Grouping SMT PCB Assembly using FCM Algorithm

K.H. Li[1], S. F. Lee[2], C. K. Li[3] and Dawei Lu[4]

(1. Hong Kong Institute of Vocational Education (Tsing Yi Campus), Hong Kong;
2. Hong Kong Productivity Council, Hong Kong;
3. Dept of Electronic & Information Engineering, The Hong Kong Polytechnic University;
4. Warwick Manufacturing Group, University of Warwick, England, United Kingdom)

Abstract: This is a further development work on grouping of printed circuit boards (PCBs) for Surface Mount Assembly in the electronic industry. The arrangement of PCBs among several surface mount machine lines is a typical kind of group technology (GT) problem. From literatures, there are various clustering techniques developed to solve the clustering problems. In this paper, fuzzy c-means clustering (FCM) is used to solve the PCBs grouping problem. Applying them in a real problem compares the results of the two methods. The result shows that there should be a systematic method to arrange the scheduling of PCB assemblies in electronic industry to improve the operations planning process.

Keywords: SMT assembly, PCB grouping, PCB-component matrix, clustering techniques, fuzzy c-means clustering.

Article No. : 1009-9492 (2002) 06-0112-05

1 Introduction

One of the characteristics of the electronic industries in Hong Kong is its flexible manufacturing environment. The type of production is typically a low volume, high mix type. Low volume means the production for each batch of production is rather small, typically 500 - 5000 units per batch. High mix means the variety of the production is large and many different types of PCBs are produced at the same time. In such situation, the changeover from one type of PCB to another type is quite frequent and unavoidable. But the time wasted in changing PCB types and component types is normally large comparing with production time [1]. As a result, the idle time of the PCB assembly machine lines is high and capital investment will be increased. So, there should be a systematic method to arrange the scheduling of PCB assemblies in electronic industry to improve the operations planning process by increasing the efficiency and hence reduce the manufacturing overhead costs.

Surface Mount Technology (SMT) was a new trend in electronics industry for about ten years. It is the assembling of surface mount components onto printed circuit boards to produce electronic products. In SMT assembly, component placement time is equal to the sum of placement times of individual component. Since SMT pick-and-place is an automated process, which is done by machine, the placement time for each component can be assumed to be constant. The utilization of the SMT machines can be increased by minimizing the set-up times of both PCB and component changeover. Component set-up time can be reduced by minimizing the number of component changeovers through grouping similar PCBs, which require the similar type of components.

If a new PCB is processed, the amount of component types to be changed depends on the amount of common component types of the new PCB with the previous PCB. If the amount of common component type is high between the previous PCB and the new PCB, the number of component changeover will be small. So, the set-up time for a PCB depends on the number of component changeover done after the earlier set-up, and the required number of component changeover depends on the existing component types in the machine feeders. The high cost of SMT PCB assembly machine justifies careful planning and control of the operations. The high assembly rate and the slow set-up time lead the particular attention to the set-up issue since the assembly rate is a few thousand components per hour while machine set-up is about an hour.

Various clustering methodologies had been widely applied

收稿日期：2002—07—16

in group technology of grouping of cellular manufacturing, such as similarity measures, rank order clustering, directive clustering algorithms and the popular one, c-means clustering, etc. They can also be applied in the planning and scheduling of SMT assembly by clustering of PCBs into groups according to the number of machine lines available in a company.

2 The SMT Assembly Environment

A typical SMT assembly line (pick & place process) used for automatic surface mount device (SMD) component assembly consists of several successive work phases (Fig. 1) [1]. At first, an empty PCB is passed to a screen printer, which deposits a thin layer of solder paste onto the PCB (or glue dispenser for inserts a glue dot) in order to solder the SMD components. Then the conveyor transfers the PCB to the chip mounter, in which the SMD chip components are mounted onto the surface of the PCB. The PCB will also go through the IC mounter for placement of integrated circuits (ICs), and finally to a reflow machine for solder reflow (soldering).

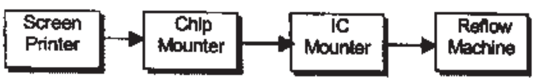

Fig. 1 SMD Pick & Place Assembly Line

Total Assembly Time Per PCB

The total assembly time for the SMT assembly can be summarized as follows:

Total assembly time = PCB set-up time (PS_t) + *Component set-up time* (CS_t) + *Component placement time* (CP_t) + *Idle time* (I_t)

Since the component placement time (CP_t) is fixed and if the idle time (I_t) is ignored. The total assembly time per PCB is therefore depends on the PCB and components set-up times. If we further assume that the PCB changing over and set-up time is the same for all PCBs, then we can conclude that the total assembly time per PCB is mainly depends on the component set-up time during each PCB type changes. So, the aim of the process engineer is to reduce the component set-up times in order to minimize the total assembly time.

3 C-means Clustering Algorithm

C-means clustering is a simple unsupervised learning method, which can be used, for data grouping or classification when the number of clusters is known. It consists of the following steps:

(1) Choose the number of clusters c.

(2) Set initial centers of clusters, v_1, v_2, $\cdots$, v_k to the arbitrarily selected c vectors from the training set.

(3) Classify each vector $x_l = [x_{l1}, x_{l2}, .. x_{ln}]^T$ (n is the dimension of input vectors) into the closest v_i by Euclidean distance measure:

$$\| x_l - v_i \| = \min (j) \ \| x_l - v_j \|$$

(4) Re-compute the estimates for the cluster v_i. Let $c_i = [v_{i1}, v_{i2}, .., v_{in}]^T$, v_{im} is computed by $v_{im} = (\sum x_{li} \in \text{cluster } i \ x_{lim}) / N_i$, where N_i is the number of vectors in the ith cluster.

(5) If none of the cluster centers (v_i, $i=1 \cdots k$) changes in step 4, stop; otherwise go to step 3.

We can make use of the c-means cluster centers and variances, which reflect the actual data distribution in the space, to generate membership functions for each input.

4 Fuzzy c-means (FCM) Algorithm

The important problem of feature extraction is the determination of the characteristics of the physical process. The task is to divide n objects $x \in X$ characterized by p indicators into c, $2 \leq c \leq n$ categorically homogenous subsets called "clusters". The objects belonging to any one of the clusters should be similar and the objects of different clusters as dissimilar as possible. The number of clusters, c, is normally not known in advance.

Fuzzy Clustering

In fuzzy clustering [2], a sample is assigned a membership function for each of the groups, so a fuzzy partition is made. One common weakness with conventional analytical methods, such as array-based clustering, hierarchical clustering (or similarity coefficient-based), and those that use mathematical programming, is that they implicitly assume that the part families are mutually exclusive and collectively exhaustive [3], i.e. each part can only belong to one part family. In reality, it is clear that some parts definitely belong to certain part families, but there exist parts whose lineages are much less evident. Fuzzy clustering is one approach proposed for a more accurate presentation of the problem in the environment of uncertain or inexact information [4].

Assume that there n parts machines to be grouped into c part families and corresponding machine cells. Conventional clustering methods implicitly assume that disjoint part families exist in the data set; therefore, a part can only belong to one part family. The classification result, thus, can be expressed as a binary matrix:

$$X=\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & \cdots & 0 \\ 0 & 1 & 1 & 0 & 1 & \cdots & 0 \\ 1 & 1 & 0 & 0 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 1 & 0 & \cdots & 1 \end{bmatrix}$$

And that

$$u_{ik} = 0 \text{ or } 1, \quad i=1,2,..,c;\ k=1,2,..,n \tag{2}$$

$$\sum_{i=1}^{c} u_{ik} = 1 \ , \ k=1,2,..,n \tag{3}$$

$$0 < \sum_{k=1}^{n} u_{ik} \leq n, \quad i=1,2,..,c \tag{4}$$

with i = part family, $i=1,2,..,c$, c = maximum clusters in the system, k = components types, $k=1,2,..,n$, and n = the n^{th} type of component

Constraint (2) ensures that u_{ik} equals 1 if the k^{th} part belongs to the i^{th} part family. Constraint (3) ensures that each part exactly belongs to one part family. Constraint (4) ensures that each part family consists of at least one part. But in many cases, part families are not completely disjoint; rather, the separation of part families is fuzzy. Consequently, the concept of fuzzy subsets could offer an advantage over conventional clustering and could allow a representation of the degree or grade of membership of a part associated with each part family. In fuzzy clustering, the classification results can be expressed as a matrix.

$$U=\begin{bmatrix} u_{11} & u_{12} & : & u_{1n} \\ u_{21} & u_{22} & : & u_{2n} \\ u_{31} & u_{32} & : & u_{3n} \\ : & & & \\ u_{c1} & u_{c2} & : & u_{cn} \end{bmatrix}$$

such that,

$$0 \leq u_{ik} \leq 1, \quad i=1,2,..,c;\ k=1,2,..,n \tag{6}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k=1,2,..,n \tag{7}$$

$$0 < \sum_{i=1}^{c} u_{ik} \leq n, \quad i=1,2,..,c; \tag{8}$$

The constraint sets (7) and (8) are similar to (3) and (4), but the u_{ik} are not restricted to values of 0 and 1 (they can be fractional value between 0 and 1). Therefore, a part can belong to several part families with different degrees of membership.

Fuzzy C-means Algorithm

The problem of fuzzy clustering has received much attention, and several algorithms for solving it have been proposed by Bezdek [4]. In this study, the generalized fuzzy c-means (FCM) algorithm is used, one that has been widely used. Since the number of possible U matrices that satisfy constraints (6) and (8) are infinite, we need an objective criterion to optimize the solution. Though it can be in any inner product norm, the sum of square error function is often used.

Fuzzy C-means Algorithm:

$$J_m(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{9}$$

$(d_{ik})^2 = |x_k - v_i|^2 = \sqrt{\sum_{j=1}^{p} (x_k - v_i)}^{\,2}$ is the desired *membership function:*

- $x_k \in X$, *where* $X = \{x_1, x_2, .., x_n\}$ *is a data set of n parts;*
- $v = (v_1, v_2, .. v_n)$ *is the centre of cluster* u_i, *i.e, the mean vector of the parts in the ith part family;*
- $U = [u_{ik}]$ *is a matrix of fuzzy c-partition of* X;
- $\{u_{ik}\}^m = \{u_1(x_k)\}^m$

Bezdek [4] proposed a Picard iteration procedure to solve for the matrix U in formulation (9). The procedure consists of eight steps:

(1) Choose the desired number of part family c, $1<c<n$.

(2) Choose a value m, $m>1$, *for* the degree of fuzziness.

(3) Choose a membership function, | . |.

(4) Choose a value ξ for the stopping criterion.

(5) Choose an initial classification matrix, $\boldsymbol{U}^{(0)}$.

(6) For iteration h = 0,1,2,⋯, calculate the mean vector $\{v_I^{(h)}\}$ for the fuzzy cluster centre.

$$v_i^{(h)} = \sum_{k=1}^{n} u_{ik}{}^m \cdot x_{kj} / \sum_{k=1}^{n} u_{ik}{}^m \tag{10}$$

(7) Update $U^{(h)}$. using $\{v_I^{(h)}\}$and:

$$u_{ik}^{(h+1)} = \left[\sum_{j=1}^{c} (d_{ik}{}^{h+1} / d_{jk}{}^{h})^{2/(m-1)} \right]^{-1}$$

(8) Compare $\boldsymbol{U}^{(h)}$ to $\boldsymbol{U}^{(h+1)}$. If $|u_{ik}^{(h+1)} - u_{ik}^{(h)}| \leq m$, stop; otherwise, go to step (6).

Classical (crisp) clustering algorithms generate partitions such that each object is assigned to exactly one cluster. Often, objects cannot adequately be assigned to strictly one cluster. In these cases, fuzzy clustering methods provide a much more adequate tool representing real-data structures. A typical flow chart of using the c-means algorithm is shown in Fig.2.

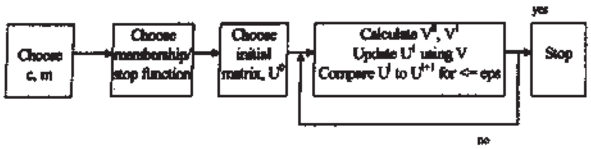

Fig. 2 the flow chart of the fuzzy c-means algorithm

5 Case Study

A set of PCB data (Appendix A) is used as a case study to illustrate the results solved by FCM. The studied company has a surface-mount facility at Dongguan, People's Republic of China. The company is a telecommunications electronic products manufacturer that has invested a large amount of capital in surface-mount equipment for the last ten years to meet the production output requirement. The company has three similar SMT assembly lines, and in order to maximize the utilization of the machines, the company operates the SMT machines lines for 24 hours a day. The surface-mount lines use batch production, with each PCB type produced at a quantity of about 1000-2000 day to meet the production requirements. So, the PCB types must be changed, on average, about 5-10 times per day. As a result, the average number of PCB changes for each line is about two-three per day. There are 8 PCB types and there are 59 component types in this case study. In the table of Appendix A, the required components for each PCB is identified with a '1' and an empty cell means the PCB do not need that component.

Fig. 3 A Typical SMT Line

The company has three SMT lines that are similar to the set-up as shown in Fig. 3. The current practice is to arbitrary select the PCB types and to divide them into three groups. The following two sections present the results of solving the grouping of SMT assemblies by ROC and FCM algorithms respectively.

Solved by FCM

The FCM algorithm was applied to the data set using group = 3, m=2, | . | = Euclidean distance, error =0.01. The initialization of $\boldsymbol{U}^{(0)}$ is obtained from the heuristic. Table 2 and 3 show the final classification matrix and the cluster centers means. The number of iterations is 20. The values in Table 2 indicate the degree of membership of each component associated with each component family. The larger the value, the higher the degree of the association. For example, the grades of membership for component 2 in part families 1, 2 and 3 are 0.19, 0.18 and 0.63 respectively. Since part 2 has the largest grade with part family 3, it will be assigned to part family 3. However, part 12 also has a fairly large grade with part family 1, it could also be assigned to part family 1.

Table 1 Grouping result using FCM algorithm

Group Number	PCB Types	Component Types
I	2	7, 16, 17, 21, 24, 25, 26, 28, 38, 40, 41, 42, 43, 44, 52, 53, 56, 58, 59
II	1, 6	1, 3, 4, 8, 9, 10, 12, 14, 15, 18, 19, 20, 22, 23, 29, 30, 31, 33, 37, 46, 49, 50, 51, 55, 57
III	3,4,5,7,8,	2, 5, 6, 11, 13, 27, 32, 34, 35, 36, 39, 45, 47, 48, 54

Table 2 Memberships for cluster

PCB Type (j)	*Memberships for cluster (i), v*			*Max. of (v)*	*1st Choice*	*2nd Choice*
	1	*2*	*3*			
1	0.23	0.60	0.51	0.60	2	3
2	0.76	0.35	0.39	0.76	1	3
3	0.30	0.34	0.70	0.70	3	2
4	0.08	0.11	0.77	0.77	3	2
5	0.09	0.14	0.82	0.82	3	2
6	0.10	0.23	0.14	0.23	2	3
7	0.20	0.28	0.47	0.47	3	2
8	0.09	0.16	0.81	0.81	3	2

6 Results

The following table (Table 2) shows the membership function for PCBs (clusters) after the FCM computation.. The FCM algorithm was programmed in C programming language complied by C++ Builder, and run on a Pentium 4 (1.4G) PC. Appendix D shows the run results after 20 iterations, and Appendix E shows the whole C program for the c-means algorithm.

7 Comparing Results

With FCM algorithm, and according to the clusters formation as shown in Table 2, the arrangement is shown in Fig. 4. PCB # 2 is the only one member in group 1 which is assigned to the SMT line #1. PCB # 1 and 6 belong to group 2 which are assigned to the SMT line #2 and finally PCB # 3, 4, 5, 7, & 8 belong to group 3 which are assigned to the SMT line #3.

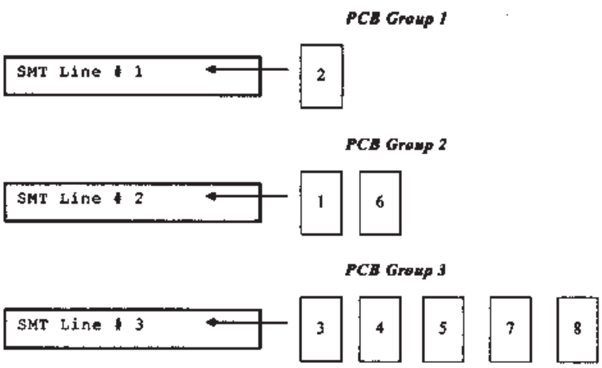

Fig. 4 Grouping result of FCM algorithm

The result of the FCM algorithm is compared with two arbitrary grouping results are shown in Table 3. Arbitrary grouping means that we select the candidates in each group (totally there are three groups as indicated in the case study) in a randomly selection basis. The last group (full component) is an extreme case (that is undesirable to occur) that we install all component types at all machine lines at the very beginning (of course it is assumed that the SMT machine has enough feeder space to install all the component types) . Since the component set-up

Table 3 Compare results

		SMT lines			Total set-up time index
		SMT #1	SMT #2	SMT #3	
FCM	PCB type	2	1,6	3,4,5,7,8	
	Comp	30	30	40	100
Arbitrary group #1	PCB type	1,2	3,4	5,6,7,8	
	Comp	45	30	37	112
Arbitrary group #2	PCB type	1,4	2,5,6	3,7,8	
	Comp	34	48	39	121
Full component	PCB type	–	–	–	
	Comp	59	59	59	177

time is directly proportional to number of component change-over during each PCB change within that group and the initial component set-up time when the PCB is first loaded for assembly (indicated as 'comp' in the table) , we can summarize the total set-up time index.

From the data in Table 3, we can roughly summarize the improvement of the FCM algorithm when comparing with the other arrangements. If we use the full component group (the last group) as the reference (that means to install types of components in all machine lines) , the improvement (% reduction in set-up time) will be equal to $(177-100)/177\times100\% = 43.5\%$ if we assume the set-up time to be one unit time. Where the improvement of using FCM is equal to $(121-100)/121\times100\% = 17.4\%$ when comparing the data for the arbitrary group #2. If we use a rough estimation of about 5 minutes to install a component type in a feeder and then install the feeder onto a SMT machine, the improvement of using FCM will be about $5\times77 = 385$ minutes per day when comparing with the full component group, and of $5\times21 = 105$ minutes per day when comparing with the arbitrary group #2 respectively.

8 Conclusion

This is a further development works on grouping of PCBs in Surface Mount Assembly in the electronic industry. In this paper, a famous methodology, fuzzy c-means clustering (FCM) is used to solve the PCBs grouping problem. Applying them in a real problem compares the results of the method. The result shows that the performance of the FCM is better than that of the arbitrary assignment. Also it is recommended that that there should be a systematic method to arrange the scheduling of PCB assemblies in electronic industry to improve the operations planning process.

References:

[1] *K.H. Li, C.K. Li, S.F. Lee and D. Lu, "Surface Mount Implementation Problems in Hong Kong" , Oct. 2000, Circuits Assembly, pp. 65-82.*

[2] *C.H. Chu, C. Hayya, "A fuzzy clustering approach to manufacturing cell formation, International" Journal of Production Research, Vol. 29, No. 7, pp.1475-1487.*

[3] *H. Xu and B. Wang, "Part family formation for GT application based on fuzzy mathematics" , International Journal of Production Research, vol. 27, No. 9, pp.1637-1651.*

[4] *J. C. Bezdek, "Pattern Recognition with fuzzy objective function algorithms" (New York Plenum Press) , pp.23-71.*