An Enhanced Grouping Genetic Algorithm for Solving the Multifunctional Design Team Formation Problem

Teerawut Tunnukij and Christian Hicks*,

Newcastle University Business School, Armstrong Building,

Newcastle University, Newcastle upon Tyne UK, NE1 7RU.

*Corresponding author: Chris.Hicks@ncl.ac.uk

Abstract

In the design or product development process, multifunctional design teams (MDTs) are frequently required to work together simultaneously to create a product that satisfies customers and market requirements. The implementation of MDTs has been shown to significantly improve the performance of the new product development proces in many companies. However, the methods in the literature are mostly applied to relatively small problems. There has been little research that has developed methods for generating effective MDTs for large complex design projects.

Previous research by the authors has developed an Enhanced Grouping Genetic Algorithm (EnGGA) for solving cell formation problems in manufacturing. It was found that the EnGGA performed better than previous methods. This paper presents a two-stage algorithm that contains the EnGGA and a local search heuristic for solving MDT formation problems. The EnGGA was used to form MDTs and groups of tasks assigned to the MDTs simultaneously. A local search heuristic was used to identify engineering liaisons that facilitated information transfer between the MDTs. The two-stage algorithm was tested using randomly generated problems. The quality of the solutions was evaluated in terms of the modified grouping efficacy. The results show that the two-stage algorithm was effective and it is likely to be a promising method for solving MDT formation problems in complex systems.

Keywords: Cell Formation, Genetic Algorithms, Design Teams.

1 Introduction

Group Technology (GT) is a philosophy that aims to exploit similarities and achieve efficiencies by grouping in three distinct ways: (i) by performing like activities together; (ii) by standardising similar tasks; and (iii) by storing and retrieving information about recurring problems [1,2]. There are various application areas including design, manufacturing, process planning and control, purchasing and sales, and costs estimation [1-3]. The formation of multifunctional design teams is the application of GT to the design or product development process. Multifunctional design teams (MDTs) are frequently required to work together simultaneously in order to create a product that satisfies customer and market requirements and improve the design process. The implementation of MDTs requires tasks with interconnected problem-solving requirements to be grouped into sets of tasks. MDTs comprise groups of individuals from different disciplines that work closely together on sets of tasks [4]. They are frequently used within concurrent engineering (CE), which is 'a systematic approach to the integrated concurrent design of products and their related processes, including manufacture and support' [5, p27]. They can improve product quality, reduce costs and lead-times [5,6]. CE aims to bring together various viewpoints and talents in the design phase to ensure that decisions made in the design process will be effective throughout the entire product life cycle and satisfy customer requirements [5,6]. The implementation of CE has reduced product development time, engineering changes, scrap and rework and has improved time to market and return on assets and increased service life [7].

The implementation of MDTs has been reported to improve the performance of new product development in many companies [8]. However, the methods in the literature have focused upon relatively small or simple problems. There has been little research that provides methods for generating effective MDTs for very complex design processes. If companies do not have well formed MDTs, together with effective communication and cooperation between and within teams, they are likely to suffer from delays to the product development process [9].

The optimisation of the MDT formation problem has been shown to be a non-deterministic polynomial (NP) complete problem [10], which means that the amount of computation required increases exponentially with problem size. Even a powerful computer can take an unacceptably long time to solve a large problem due to combinatorial diffusion. Stochastic search methods are particularly suitable for solving complex combinatorial optimization problems. They are able to search large regions of the solution space without becoming trapped in local optima. Commonly used methods include Genetic Algorithms [11], Tabu search [12] and Simulated Annealing [13].

Genetic Algortithms (GAs) are derived from an analogy with biological evolution, in which the fitness of an individual determines its ability to survive and reproduce [14].

Falkenauer [15] developed a Grouping Genetic Algorithm (GGA) that suited the structure of grouping problems. Brown and Sumichrast [16] evaluated the performance of GGAs and suggested that GGAs are generally better than GAs for solving grouping optimisation problems because they are more computationally efficient. Tunnukij and Hicks [17] developed the Enhanced Grouping Genetic Algorithm (EnGGA) for solving the cell formation problem in manufacturing. It was found that the EnGGA outperformed other methods when applied to a wide range of cell formation problems.

The objectives of this paper are to: i) apply the Enhanced Grouping Genetic Algorithm (EnGGA) to the multifunctional design team formation (MDTF) problem; ii) describe the development of the two-stage algorithm that employs the EnGGA to identify effective multifunctional design teams and groups of tasks that should be assigned to the teams. It uses a local search heuristic to identify engineering liaisons who facilitate information transfer between the teams; and iii) report the results of experiments that tested the algorithm using randomly generated problems;

Section 2 reviews the literature relating to the MDT formation problem. Section 3 describes the development of the two-stage algorithm for solving the multifunctional design team formation problem. Section 4 presents the computational results. The conclusions are presented in section 5.

2 The Multifunctional Design Team Formation Problem

The multifunctional design team formation problem groups individuals from different disciplines into multifunctional design teams (MDTs) and tasks into groups of tasks. Well designed MDTs should maximise interactions between team members within each team and minimise interactions between teams. However, each team should maintain a frequent dialogue with interacting teams in order to raise and discuss technical conflicts [18]. A number of methods have been developed for solving the MDT formation problem. However, there has been little research that provides methods for generating effective MDTs for complex problems. All of the methods in the literature have been developed for relatively simple or small problems. These methods can be classified into three categories: (i) personal characteristic-based methods; (ii) product characteristic-based methods; and (iii) clustering-based methods.

Personal characteristic-based methods consider important characteristics of team members as the criteria for forming MDTs. Chen and Lin [19] identified the single best MDT by considering three major attributes of team members: (i) multifunctional knowledge; (ii) teamwork capability (teamwork experience, communication skills and flexibility); and (iii) working relationships. The Analytic Hierarchy Process (AHP) [20] is a multi-criteria decision analysis method that evaluates qualitative information. It uses quantitative measures together with comparison techniques. It has been used to define the quantifiable measures for team members' multifunctional knowledge and teamwork capability. Working relationships were evaluated by using the Myers–Briggs test, which classifies individuals into sixteen personality types [21]. This was used to classify team members' personalities, which was used to predict their ability to work with others. However, this approach has some limitations. Firstly, it only provides a mechanism for forming the single best MDT. In complex projects or systems, a single design team would be too large to be effective at facilitating communication among all team members. In practice, it may be necessary to have several MDTs working on different groups of tasks due to the complexity of a project [18]. Secondly, personality-based approaches do not provide a mechanism for identifying groups of tasks; additional methods must be employed to complete the formation of MDTs. Thirdly, the best solution may be highly sensitive to particular people involved, which may change over time. If each member's personal characteristics are evaluated improperly, this method may produce an ineffective MDT.

Product characteristic-based methods have considered product design characteristics, which are generally translated from customer requirements, to form MDTs. Zakarian and Kusiak [22] proposed a quantitative methodology for forming MDTs based upon product characteristics. The Quality Function Deployment (QFD) method, also known as the 'House of Quality' [23], was used to identify customer requirements relating to a new product which were translated into relevant product design characteristics. The relationship between product (engineering) characteristics and the team members responsible for those characteristics was then determined. The AHP method was used to prioritise the allocation of team members based upon importance measures assigned to customer requirements and product characteristics via the pair-wise comparison technique [20]. A mathematical programming model was then used to determine the optimal composition of MDTs. However, there were two main limitations. First, the number of MDTs had to be predetermined in advance. Therefore, if the number was selected incorrectly, it would have produced unsatisfactory results. Second, the pair-wise comparison technique in the AHP can take a long time to complete the comparisons, particularly for large problems. For prioritising n elements, $n(n-1)/2$ paired comparisons need to be considered [19,22]. Therefore, the AHP method is likely work best for relatively small problems as the execution time for the pair-wise comparison will increase in proportion to n^2.

Clustering-based methods have solved grouping problems by maximising the interactions between elements within clusters (groups) whilst minimising interactions between clusters [24,25]. A number of clustering-based methods have used the Design Structure Matrix (DSM) [26] to represent the relationships among project tasks for identifying MDTs. The DSM is an information exchange model that represents the relationships between tasks or physical components within a system. The DSM is a square matrix with identical row and column labels [27,28]. McCord and Eppinger [18] used the DSM to identify interactions between components to structure groups of MDTs as system teams in a development project. In their research, MDTs that performed the tasks related to components were predetermined in advance. A clustering heuristic based upon a visual inspection of the DSM together with logical reasoning was used to rearrange the matrix in order to identify groups of highly coupled components. As a result, system teams were identified based upon groups of components that correspond to groups of MDTs. Engineering liaison was also identified by rearranging the DSM to make some components (corresponding to MDTs) formal members of more than one group of MDTs (corresponding to a system team). However, this method did not provide a mechanism for assigning individuals to MDTs.

Some researchers have partitioned project tasks in order to identify independent (parallel) tasks, dependent tasks and interdependent (coupled) tasks before forming MDTs. Independent tasks can be performed in any order without affecting other tasks. Dependent tasks can be performed sequentially or by overlapping tasks. Interdependent tasks are frequently required to be performed by various experts from different disciplines as MDTs [9]. Chen and Lin [9] use a partitioning algorithm proposed by Steward [29] that uses visual inspection and logical reasoning to rearrange the DSM. The AHP method was then used to transform the binary form of task relationships in the matrix into the quantifiable task coupling strength. As a result, large groups of interdependent tasks could be decomposed into smaller and manageable task groups. The structure of task groups provided initial information for creating MDTs based upon groups of tasks. However, this methodology did not provide a mechanism for assigning individuals to MDTs to perform groups of tasks. Chen [30] integrated Chen and Lin's task partitioning model [9] with Chen and Lin's personal characteristic-based method [19] for identifying groups of tasks and assigning individuals into MDTs. A task-member model was developed for assigning individuals to tasks based upon the results of the task partitioning model and the personal characteristic-based method. However, for complex design projects, it may be difficult to identify the sequence of tasks by rearranging a large DSM based upon visual inspection and logical reasoning due to the complexity. In addition, personal characteristics may be difficult to evaluate. This method may produce inconclusive results for large complex problems.

Braha [4] proposed a mathematical formulation for partitioning tasks by determining partitioning costs. This was based upon an assumed communication cost for each product attribute associated with a related task. The objective was to assign tasks and attributes to MDTs so as to minimise the costs of the interactions across teams. However, the number of MDTs needs to be predetermined in advance by specifying the maximum number of attributes that can be assigned to a team. Therefore, if this number is selected improperly, the optimal solution may not be found. In addition, this method does not provide a mechanism for assigning individuals to MDTs.

Hierarchical clustering methods are an alternative clustering approach for solving the MDT formation problem. Componation and Byrd [7] applied hierarchical clustering methods using single linkage, complete linkage, average linkage, the centroid method, and Ward's method to group tasks by using the Statistical Analysis Software (SAS) package. MDTs were formed by assigning individuals to groups of tasks that required their skills. The project duration and total project risk were used to evaluate the quality of the solutions. The results showed that the average linkage method was most effective in developing MDTs in terms of reducing the project duration and risk levels. The methodology was applied to a 23-person team working on a product development project that was divided into 18 primary design tasks. However, in practice, when some large complex design projects are considered, the number of tasks and interactions that must be evaluated can become too time consuming and difficult to manage [7].

Wang et al. [10] proposed a mathematical model to form MDTs by considering three main requirements: (i) similarities between tasks (each task required in the current project and all the completed tasks for which each team member has been responsible); (ii) team members' abilities; and (iii) the number of team members, their workloads and utilisation rates. They decomposed the MDT formation problem into a series of independent sub-problems and employed Tabu Search to solve the sub-problems. The objective was to choose the most competent individuals for each task. However, this method did not provide a mechanism for identifying groups of tasks that should be assigned to each team – it assumed that the groups of tasks that needed to be undertaken by each team were already known. In addition, it could not solve the whole problem directly – the problem needed to be decomposed into sub-problems. Furthermore, team members' abilities may be difficult to evaluate and an optimum solution is not guaranteed.

The optimisation of the MDT formation problem has been shown to be an NP-complete problem [10]. However, it can be seen that all of the methods that have been developed in the literature are likely to suit for relative smalls or simple problems. In addition, additional methods are required to complete the formation of MDTs. None of the previous methods can group individuals into MDTs and tasks into groups of tasks simultaneously. Therefore, there is a lack of effective methods for creating optimal MDTs. This is a particular problem with the design of complex systems which may involve hundreds of people organised into many MDTs.

3 Problem Definition and Performance Measurement

The relationships between team members and tasks may be represented as a member-task incidence matrix (see Fig. 1a). For example, in a, task 1 are performed by members (individuals) 1, 2 and 4, whilst member 1 is required to perform tasks 1, 3 and 6. The objective is to rearrange the matrix to perform a block diagonal structure from which groups of tasks and the members required to operate them can be selected, such as shown in Fig. 1b. The block diagonal matrix can be then measured the quality of the solution by a performance measure.

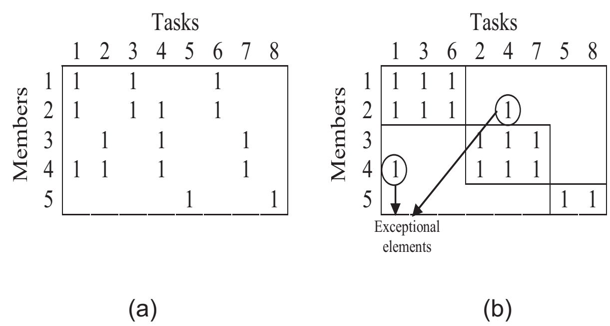


Fig. 1 A member-task incidence matrix: (a) the original matrix; (b) a rearranged matrix into block diagonal form

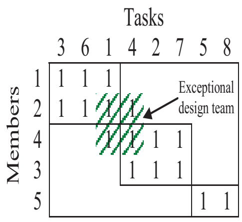


Fig. 2 A rearranged matrix that identifies an exceptional element team.

The grouping efficacy (Γ), proposed by Kumar and Chandrasekharan [31], has been widely used as a quantitative criterion for measuring the quality of block diagonal forms in machine-part incidence matrices. It can be used for measuring the quality of block diagonal forms in the member-tasks incidence matrices. The research presented in this paper employed the grouping efficacy as a quantitative criterion to measure the quality of solutions in member-task incidence matrices. The grouping efficacy was modified in order to take into account exceptional design teams (EDTs), which contain engineering liaisons and exceptional tasks performed by more than one MDT. For example, the rearranged matrix in Fig. 2 shows the solution where eight tasks and five members are allocated to three design teams with one EDT. The first team contains members 1 and 2 together with tasks 1, 3 and 6. The second team contains members 3 and 4 together with tasks 2, 4, and 7. The final team contains member 5 together with tasks 5 and 8. The EDT contains members 2 and 4 assigned as engineering liaisons for facilitating communication between the first team and the second team, together with exceptional tasks 1 and 4 that require members from the first and second teams to perform them. The modified grouping efficacy (Γ_m) was used as a quantitative criterion to measure the quality of solutions.

$$\Gamma_m = 1 - \frac{e_{ob} + e_{vb}}{e + e_{vb}} \qquad (1)$$

where e is the total number of operations (number of 1's in the matrix), e_{ob} is the number of 1's outside the diagonal and exceptional element blocks and e_{vb} is the number of voids in the diagonal and exceptional element blocks.

Using the modified grouping efficacy measure, the clustering algorithm developed in this paper aims to maximise interactions between team members within each MDT by minimising the number of voids in the diagonal and exceptional element blocks whilst minimising interactions between MDTs by minimising the number of exceptional element (1's) outside the diagonal and exceptional element blocks.

4 Two-stage Algorithm

The algorithm proposed in this paper can solve the MDT formation problem by grouping individuals from different disciplines into MDTs and tasks into groups of tasks simultaneously. It can also identify engineering liaisons between MDTs. The algorithm consists of two stages: (i) clustering members and tasks into MDTs; and (ii) clustering exceptional elements into exceptional design teams (EDTs) in order to identify engineering liaisons. The Enhanced Grouping Genetic Algorithm (EnGGA), proposed by Tunnukij and Hicks [17], was applied to group members and tasks into MDTs because it outperformed other methods when applied to a wide range of cell formation problems, which are typical grouping optimisation problems. A local search heuristic, proposed by Goncalves and Resende [32], was then used to identify engineering liaisons by clustering exceptional elements into EDTs.

The general structure of the EnGGA [17] applied in this paper is shown in Fig. 3. The EnGGA uses the 0-1 member-task incidence matrix to represent the initial configuration. The EnGGA can solve the MDT formation problem without predetermining the number of MDTs or the number of team members and tasks assigned to each MDT. However, there is no point in clustering all the team members (individuals) (X) and all the tasks (T) into only one MDT or having only one member in each MDT. Therefore, the possible number of MDTs (DT) is defined as $2 \leq DT \leq min(X\text{-}1, T\text{-}1)$.

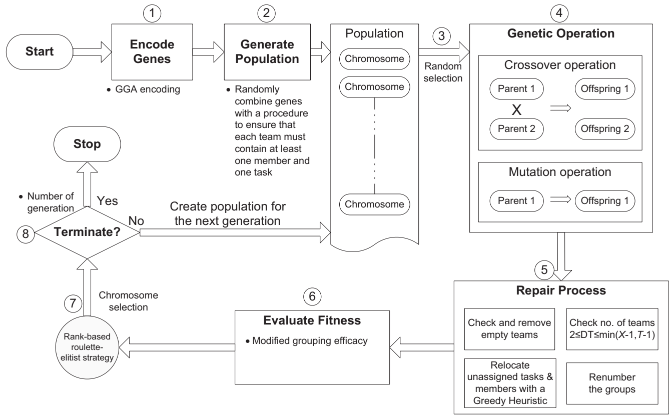


Fig. 3 The general structure of the EnGGA (adapted from Tunnukij and Hicks [17]).

4.1 Genetic representation.

The EnGGA [17] used the GGA encoding scheme [15]. The chromosome representation (shown in Fig. 4) consists of three sections: (i) the task section; (ii) the member section; and (iii) the group section. Each gene in the task and member sections contains an integer that represents the MDT number. The task and member numbers are represented by the position of the genes within the appropriate section. The length of individual chromosomes may differ because the number of MDTs in alternative solutions may vary. The chromosome length is therefore equal to the sum of the number of tasks (T), the number of members (X), and the number of MDTs (DT), where DT varies from 2 to $min(X\text{-}1, T\text{-}1)$. This representation allows the EnGGA to form MDTs and groups of tasks simultaneously. It also allows genetic operators to be performed on the group portion of the chromosome. As a result, the groups are modified as a whole, rather than by modifying individual members. This is a computationally efficient approach. The order in which the MDTs in the group section are listed does not matter. Fig. 4 illustrates this representation with a chromosome that represents a possible solution to the member-task grouping problem shown in Fig. 1a. The group section shows that the members and tasks are allocated to three MDTs. The first team contains members 1, 2 and 6 together with tasks 3 and 5. The second team contains members 3 and 5 together with tasks 2 and 4. The final team contains members 4, 7, 8 and task 1.

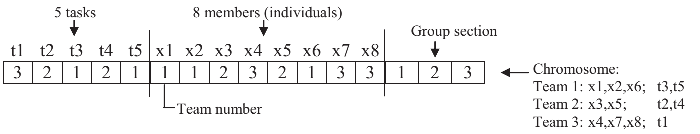


Fig. 4 A chromosome representation of the EnGGA for the MDT problem.

4.2 Method for generating the initial population.

The initial population of chromosomes is generated randomly. This process is as follows. The number of design teams *(DT)* teams is randomly generated, where *DT* is a random positive integer where $2 \leq DT \leq X-1$ if $X<T$, otherwise $2 \leq DT \leq T-1$. Next, *DT* members and *DT* tasks are randomly selected; the members and tasks are then assigned to teams so that each team contains at least one member and one task. The remaining members and tasks are then randomly allocated into the teams; The steps are repeated until a population of the required size *(Pop)* is produced.

4.3 Reproduction selection scheme.

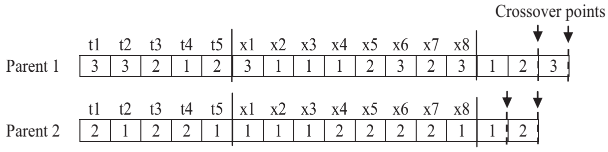
Chromosomes are randomly selected for the crossover and mutation operations; all chromosomes have an equal probability of selection. The probabilities of crossover (P_c) and mutation (P_m) are pre-specified experimental parameters.

4.4 Genetic operators.

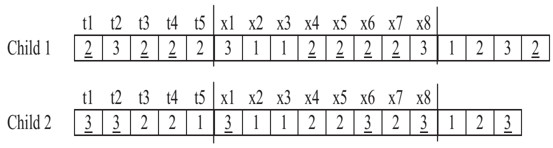
Falkenauer's [15] crossover, elimination mutation and division mutation operators were adopted (with minor modifications). They were integrated with a repair process that rectifies infeasible chromosomes produced by genetic operations.

The crossover operator includes two steps, which are shown in Fig. 5: i) two parents are randomly chosen from the population; ii) two crossover points are then randomly selected from the group section of each parent. All the genes from the first parent are initially copied to the first child. Likewise, all the genes from the second parent are initially copied to the second child. The section within the crossover points of the second parent is appended to the first child; likewise, the section within the crossover points of the first parent is appended to the second child. When genetic information is copied from the second parent to the first child, or from the first parent to the second child, it is shown in underlined text. All the members and tasks that belong to the teams within the appended section are inherited by the child. For example, in Fig. 5b, the first child has inherited team 2 from the second parent. This team contains members 4, 5, 6 and 7 together with tasks 1, 3 and 4; they are all inherited by the first child, which replace the genes initially inherited from the first parent.

Falkenauer's crossover operator produces children that are identical to the parents if the MDT formations represented by the two parents are the same. This phenomenon will trap the search into a local optimum. Therefore, in the EnGGA the two selected parents are compared before they are processed by the crossover operator. The algorithm attempts to randomly choose a parent that has a different MDT formation. If the randomly chosen chromosomes are the same, the process is repeated until either a different chromosome has been chosen, or until 30% of the population has been sampled.



(a)



(b)

Fig. 5 Faulkner's crossover operator: (a) select crossover points; (b) injection

The standard GGA elimination mutation operator and division mutation operator [15] were used with minor modifications. The mutation steps, which are shown in Fig. 6, are as follows: i) a parent is chosen from the population randomly; ii) the number of teams is checked: iii) if the number of teams is more than two, the standard elimination mutation operator will be used. One of the teams in the group section is randomly selected and all of its elements are eliminated. The remaining elements are inherited by the child (see Fig. 6a); iv) if the number of teams is two or less, the modified division mutation operator will be used. A team that contains at least two members and two tasks is randomly selected and then divided into two new teams. Two members and two tasks within the selected team are randomly selected and are split between the two new teams. This ensures that each new team contains at least one member and one task. Fig. 6b illustrates this process. In this case, team 2 has been randomly selected and it is then divided into team 2 and team 3. The underlined team numbers indicates that the teams have been created by the division mutation. The next step is to randomly select two members and two tasks from team 2 to be assigned to teams 2 and 3. In this case, member 4 and task 3 have been assigned to team 2, whilst member 6 and task 1 have been allocated to team 3. The remaining unassigned elements (members 5, 7 and task 4) are allocated by the repair process.

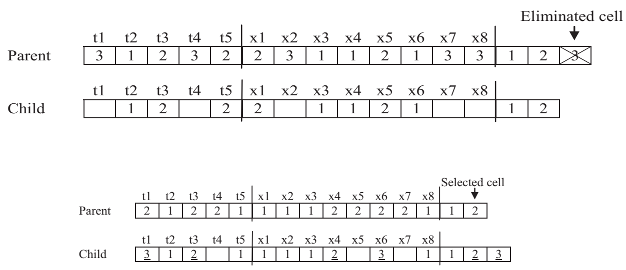


Fig. 6 Faulkner's mutation operators: (a) elimination mutation; (b) division mutation

4.5 Repair process.

The chromosomes produced by the genetic operations may represent infeasible solutions. The EnGGA repair process [17] was applied to rectify infeasible chromosomes. The repair process consists of four stages: i) checking and removing empty teams – each team must contain at least one member and one task. For example, in Fig. 5b, child 1 contains empty teams. Team 1 has two members but no tasks, whilst team 2 has one member but no teams. The repair process identifies and then removes the empty teams (see Fig. 7a); ii) checking the number of teams – the possible number of teams (DT) is defined as $2 \le DT \le min(X\text{-}1, T\text{-}1)$; iii) if the number of teams within the child produced after step 1 is one, a new team number will be inserted and unassigned members and tasks will be relocated into the new team; if the number of teams is more than $min(X\text{-}1, T\text{-}1)$, a team will be randomly selected and eliminated until the number of teams is equal to $min(X\text{-}1, T\text{-}1)$; iv) Unassigned members and tasks will then be relocated into the existing teams by the Greedy Heuristic.

	t1	t2	t3	t4	t5	x1	x2	x3	x4	x5	x6	x7	x8		
Child 1	2	3	2	2		3			2	2	2	2	3	3	2

(a)

	t1	t2	t3	t4	t5	x1	x2	x3	x4	x5	x6	x7	x8		
Child 1	2	3	2	2	3	3	2	3	2	2	2	2	3	3	2

(b)

	t1	t2	t3	t4	t5	x1	x2	x3	x4	x5	x6	x7	x8		
Child 1	2	1	2	2	1	1	2	1	2	2	2	2	1	1	2

(c)

Fig. 7 The EnGGA repair process: (a) remove the empty teams; (b) relocate unassigned members and/or tasks by the greedy heuristic; (c) renumber the groups.

4.6 Greedy Heuristic

Unassigned members and tasks are assigned to the existing teams by a Greedy Heuristic. The Greedy Heuristic evaluates the fitness of all the possible chromosomes that could be produced by all the alternative allocations of unassigned members and tasks. Fitness is measured in terms of the modified grouping efficacy. Fig. 7a and Fig. 7b illustrate this procedure. Child 1 represents a team formation where team 2 contains members 4, 5, 6, 7 and tasks 1, 3 and 4; team 3 contains members 1, 8 and task 2. However, members 2, 3 and task 5 are unassigned and need to be relocated into either teams 2 or 3. If the original member-task incidence matrix shown in Fig. 1a was rearranged to reflect this configuration and member 2 was relocated into team 3, the modified grouping efficacy would be 24.00. If member 2 was relocated into team 2, the grouping efficacy would be 32.00. Therefore, the Greedy Heuristic would place member 2 into team 2 because that would generate the highest modified grouping efficacy. After relocating member 2 into team 2, member 3 would then be relocated into team 3 because that would generate the highest modified grouping efficacy of 36.00 rather than placing it into team 2 which would generate a modified grouping efficacy of 33.33. Finally, task 5 would be relocated into team 3 which would generate a highest modified grouping efficacy of 37.04 rather than placing it into team 2 which would generate a modified grouping efficacy of 34.48. Fig. 7b shows the solution after relocating unassigned members and tasks using the Greedy Heuristic. Renumbering the groups to simplify interpretation. This is illustrated by Fig. 7c. In this example, team 3 in child 1 has been renumbered as team 1, whilst team 2 has been renumbered as team 2.

4.7 Evaluation criteria.

This paper developed and used the modified grouping efficacy (Γ_m) (see equation (1) in section 3) as the objective function for measuring the quality of block forms which takes into account both diagonal blocks and exceptional element blocks. The best solution minimises the number of voids (zeros) in the diagonal and exceptional element blocks and the number of 1's outside the diagonal and exceptional element blocks.

4.8 Mechanism for creating successive generations.

The EnGGA used the rank-based roulette-elitist strategy with 15% of the best chromosomes surviving to the next generation for creating successive generations [17]. The rank-based roulette-elitist strategy employs the elitist strategy [14] to select the fittest chromosomes, and it uses the rank-based roulette wheel [33] to select the remaining chromosomes.

4.9 Stopping criteria.

The EnGGA terminates when a fixed number of generations have been completed. The team formation configuration associated with the highest fitness chromosome is then shown.

The local search heuristic, proposed by Goncalves and Resende [32], was applied to the best team formation produced by the EnGGA in order to choose the best choice available. It then allocated the exceptional elements (EEs) to exceptional design teams (EDTs) whenever possible in order to identify engineering liaisons. The EEs that were assigned into an EDT would belong to both a MDT and an EDT(Fig. 2). The local search heuristic allocated the EEs into EDTs that generate the best fitness value (the modified grouping efficacy). This research has assumed that each EDT would be shared between two teams, engineering liaisons within each EDT facilitate communication between two MDTs.

The local search heuristic works by first considering each pair of MDTs produced by the EnGGA starting from the first two MDTs to the last two MDTs. For example, if there were four MDTs (teams 1, 2, 3 and 4) produced by the EnGGA, the local search heuristic would consider pairing teams 1 and 2, teams 2 and 3, and teams 3 and 4, respectively. A designer may renumber team numbers (see step 4 in the repair process within the EnGGA) to enable the MDT structure to reflect the product structure (associated with the assigned tasks). The local search heuristic consists of an improvement procedure that is repeatedly applied. It consists of the following four steps: i) the initial solution is generated from the best MDT formation produced by the EnGGA. Each exceptional task is assigned to an EDT that is shared by a pair of MDTs. Each exceptional member is then assigned to an EDT if there is an increase in the fitness value (modified grouping efficacy); ii) exceptional tasks are reassigned into EDTs. With a set of exceptional members that has been assigned into EDTs in step 1, each exceptional task is reassigned into the EDT that maximises the fitness value. In this step, the heuristic generates a new set of exceptional task assignments to EDTs; iii) exceptional members are reassigned into EDTs. With the new set of exceptional tasks that has been assigned into EECs in step 2, each exceptional member is reassigned into the EDT that maximises the fitness value. In this step, the heuristic generates a new assignment of exceptional members assigned into EDTs; iv) steps 2 and 3 above are repeated until there is no improvement of the fitness value, at which point the algorithm terminates and returns the best solution found.

5 Analysis of performance using randomly generated data sets

The two-stage algorithm that contains the EnGGA [17] and the local search heuristic [32] was tested using two randomly generated problems. This was necessary because the problems described in the literature have grouped tasks into groups of tasks or individuals into multifunctional design teams (MDTs). There are no examples that have considered the MDT formation problem as a whole i.e. assigning members to MDTs and also grouping tasks into sets of tasks. A full factorial experiment that considered the parameter settings, proposed by Tunnukij and Hicks [17], was used with a population size of 1000. The two-stage algorithm was written in C and was tested on a laptop with a 1.66GHz processor.

In order to generate problems, this paper assumed that the number of tasks that could be assigned to a member (individual) was between two and three. Each member was then allocated to perform tasks randomly. In practice, the number of tasks that can be assigned to each member may be determined by evaluating each member's workload and the skills required to perform the tasks [7,10].

Table 1 The best results for problems A and B obtained by the two-stage algorithm

Problem	Team configuration			r_m	Time(s)	Best Generation
	No. of teams	No. of EDTs	No. of eng. liaisons			
A (40 x 20)	12	6	7	57.80	34	13
B (100 x 50)	30	7	7	51.34	190	22

Table 1 shows the best solutions found by the two-stage algorithm for problems A and B. The computational time in seconds and the generation when the best solution was found are also reported. The block diagonal matrices for the best solutions for problems A and B were evaluated and the results showed that the two-stage algorithm effectively solved the two randomly generated problems. MDTs and groups of assigned tasks were identified as well as engineering liaisons among MDTs. For problem A, a small problem, the computational time required to run the two-stage algorithm with 50 generations was about 30 seconds; for problem B, a medium-large problem, took about three minutes even with a large population size of 1000. The best solutions for both problems were found within 30 generations. Please note that the two-stage algorithm allows the presence of singletons (teams containing only one member or one task) in order to maximise the quality of solutions. However, it does not mean this team should work independently. Interactions between each team should be maintained in order to communicate and discuss technical conflicts between teams [18].

6 Conclusions

A number of methods have been developed for solving the multifunctional design team (MDT) formation problem during the design process or the product development process. However, there has been little research that has provided methods for generating effective MDTs in complex systems. All of the methods in the literature are likely to be suitable for relatively small or simple problems. Additional methods are also required to complete the formation of MDTs. None of them can group individuals into MDTs and tasks into groups of tasks simultaneously.

This paper has applied a two-stage algorithm that contains the Enhanced Grouping Genetic Algorithm (EnGGA) and a local search heuristic for solving MDT formation problems. The EnGGA can form MDTs and groups of tasks simultaneously whilst a local search heuristic was used to identify engineering liaisons that facilitate information transfer between the MDTs. The two-stage algorithm was tested using two randomly generated problems. The modified grouping efficacy was used to measure the quality of the solutions. The results show that the two-stage algorithm was effective when applied to the MDT formation problems. It can group individuals into MDTs and tasks into groups of tasks simultaneously without predetermining the number of MDTs or the number of team members and tasks within each MDT. However, designers may specify the maximum number of MDTs and the minimum number of team members and tasks within a MDT based upon their requirements when applying the two-stage algorithm in practice; this may reduce the quality of the solutions.

References

[1] N. L. Hyer and U. Wemmerlov, "Group technology in the US manufacturing industry: a survey of current practices", *International Journal of Production Research,* vol. 27, part 8, pp. 1287-1304, 1989.

[2] N. L. Hyer and U. Wemmerlov, "Group Technology and Productivity", *Harvard Business Review,* vol. 62, part 4, pp. 140-149, 1984.

[3] N. C. Suresh and J. M. Kay, *Group Technology and Cellular Manufacturing: A State-of-the-Art Synthesis of Research and Practice*, Kluwer Academic, Boston, 1998.

[4] D. Braha, "Partitioning tasks to product development teams", in *Proc. 14th International Conference on Design Theory and Methodology Integrated Systems Design Theme: Aircraft Systems, Sep 29-Oct 2 2002*, Montreal, Quebec,

Canada, 2002.

[5] G. E. Dieter, *Engineering Design : A Materials and Processing Approach*, McGraw-Hill, Boston, 2000.

[6] P. G. Ranky, *Concurrent/Simultaneous Engineering (Methods, Tools & Case Studies)*, CIMware, Guildford, England, 1994.

[7] P. J. Componation and J. J. Byrd, "Utilizing cluster analysis to structure concurrent engineering teams", *IEEE Transactions on Engineering Management,* vol. 47, part 2, pp. 269-280, 2000.

[8] M. L. Swink, "Tutorial on implementing concurrent engineering in new product development programs", *Journal of Operations Management,* vol. 16, part 1, pp. 103-116, 1998.

[9] S.-J. Chen and L. Lin, "A project task coordination model for team organization in concurrent engineering", *Concurrent Engineering Research and Applications,* vol. 10, part 3, pp. 187-202, 2002.

[10] Z. Wang, H.-S. Yan and X.-D. Ma, "A quantitative approach to the organisation of cross-functional teams in concurrent engineering", *International Journal of Advanced Manufacturing Technology,* vol. 21, part 10-11, pp. 879-888, 2003.

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.

[12] F. Glover, "Tabu search - part I", *ORSA Journal on Computing,* vol. 1, part 3, pp. 190-206, 1989.

[13] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", *Science,* vol. 220, part 4598, pp. 671-680, 1983.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[15] E. Falkenauer, *Genetic Algorithms and Grouping Problems*, John Wiley & Sons, New York, 1998.

[16] E. C. Brown and R. T. Sumichrast, "Evaluating performance advantages of grouping Genetic Algorithms", *Engineering Applications of Artificial Intelligence,* vol. 18, part 1, pp. 12, 2005.

[17] T. Tunnukij and C. Hicks, "An enhanced grouping Genetic Algorithm for solving the cell formation problem", *International Journal of Production Research,* vol. in press, part DOI: 10.1080/00207540701673457, pp. 2008.

[18] K. R. McCord and S. D. Eppinger, "Managing the integration problem in concurrent engineering", MIT Sloan School of Management, Cambridge, MA, Working Paper No. 3594, 1993.

[19] S.-J. G. Chen and L. Lin, "Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering", *IEEE Transactions on Engineering Management,* vol. 51, part 2, pp. 111-124, 2004.

[20] T. L. Saaty, "How to make a decision - the Analytic Hierarchy Process", *Interfaces,* vol. 24, part 6, pp. 19-43, 1994.

[21] I. B. Myers, *Introduction to Type*, CA: Consulting Psychologists Press, Palo Alto, 1998.

[22] A. Zakarian and A. Kusiak, "Forming teams: an analytical approach", *IIE Transactions (Institute of Industrial Engineers),* vol. 31, part 1, pp. 85-97, 1999.

[23] L. Cohen, *Quality Function Deployment: How to Make QFD Work for You.*, Addison-Wesley Publishing Company, Reading, Masschusetts, 1995.

[24] E. Rechtin, *Systems Architecting*, Prentice Hall, Englewood Cliffs, N.J., 1991.

[25] R. Sanchez and J. T. Mahoney, "Modularity, flexibility and knowledge management in product and organization design", *IEEE Engineering Management Review,* vol. 25, part pp. 50-61, 1997.

[26] D. V. Steward, "The design structure system: a method for managing the design of complex systems", *IEEE Transactions on Engineering Management,* vol. EM28, part 3, pp. 71-74, 1981.

[27] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions", *IEEE Transactions on Engineering Management,* vol. 48, part 3, pp. 306, 2001.

[28] A. Yassine and D. Braha, "Complex Concurrent Engineering and the Design Structure Matrix Method", *Concurrent Engineering Research and Applications,* vol. 11, part 3, pp. 165-176, 2003.

[29] P. M. Braiden, C. Hicks, P. Pongcharoen and D. J. Stewardson, "Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products", *International Journal of Production Economics,* vol. 78, part 3, pp. 311-322, 2002.

[30] S.-J. Chen, "An integrated methodological framework for project task coordination and team organization in concurrent engineering", *Concurrent Engineering Research and Applications,* vol. 13, part 3, pp. 185-197, 2005.

[31] C. S. Kumar and M. P. Chandrasekharan, "Grouping Efficacy - a quantitative criterion for goodness of block diagonal forms of binary matrices in Group Technology", *International Journal of Production Research,* vol. 28, part 2, pp. 233-243, 1990.

[32] J. F. Goncalves and M. G. C. Resende, "An evolutionary algorithm for manufacturing cell formation", *Computers and Industrial Engineering,* vol. 47, part 2-3, pp. 273, 2004.

[33] C. R. Reeves, "Genetic Algorithm for flowshop sequencing", *Computers and Operations Research,* vol. 22, part 1, pp. 5-13, 1995.