AN IMPROVED SIMULATED ANNEALING FOR FACILITY LAYOUT PROBLEMS IN CELLULAR MANUFACTURING SYSTEMS

TAI-YUE WANG,* HER-CHANG LIN and KUEI-BIN WU
Department of Industrial Management Science, National Cheng Kung University, Tainan, Taiwan, the Republic of China

(Received 16 October 1997)

Abstract—In this paper, we formulate a model solving both inter-cell and intra-cell facility layout problems for cellular manufacturing systems. This model minimizes the total material handling distance on the shop floor. Due to the complexity of the problem, we propose an improved simulated annealing algorithm to solve this problem. This algorithm modifies the generation mechanism of neighborhood configurations. This new generation mechanism can always generate a neighborhood configuration that satisfies all of the zoning constraints. Then, the comparison between the improved simulated annealing algorithm and Kouvelis’s is conducted. The results show that the improved simulated annealing algorithm produces the same solution quality while requiring less computation time as the problem size is increased. © 1998 Elsevier Science Ltd. All rights reserved.
2-way exchange algorithm;
3-way exchange algorithm;
Computerized Relative Allocation of Facilities Technique (CRAFT);
Modified Penalty (MP) algorithm;
Tabu Search (TS) algorithm;
Simulated Annealing (SA) algorithm.

Skorin-Kapov [5] made a comparison between TS and SA. The SA algorithm not only generated solutions as good as the TS algorithm but required less computation time for larger problems. Heragu and Alfa [6] compared the 2-way exchange algorithm, 3-way exchange algorithm, MP algorithm, and SA algorithm. The SA algorithm was better than the others, on both solution quality and efficiency. Jajodia et al. [7] compared CRAFT, SA algorithm, and other heuristic algorithms. They also arrived at the same conclusion. Similar comparisons in the literature can be found in Kouvelis and Chiang [8].

The facility layout problem in CMS hasn’t captured researchers’ attention as much as cell formation in the past two decades. The layout problem, however, plays an important role on the shop floor. A poorly designed layout will result in poor productivity, increased work in process, disordered material handling, and so on. Only a few researches have dealt with this subject. Tzeng [9], and Twu [10] formulated models for solving facility layout problems in a single cell. Alfa et al. [11] formulated a model for solving facility layout problems in CMS. Alfa’s model assumes that the location of the cells is predetermined, hence it only solves the intra-cell layout problems. Das [12] also formulated a model for solving inter-cell layout problems. This model doesn’t consider intra-cell layout, however it does influence the final inter-cell layout. To date, no researcher has demonstrated any model to solve both inter-cell and intra-cell facility layout problems for CMS.

In this paper, we present a model that minimizes the total material handling distance on the shop floor. This model solves both inter-cell and intra-cell facility layout problems in CMS simultaneously. We also present an improved simulated annealing algorithm that modifies the generation mechanism of neighboring configurations to find the solution. Then, the solution quality (measured in total distance) and computational efficiency (measured in CPU time) with Compulsion simulated annealing algorithm [13] are compared. The rest of this paper is as follows. In Section 2, the formulation of the facility layout problem is presented. Section 3 reviews the simulated annealing algorithm and presents the Improved simulated annealing algorithm. Computational results and performance comparisons are given in Section 4. The conclusions of this study are given in Section 5.
2. CMS FACILITY LAYOUT PROBLEM

In this section, we will formulate the model for the facility layout problem in CMS. The concepts of the formulation are shown in Fig. 2, and the assumptions are described as follows:

1. The cell formation is completed first, i.e. what kinds of machines belong to which cells is known.
2. Each cell is laid out in a U-shape. The length and width of the U-shape is known. Besides, material handling flows along the U-shape.
3. Workstations are regarded as cells.
4. The distance between any two neighboring machines in the same cell is equal. The distance between any two neighboring cells is equal.
5. The loading and unloading point is at the center of each machine or workstation.
6. The loading point for the cell is at the entry point of the cell and the unloading point for the cell is at the leaving point of the cell.
7. The shape and space of the shop floor is not restricted.

The variables and symbols used in the model are defined as follows:

- \( c \): total number of cells, including workstations
- \( m \): total number of machines in the system including workstations
- \( MF_{ij} \): material flow between machines \( i \) and \( j \)
- \( NMC(p) \): the number of machines assigned to cell location \( p \)
- \( E \): the cell to which machine \( i \) is assigned
  \[ X_{ik} = \begin{cases} 1, & \text{if machine } i \text{ is assigned to machine location } k \\ 0, & \text{otherwise} \end{cases} \]
- \( Y_{bp} \): the cell to which cell \( b \) is assigned to cell location \( p \)
  \[ Y_{bp} = \begin{cases} 1, & \text{if cell } b \text{ is assigned to cell location } p \\ 0, & \text{otherwise} \end{cases} \]
- \( DR_{ikjl} \): the inter-cell material handling distance between machine \( i \) and \( j \) when machine \( i \) is assigned to machine location \( k \) and machine \( j \) is assigned to machine location \( l \)
- \( DA_{ikjl} \): the intra-cell material handling distance between machine \( i \) and \( j \) when machine \( i \) is assigned to machine location \( k \) and machine \( j \) is assigned to machine location \( l \)
- \( D_{ikjl} \): total material handling distance between machine \( i \) and \( j \), \( D_{ikjl} = DR_{ikjl} + DA_{ikjl} \)

The facility layout problem for CMS is then formulated as follows:

\[
\text{Minimize } \sum_{p=1}^{c} \sum_{q=1}^{c} \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{l=1}^{m} MF_{ij} D_{ikjl} X_{ik} X_{jl} Y_{Ep} Y_{Eq}
\]

(1)

Fig. 2. The concepts of facility layout in CMS (for \( c = 4 \), \( m = 9 \)).
subject to \( \sum_{i=1}^{m} X_{ik} = 1 \), \( \forall k = 1, 2, \ldots, m \) (2)

\[ \sum_{k=1}^{m} X_{ik} = 1 , \quad \forall i = 1, 2, \ldots, m \] (3)

\[ \sum_{b=1}^{c} Y_{bp} = 1 , \quad \forall p = 1, 2, \ldots, c \] (4)

\[ \sum_{p=1}^{c} Y_{bp} = 1 , \quad \forall b = 1, 2, \ldots, c \] (5)

\[ \sum_{r=1}^{p-1} \sum_{b=1}^{c} Y_{br}NMC(b) < \sum_{k=1}^{m} kX_{ik} , \quad \forall i = 1, 2, \ldots, m \] (6)

\[ \sum_{k=1}^{m} kX_{ik} \leq \sum_{r=1}^{p} \sum_{b=1}^{c} Y_{br}NMC(b) , \quad \forall i = 1, 2, \ldots, m \] (7)

\[ X_{ik} = \begin{cases} 1, & \text{if machine } i \text{ is assigned to machine location } k \\ 0, & \text{otherwise} \end{cases} \]

\[ Y_{bp} = \begin{cases} 1, & \text{if cell } b \text{ is assigned to cell location } p \\ 0, & \text{otherwise} \end{cases} \]

Objective function (1) minimizes the total material handling distance on the shop floor for this system. Constraints (2) and (3) ensure that each machine location is assigned to one machine and each machine is assigned to one machine location. Then constraints (4) and (5) ensure that each cell location is assigned to one cell and each cell is assigned to one cell location. Lastly, constraints (6) and (7), the zoning constraints, ensure that machines in the same cell \( E_i \) are assigned to the same cell location. In other words, the \( k \)th machine location must be between the total number of machines in the first \( p - 1 \) cells and the total number of machines in the first \( p \) cells.

3. SIMULATED ANNEALING

3.1. Traditional simulated annealing

Kirkpatrick et al. [14] introduced the concept of a simulated annealing algorithm in 1983. It is an analogy between the physical annealing process of solids and the problem of solving large combinatorial optimization problems. Solutions in a combinatorial problem are equivalent to the energy of a state [15]. In the searching process, the simulated annealing algorithm accepts not only better but also worse neighboring solutions with a certain probability. This means that the simulated annealing algorithm has the ability to escape from local minima. Therefore, it can find high quality solutions that do not strongly depend upon the choice of the initial solution compared to local search algorithms. In other words, the algorithm is effective and robust. Furthermore, it has been proven that the computation time of this algorithm has a polynomial upper bound [15].

The simulated annealing procedure includes four basic components [16]:

1. Configurations: all of the possible solutions for the combinatorial problem, i.e. the states.
2. Move set: a set of allowable transitions. These transitions must be able to reach all of the configurations.
3. Cost function: a measure of how good any given configuration is.
4. Cooling schedule: to anneal the problem from random to a good, frozen solution.

If one wants to use the simulated annealing algorithm, the initial temperature, rules for decreasing the value of the temperatures, the number of transitions at each value of temperature, and when annealing should be stopped must first be decided.

Then one can use Sridhar and Rajendran’s [2] annealing procedure as follows to find the solution:

Step 1. Generate an initial configuration $S$.
Step 2. Get an initial temperature $T > 0$.
Step 3. While not yet satisfy the stop criterion.
   Step 3.1 Perform the following loop $L$ times.
      Step 3.1.1 Pick a random neighbor $S'$ of $S$.
      Step 3.1.2 Let $\Delta = \cos(S') - \cos(S)$.
      Step 3.1.3 If $\Delta \leq 0$ then set $S = S'$.
      Step 3.1.4 If $\Delta > 0$ then set $S = S'$ with probability $\exp(-\Delta/T)$.
Step 3.2 Set $T = rT$.
Step 4. Return $S$.

Step 3.1.1 is known as the generation mechanism of neighboring configurations. Generally, the generation mechanism for facility layout problems without zoning constraints randomly selects two machines and exchanges the location of the selected machines. Kouvelis et al. [13] presented two different approaches for the presence of zoning constraints in the facility layout problem. The first approach is referred to as Compulsion simulated annealing. It takes the zoning constraints into account when generating the initial and neighboring configurations. They also modified the generation mechanism as follows [13]:

Step 3.1.1 Pick a random neighbor $S'$ (randomly select two machines then exchange the location of the selected machines) of $S$. If $S'$ satisfies the zoning constraints, continue with Step 3.1.2. Otherwise, repeat Step 3.1.1.

The second approach is referred to as Penalty simulated annealing. Instead of using Lagrange multipliers, the zoning constraints are taken into account in the cost function. It imposes high penalty terms to each configuration that violates the constraints. This forces the algorithm to choose configurations that satisfy the constraints most of the time. They concluded that the Compulsion method outperforms the Penalty method with respect to solution quality as well as computational efficiency.

3.2. Cooling schedule

The cooling schedule of this paper is similar to Kouvelis et al. [13] except the stop criterion.

3.2.1. Initial temperature. In physical analogy, the initial temperature should be large enough to heat up the solid until all particles are randomly arranged in the liquid phase. This means that at the beginning of the annealing process, the transitions are able to reach all of the configurations. By this property, the algorithm can find a solution that does not strongly depend upon the initial configuration. The initial temperature $T_0$ can be determined by means of a cost-increasing transition which would be accepted in the beginning of the annealing process with probability $P_0$. The mean cost increasing $\bar{\Delta}$ of the cost-increasing transitions is then computed. In the calculation, $T_0$ is calculated as follows.

$$T_0 \approx \frac{\bar{\Delta}}{\ln(P_0^{-1})}$$

(8)

The number of transitions for calculating $\bar{\Delta}$ is a fraction (SAMPLE) of the total number of neighboring configurations. The total number of neighboring configuration is the order of $m^2$.

3.2.2. Length of the Markov chains. The annealing process transfers from one configuration to one of its neighbors with a certain probability, this is equivalent to a Markov chain. Thus, one can consider each value of temperature as a Markov chain. In fact, the number of accepted transitions are smaller with lower values of the temperature. Therefore, we should determine the upper bound of the Markov chain length. The upper bound can be a proportion (LENPERCENT) of the total number of neighboring configurations.
3.2.3. Rules for decreasing the temperature. For a certain value of temperature, the temperature is reduced when the numbers of transitions reach the upper bound of the Markov chain length. The control parameter, i.e. the reduction ratio of temperature, usually is chosen for small temperature changes. The Markov chain more easily leads to an equilibrium state if the temperature change is small. Hence, we use the decrement rule as follows.

\[ T_k = rT_{k-1} \quad k = 1, 2, 3, \ldots \]  

The control parameter \( r \) (RATIO) is small but close to 1.

3.2.4. Stop criterion. The annealing process is terminated when the system is frozen, i.e. the value of the cost function of the solution does not improve after a certain number of consecutive Markov chains. In this paper, the annealing process is terminated if either of the following two conditions is satisfied. (1) The process will stop if the number of accepted transitions is less than a fraction (MINPERCENT) of the total number of attempted transitions. (2) The annealing process will also stop if the current best configuration remains unchanged for \( \ln(\Theta) \) number of temperature reduction steps. Aarts and Korst [15] have proven that the upper bound of the total number of temperature reduction steps (i.e. the number of Markov chains) is proportional to \( \ln(\Theta) \); \( \Theta \) is the solution space that denotes the finite set of all possible solutions. In our problem, \( \Theta \) is equivalent to the factorial of \( m \). Most of the elements in \( \Theta \), however, are infeasible solution because there are too many zoning constraints, so we use \( \ln(\Theta) \) as the upper bound of the number of Markov chains.

3.3. Improved simulated annealing algorithm

The facility layout problem in CMS shown in Section 2 is too complex to generate a neighboring configuration that satisfies all of the zoning constraints. Thus the Compulsion method becomes inefficient, because it is easily trapped into the loop generating a neighboring configuration, and spends a significant proportion of time testing if the configuration is feasible (i.e. solutions that satisfy all of the zoning constraints). To escape from the trap, we introduce a new generation mechanism that can always generate a neighboring configuration which satisfies all of the zoning constraints. The new generation mechanism is described below:

1. Pick a random neighbor \( S' \) (randomly select two machines then exchange the location of the selected machines) of \( S \). If \( S' \) satisfies the zoning constraints, continue with Step 3.1.2. Otherwise, restore the original condition, exchange the location of the cells that the selected machines belong to, replace \( S' \) with this new configuration, continue with Step 3.1.2.

The annealing procedure of the Improved simulated annealing is as shown in Fig. 3.

3.4. Parameter analysis

To find the appropriate parameters for these two annealing procedures, we use the parameters provided by Kouvelis et al. [13] at first then change one parameter at a time. The value of parameters provided by Kouvelis et al. are as follows: \( P_0 = 0.4 \), \( \text{SAMPLE} = 0.08 \), \( \text{RATIO} = 0.99 \), \( \text{LENPERCENT} = 0.95 \), \( \text{MINPERCENT} = 0.01 \). The problem size used to do the parameter analysis is \( m = 8 \), \( c = 3 \). Ten replications with different random seed are run for each combination of parameters. We choose the value of the parameter that has the best average solution quality. If a tie occurs, we choose the value with less average CPU time. The parameter analysis results are shown in Tables 1–5. The asterisk (*) indicates the incumbent parameter for searching the standard parameter set. The standard parameter set used for these two simulated annealing procedures are shown in Table 6.

4. COMPUTATIONAL RESULTS

The above two different simulated annealing procedures use the cooling schedule and parameters in accordance with Section 3.2 and Table 6. These two procedures were programmed in C language and ran on a PC with 100 MHz Pentium® CPU. The test problems were randomly generated with feasible cell dimension in mind. We ran each test problem ten times with different random seeds. To provide the comparisons of solution quality, the optimal solutions of test problems are found by enumeration except the problem \( m = 16 \), \( c = 6 \) (Table 7).
The computational results are shown in Table 8. The solution quality from either Compulsion or Improved procedure is rather good. In each test problem, the best and average solution of the ten runs is almost equal to the optimal solution. Even in the worst case ($m = 14, c = 5$), the
### Table 1. Parameter $P_0$

<table>
<thead>
<tr>
<th>$P_0$</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsion procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78906.25</td>
<td>79094.25</td>
<td>78906.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>10.13</td>
<td>8.59</td>
<td>8.99</td>
<td>9.79</td>
<td>10.55</td>
<td>12.66</td>
</tr>
<tr>
<td>Improved procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78826.25</td>
<td>78746.25</td>
<td>78906.25</td>
<td>78906.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>10.35</td>
<td>10.53</td>
<td>10.19</td>
<td>9.91</td>
<td>10.61</td>
<td>11.17</td>
</tr>
</tbody>
</table>

### Table 2. Parameter SAMPLE

<table>
<thead>
<tr>
<th>SAMPLE</th>
<th>0.04</th>
<th>0.08</th>
<th>0.12</th>
<th>0.16</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsion procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78906.25</td>
<td>79094.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>9.41</td>
<td>9.69</td>
<td>11.57</td>
<td>9.31</td>
<td>10.39</td>
</tr>
<tr>
<td>Improved procedure</td>
<td>78826.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78906.25</td>
<td>79159.50</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>10.10</td>
<td>10.62</td>
<td>10.01</td>
<td>10.32</td>
<td>10.56</td>
</tr>
</tbody>
</table>

### Table 3. Parameter RATIO

<table>
<thead>
<tr>
<th>RATIO</th>
<th>0.99</th>
<th>0.98</th>
<th>0.96</th>
<th>0.92</th>
<th>0.85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsion procedure</td>
<td>78826.25</td>
<td>78746.25</td>
<td>78826.25</td>
<td>78746.25</td>
<td>78746.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>11.13</td>
<td>9.40</td>
<td>9.32</td>
<td>10.16</td>
<td>9.16</td>
</tr>
<tr>
<td>Improved procedure</td>
<td>79211.25</td>
<td>78906.25</td>
<td>78906.25</td>
<td>78746.25</td>
<td>78746.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>9.36</td>
<td>11.81</td>
<td>14.31</td>
<td>10.19</td>
<td>9.62</td>
</tr>
</tbody>
</table>

### Table 4. Parameter LENPERCENT

<table>
<thead>
<tr>
<th>LENPERCENT</th>
<th>1.00</th>
<th>0.95</th>
<th>0.90</th>
<th>0.85</th>
<th>0.80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsion procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>9.48</td>
<td>9.68</td>
<td>8.90</td>
<td>8.80</td>
<td>8.26</td>
</tr>
<tr>
<td>Improved procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>9.63</td>
<td>9.17</td>
<td>9.88</td>
<td>9.98</td>
<td>10.25</td>
</tr>
</tbody>
</table>

### Table 5. Parameter MINPERCENT

<table>
<thead>
<tr>
<th>MINPERCENT</th>
<th>0.01</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsion procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>79639.25</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>8.49</td>
<td>8.32</td>
<td>7.21</td>
<td>6.46</td>
<td>3.35</td>
<td>3.21</td>
</tr>
<tr>
<td>Improved procedure</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>78746.25</td>
<td>79449.38</td>
</tr>
<tr>
<td>AVG time (sec)</td>
<td>10.46</td>
<td>8.52</td>
<td>8.64</td>
<td>9.30</td>
<td>7.76</td>
<td>4.04</td>
</tr>
</tbody>
</table>

### Table 6. The standard parameter set used in this paper

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Compulsion procedure</th>
<th>Improved procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>RATIO</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Total number of neighbors</td>
<td>$O(m^2)$</td>
<td>$O(m^2)$</td>
</tr>
<tr>
<td>Transitions for calculation $\Delta$</td>
<td>0.16$m^2$</td>
<td>0.12$m^2$</td>
</tr>
<tr>
<td>UPPERLEN</td>
<td>0.80$m^2$</td>
<td>0.95$m^2$</td>
</tr>
<tr>
<td>MINPERCENT</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>TIME</td>
<td>$\ln[m^2]$</td>
<td>$\ln[m^2]$</td>
</tr>
</tbody>
</table>
error is only around 6% compared to the optimal solution. The solution quality has no significant difference in these two procedures. The results are depicted in Fig. 4.

Although there is no significant difference in solution quality, the difference in computational efficiency is quite significant. Table 9 shows the computational time for both procedures, and Fig. 5 gives the graphical presentation. One can see that the computational efficiency of the Compulsion procedure, however deteriorates quickly when the problem size becomes large. One can explain the
results as follows: the Compulsion procedure, although it maintains a higher degree of randomness than the Improved procedure, is easily trapped into the loop generating a neighboring configuration and spends a significant proportion of time testing if the configuration is feasible when the problem size becomes large. The advantage of the Improved procedure is that it always generates a neighboring configuration that satisfies all of the zoning constraints. Thus, the Improved procedure will require less computational time than the Compulsion procedure as the problem size increases.

5. CONCLUSION

In this research, we develop a model for facility layout problems for CMS. We also present an Improved simulated annealing algorithm for the solution of this model. The computational results show that the Improved simulated annealing algorithm can produce a solution as good as the Compulsion simulated annealing algorithm, while having less randomness. Also the Improved simulated annealing algorithm has very good computational efficiency (i.e. requires less CPU time) when the problem increases. This implies that the Improved simulated annealing algorithm provides an opportunity to find a better solution with the same time period than the Compulsion simulated annealing algorithm. This goal is achieved because the Improved simulated annealing algorithm can search longer Markov chain lengths or search a greater number of times than the Compulsion simulated annealing algorithm. Thus, the Improved simulated annealing algorithm will provide a new direction for other problems solved by simulated annealing algorithms.

REFERENCES