Clustering Parts Based on the Design and Manufacturing Similarities Using a Genetic Algorithm

Sungyoul Lee*

Abstract The part family (PF) formation in a cellular manufacturing has been a key issue for the successful implementation of Group Technology (GT). Basically, a part has two different attributes; i.e., design and manufacturing. The respective similarity in both attributes is often conflicting each other. However, the two attributes should be taken into account appropriately in order for the PF to maximize the benefits of the GT implementation.
This paper proposes a clustering algorithm which considers the two attributes simultaneously based on pareto optimal theory. The similarity in each attribute can be represented as two individual objective functions. Then, the resulting two objective functions are properly combined into a pareto fitness function which assigns a single fitness value to each solution based on the two objective functions. A GA is used to find the pareto optimal set of solutions based on the fitness function. A set of hypothetical parts are grouped using the proposed system. The results show that the proposed system is very promising in clustering with multiple objectives.

Key Words : Part family, Genetic algorithm, Pareto optimization, Group technology

1. Introduction

In cellular manufacturing system (CMS), a set of parts, usually dissimilar in shapes, are grouped into a cell. Such a cell should ideally be responsible for the complete processing the group of parts called a part family (PF). Grouping parts can be viewed as a clustering problem.

The major issue in the design of CMS is to form a proper PF. Parts have design and manufacturing attributes which are often independent and conflicting each other in terms of similarity. However, the two attributes should be properly considered in forming a PF to maximize the benefits of the GT implementation.

Therefore, this study presents a pareto optimal model with a solution procedure based on a genetic algorithm (GA). In a pareto optimal solution set, there cannot be another single solution which is better in the two objectives. There may exist a solution which is better in one objective, and another solution which is better in another objective, but no single solution exists which is better in two objectives. [Lee, 2009]

There is no unique solution for a multi objective decision making problem with conflicting objectives. Instead, a set of satisfactory solutions or compromise solutions will have to be found. A solution is satisfactory if there exists no other solution which can improve one or more objectives without detriment to any other objectives.

In general, when multiple objectives exist, the

* Department of Computer Science & Engineering, Kwandong University

objectives must be combined into a single fitness value using weighting factors for the respective objectives. It can be very difficult to decide a proper set of weighting factors. [Cho & Gen, 2010] In this paper, the problem is solved using GA based on a pareto fitness function which eliminates the need to weight the objectives.

When the overall pareto set of solutions is found, the decision makers can choose one among themselves as to the relative importance of the objectives.

A numerical example has been tested to verify the proposed system. The results show that the proposed system is promising approach to group parts based on design and manufacturing attributes simultaneously.

2. Genetic Algorithm

A GA is a general adaptive search method requiring domain specific knowledge to solve a problem. The GA mimics the process of natural evolution by combining the survival of the fittest among solution population gradual information exchange by generation and creates offspring. The offspring displace weak solutions during each generation.

A GA is used to search the solution space and find pareto set of solutions.

2.1 Modified GA

Modified GA proposed by Michalewicz *et al.* [1992] is basically applied in this study. The modified GA is summarized as follows:

Step 1. Set the generation as 0 and t=0.

Step 2. Produce the initial population, P(0).

Step 3. Evaluate each chromosome of the population.

Step 4. Increase the generation number by 1, and t=t+1.

Step 5. Copy (population size - R) chromosomes in P(t-1) into P(t) without repetition.

Step 6. Select R chromosomes in P(t-1) with repetition.

Step 7. Do crossover and mutate for the R chromosomes. That is, apply crossover for the integer number of the (P_c x R) chromosomes and mutate for the (R-(P_c x R)) chromosomes.

Step 8. Add the new R chromosomes into P(t).

Step 9. Evaluate each chromosome in the new population.

Step 10. If termination condition (if a given number of generations is reached) is satisfied, then stop. Otherwise, go to *Step 4.*

2.2 Chromosomal Representation

In this study, a gene stands for a cell number. An integer string of cell numbers constitutes a chromosome. Hence, the length of a chromosome is equal to number of parts to be clustered. The position of a gene in a chromosome represents the part number while gene value is the cell number to which that part has been assigned. For instance, say there are five parts numbered from 1 to 5 and two cells identified by integers 1, and 2. Then, a C = (2, 2, 1, 2, 1) is a chromosomal representative of a possible solution in which parts 3 and 5 are assigned into cell #1 and parts 1, 2, and 4 are assigned into cell #2.

2.3 Initial Population

For initial population, two feasible chromosomes are individually calculated based on K means algorithm for the each of the design similarity and the manufacturing similarity. Then, each chromosome is permuted randomly to produce the first half and the last half of the given number of initial population. Finally, the resulted population is reordered randomly. Here, whenever a chromosome violates the given number of groups constraint, a forced repairing process takes place. That is, if some group numbers are missing either through the initial population generating process or after series of genetic process, the missed group numbers are forced to put into the two random chromosomal positions.

2.4 Pareto Fitness Function

A generic pareto fitness function [Schaumann *et al* 1998] is used to determine whether or not a solution is a member of the pareto set for a generation. This function considers all the objective values to determine a scalar value representing the pareto optimality of each solution. The following function calculates the pareto fitness for the i th solution in the generation:

$$F_i = [1 - \max_{j \neq i}(\min(f1_i - f1_j, f2_i - f2_j, \ldots))]^p$$

where:

F_i = pareto fitness value of i th solution

$f1_i$ = first objective function value (geometrical characteristic vector)

$f2_j$ = second objective function value (manufacturing characteristic vector)

p = pareto exponent

Note that the *max* is over all other solutions j ≠ i in the generation, and the *min* is over all the objectives. Here, the two objectives are all minimization problems. In this paper, the first objective value is a characteristic vector which is extracted from the boundary information of a closed planar shape. Then the extracted data is reduced by using some image processing techniques. For more details, please refer to Reference [Lee and Fischer, 1999]. Similarly, the second objective value is a part machine incidence matrix multiplied by other important manufacturing similarities such as production volume and processing time. The resulted two objective values must be scaled respectively to be equally compared in the same function. The following simple formular can be applied to scale each objective value for solution i:

$$f1_i = \frac{rawf1_i - rawf1_{min}}{rawf1_{max} - rawf1_{min}}$$

where:

$f1_i$ = scaled first objective value

$rawf1_i$ = raw(unscaled) first objective value for solution i

$rawf1_{max}$ = maximum raw first objective value among all solutions in a generation

$rawf1_{min}$ = minimum raw objective value among all solutions in a generation

The resulting scaled value range between zero and one in such a way that the best ones in the current population are mapped onto 0 and the worst onto 1. The others are proportionally mapped onto a value between 0 and 1. Pareto optimal solutions in a generation must have a pareto fitness value of one or greater. Non pareto solutions have a pareto fitness value between zero and one. The pareto exponent, p, amplifies the difference between pareto and non pareto solutions. [Schaumann *et al,* 1998]

2.5 Modified Fitness Function

A constraint is used in this system to limit maximum number of parts which belongs to a cell. The constraint value is a penalty and 1 here. The constraint value is used in the following equation to modify the pareto fitness:

$$F_{new} = F_{old} \times (\frac{1}{1+C_i})$$

Where C_i = 1, if i th chromosome violates the constraint.

F_{old} = old pareto fitness value.

F_{new} = new pareto fitness value.

The pareto fitness is decreased if the constraint is violated, but remains the same if no constraint is violated.

A specific number of 'very fit' chromosomes are chosen and copied directly into the succeeding generation without any genetic processing. These are the ones which give the lowest value of each of the two objective functions, and the one with the highest fitness value in the generation.

2.6 Selection

A Roulette Wheel method, which is selected in proportion to its fitness value, is adopted as a selection method in this study.

2.7 Genetic Operators

Genetic operators should efficiently produce an offspring that reflects the property of the problem well from their parents and keeps feasible solutions. Two basic genetic operators, crossover and mutation are used here.

2.7.1 Crossover Operator

A number of crossover operators have been investigated to deal with the permutation representation. In this study, the partially matched (or mapped) crossover (PMX) operator which has been well proven in this field was used.

The PMX is defined as follows: first, randomly select two crossover points in parent 1 (P1) and parent 2 (P2) simultaneously. These two points define a matching section that is used to effect a cross through position by position exchange operation. For instance, two random points (pos1=3, pos2=5) in a parent chromosome are assumed to be selected as shown below.

Parent 1 (P1) = (3 4 | 6 7 1 | 2 5)
Parent 2 (P2) = (7 3 | 4 2 5 | 6 1)

PMX proceeds by positionwise exchanges. First, switch elements between the two points. Then the 4 and 6, and the 2 and 7, and the 5 and 1 exchange places each other for the rest of the elements. Finally, the following two offsprings are produced:

Offspring 1 (O1) = (3 6 | 4 2 5 | 7 1)
Offspring 2 (O2) = (2 3 | 6 7 1 | 4 5)

where each chromosome contains information partially determined by each of its parents.

2.7.2 Mutation Operator

A mutation operator randomly selects one point in a chromosome and substitutes the value in that position by a random feasible group number which is not equal to the previous value.

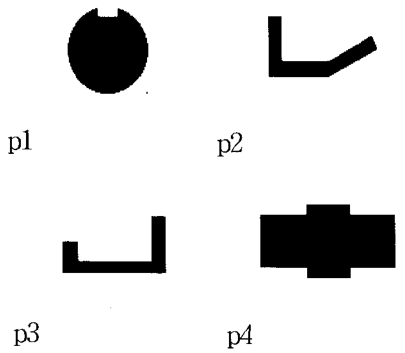
3. Numerical Illustration

To confirm the validity of the proposed system, a set of hypothetical 16 parts which is obtained from the other paper [Lee and Fischer, 1999] has been used and analyzed. The set contains geometric shapes and products with three important manufacturing attributes (process routes, process times, and production volumes). The part shapes are shown in Figure 1.

The three manufacturing attributes data can be combined by multiplying process times and

production volumes to the 0 or 1 part machine incidence matrix. Here, these values are randomly but meaningfully given by the user. In Fig. 2, each row implies the number of input parts. Five columns on the left imply shape attributes and eight columns on the right for manufacturing attributes.
Parameter values for a population size and the maximum number of generation are used as 100 and 100 respectively in this example. A pareto exponent value of 3 is used. R is 70. The number of parts is 16 and the number of desirable groups is given as 5. Crossover and mutation rates are 0.6 and 0.24 (0.6 x 0.4). These rates are obtained through the several preliminary experiments.

The proposed algorithm was programmed using Matlab software and operated at the Pentium 233 and 40MB RAM PC. The run time took about 6 min.



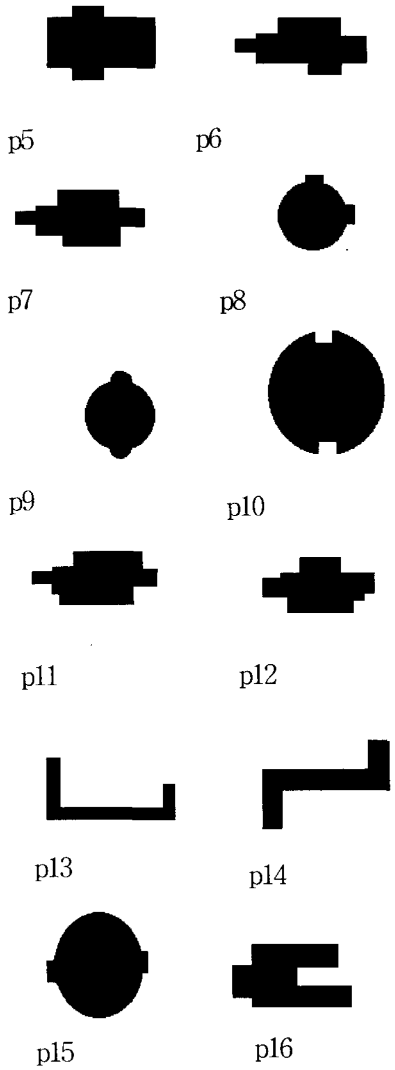


Fig. 1. Sixteen part shapes

0.9934	0.0040	0.0011	0.0001	0.0015	0	0	12	0	0	60	0	36
0.8500	0.1371	0.0091	0.0002	0.0037	0	0	0	1500	0	0	200	100
0.8873	0.1018	0.0065	0.0005	0.0039	0	0	0	1600	0	0	80	40
0.9565	0.0425	0.0006	0.0004	0.0000	60	0	50	0	60	0	0	0
0.9651	0.0305	0.0036	0.0007	0.0000	70	0	25	0	0	0	0	0
0.9131	0.0811	0.0049	0.0007	0.0002	280	91	0	210	84	140	0	0
0.9328	0.0602	0.0055	0.0015	0.0001	420	70	0	70	0	0	0	0
0.9923	0.0032	0.0033	0.0001	0.0011	0	100	100	0	0	75	0	0
0.9922	0.0053	0.0022	0.0003	0.0000	0	60	120	60	0	180	0	0
0.9948	0.0033	0.0015	0.0004	0.0000	0	12	18	0	0	120	0	42
0.9277	0.0641	0.0063	0.0016	0.0003	280	160	0	320	160	160	0	0
0.9284	0.0635	0.0067	0.0008	0.0004	560	168	0	240	80	0	0	0
0.8851	0.1045	0.0063	0.0004	0.0037	0	0	0	1400	0	0	120	80
0.8378	0.1561	0.0053	0.0007	0.0001	0	0	0	3000	0	0	180	120
0.9968	0.0020	0.0010	0.0001	0.0000	0	150	150	0	0	120	0	0
0.8781	0.0601	0.0412	0.0005	0.0200	200	100	0	100	0	50	0	0

Fig. 2. Input data for the sixteen parts (five columns on the left for shapes and eight columns on the right for manufacturing attributes)

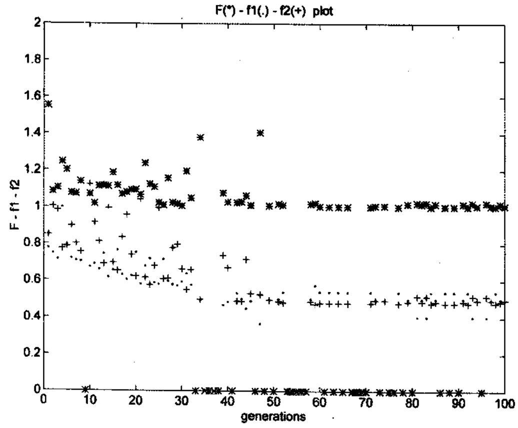


Fig. 3. The F, f1, and f2 plots generated from one to 100 generations

Table 1. Two grouping results among a set of pareto solutions obtained at the end of 100 generations.

	Quasi optimal solutions	f1	f2	F
Solution #1	3 5 5 1 1 4 4 3 3 3 4 4 5 5 3 2	0.3795	0.5156	1.1378
Solution #2	5 1 4 3 3 2 2 5 5 5 2 2 4 1 5 2	0.4945	0.4983	1.0102

Table 2. Part family list rearranged from the two grouping results

Part Family	Solution #1	Solution #2
PF #1	4, 5	2, 14
PF #2	16	6, 7, 11, 12, 16
PF #3	1, 8, 9, 10, 15	4, 5
PF #4	6, 7, 11, 12	3, 13
PF #5	2, 3, 13, 14	1, 8, 9, 10, 15

From the Fig. 3, it is notable that the program gradually converges at around 50 generations. F=0, which is marked as '*' at the bottom line, means non pareto solutions. Table 1 shows two example results among a set of pareto solutions obtained at the end of 100 generations. Table 2 shows the part family list rearranged from the two results. By carefully looking at the two results and Fig. 1 and Fig. 2, it is recognized that these results are feasible and good solution that satisfies both the given design and manufacturing attributes.

4. Conclusions

The most benefits from implementing a GT method can be realized when the parts are grouped based on both design and manufacturing attributes. In order to group parts based on both of the two similarities, a GA, which adopts a pareto fitness function, has been used and tested. Through the numerical experiments, the proposed system has proven to be a feasible approach to classify parts based on both the design and manufacturing similarities.

References

[1] Cho, J.B., and Gen, Mitsuo, "The Optimization of Multi-Objective Supply Chain Network Using Priority Based GA," *Proceedings of the KIIE/KORMS Spring Joint Conference*, 2010.

[2] Joines, J.A., King, R.E., and Culbreth, C.T., "Manufacturing Cell Design Using Hybrid Genetic Algorithms," *Proceedings of the Second Asia Pacific Conference on Industrial Engineering and Management System*, Kanazawa, Japan, 1999.

[3] Lee, S.Y., "PS NC Genetic Algorithm Based Multi Objective Process Routing," *Journal of the Korea Industrial Information System Research*, Vol. 14, No. 4, pp. 1 7, 2009.

[4] Lee, S.Y., and Fischer, G.W., "Grouping Parts on Geometrical Shapes and Manufacturing Attributes

using a Neural Network," *Journal of Intelligent Manufacturing*, Vol. 10, pp. 199 209, 1999.
[5] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, Berlin, 1992.
[6] Schaumann, E.J., Balling, R.J., and Day, Kirstin, "Genetic Algorithms with Multiple Objectives," *Proceedings of the 7th AIAA/USAF/NASA/ ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part 3, pp. 2114 2123, 1998.
[7] Venugopal, V. and Narendran, T.T., "A Genetic Algorithm Approach to The Machine Grouping Problem with Multiple Objectives," *Computers & Industrial Engineering*, Vol. 22, No. 4, pp. 469 480, 1992.

Sungyoul Lee

• BS, In-Ha University (Mechanical Engineering)
• MS, University of Texas at Arlington (Industrial Engineering)
• Ph. D., North Dakota State University (Industrial Engineering)
• Research Fields: Data Mining, Meta Heuristic Application, HCI
• Present: Professor, Department of Computer Science & Engineering, Kwandong University

논 문 접 수 일 : 2011년 07월 15일
1차수정완료일 : 2011년 11월 10일
게 재 확 정 일 : 2011년 12월 15일