=v32(q)/counter(q)-0.6
  if (v32(q)<0) then
   v32(q)=0
  end if
   v33(q)=v33(q)/counter(q)-0.3
  if (v33(q)<0) then
   v33(q)=0
  end if
   v41(q)=v41(q)/counter(q)-0.3
  if (v41(q)<0) then
   v41(q)=0
  end if
```

```
  v42(q)=v42(q)/counter(q)-0.6
 if (v42(q)<0) then
  v42(q)=0
 end if
  v43(q)=v43(q)/counter(q)-0.3
 if (v43(q)<0) then
  v43(q)=0
 end if
bnd2(q)=v11(q)/4+v21(q)/4+v31(q)/4+v41(q)/4+v12(q)/4+v22(q)/4+v32(q)/4+&
     v42(q)/4+v13(q)/4+v23(q)/4+v33(q)/4+v43(q)/4
end do

maxbnd2=maxval(bnd2(1:reqteam))
avgbnd2=sum(bnd2(1:reqteam))/reqteam

maxbnd=0.55*maxbnd1+0.45*maxbnd2
avgbnd=0.55*avgbnd1+0.45*avgbnd2
write(unit=6, fmt="(a, 3f8.4)") "bounds trial 12x12 ", bound1, bound2, bound
write(unit=6, fmt="(t4, a, 3f8.4)") "actualmax 12x12 ", maxbnd1, maxbnd2, maxbnd
write(unit=6, fmt="(t4, a, 3f8.4)") "actualavg 12x12 ", avgbnd1, avgbnd2, avgbnd

print*, " "
close(unit=3)
end subroutine Score

!*********************************************************************
end module TB
```

APPENDIX E – Program Members

Member generation program:

```
program Members

use Generate

call RandNumbers()

end program Members
```

APPENDIX F – Module Generate

```
module Generate

public :: RandNumbers

real, public :: rand1, rand2, rand3, rand4
integer, public :: FF, FT, QS, IM

contains

 subroutine RandNumbers ()

 integer :: k, emid, sgrp, tmid, a11, a12, a13,&
        a21, a22, a23, a31, a32, a33, a41, a42, a43, counter,N
 real :: norm

 open(unit=5, file="member.txt", status="replace", action="write", form="formatted")


 emid=1001
 sgrp=1
 tmid=0
 counter=0

 do k=1,1728

 if (sgrp==5) then
    sgrp=1
 end if

 call random_number(rand1)
 rand1=10*rand1
 call random_number(rand2)
 rand2=10*rand2
 call random_number(rand3)
 rand3=10*rand3
 call random_number(rand4)
 rand4=10*rand4

 norm=rand1+rand2+rand3+rand4
 rand1=20*(rand1/norm)
```

```
rand2=20*(rand2/norm)
rand3=20*(rand3/norm)
rand4=20*(rand4/norm)
FF=int(rand1+0.5)
FT=int(rand2+0.5)
QS=int(rand3+0.5)
IM=int(rand4+0.5)


if ((FF<11).and.(FF>0)) then
if ((FT<11).and.(FT>0)) then
if ((QS<11).and.(QS>0)) then
if ((IM<11).and.(IM>0)) then

counter=counter+1

if (counter==1446) then
  exit
endif

if (FF<4) then
  a11=1
  a12=0
  a13=0
else if (FF<7) then
  a11=0
  a12=1
  a13=0
else
  a11=0
  a12=0
  a13=1
end if

if (FT<4) then
  a21=1
  a22=0
  a23=0
else if (FT<7) then
  a21=0
  a22=1
  a23=0
else
  a21=0
```

```
    a22=0
    a23=1
  end if

  if (QS<4) then
    a31=1
    a32=0
    a33=0
  else if (QS<7) then
    a31=0
    a32=1
    a33=0
  else
    a31=0
    a32=0
    a33=1
  end if

  if (IM<4) then
    a41=1
    a42=0
    a43=0
  else if (IM<7) then
    a41=0
    a42=1
    a43=0
  else
    a41=0
    a42=0
    a43=1
  end if


  write (unit=5, fmt="(i5,14i3)") emid, sgrp, tmid, a11, a12, a13,&
       a21, a22, a23, a31, a32, a33, a41, a42, a43

  emid=emid+1
if (emid>1144) then
  emid=1001
end if
  sgrp=sgrp+1

  end if
  end if
```

```
  end if
  end if


  end do

  close(unit=5)

  end subroutine RandNumbers

end module Generate
```

APPENDIX G – Sample Output: Member

1001	1	0	1	0	0	0	0	1	1	0	0	0	0	1
1002	2	0	0	0	1	1	0	0	0	0	1	1	0	0
1003	1	0	0	1	0	1	0	0	0	0	1	1	0	0
1004	2	0	0	0	1	1	0	0	0	0	1	1	0	0
1005	1	0	0	0	1	0	1	0	0	1	0	0	0	1
1006	2	0	1	0	0	1	0	0	0	1	0	0	0	1

APPENDIX H – Sample Output: Final

EmpID, SkGrp, TeamID

1001	1	1
1002	2	1
1004	2	2
1005	1	2

APPENDIX I – Sample Output: Score

bounds	0.1250	0.0656	0.0983
actualmax	0.5000	0.2000	0.3650
actualavg	0.3125	0.1750	0.2506

APPENDIX K – Sample Output: Trace

GetData
1 1 0 0
1 1 0 0
4
CalcWeights1
The weights are:
1.33 2.00 0.00 0.00 0.50 0.17 0.00 0.00 1.67 1.67 0.00 0.00
SelectMax1
The selected weight is:
2.00
The operating zone, skill group:
1 2
Place1
The team for 2 is
1
The team for 4 is
2
RemoveSkGroup
The current weights are:
1.33 0.00 0.00 0.00 0.50 0.00 0.00 0.00 1.67 0.00 0.00 0.00
SelectMax1
The selected weight is:
1.67
The operating zone, skill group:
3 1
RemoveSkGroup
The current weights are:
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
CalcTeamWeights
The team weights are:
1.00
1.00
SelectTeam
The lowest team weight is
1.00
The team is
1
ChooseMember
The team sums are
0 0 1 1 0 0 0 0 1 1 0 0

The total deviation for 1 is 8
The single max deviation is
1
The total deviation for 3 is 3
The single max deviation is
2
The total deviation for 5 is 8
The single max deviation is
2
maxdev= 8
The maximum total deviation is
8
q= 1p= 1
q= 2p= 5
The member chosen is
1
and placed in team
1
Iterate
SelectTeam
The lowest team weight is
1.00
The team is
2
ChooseMember
The team sums are
0 0 1 1 0 0 0 0 1 1 0 0
The total deviation for 3 is 3
The single max deviation is
2
The total deviation for 5 is 8
The single max deviation is
2
maxdev= 8
The maximum total deviation is
8
q= 1p= 5
The member chosen is
5
and placed in team
2
Iterate
Score

APPENDIX K – Raw Output: Test Problems

	Synergy	Inertia	Weighted
bounds trial6x4	0.0833	0.2167	0.1433
actualmax 6x4	0.375	0.375	0.375
actualavg 6x4	0.1875	0.2562	0.2184
bounds trial6x4	0.0833	0.2167	0.1433
actualmax 6x4	0.2857	0.3393	0.3098
actualavg 6x4	0.1771	0.2579	0.2134
bounds trial4x4	0.125	0.0656	0.0983
actualmax 4x4	0.5	0.2	0.365
actualavg 4x4	0.3125	0.175	0.2506
bounds trial4x4	0	0.05	0.0225
actualmax 4x4	0.25	0.1875	0.2219
actualavg 4x4	0.2188	0.1281	0.178
bounds trial4x4	0.125	0.1187	0.1222
actualmax 4x4	0.5	0.2375	0.3819
actualavg 4x4	0.25	0.1719	0.2148
bounds trial4x4	0	0.0656	0.0295
actualmax 4x4	0.25	0.1875	0.2219
actualavg 4x4	0.125	0.125	0.125
bounds trial4x4	0.125	0.0656	0.0983
actualmax 4x4	0.375	0.2	0.2962
actualavg 4x4	0.25	0.1469	0.2036
bounds trial4x4	0.1563	0.0656	0.1155
actualmax 4x4	0.375	0.2375	0.3131
actualavg 4x4	0.25	0.1687	0.2134
bounds trial4x4	0	0.0344	0.0155
actualmax 4x4	0.125	0.1375	0.1306
actualavg 4x4	0.0313	0.1031	0.0636
bounds trial4x4	0.125	0.0312	0.0828
actualmax 4x4	0.375	0.15	0.2738
actualavg 4x4	0.25	0.1094	0.1867
bounds trial4x4	0.0625	0.0469	0.0555
actualmax 4x4	0.25	0.2125	0.2331
actualavg 4x4	0.2188	0.1406	0.1836
bounds trial4x4	0.1875	0.1437	0.1678
actualmax 4x4	0.625	0.2625	0.4619

actualavg 4x4	0.4063	0.2156	0.3205
bounds trial4x8	0.0625	0.0391	0.052
actualmax 4x8	0.25	0.1375	0.1994
actualavg 4x8	0.1875	0.1109	0.153
bounds trial4x8	0.0625	0.0484	0.0562
actualmax 4x8	0.3125	0.1437	0.2366
actualavg 4x8	0.1563	0.1203	0.1401
bounds trial4x8	0.1406	0.0469	0.0984
actualmax 4x8	0.3125	0.1875	0.2562
actualavg 4x8	0.25	0.1281	0.1952
bounds trial4x8	0.0625	0.0203	0.0435
actualmax 4x8	0.25	0.1437	0.2022
actualavg 4x8	0.1406	0.1266	0.1343
bounds trial4x8	0.125	0.0828	0.106
actualmax 4x8	0.375	0.1687	0.2822
actualavg 4x8	0.3125	0.1453	0.2373
bounds trial4x8	0.1094	0.0672	0.0904
actualmax 4x8	0.375	0.1562	0.2766
actualavg 4x8	0.1875	0.1141	0.1545
bounds trial4x8	0.1094	0.0312	0.0742
actualmax 4x8	0.375	0.1312	0.2653
actualavg 4x8	0.2031	0.0922	0.1532
bounds trial4x8	0.0469	0.1016	0.0715
actualmax 4x8	0.4375	0.25	0.3531
actualavg 4x8	0.1719	0.1422	0.1585
bounds trial4x8	0.0625	0.0516	0.0576
actualmax 4x8	0.3125	0.1562	0.2422
actualavg 4x8	0.1563	0.1016	0.1316
bounds trial4x8	0	0.0219	0.0098
actualmax 4x8	0.25	0.15	0.205
actualavg 4x8	0.0938	0.0906	0.0923
bounds trial 4x12	0.0833	0.0542	0.0702
actualmax 4x12	0.3333	0.1458	0.249
actualavg 4x12	0.2188	0.1177	0.1733
bounds trial 4x12	0.0938	0.0333	0.0666
actualmax 4x12	0.3333	0.1542	0.2527

actualavg	4x12	0.1875	0.0979	0.1472
bounds trial	4x12	0.0625	0.0146	0.0409
actualmax	4x12	0.2917	0.1083	0.2092
actualavg	4x12	0.1875	0.0854	0.1416
bounds trial	4x12	0.1354	0.0802	0.1106
actualmax	4x12	0.375	0.1958	0.2944
actualavg	4x12	0.25	0.15	0.205
bounds trial	4x12	0.0833	0.0302	0.0594
actualmax	4x12	0.2917	0.1375	0.2223
actualavg	4x12	0.1667	0.0927	0.1334
bounds trial	4x12	0.0625	0.0542	0.0587
actualmax	4x12	0.2917	0.15	0.2279
actualavg	4x12	0.1667	0.1156	0.1437
bounds trial	4x12	0.0104	0.0031	0.0071
actualmax	4x12	0.2917	0.1042	0.2073
actualavg	4x12	0.1875	0.0604	0.1303
bounds trial	4x12	0.0521	0.0354	0.0446
actualmax	4x12	0.3333	0.125	0.2396
actualavg	4x12	0.1667	0.0781	0.1268
bounds trial	4x12	0.0625	0.0167	0.0419
actualmax	4x12	0.1667	0.1125	0.1423
actualavg	4x12	0.1146	0.0677	0.0935
bounds trial	4x12	0.0417	0.0354	0.0389
actualmax	4x12	0.25	0.1583	0.2087
actualavg	4x12	0.1042	0.0948	0.0999
bounds trial	8x4	0.0625	0.0391	0.052
actualmax	8x4	0.375	0.1875	0.2906
actualavg	8x4	0.1875	0.1219	0.158
bounds trial	8x4	0.0625	0.0484	0.0562
actualmax	8x4	0.375	0.1875	0.2906
actualavg	8x4	0.1719	0.1391	0.1571
bounds trial	8x4	0.1406	0.0469	0.0984
actualmax	8x4	0.375	0.2625	0.3244
actualavg	8x4	0.2656	0.1375	0.208
bounds trial	8x4	0.0625	0.0203	0.0435
actualmax	8x4	0.25	0.175	0.2162

actualavg 8x4		0.1719	0.1187	0.148
bounds trial 8x4		0.125	0.0828	0.106
actualmax 8x4		0.5	0.2	0.365
actualavg 8x4		0.3125	0.1578	0.2429
bounds trial 8x4		0.1094	0.0672	0.0904
actualmax 8x4		0.5	0.2625	0.3931
actualavg 8x4		0.2656	0.1516	0.2143
bounds trial 8x4		0.1094	0.0312	0.0742
actualmax 8x4		0.375	0.25	0.3187
actualavg 8x4		0.2031	0.1516	0.1799
bounds trial 8x4		0.0469	0.1016	0.0715
actualmax 8x4		0.375	0.2125	0.3019
actualavg 8x4		0.2031	0.1656	0.1863
bounds trial 8x4		0.0625	0.0516	0.0576
actualmax 8x4		0.375	0.2	0.2962
actualavg 8x4		0.1875	0.1453	0.1685
bounds trial 8x4		0	0.0219	0.0098
actualmax 8x4		0.25	0.1625	0.2106
actualavg 8x4		0.1458	0.1026	0.1264
bounds trial	8x8	0.0625	0.032	0.0488
actualmax	8x8	0.4375	0.1875	0.325
actualavg	8x8	0.2031	0.118	0.1648
bounds trial	8x8	0.1016	0.0203	0.065
actualmax	8x8	0.375	0.175	0.285
actualavg	8x8	0.2188	0.1148	0.172
bounds trial	8x8	0.1172	0.0594	0.0912
actualmax	8x8	0.4375	0.175	0.3194
actualavg	8x8	0.25	0.1289	0.1955
bounds trial	8x8	0.0781	0.0484	0.0648
actualmax	8x8	0.3125	0.1562	0.2422
actualavg	8x8	0.2031	0.1055	0.1592
bounds trial	8x8	0.0313	0.0289	0.0302
actualmax	8x8	0.1875	0.175	0.1819
actualavg	8x8	0.1172	0.0789	0.1
bounds trial	8x8	0.0469	0.0031	0.0272
actualmax	8x8	0.3125	0.1437	0.2366

actualavg	8x8	0.1328	0.0852	0.1114
bounds trial	8x8	0.0313	0.0102	0.0218
actualmax	8x8	0.3125	0.1875	0.2562
actualavg	8x8	0.2031	0.0953	0.1546
bounds trial	8x8	0.0547	0.0219	0.0399
actualmax	8x8	0.25	0.125	0.1937
actualavg	8x8	0.125	0.0859	0.1074
bounds trial	8x8	0.0391	0.0289	0.0345
actualmax	8x8	0.25	0.1562	0.2078
actualavg	8x8	0.1406	0.0898	0.1178
bounds trial	8x8	0.0703	0.0172	0.0464
actualmax	8x8	0.3125	0.1437	0.2366
actualavg	8x8	0.2031	0.1078	0.1602
bounds trial	8x12	0.0885	0.0255	0.0602
actualmax	8x12	0.375	0.1458	0.2719
actualavg	8x12	0.2187	0.1208	0.1747
bounds trial	8x12	0.099	0.0276	0.0668
actualmax	8x12	0.4167	0.1583	0.3004
actualavg	8x12	0.2188	0.1125	0.1709
bounds trial	8x12	0.0729	0.0359	0.0563
actualmax	8x12	0.4167	0.1917	0.3154
actualavg	8x12	0.1927	0.1125	0.1566
bounds trial	8x12	0.0313	0	0.0172
actualmax	8x12	0.25	0.1458	0.2031
actualavg	8x12	0.1302	0.0766	0.1061
bounds trial	8x12	0.0521	0.0141	0.035
actualmax	8x12	0.2917	0.125	0.2167
actualavg	8x12	0.1146	0.0896	0.1033
bounds trial	8x12	0.0313	0.0094	0.0214
actualmax	8x12	0.2083	0.25	0.2271
actualavg	8x12	0.1354	0.1016	0.1202
bounds trial	8x12	0.0521	0.0172	0.0364
actualmax	8x12	0.3333	0.1708	0.2602
actualavg	8x12	0.1615	0.0995	0.1336
bounds trial	8x12	0.1198	0.062	0.0938
actualmax	8x12	0.5	0.2417	0.3837

actualavg	8x12	0.224	0.1229	0.1785
bounds trial	8x12	0.1042	0.0182	0.0655
actualmax	8x12	0.4167	0.1542	0.2985
actualavg	8x12	0.1927	0.0948	0.1486
bounds trial	8x12	0.026	0.0302	0.0279
actualmax	8x12	0.2083	0.1458	0.1802
actualavg	8x12	0.1365	0.1045	0.1221
bounds trial	12x4	0.0833	0.0542	0.0702
actualmax	12x4	0.375	0.2375	0.3131
actualavg	12x4	0.2083	0.1292	0.1727
bounds trial	12x4	0.0938	0.0333	0.0666
actualmax	12x4	0.375	0.2125	0.3019
actualavg	12x4	0.1875	0.1115	0.1533
bounds trial	12x4	0.0625	0.0146	0.0409
actualmax	12x4	0.375	0.15	0.2738
actualavg	12x4	0.2097	0.1021	0.1613
bounds trial	12x4	0.1354	0.0802	0.1106
actualmax	12x4	0.5	0.2625	0.3931
actualavg	12x4	0.3069	0.166	0.2435
bounds trial	12x4	0.0833	0.0302	0.0594
actualmax	12x4	0.375	0.2125	0.3019
actualavg	12x4	0.1771	0.1167	0.1499
bounds trial	12x4	0.0625	0.0542	0.0587
actualmax	12x4	0.375	0.25	0.3187
actualavg	12x4	0.1458	0.1271	0.1374
bounds trial	12x4	0.0104	0.0031	0.0071
actualmax	12x4	0.25	0.1875	0.2219
actualavg	12x4	0.0938	0.0792	0.0872
bounds trial	12x4	0.0521	0.0354	0.0446
actualmax	12x4	0.375	0.2125	0.3019
actualavg	12x4	0.1771	0.1333	0.1574
bounds trial	12x4	0.0625	0.0167	0.0419
actualmax	12x4	0.375	0.2625	0.3244
actualavg	12x4	0.1771	0.1094	0.1466
bounds trial	12x4	0.0417	0.0354	0.0389
actualmax	12x4	0.5	0.2	0.365

actualavg	12x4	0.1979	0.1208	0.1632
bounds trial	12x8	0.0885	0.0255	0.0602
actualmax	12x8	0.3125	0.2062	0.2647
actualavg	12x8	0.2083	0.1094	0.1638
bounds trial	12x8	0.099	0.0276	0.0668
actualmax	12x8	0.5625	0.25	0.4219
actualavg	12x8	0.25	0.1193	0.1912
bounds trial	12x8	0.099	0.0276	0.0668
actualmax	12x8	0.5625	0.25	0.4219
actualavg	12x8	0.25	0.1193	0.1912
bounds trial	12x8	0.0313	0	0.0172
actualmax	12x8	0.25	0.1562	0.2078
actualavg	12x8	0.1042	0.0906	0.0981
bounds trial	12x8	0.0521	0.0141	0.035
actualmax	12x8	0.5	0.2312	0.3791
actualavg	12x8	0.2083	0.1073	0.1629
bounds trial	12x8	0.0313	0.0094	0.0214
actualmax	12x8	0.25	0.1562	0.2078
actualavg	12x8	0.1094	0.0828	0.0974
bounds trial	12x8	0.0521	0.0172	0.0364
actualmax	12x8	0.3125	0.1625	0.245
actualavg	12x8	0.1406	0.0844	0.1153
bounds trial	12x8	0.1198	0.062	0.0938
actualmax	12x8	0.375	0.2	0.2962
actualavg	12x8	0.2344	0.1245	0.1849
bounds trial	12x8	0.1042	0.0182	0.0655
actualmax	12x8	0.3125	0.175	0.2506
actualavg	12x8	0.2083	0.1156	0.1666
bounds trial	12x8	0.026	0.0302	0.0279
actualmax	12x8	0.1875	0.175	0.1819
actualavg	12x8	0.1042	0.1068	0.1053
bounds trial	12x12	0.0799	0.0128	0.0497
actualmax	12x12	0.2917	0.1792	0.241
actualavg	12x12	0.1944	0.1194	0.1607
bounds trial	12x12	0.0938	0.0128	0.0573
actualmax	12x12	0.375	0.2542	0.3206

actualavg	12x12	0.1944	0.108	0.1555
bounds trial	12x12	0.0417	0	0.0229
actualmax	12x12	0.2917	0.175	0.2392
actualavg	12x12	0.1146	0.1059	0.1107
bounds trial	12x12	0.0347	0	0.0191
actualmax	12x12	0.2083	0.1417	0.1783
actualavg	12x12	0.1146	0.0823	0.1001
bounds trial	12x12	0.0972	0.016	0.0607
actualmax	12x12	0.5	0.2292	0.3781
actualavg	12x12	0.2153	0.1444	0.1834
bounds trial	12x12	0.0868	0.0354	0.0637
actualmax	12x12	0.3333	0.1458	0.249
actualavg	12x12	0.1806	0.1035	0.1459
bounds trial	12x12	0.0243	0.0066	0.0163
actualmax	12x12	0.25	0.1875	0.2219
actualavg	12x12	0.1701	0.1094	0.1428
bounds trial	12x12	0.0903	0.0247	0.0607
actualmax	12x12	0.375	0.2167	0.3038
actualavg	12x12	0.1944	0.1198	0.1609
bounds trial	12x12	0.0938	0.0302	0.0652
actualmax	12x12	0.4167	0.225	0.3304
actualavg	12x12	0.2083	0.1253	0.171
bounds trial	12x12	0.0114	0.0477	0.0277
actualmax	12x12	0.75	0.375	0.5813
actualavg	12x12	0.2	0.1667	0.185

REFERENCES

[1] R. G. Askin, and S. Estrada, "A Survey of Cellular Manufacturing Practices", in Handbook of Cellular manufacturing Systems, S. Irani, (ed.), John Wiley & Sons, New York, 1999.

[2] R. G. Askin, and Y. Huang, "Forming effective worker teams for cellular manufacturing", Working Paper, Systems & Industrial Engineering, The University of Arizona, 2000.

[3] R. G. Askin, and M. Sodhi, "Organization of Teams in Concurrent Engineering", Handbook of Manufacturing and Automation, R. Dorf and A. Kusiak (eds.), John Wiley & Sons, New York, 1994.

[4] R. G. Askin, and A. Vakharia, "Group Technology Planning and Operation", in The Automated Factory Handbook, D. Cleland and B. Bopaya (eds.), TAB Books, PA, pp. 317-366, 1990.

[5] D. E. Bailey, "Manufacturing improvement team programs in the semiconductor industry", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, pp. 1-10, 1997.

[6] D. E. Bailey, "Modeling work group effectiveness in high-technology manufacturing environments", *IIE Transactions*, Vol. 32, pp. 361-368, 1999.

[7] M. R. Barrick and M. K. Mount, "The big five personality dimensions and job performance: a meta-analysis", *Personnel Psychology*, Vol. 44, No. 1, pp. 1-26, 1991.

[8] R. M. Belbin, "Management Teams: Why They Succeed or Fail", Heinemann, London, 1981.

[9] R. M. Belbin, B. R. Aston, and R. D. Mottram, "Building Effective Management Teams", *Journal of General Management*, Vol. 3, No. 3, pp. 23-29, 1976.

[10] R. M. Belbin, B. Watson, and C. West, "True Colours", *People Management*, Vol. 3, No. 5, pp. 34-38, 1997.

[11] J. L. Burbidge, The Introduction of Group Technology, John Wiley, New York, 1975.

[12] P. D. Carr, and G. Groves, "Teams and Cellular Manufacturing, The Role of the Team Leader", in Group Technology and Cellular Manufacturing, N. C. Suresh, and J. M. Kay (eds.), Kluwer Academic Publishers, Boston, 1998.

[13] A.C. Ebeling, and C.-Y. Lee, "Cross training Effectiveness and Profitability", *International Journal of Production Research*, Vol. 32, pp. 2843-2859, 1994

[14] A. Furnham, H. Steele, and D. Pendleton, "A psychometric assessment of the Belbin Team-Role Self-Perception Inventory", Journal of Occupational and Organizational Psychology, Vol. 55, pp. 245-257, 1993.

[15] A. L. Hammer, and G. E. Huszczo, "Teams", in MBTI Applications, A. L. Hammer (ed.), Consulting Psychologists Press, Palo Alto, California,1996.

[16] S. M. Henry, and K. T. Stevens, "Using Belbin's leadership role to improve team effectiveness: An empirical investigation", *The Journal of Systems & Software*, Vol. 44, pp. 241-250, 1999.

[17] J. Hut and E. Molleman, "Empowerment and team development", *Team Performance Management*, Vol. 4, No. 2, pp. 53-66, 1998.

[18] D. C. Juran, K. L. Schultz, T. H. Hammer, J. O. McClain, and L.W. Schruben, " Personality and the Performance of Production Systems", http://www.johnson.cornell.edu/faculty/mcclain/Papers/Papers.htm#Personality., 1997.

[19] J. R. King, and V. Nakornchai, "Machine Component Group Formation in Group Technology – Review and Extension", *International Journal of Production Research*, 20, pp. 117-133, 1982.

[20] K. Kolbe, "Pure Instinct", Random House, New York, 1993.

[21] K. Kolbe, "The Conative Connection", Addison-Wesley, New York, 1989.

[22] A. Kusiak, "The Generalized Group Technology Concept", *International Journal of Production Research*, 25(4), pp. 561-569, 1987.

[23] R. Lingard and E. Berry, "Improving team performance in software engineering", http://www.ecs.scun.edu/~rlingard/Publications/TeamLearning/TeachLearnPaper.html, 2000.

[24] H.Min and D.Shin, "Simultaneous Formation of Machine and Human Cells in Group Technology: A Multiple Objective Approach", *International Journal of Production Research*, Vol. 31, pp. 2307-2318, 1993.

[25] B. Norman, W. Tharmmaphornphilas, K. LaScola Needy, B. Bidanda and R. Colosimo Warner, "Assigning Workers to Tasks Considering Technical and Human Skills", *IE Research Conference*, Cleveland, OH, 2000.

[26] N. Singh, "Design of Cellular Manufacturing Systems: An Invited Review", *European Journal of Operational Research*, 69(3), pp. 284-291, 1993.

[27] G. A Suer, "Optimal Operator Assignment and Cell Loading in Labor-intensive Manufacturing Cells", *Computers and Industrial Engineering*, Vol. 26, pp. 155-158, 1996.

[28] N. C. Suresh, "Partitioning Work Centers for Group Technology : Insights from An Analytical Model", *Decision Sciences*, 22(4), pp. 772-791, 1991.

[29] N. C. Suresh, "Partitioning Work Centers for Group Technology : Analytical Extension and Shop-Level Simulation Investigation", *Decision Sciences*, 23(2), pp. 267-290, 1992.

[30] N. C. Suresh, and J. M. Kay, "Group Technology and Cellular Manufacturing: Updated Perspectives", in Group Technology and Cellular Manufacturing, N. C. Suresh, and J. M. Kay (eds.), Kluwer Academic Publishers, Boston, 1998.

[31] D. Tepke, and H. Davis, University of Chicago Study, Working Paper, 1990.

[32] R. C. Warner, K. L. Needy, and B. Bidanda, "Worker Assignment in Implementing Manufacturing Cells", *Proceedings of the IERC*, pp. 240-245, 1997.

[33] D. Wechsler, "Cognitive, Conative, and Non-Intellective Intelligence", in The Evolution of Psychology: Fifty Years of the American Psychologist, J. M. Notterman (ed.), American Psychological Association, Washington, DC, 1997.