Lead Facility. To determine the optimal position of the Follower Facility, one-dimensional (1-D) search is carried out, using the modified quadratic-fit procedure [36], along all the four sides of the Lead Facility and for both possible orientations of the Follower Facility. Among the eight optimal positions, the Follower Facility is placed at a position (with appropriate orientation) that corresponds to the minimum value of the cost function. Thereafter, the boundary of the cluster formed by these two facilities is determined. Also, an overlap-free search path along the cluster boundary is defined for the next facility (in order) to be placed.

Step #2 (Placement of Remaining Follower Facilities)

For the next Follower Facility in order, 1-D search is carried out along the overlap-free search path based on the previous cluster boundary. This improves the efficiency of the procedure as it does not require checking of overlaps following 1-D search. This process is repeated after changing the orientation of the new facility. The new facility is then placed at a position (with appropriate orientation) that corresponds to the minimum (best) value of the cost function. Now, instead of recalculating the boundary of the cluster after the placement of the new facility, as is carried out in the original cluster boundary technique [30], the previous cluster boundary is simply updated to reflect the addition of new facility. This significantly reduces the computation cost needed for recalculating the new boundary, especially for large-size problems, and thereby allows multiple firing orders to find a better quality layout without excessive computation cost. For the updated cluster boundary, the new overlap-free search path is defined for the next facility (in order) to be placed.

Step #2 is repeated for all remaining facilities, one at a time, until all the facilities have been placed at their optimal positions with optimal orientations. This completes the optimization procedure if the only constraint was non-overlapping of facilities. However, if a constraint on placing the facilities within an enclosing rectangle is specified either directly or by specifying the maximum permissible white area (additional area in the enclosing rectangle other than the summation of actual areas of all facilities), then the following procedure is also included in step #2.

If at any stage, placement of a new facility violates the enclosing rectangle constraint, that is, the facility is placed outside the permitted area of enclosing rectangle, a penalty function is applied and accordingly that placement position is excluded from further consideration. Furthermore, to minimize the accumulation of white space (area), if two or more placements of a new facility produce cost functions whose difference is within a specified relative threshold, then the placement with the least contribution of white space (area) is given preference.

The above-mentioned procedure was implemented in a computer program written in C++ and C#. The modified Cluster Boundary Technique will be referred to as mCBT in the discussion of test results.

IV. TEST RESULTS AND DISCUSSION

The presented technique (mCBT) was tested with UA-FLP problems taken from published benchmark or test problems. For the first set of test problems, benchmark problems presented by a general-purpose block layout design package, VIP-PLANOPT [37], were tried and compared with its optimal layouts. VIP-PLANOPT is a well-known commercially available software package for general-purpose block layout optimization. Its optimization technique is based on a pseudo-exhaustive search procedure.

The performance of the presented technique was compared with three benchmark problems of VIP-PLANOPT involving 8, 28, and 100 blocks (or facilities) of unequal-areas. For these benchmark problems the only constraint was non-overlapping of facilities. The results are shown in Table 1 below. As can be observed from this table, mCBT always produced a better quality optimal layout as compared to the best results obtained by VIP-PLANOPT. Also, the improvement in cost function increases as the number of facilities increase. For the 100-facility problem, there is 9.77% improvement in the cost function value as compared to the best value achieved by VIP-PLANOPT. This shows that the presented technique is especially suited for solving large-size layout problems. For the 100-facility problem, mCBT took only 56s on Acer laptop using Intel Core i5 CPU running at 2.27 GHz and having 4GB RAM. The layout designs obtained by mCBT for the three benchmark problems are given in Figures 1 to 3.

The data for another layout problem involving 30 unequal-area facilities were taken from [11] where it was solved by using individual Tabu search and particle swarm optimization techniques as well as by hybrid techniques combining both Tabu search and particle swarm optimization. The layouts were obtained without imposing any constraint on enclosing rectangle or specifying any upper bound on white area. The same problem was solved by mCBT with three different values (15%, 25%, 40%) of upper bound on white area in the final layout. It may be mentioned here that the objective function defined in [11] is twice the cost function defined by Eqn. (1). Accordingly, the cost obtained by mCBT was doubled for