Genetic Search for Facility Layout Design under Interflows Uncertainty

Runwei CHENG^{*1} Mitsuo GEN^{*2} Tatsumi TOZAWA^{*1}

1. Introduction

Facility layout design has been the subject of an interdisciplinary interest for over two decades. It is usually viewed as an optimization problem and the best layout is picked out by optimizing some measure of performances subject to a set of constraints[1].

Most traditional formulations have assumed that the volume of flow between facility pairs is a known fixed quantity and facility layout is treated as a static problem over planning horizon [2]. However such a deterministic assumption is not suitable for modern manufacturing systems. The flows among facilities may change from period to period due to the dynamic nature of businesses, growth, demand fluctuation, and product mix. If some of these changes are planned and hence known *a priori*, the *dynamic layout model* of Rosenblatt can be used to handle such changes[3]. Unfortunately, changes in product mix, machine breakdowns, seasonal fluctuations and demand are uncertainty in nature. Under these circumstances, designers tend to get a satisfactory layout rather than an optimal layout.

There are two approaches proposed for modeling such uncertainties. One is flexible or probabilistic approach [4] and another is robustness approach [5,6]. If we adopt the flexible approach, we must specify probability distribution for material flow; if we adopt the robustness approach, we must provide several demand scenarios and the optimal solutions for each scenario. As we know that the optimal design of physical layout must be resolved in the early stage of manufacturing system design, and product mix is highly uncertainty during design phase. Giving an exact probability distribution or giving some precise demand scenarios are as difficult as the optimal

design of facility layout itself. In this case, the estimation of material flow among facilities depends highly on the subjective judgments and/or preference of manager or designer, which are of imprecision in nature.

In this paper, we propose an alternative for handling such uncertainties: fuzziness approach [7]. Fuzzy set theory is a perfect means for modeling uncertainty or imprecision arising from mental phenomena. Its capability of handling uncertainty and imprecision is crucial to all situations when the available information is not precise and the associated uncertainty cannot be ignored[8]. The subjective or preference information can usually (but not always) be represented as a convex fuzzy number[9]. Thus we can use the mathematics of fuzzy calculus to handle these uncertainty and imprecise. The practical merit of fuzziness approach is that there is no need to force manager or designer to give a precise product mix or a probability distribution.

Because the combinatorial nature of facility layout problem, heuristic approach is the most promising way for solving practical size layout problem. Recently, the interesting in the application of Genetic Algorithms to facility layout design has been growing rapidly[20]. Tate and Smith applied genetic algorithm to shape-constrained unequal area facility layout problem[10]. Cohoon *et al.* proposed a distributed genetic algorithms for floorplan design problem [11]. Tam reported his experiences of applying genetic algorithm to facility layout problem [12]. We apply genetic algorithms to fuzzy facility layout problem. Polish expression is adopted as the coding scheme of chromosome. A penalty to the violation of aspect ratios for facilities is used to guide genetic search towards to the promising part of solution space. Simulation results demonstrate that genetic algorithm and fuzziness approach can be efficient tools for facility layout design problems.

^{*1} Graduate School of Engineering, Utsunomiya University

^{*2} Dept. of Industrial and Systems Engg., Ashikaga Institute of Technology

2. Fuzzy Facility Layout Problem

2.1 Facility layout problem

Facility layout design problem is formulated as follows[13]: there is a set of m facilities, denoted by M_i , $i=1, 2, \dots, m$. Each facility is restricted to be rectangular and characterized by a triple (A_i, l_i, u_i) , where A_i is the area of the facility, l_i and u_i are lower and upper bounds on aspect ratio, respectively. The aspect ratio is defined as the ratio of the height h_i to the width w_i of the facility. This specification leads to following relationship:

$$w_i h_i = A_i, \quad i=1, 2, \dots, m \quad (1)$$

$$l_i \leq \frac{h_i}{w_i} \leq u_i, \quad i=1, 2, \dots, m \quad (2)$$

The aspect ratio can also be defined as the inverse. Figure 1 shows the relationship of the height, width and aspect ratio for a facility.

A facility layout for given m facilities consists of a bounding rectangular R , partitioned by horizontal and vertical line segments into m nonoverlapping rectangle regions, denoted $\{r_i\}$, $i=1, 2, \dots, m$. Each region r_i , with width x_i and height y_i , must be large enough to accommodate its facility M_i .

2.2 Fuzzy interflow

Generally, the uncertainty of material flows among facilities can be represented as a convex fuzzy number. We call it *fuzzy interflow*. Here, we use a trapezoidal fuzzy number (TrFN) to represent the fuzzy interflow[14]. A TrFN can be defined by a quadruple (a, b, c, d) as shown in Figure 2.

Since a fuzzy number represents many possible real numbers that have different membership values, it is not easy to compare the final ratings to determine which alternatives are preferred. Many fuzzy ranking methods have been proposed to compare fuzzy numbers[15]. In this study, we adopt Lee and Li's approach, which suggests the use of generalized mean and standard deviation based on the probability measures of fuzzy events to rank fuzzy number[16]. When fuzzy number $\tilde{M}$ is a TrFN, the generalized mean value with a uniform density is calculated as follows:

$$m(\tilde{M}) = \frac{-a^2 - b^2 + c^2 + d^2 - ab + cd}{3(-a - b + c + d)} \quad (3)$$

The standard deviation is defined as follows:

$$\sigma(\tilde{M}) = \left[\frac{1}{b-a} \left(\frac{b^4}{4} - \frac{ab^3}{3} + \frac{a^4}{12} \right) + \frac{1}{3} (c^3 - b^3) + \frac{1}{d-c} \left(\frac{d^4}{12} - \frac{c^3 d}{3} + \frac{c^4}{4} \right) \right] / \left[\frac{1}{2} (-a - b + c + d) \right] - m(\tilde{M})^2 \quad (4)$$

Figure 1: The relationship of the height, width and aspect ratio for a facility

Figure 2: Trapezoidal fuzzy number

3. Genetic Algorithms

3.1 Representation

A facility layout can be represented as a slicing structure constructed by recursively partitioning a rectangular R . Let the operations of a horizontal cut and a vertical cut be denoted by the operators $+$ and $*$, respectively. Slicing structures comprising the m given facilities (called operands) can be represented by slicing trees or Polish expressions over the alphabet set $\Sigma = \{1, 2, \dots, m, *, +\}$. In a slicing tree, operators are internal nodes and operands are leaves. An example of slicing structure is shown in Figure 3(a) and its slicing tree and Polish expression are shown in 3(b).

Figure 3: A slicing structure and its representations

We use Polish expression as the coding scheme of chromosome in our genetic algorithms. A chromosome must have m different operands and $(m-1)$ operators where m is the number of facilities.

3.2 Initialization

We use a guided random approach to initialize genetic system. It constructs a chromosome from left to right by picking one element at a time from the set Σ until a complete chromosome is formed. Because the random permutation of operands and operators may yield an illegal chromosome, we must check its legality at each random picking.

Firstly we define the concept of *prefix* for a chromosome. For a given chromosome with a total size of $(2m-1)$, a prefix is a partial expression (may not be a legal Polish expression) containing the first i elements of the chromosome with the same order as they are in the chromosome, where $i \leq 2m-1$. An example is shown in Figure 4.

Let Ψ denote the set of all possible prefix for a given chromosome. For a prefix $\psi \in \Psi$, let $N(\psi)$ denote the total number of operands the prefix ψ contains and $M(\psi)$ the total number of operators the prefix ψ contains. The condition of legality for a chromosome (or a Polish expression) is given in following proposition and corollary.

Proposition 1 For a given chromosome containing m operands and $(m-1)$ operators, if the following equation holds,

$$N(\psi) \geq M(\psi) + 1, \psi \in \Psi \quad (5)$$

then the chromosome is a legal Polish expression.

It is easy to verify that the chromosome given in Figure 4 is a legal one.

Corollary 1 For a given prefix ψ , if it does not meet the condition of Eq.5, it is impossible to develop a legal Polish expression from the prefix ψ with a left to right generating procedure.

```

chromosome
1 2 * 3 4 5 * 6 + + *
possible prefixes
1 2 * 3 4 5 * 6 + +
1 2 * 3 4 5 * 6 +
1 2 * 3 4 5 * 6
1 2 * 3 4 5 *
1 2 * 3 4
1 2 * 3
1 2 *
1 2
1

```

Figure 4: Possible prefix

For example, prefix (1 2 * 3 4) meets Eq.5, so we can develop a legal Polish expression from it with the left-to-right generating procedure as (1 2 * 3 4 + +); prefix (1 2 * + +) does not meet Eq.5, so we can not develop a legal Polish expression from it with the left-to-right generating procedure.

Initial procedure constructs a chromosome from left to right by picking up a random element from the set $\Sigma = \{1, 2, \dots, m, *, +\}$ once a time until a complete chromosome is formed. The legality check is performed at each random picking. Suppose we have a prefix (or a partially-generated chromosome) ψ . According to Corollary 1, if $N(\psi) = M(\psi) + 1$, the next element to be picked up must be an operand; otherwise the next one can be an operand or an operator. The overall procedure is shown below:

procedure:guided random initialization

```

begin
   $\Sigma_1 \leftarrow \{*, +\};$ 
   $i \leftarrow 1;$ 
  while ( $i \leq pop\_size$ ) do
    begin
       $\Sigma_0 \leftarrow \{1, 2, \dots, m\};$ 
       $\psi_i \leftarrow \phi;$ 
      select a random element  $\sigma_1$  from  $\Sigma_0;$ 
       $\Sigma_0 \leftarrow \Sigma_0 / \{\sigma_1\};$ 
       $\psi_i \leftarrow \psi_i \cup \{\sigma_1\};$ 
       $j \leftarrow 1;$ 
      while ( $j \leq 2m-1$ ) do
        begin
          if  $N(\psi_i) = M(\psi_i) + 1$ 
            select  $\sigma_j$  from  $\Sigma_0;$ 
          else
            select  $\sigma_j$  from  $\Sigma_0 \cup \Sigma_1;$ 
          endif
           $\Sigma_0 \leftarrow \Sigma_0 / \{\sigma_j\};$ 
           $\psi_i \leftarrow \psi_i \cup \{\sigma_j\};$ 
           $j \leftarrow j + 1;$ 
        end
      end
       $i \leftarrow i + 1$ 
    end
  end
end

```

3.3 Crossover

We use the crossover proposed by Cohoon *et al.* [11]. It performs with two steps:

1. get operands from one parent and
2. get operators from the other parent.

Suppose we have two parents p_1 and p_2 . Crossover operator copies the operands from parent p_1 into the corresponding positions in an offspring

o; and then copies operators from p_2 , by making a left-to-right scan, to complete the offspring o . Crossover performed in this way can guarantee to generate legal offspring. Figure 5 gives an illustration of crossover operation and Figure 6 shows the layouts of parents and offspring.

Figure 5: Crossover operator

Figure 6: Layouts after crossover

We use random *swapping*, *inverting* and *altering* as our mutation, that is, swapping two adjacent operands or two adjacent operators as shown in Figure 7(a), inverting a sequence of adjacent operators or a sequence of adjacent operands as shown in Figure 7(b), and altering an operator to the opposite one as shown in Figure 7(c). Mutation performed in this way can also guarantee to generate legal offspring. Figure 8 shows the layouts of parents and offspring.

The overall procedure is shown below:

procedure: mutation

begin

$i \leftarrow 0$;

while ($i \leq pop_size \times p_m$) **do**

select a chromosome (not mutated) randomly;

pick up a gene randomly;

if (it is an operator) **do**

select *swapping*, or *inverting* or *altering*

mutation with equal chance;

else

select *swapping* or *inverting* mutation with equal chance;

endif

perform the selected mutation;

$i \leftarrow i + 1$;

end

end

3.5 Constructing a layout from a chromosome

Before evaluating each chromosome, we must convert it to a slicing structure. This is done in a top-down fashion starting from the root of the slicing tree and passing down to the children nodes recursively. For a given chromosome (or a slicing tree), first we separate it into two subtrees: left

Figure 7: Mutation operator

Figure 8: Layouts after mutation

subtree and right subtree. The subtrees are also legal slicing tree. For each subtree we further separate it into two subtrees. The process is repeated until each subtree just contains one node.

For a chromosome, there is a position at which the chromosome will be separated into left subtree and right subtree. We call such position as a *cut point*. For a slicing tree ψ with total size $2m-1$, let $i(1 \leq i \leq 2m-2)$ be an arbitrary cut point (or position) in the tree and let $n_i(\psi)$ denote the total number of operands contained in the right part from the cut point to the right most of the tree and $m_i(\psi)$ the total number of operators contained in the same right part of the tree. The condition for searching such a cut point in a chromosome (or a slicing tree) is given by the following proposition.

proposition 2 *If a cut point i is the first position, scanning from right to left ranging from $2m-1$ to 1, which let the following equation holds,*

$$n_i(\psi) = m_i(\psi) \quad (6)$$

then at cut point i the slicing tree can be separated into left subtree containing the elements from 1 to $i-1$ of the given slicing tree and right subtree containing elements from i to $2m-2$. Each subtree is a legal slicing tree.

The set of possible dimensions (x_i, y_i) of region r_i accommodating M_i can be determined along with the recursively separating process using area requirement information. For a region r_i , we use x_i^r and x_i^l to denote the left and right boundary of the region respectively, and y_i^u and y_i^b to denote the upper and bottom boundary of the region respectively. The relation between dimensions and boundary lines for a region is calculated as follows:

$$x_i = x_i^r - x_i^l, \quad i=1, 2, \dots, m \quad (7)$$

$$y_i = y_i^u - y_i^b, \quad i=1, 2, \dots, m \quad (8)$$

Note that a simple tree is a special slicing tree with a length of 3. The overall procedure of constructing a layout from a chromosome is shown below:

```

procedure:construct(tree, length)
begin
   $i \leftarrow \text{length};$ 
  while ( $i > 0$ ) do
    begin
      separate tree to two subtrees;
      calculate  $x_i^r, x_i^l, y_i^u, y_i^b$ ;
      if the right subtree is a simple tree
        calculate  $x_i^r, x_i^l, y_i^u, y_i^b$ ;
         $i \leftarrow i-2$ ;

```

```

      if neighbor left tree is a simple tree
        calculate  $x_i^r, x_i^l, y_i^u, y_i^b$ ;
         $i \leftarrow i-2$ ;
      else
        construct(lefttree, length);
         $i \leftarrow i - \text{length};$ 
      endif
    end
    construct(righttree, length);
     $i \leftarrow i - \text{length};$ 
    if neighbor left is a simple tree
      calculate  $x_i^r, x_i^l, y_i^u, y_i^b$ ;
       $i \leftarrow i-2$ ;
    else
      construct(lefttree, length);
       $i \leftarrow i - \text{length};$ 
    endif
  endif
  calculate  $x_i$  and  $y_i$  using  $x_i^r, x_i^l, y_i^u, y_i^b$ ;
end

```

Figure 9 shows an example of converting chromosome (1 2 * 3 4 5 * 6 +++) into a layout.

3.6 Evaluation and selection

Each layout in the enlarged population (parents and offspring) is then evaluated by an objective function of total traffic cost and assigned a fitness value. Let v_k denote the k th chromosome in the enlarged population, $\tilde{C}_k(v_k)$ be the total cost, $d_{ij}(v_k)$ be a real number to denote the Manhattan distance between the centers of each pair of facilities i and j , and $\tilde{c}_{ij}(v_k)$ be a trapezoidal fuzzy number to denote the fuzzy interflow between the facilities i and j . The overall evaluation of a particular layout is given as follows:

$$\tilde{C}_k(v_k) = \sum_{i=1}^m \sum_{j=1}^m \tilde{c}_{ij}(v_k) d_{ij}(v_k) \quad (9)$$

Because the total cost is a trapezoidal fuzzy number, it is impossible to directly use it as fitness. We use Lee and Li's ranking approach to calculate the generalized mean values of the total costs for each layout. Because we treat with a minimization problem, we must convert this mean value into fitness value, such that a fitter chromosome has a larger fitness value. The conversion can be simply done by the inverse of the mean value.

Because there are aspect ratio constraints on facilities, a layout generated in this way may be illegal in the sense what there may exist one or

Figure 9: Constructing a layout

more than one facilities which violate the aspect ratio constraints. A penalty approach is adopted to handle such illegitimation[17]. Let $m(\tilde{C}_k(v_k))$ denote the generalized mean value, λ_k denote the total number of facilities which violate the aspect ratio constraints within the k th chromosome, and P be a big penalty value. We add the penalty term into the fitness function and then we have:

$$\text{eval}(v_k) = \frac{1}{m(\tilde{C}_k(v_k)) + \lambda_k P} \quad (10)$$

where $\text{eval}(v_k)$ is the fitness function for the k th chromosome.

We then make a roulette wheel with the fitness values to reproduce next generation. In our experiment, the *elitist way* was embedded within roulette wheel selection in order to enforce preserving the

best chromosome in the next generation and overcome the stochastic errors of sampling.

4. Numerical Example

The proposed algorithm has been coded in C language and run on NEC EWS 4800/260 workstation. Because there is no standard test problem available for fuzzy facility layout problem, we make a test problem with 15 facilities [18]. The area and aspect ratio of each facility is given in Table 1 and the fuzzy interflow represented as TrFNs is given in Table 2.

The solution of fuzzy layout problem is further compared to the solutions of equivalent non-fuzzy problems by choosing all the best-case and worst-case values for the interflows.

4.1 A typical solution to fuzzy case

The evolutionary environment of our experiment is given as follows: population size is 40, the ratio of crossover is 0.4, the ratio of mutation is 0.4, penalty value is 5000 and maximum generation is 200. A typical solution (the best chromosome obtained from one run of the genetic system) is shown as follows:

(13 1 + 2 6 8 3 + + 15 11 14 9 * + + *
* 4 7 * 5 10 + 12 * * + *)

The layout is depicted in Figure 10(a) and its tree representation is depicted in Figure 10(b). The evolutionary process is shown in Figure 11.

From the evolutionary process we can know that before the 10th generation, all chromosomes are illegal in the sense of violating the aspect ratios.

Table 1: Geometric constraints of facilities

Facility id	Area	Aspect ratio	
		lower bound	upper bound
1	100	0.7	1
2	80	1	1
3	50	0.7	1.3
4	60	0.5	0.8
5	120	0.9	1
6	40	0.6	1
7	20	0.7	1.4
8	40	1	1
9	150	0.8	1.1
10	120	0.5	1.5
11	50	0.7	1.1
12	10	0.8	1.2
13	20	0.95	1.5
14	30	0.75	1.25
15	50	0.9	1.1

Table 2: Fuzzy interflow among facilities

	1	2	3	4	5
1	(0,0,0,0)	(8,10,15,18)	(0,0,0,0)	(2,5,8,10)	(0,1,4,12)
2	(8,10,15,18)	(0,0,0,0)	(0,1,2,3)	(2,3,6,8)	(1,2,5,6)
3	(0,0,0,0)	(0,1,2,3)	(0,0,0,0)	(8,10,14,16)	(0,2,5,6)
4	(2,5,8,10)	(2,3,6,8)	(8,10,14,16)	(0,0,0,0)	(0,1,4,6)
5	(0,1,4,12)	(1,2,5,6)	(0,2,5,6)	(0,1,4,6)	(0,0,0,0)
6	(0,0,0,0)	(0,2,4,5)	(0,0,0,0)	(0,1,3,5)	(2,3,5,8)
7	(0,1,2,3)	(0,2,3,6)	(1,2,3,5)	(2,5,6,9)	(4,5,7,9)
8	(1,2,5,10)	(2,3,5,7)	(3,5,6,7)	(0,0,0,0)	(3,5,7,9)
9	(1,2,3,4)	(1,2,3,5)	(2,4,6,8)	(0,0,0,0)	(0,5,8,10)
10	(1,2,4,5)	(0,0,0,0)	(1,5,8,10)	(0,2,4,6)	(0,1,4,6)
11	(0,2,3,6)	(0,2,7,8)	(0,2,4,5)	(0,1,2,3)	(0,0,0,0)
12	(0,0,0,0)	(0,0,0,0)	(1,2,3,6)	(0,0,0,0)	(2,3,6,7)
13	(2,4,5,9)	(6,10,12,15)	(3,5,7,9)	(1,2,5,8)	(0,0,0,0)
14	(0,0,0,0)	(4,5,8,9)	(0,5,8,9)	(2,5,8,9)	(3,5,8,10)
15	(0,0,0,0)	(0,0,0,0)	(3,5,10,12)	(0,0,0,0)	(2,5,9,10)
	6	7	8	9	10
1	(0,0,0,0)	(0,1,2,3)	(1,2,5,10)	(1,2,3,4)	(1,2,4,5)
2	(0,2,4,5)	(0,2,3,6)	(2,3,5,7)	(1,2,3,5)	(0,0,0,0)
3	(0,0,0,0)	(1,2,3,5)	(3,5,6,7)	(2,4,6,8)	(1,5,8,10)
4	(0,1,3,5)	(2,5,6,9)	(0,0,0,0)	(0,0,0,0)	(0,2,4,6)
5	(2,3,5,8)	(4,5,7,9)	(3,5,7,9)	(0,5,8,10)	(0,1,4,6)
6	(0,0,0,0)	(1,2,3,4)	(0,2,3,4)	(0,1,2,4)	(4,5,6,8)
7	(1,2,3,4)	(0,0,0,0)	(4,6,8,10)	(0,0,0,0)	(0,1,2,3)
8	(0,2,3,4)	(4,6,8,10)	(0,0,0,0)	(3,5,7,8)	(0,2,4,6)
9	(0,1,2,4)	(0,0,0,0)	(3,5,7,8)	(0,0,0,0)	(0,0,0,0)
10	(4,5,6,8)	(0,1,2,3)	(0,2,4,6)	(0,0,0,0)	(0,0,0,0)
11	(0,0,0,0)	(4,5,8,10)	(7,10,15,18)	(2,10,12,15)	(0,0,0,0)
12	(0,0,0,0)	(3,5,7,9)	(0,0,0,0)	(1,5,8,9)	(2,4,6,8)
13	(1,2,4,5)	(2,5,6,7)	(2,5,9,10)	(5,10,15,18)	(0,0,0,0)
14	(3,5,8,9)	(0,1,3,4)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
15	(8,10,12,15)	(0,0,0,0)	(0,0,0,0)	(0,2,7,8)	(4,5,8,10)
	11	12	13	14	15
1	(0,2,3,6)	(0,0,0,0)	(2,4,5,9)	(0,0,0,0)	(0,0,0,0)
2	(0,2,7,8)	(0,0,0,0)	(6,10,12,15)	(4,5,8,9)	(0,0,0,0)
3	(0,2,4,5)	(1,2,3,6)	(3,5,7,9)	(0,5,8,9)	(3,5,10,12)
4	(0,1,2,3)	(0,0,0,0)	(1,2,5,8)	(2,5,8,9)	(0,0,0,0)
5	(0,0,0,0)	(2,3,6,7)	(0,0,0,0)	(3,5,8,10)	(2,5,9,10)
6	(0,0,0,0)	(0,0,0,0)	(1,2,4,5)	(3,2,8,9)	(8,10,12,15)
7	(4,5,8,9)	(3,5,7,9)	(2,5,6,7)	(0,1,3,4)	(0,0,0,0)
8	(7,10,15,18)	(0,0,0,0)	(2,5,9,10)	(0,0,0,0)	(0,0,0,0)
9	(2,10,12,15)	(1,5,8,9)	(5,10,15,18)	(0,0,0,0)	(0,2,7,8)
10	(0,0,0,0)	(2,4,6,9)	(0,0,0,0)	(0,0,0,0)	(4,5,8,10)
11	(0,0,0,0)	(2,5,8,10)	(0,0,0,0)	(3,5,8,12)	(0,0,0,0)
12	(2,5,8,10)	(0,0,0,0)	(2,3,5,6)	(1,3,5,7)	(0,0,0,0)
13	(0,0,0,0)	(2,3,5,6)	(0,0,0,0)	(6,10,14,17)	(0,2,5,8)
14	(3,5,8,12)	(1,3,5,7)	(6,10,14,17)	(0,0,0,0)	(3,4,6,7)
15	(0,0,0,0)	(0,0,0,0)	(0,2,5,8)	(3,4,6,7)	(0,0,0,0)

(a)

(b)

Figure 11: Evolutionary process

Penalty function guides the genetic search to feasible region. After entering the feasible region, penalty loses its function and natural selection forces chromosomes to evolve gradually. In the 63

th generation the best solution is found.

The fuzzy cost measure of the best layout is $\tilde{C} = (2946.91, 5841.40, 9561.81, 12613.53)$. We can use possibility theory and fuzzy integrals to inter-

pret the fuzzy performance. The possibility of a member of the support set of a fuzzy number taking on a certain exact value is defined to be the degree of ease[19]. In most cases, the possibility is exactly the membership function value of that member. For example, the possibility that the layout will have total cost of 4394 units is 0.5, as illustrated in Figure 12. The concept of possibility may be difficult to understand in an absolute sense, but it can be very useful in a relative context. For example, when comparing one support set value to another, the manager can identify which value is more possible based on their possibilities. Genetic algorithms can provide a set of superior solutions to managers for their judgment.

Perhaps of more use to the managers is a proportional measure of the distribution to the density of the membership function. It is calculated by taking fuzzy integral over interest area and dividing it by the integral of fuzzy number over support set shown as follows:

$$\tilde{I}(x \leq \beta) = \frac{\int_a^\beta \mu(x) dx}{\int_a^\beta \mu(x) dx} \quad (11)$$

For example, the fuzzy integral $\tilde{I}$ of 4394 is 0.13 using Eq.(11). The manager can interpret this result as the fuzzy expectation of the layout yielding total cost less than or equal to 4394 is 0.13. The fuzzy integral $\tilde{I}$ of 9562 is 0.91, so the fuzzy expectation of the layout yielding total cost less than or equal to 9562 is 0.91.

Figure 12: Possibility of $\tilde{C}=4394$

4.2 Comparison with non-fuzzy cases

Fuzzy interflow is characterized by a quadruple (a, b, c, d) where a represents the optimistic estimation on material flow among facilities (the best-case), d the pessimistic estimation (the worst-case), b one average estimation (near best-case), and c another average estimation (near worst-case). According to the four cases, we have made four equivalent non-fuzzy problems using the same fuzzy data used above and solved by the proposed algorithms (with slight modification to make it to be suitable for non-fuzzy case). We use the same parameter setting for each run of the four problems and the best solutions for each case are given below:

best-case: (5 14 3 * 9 * * 15 10 + 12 13 * 1 *
* 6 7 8 11 + * * 4 * 2 * + +)
near best: (6 7 + 8 * 9 1 2 + * * 5 15 11 13 *
10 3 * 14 * 4 12 * * + * * +)
near worst: (9 15 * 14 * 4 13 1 * * * 10 12 6
* + 11 3 8 7 + + 5 * 2 * * * +)
worst-case: (7 3 * 14 * 4 * 11 6 * + 2 * 12 * 9
13 8 * * + 5 * 10 15 1 + + *)

Figure 13 depicts the layout and its tree representation for the best-case, Figure 14 for the near best-case, Figure 15 for the near worst-case, and Figure 16 for the worst-case.

Figure 13: layout and tree representation for the best-case

Figure 14: layout and tree representation for the near best-case

Figure 15: layout and tree representation for the near worst-case

Figure 16: layout and tree representation for the worst-case

For each layout, we calculate the total costs under four kinds of material flows (best, near best, near worst and worst) and the comparison with fuzzy solution is summarized in Table 3.

From the comparison results we know that for an equivalent non-fuzzy case, we can get a layout with a minimum cost under the considered material flows, but for the other cases, this layout will yield large costs than the solution obtained by the fuzzy approach. That is, non-fuzzy approach just can get a layout suitable for its considered case while fuzzy approach can get a reasonable solution suitable for all cases ranging from the best-case to the worst-case. For example, the total cost of an equivalent non-fuzzy problem for the best-case of material flows is 2794 which is smaller than the one in fuzzy case (2947). Using the same layout, we calculate the costs for other three cases: the near best-case, the near worst-case and the worst-case. The costs (6068, 9953 and 13215) are larger than those in fuzzy case. The relative error with respect

to the solution obtained by fuzzy approach is given in Figure 17.

5. Conclusion

Facility layout design problems are strategically critical in modern manufacturing system. Because the optimal design of the physical layout must be resolved in the early stage of the manufacturing system design and product mix is highly uncertainty during design phase, layout designing is greatly depended on the subjective judgments and/or preference of manager or designer, which are of imprecision in nature.

In this paper, we introduce the concept of fuzzy interflow into facility layout design problem and address fuzzy facility layout problem, where the uncertainty of material flows among facilities is represented as trapezoidal fuzzy numbers. The practical merit of the proposed fuzziness approach, comparing with conventional flexible approach and robustness approach, is that there is no need to

Table 3: Comparison Results

solutions	a	b	c	d
fuzzy	2947	5841	9562	12614
best-case	2795	6068	9953	13215
near best-case	2895	5083	9625	12922
near worst-case	2971	5869	9581	12685
worst-case	3012	6257	10024	13134

Figure 17: Relative error with respect to fuzzy solution

force the designers to give a precise interflow or a probability distribution. Genetic algorithms are applied to such hard fuzzy combinatorial problem. Polish expression is adopted as the coding scheme of chromosome. The condition of legality of Polish expression coding for chromosome and the condition for searching cut point in a chromosome are given. Based on these conditions, effective initialization procedure and construction procedure from a chromosome to a layout are built. A penalty to the violation of aspect ratio for each facilities is used to guide genetic search effectively towards to promising part of solution space. Lee and Li's ranking approach is used to select the best layout in the fuzzy context. The possibility theory and fuzzy integral are used to meaningfully interpret the fuzzy results.

The solution of fuzzy layout problem is further compared to the solutions of equivalent non-fuzzy problems by choosing all the best-case and worst-case values for the interflows. From the comparison results we know that non-fuzzy approach just can get a layout suitable for its considered case while fuzzy approach can get a reasonable solution suitable for all cases ranging from the best-case to the worst-case. Simulation results demonstrate that genetic algorithm and fuzziness approach can be the efficient tools to solve large-scale layout problem.

Acknowledgment

This work was supported in part by a research grant from the Ministry of Education, Science and Culture, Japanese Government: Grant-in-Aid for Scientific Research, the International Scientific Research Program (No. 07045032).

References

- [1] Francis, R. L., L. F. McGinnis, and J. A. White, *Facility Layout and Location: An Analytical Approach*, second edition, Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [2] Kusiak, A. and S. Heragu, The facility layout problem, *European Journal of Operational Research*, vol.29, pp.229--251, 1987.
- [3] Rosenblatt, M. J., The dynamics of plant layout, *Management Sciences*, vol.32, pp.76--86, 1986.
- [4] Shore, R. H. and J. A. Tompkins, Flexible facilities design, *IIE Transactions*, vol.12, pp.200--205, 1980.
- [5] Rosenblatt, M. J. and H. L. Lee, A robustness approach to facilities design, *International Journal of Production Researches*, vol.25, pp.479--486, 1987.
- [6] Kouvelis, P., A. A. Kurawarwala, and G. J. Gutierrez, Algorithms for robust single and multiple period layout models for manufacturing systems, *European Journal of Operational Research*, vol.63, pp.287--303, 1992.
- [7] Cheng, R., M. Gen, and T. Tozawa, Fuzzy facility layout problem using genetic algorithm, In *The 11th Fuzzy System Symposium*, Okigawa, Japan, 1995.
- [8] Pedrycz, W., Fuzzy modeling: methodology, algorithms and practice, In Zurada J.M., P. M. I. and C. Robinson, editors, *Computational Intelligence: Imitating Life*, section 2.3, IEEE Press, New York, 1994.
- [9] Otto, K. N. and E. K. Antonsson, Modeling imprecision in product design, In *FUZZ-IEEE'94*, pp. 346--356, 1994.
- [10] Tate, D. M. and A. E. Smith, Unequal area facility layout using genetic search, to appear in *IIE Transactions*.
- [11] Cohoon, J. P., S. U. Hegde, and N. Martin, Distributed genetic algorithms for the floorplan design problem, *IEEE Tran. on Computer-aided Design*, vol.10, pp.483--491, 1991.
- [12] Tam, K. Y., Genetic algorithms, function optimization, and facility layout design, *European Journal of Operational Research*, vol.63, pp.322--346, 1992.
- [13] Wong, D. F. and C. L. Liu, A new algorithm for floorplan design, In Wizard, V. and M. Yannakakis editors, *Proc. 23rd ACM-IEEE Design Automation Conf*, pp. 101--107, Boston, 1986.
- [14] Kaufmann, A. and M. M. Gupta, *Fuzzy Mathematical Models in Engineering and Management Science*, North-Holland, Amsterdam, 1988.
- [15] Chen, S. J. and C. L. Hwang, *Fuzzy Multiple Attribute Decision Making*, Springer-Verlag, Berlin, 1992.
- [16] Lee, E. S. and R. L. Li, Comparison of fuzzy numbers based on the probability measure of fuzzy events, *Operations Research*, vol.15, pp.887--896, 1988.
- [17] Smith, A. E. and D. Tate, Genetic optimization using a penalty function, In Forrest, S. editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 499--505, Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [18] Cheng, C., Study on optimal design of fuzzy facility layout, Master's thesis, Ashikaga Institute of Technology, Ashikaga, Japan, March 1995.

- [19]Zadeh, L. A., Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, vol.1, pp.3--28, 1978.
- [20]Gen, M. and R. Cheng: "Genetic Algorithms and Engineering Design", John Wiley & Sons, New York, 1996.

(1995年2月9日 受 付)

(1995年7月3日 再受付)

[問い合わせ先]

〒326 足利市大前町268

足利工業大学

玄 光男

TEL :+81(284)62-0605

FAX :+81(284)64-1071

E-mail : gen@genlab.ashitech.ac.jp

著 者 紹 介

程 潤偉 (てい じゅんい)

宇都宮大学大学院工学研究科博士後期課程生産・情報工学専攻

昭和57年 中国・東北大学工業自動化学工学科卒業。昭和59年同大学院工学研究科修士課程システム工学専攻修了。同年同大システム工学研究所・助手。昭和62年講師。平成5年助教授。平成4年10月～平成5年3月足利工業大学訪問研究員。平成5年4月～現在 足利工業大学非常勤講師。平成6年4月宇都宮大学大学院工学研究科博士後期課程生産・情報工学専攻入学。現在に至る。遺伝的アルゴリズムの応用の研究開発に従事。著書 "Genetic Algorithms and Engineering Design", John Wiley & Sons, 1996。日本経営工学会、日本ファジイ学会の会員。

杜澤 達美 (とざわ たつみ)

宇都宮大学大学院工学機械システム工学科

昭和32年 工学院大・工・工業化学化卒。昭和33年 東京大学・工・応用物理学科助手。昭和36年 宇都宮工業短期大学・機械科講師。昭和40年 宇都宮大学・工・機械工学科助教授。昭和56年 教授。工博。金属材料の変形及び破壊挙動への非破壊評価法の適用。材料組織の三次元的結晶粒度分布の評価などに関する研究に従事。応用物理学会、計測自動制御学会、軽金属学会、日本金属学会、日本鉄鋼協会等各会員。

玄 光男 (げん みつお)

足利工業大学工学部経営工学科

昭和44年 工学院大学・工・電子卒業。昭和49年 同大学院博士課程修了。同年 足利工業大学・工・経営工学講師。昭和55年 同助教授。昭和62年 同教授。現在に至る。平成元年～7年 足利工大・情報科学(前 AITnet)センター長。昭和56年～57年 米国 Univ. of Nebraska-Lincoln 客員助教授。工学博士。ファジィシステム工学、ファジィ多目的計画法、遺伝的アルゴリズムの応用等の研究開発に従事。著書「線形計画・目標計画プログラム」(昭和59年 電気書院)、「Genetic Algorithms and Engineering Design', John Wiley & Sons, 1996 など。日本ファジイ学会、日本オペレーションズ・リサーチ学会、電子情報通信学会、日本経営工学会、情報処理学会、IEEE、IIE 等各会員。

Genetic Search for Facility Layout Design under Interflows Uncertainty

by

Runwei CHENG, Mitsuo GEN and Tatsumi TOZAWA

Abstract:

In this paper, we introduce the concept of fuzzy interflow into facility layout design problem and address fuzzy facility layout problem, where the uncertainty of material flows among facilities is represented as trapezoidal fuzzy numbers. Genetic algorithms are applied to such hard fuzzy combinatorial problem. Polish expression is adopted as the coding scheme of chromosome. The condition of legality for Polish expression coding and the condition for searching cut point in a chromosome are given. Based on these conditions, effective initialization procedure and layout construction procedure are built. Fuzzy ranking method is used to select the best layout in fuzzy context. A penalty to the violation of aspect ratio for each facilities is used to guide genetic search effectively towards to the promising part of solution space. The possibility theory and fuzzy integral are used to meaningfully interpret the fuzzy results. The simulation results demonstrate that genetic algorithm and fuzziness approach can be the efficient tools to solve large-scale layout problem.

Keywords :Genetic Algorithms, Facility Layout Problem, Interflows Uncertainty, Fuzzy Number Theory, Possibility Theory

Contact Address: **Prof. Mitsuo Gen**

Dept. of Industrial and Systems Engg., Ashikaga Institute of Technology
Ashikaga, 326 Japan
TEL :+81(284)62-2985
FAX :+81(284)64-1071
E-mail : gen@genlab.ashitech.ac.jp