



**Lancaster University**  
MANAGEMENT SCHOOL

# Facility layout design with random demand and capacitated machines

**Yifei Zhao**

Department of Management Science  
Lancaster University Management School

**Stein W. Wallace**

Department of Business and Management Science  
Norwegian School of Economics

Working Paper 2012:9

# Facility layout design with random demand and capacitated machines

Yifei Zhao

Department of Management Science  
Lancaster University Management School, Lancaster LA1 4YX, UK

Stein W. Wallace

Department of Business and Management Science  
Norwegian School of Economics, NO-5045 Bergen, Norway

## **Abstract**

We consider the facility layout problem with machine capacities (and hence multiple machines of each type) and random demand. We develop a very efficient heuristic allowing us to find good solutions to the stochastic case whenever good solutions can be found for the corresponding deterministic problem with uncapacitated machines of the same dimension. Hence, this problem represents a case where a specific deterministic model, even when solved heuristically, produces a very good solution to a stochastic model.

**Keywords:** Facility layout, Random demand, Machine capacities

# 1 Introduction

The facility layout problem (FLP) is to arrange physical locations for facilities (such as machine tools, work centres, manufacturing cells, departments, and warehouses) for a production or delivery system. The layout of facilities is one of the most fundamental and strategic issues in many manufacturing industries. Most modifications or re-arrangements of existing layouts involve substantial financial investments and planning efforts (Singh and Singh 2010). An efficient layout of facilities can reduce operational cost and contribute to the overall production efficiency (Tomkins et al 2003).

Given information such as capacities of machines, processing routes and demand, the facility layout planner's objective is to determine an efficient layout according to some design criterion. One of the most frequently considered criteria is the minimization of material handling distances/costs. It is claimed that material handling costs contribute from 20 to 50 percent of total operating expenses in manufacturing (Francis and White 1974). Material handling costs are incurred when moving finished parts, work-in-process parts, spares, tools and other equipments or supplies between machines or machines and storage locations to meet different demands during production (Heragu and Kusiak 1988). Manned and unmanned material handling equipment, such as conveyors, automatic guided vehicles (AGVs), cranes and robots are deployed for delivery activities. To minimize material handling costs in layout design, distances travelled for both material, equipment, and personnel will need to be taken into consideration (Taghavi and Murat 2011).

The FLP was initially formulated as a quadratic assignment problem (QAP) which is NP-hard and has been widely used to solve FLPs (Koopmans and Beckman 1957). Exact methods (e.g. branch-and-bound (Salimanpur and Jafari 2008) and cutting planes (Bukard and Bonniger 1983)) are only efficient for layout problems of sizes up to about 16 (Singh and Sharma 2006). To achieve solutions for larger cases, heuristic methods such as simulated annealing (Baykasoglu and Cindy 2001, Misevicius 2003), genetic algorithm (Balakrishnan et al 2003, Lee et al 2003), tabu search (Chiang and Chiang 1998, Helm and Hadley 2000) and other methods based on the QAP have been proposed. Performances of SA and GA for solving layout problems have been compared by Heragu and Alfa (2003) and Tavakkoli-Moghaddain and Shanyan (1998).

Urban et al (2000) studied an integrated facility layout and flow assignment problem (IFLFAP) which is closer to reality by considering capacitated machines. Being different from QAPs, flows between any pair of machines in IFLFAPs are decision variables determined by solving a flow assignment problem to satisfy product demand. Hence the facility design from IFLFAP is obtained by joint decisions on machine locations and flow assignments. The resulting formulation of IFLFAP was then solved by using two metaheuristics, GRASP and Tabu search (Taghavi and Murat 2011). A layout obtained from the IFLFAP is called a distributed layout, and is commonly used in Japan (Suzaki 1985).

A stochastic version of IFLFAP was considered by Benjaafar and Sheikhzadeh (2000) who took into account uncertain factors such as changes in technology and market requirements in the production environment. The objective of a stochastic IFLFAP is to design a robust layout which adapts to changes of product mix and demand. An extended QAP was applied to formulate the stochastic IFLFAP, which is also NP-hard. Optimal solutions could only be found for small-scale problems. Large cases (up to 40 considered by Benjaafar and Sheikhzadeh) were solved by an iterative heuristic method after decomposing the problem into facility layout and flow assignment sub-problems. Distributed layouts obtained from the stochastic IFLFAP were then compared with three other a priori layout strategies including

functional layouts, maximally distributed layouts and random layouts. The criterion of minimizing expected material handling costs over different production scenarios was used for the comparison of the four layout strategies. Computational results indicated that distributed layouts outperformed the other three layout strategies by 10% - 40%.

This paper studies a specific case of the stochastic IFLFAP in which only demand varies. The formulation of this specific stochastic IFLFAP is similar to the one proposed by Benjaafar and Sheikhzadeh (2000). A new heuristic approach is presented to solve this problem in a much more efficient way than before.

Most decisions are made under relevant uncertainty about the future. If the starting model, before uncertainty is added, is very hard to solve – such as the extremely difficult QAP we are facing here – adding uncertainty (and thereby normally some dynamics) might seem numerically hopeless. On the other hand, not doing so, may lead to rather useless results. This is a classical dilemma of many planning problems, not the least in logistics. So, rather than adding uncertainty to an already "hopeless" model, we may look for other approaches. Sometimes deterministic solutions, despite being very bad in their own rights, see for example Thapalia et al (2012), can turn out to contain very useful information, while in other similar situations, this is not at all the case, see for example Lium et al (2009). In this paper we demonstrate that a heuristic solution to a deterministic problem can lead to extremely good solutions to a numerically hopeless stochastic QAP. This way, whenever the deterministic problem can be solved, randomness (and in this case also capacitated machines) can be added to the model without any serious numerical effects, making the models much more relevant and useful.

## 2 Model development

### 2.1 Problem statement

We study the problem of designing plant layouts with capacitated machine duplicates in a stochastic environment. It is assumed that plants can only produce one product or one fixed product mix, but the actual demand is random. From period to period, demand fluctuates according to a known continuous distribution. Reconfigurations of plant layouts are considered infeasible since both the frequency of demand changes and costs of re-layouts are too high for any redesign to be economic and efficient. Hence the objective of this problem is to design a fixed plant layout for all production periods which minimizes the total expected (or average) material handling costs. Copies of one machine type have the same capacities while for those of different types, capacities can be different. The total production capacity of a plant can easily be found from the capacities of the machine copies. It is also assumed that maximal demand equals total production capacity. The objective is to obtain a layout which minimizes the expected/average material handling costs over all possible demands.

Compared with the classical FLP, the unique aspect of the IFLFAP is that under each production scenario we are only given flow volumes (equal to demand) between machine *types*. Flows between pairs of individual machine copies are not known. These are decision to be determined together with machines locations. This is the reason why this problem is called the integrated facility layout and flow assignment problem. However flow assignments are not really in focus. The final result is the layout of the facility.

The following are the IFLFAP's assumptions. The first three are inherited from the classical FLP.

- All machines of the same type have the same capacity
- The number of machines equals the number of locations
- The unit flow cost between machines only depends on their locations
- The (continuous) distribution function for demand is known and given
- No redundant machines are in the production system, implying that all copies are utilized when maximal demand occurs

## 2.2 Problem formulation

### 2.2.1 Notations

parameters:

- $N$  = total number of machine types
- $N_i$  = total number of machine copies of type  $i$
- $n_i$  =  $n$ th machine of type  $i$
- $K$  = total number of locations
- $C_j$  = capacity of machine copies of type  $j$
- $d_{kl}$  = distance between location  $k$  and location  $l$
- $h$  = uncertain demand which follows a continuous distribution with probability density function  $g(h)$
- $h_{max}$  = the maximal demand (equal to the production capacity)
- $V_{ij}^h$  = required flow amount from machine type  $i$  to type  $j$  given demand is  $h$
- $$V_{ij}^h = \begin{cases} h, & \text{if production requires sequence from machine type } i \text{ to type } j \\ 0, & \text{otherwise} \end{cases}$$

decision variables:

- $x_{n_i k} = \begin{cases} 1, & \text{if } n\text{th copy of type } i \text{ is assigned to location } k \\ 0, & \text{otherwise} \end{cases}$
- $v_{n_i m_j}$  = flow volume from  $n$ th copy of type  $i$  to  $m$ th copy of type  $j$ .

### 2.2.2 Formulation

$$\min z = \int_0^{h_{max}} \pi(X, h)g(h) dh \quad (1)$$

**subject to:**

$$\sum_{k=1}^K x_{n_i k} = 1 \quad n_i = 1, \dots, N_i \quad i = 1, \dots, N \quad (2)$$

$$\sum_{i=1}^N \sum_{n_i=1}^{N_i} x_{n_i k} = 1 \quad k = 1, \dots, K \quad (3)$$

$$x_{n_i k} \in \{0, 1\} \quad n_i = 1, \dots, N_i \quad i = 1, \dots, N \quad k = 1, \dots, K \quad (4)$$

**where:**

$$\pi(X, h) = \min\{v_{n_i m_j} d_{kl} x_{n_i k} x_{m_j l}\} \quad (5)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^N \sum_{n_i=1}^{N_i} v_{n_i m_j} \leq C_j \quad m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (6)$$

$$\sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} v_{n_i m_j} = V_{ij}^h \quad i, j = 1, \dots, N \quad (7)$$

$$\sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{n_i m_j} = \sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{m_j n_i} \quad m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (8)$$

$$v_{n_i m_j} \geq 0 \quad i, j = 0, 1, \dots, N \quad n_i = 1, \dots, N_i \quad m_j = 1, \dots, N_j \quad (9)$$

The IFLFAP is formulated as a two-stage stochastic program where the first stage problem is to decide the allocation of machines (Constraint set (2) to Constraint set (4)) and the second stage problem is to assign flows between machines (Constraint set (6) to Constraint set (8)). The objective (1) is a cubic function to minimize the expected material handling costs over all possible demands from 0 to  $h_{max}$ . Constraint sets (2) and (3) are the assignment constraints. Constraint set (6) guarantees that the capacity of each machine copy is not exceeded. Constraint set (7) ensures that the total flow from copies of type  $i$  to copies of type  $j$  equals the required amount  $V_{ij}^h$  of the two types, where  $V_{ij}^h$  is decided by demand  $h$  under each scenario. Conservation of material flow among machines is guaranteed in (8).

Two decision variables are shown from the above formula, the layout variables ( $x_{n_i k}, x_{m_j l}$ ), and the flow variables ( $v_{n_i m_j}$ ) for each possible demand. But the ultimate goal is to determine the machine layout, not the flows.

### 2.3 MILP formulation

The objective function of the formulation in Section 2.2.2 contains a cubic function  $\pi(X, h)$  which is the product of three decision variables, one flow assignment decision ( $v_{n_i m_j}$ ) and two machine allocation decisions ( $x_{n_i k}, x_{m_j l}$ ). In this section, we generate a new formulation which linearises the cubic objective function by introducing a new decision variable:

$$w_{n_i k m_j l} = v_{n_i m_j} * x_{n_i k} * x_{m_j l} \quad (10)$$

This decision variable  $w_{n_i k m_j l}$  denotes the amount of flow between machine  $n_i$  and machine  $m_j$  if these two machines have been allocated to locations  $k$  and  $l$ , respectively. By using this decision variable, the IFLFAP is formulated as a mix integer linear problem (MILP):

$$\min z = \int_0^{h_{max}} \pi(W, h)g(h) dh \quad (11)$$

**subject to:**

constraint sets (2) to (4)

**where:**

$$\pi(W, h) = \min\{d_{kl}w_{n_i k m_j l}\} \quad (12)$$

$$\sum_{i=0}^N \sum_{n_i=1}^{N_i} \sum_{k=1}^K \sum_{l=1}^K w_{n_i k m_j l} \leq C_j \quad m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (13)$$

$$\sum_{n_i=1}^{N_i} \sum_{k=1}^K \sum_{m_j=1}^{N_j} \sum_{l=1}^K w_{n_i k m_j l} = V_{ij}^h \quad i, j = 1, \dots, N \quad (14)$$

$$\sum_{i=0}^N \sum_{n_i=1}^{N_i} \sum_{k=1}^K \sum_{l=1}^K w_{n_i k m_j l} = \sum_{q=0}^N \sum_{r_q=1}^{N_q} \sum_{l=1}^K \sum_{u=1}^K w_{m_j l r_q u} \quad (15)$$

$$m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (16)$$

$$x_{n_i k} = \{0, 1\} \quad n_i = 1, \dots, N, \quad i = 1, \dots, N, \quad k = 1, \dots, K \quad (17)$$

$$w_{n_i k m_j l s} \geq 0 \quad (18)$$

$$w_{n_i k m_j l s} \leq Mx_{n_i k} \quad (19)$$

$$w_{n_i k m_j l s} \leq Mx_{m_j l} \quad (19)$$

$$n_i = 1, \dots, N_i \quad m_j = 1, \dots, N_j \quad k, l = 1, \dots, K \quad i, j = 1, \dots, N$$

where  $M = h_{max}$

The above formulation is for continuous demand. For the case with a number of discrete demand scenarios, we only need to change the objective function to (20).

$$\min z = \sum_{s=1}^S \pi(W, s)g(s) \quad (20)$$

where  $S$  is the total number of scenarios.

Compared with the formulation in Section 2.2.2, two new constraints ((18) and (19)) are added in the MILP formulation. They guarantee that the flow variable  $w_{n_i k m_j l s}$  will be zero if machine  $n_i$  and machine  $m_j$  are not allocated to places  $k$  and  $l$ , respectively. The other constraints are similar to those in Section 2.2.2, with the only difference being that decision variables of flow assignments are changed to  $w_{n_i k m_j l}$ .

The MILP model linearizes the original formulation in Section 2.2.2 by increasing the number of variables and constraints in the second-stage flow assignment problem, which contains  $K^2$  binary variables,  $N_a K(K-1)$  continuous variables and  $1 + 4K + N(N-1) + 2N_a K(K-1)$  constraints (where  $K$  is the total number of machines and locations,  $N$  is the number of machine types, and  $N_a$  is the number of arcs in the network). As problem

size increases, the computational effort needed to obtain optimal solutions increases quickly. Only small to medium sized problems can be solved optimally. From our experiments, the limitation of problem size is around 15 machines of four types. Hence, heuristics are required for solving larger problems.

Urban et al (2000) have made a similar linearisation by introducing the variable  $w_{n_i k m_j l}$ , using

$$w_{n_i k m_j l} \geq v_{n_i m_j} + Mx_{n_i k} + Mx_{m_j l} - 2M \quad (21)$$

to replace the two constraint sets (18) and (19) used in our formulation. Compared with Urban et al's model, our formulation has more constraints. But according to our computational experience, less computational effort is required to solve our model. Hence we keep the two constraint sets ((18) and (19)) in the formulation of Section 2.3.

### 3 A heuristic approach

A new heuristic to solve stochastic IFLFAPs is proposed in this section. The approach decomposes the stochastic IFLFAP into two sub-problems, a flow map problem and a facility layout problem. The unique aspect of this approach is that the two sub-problems only need to be solved once. The idea of the heuristic comes from some observation of optimal layouts which were found from small-to-medium sized problems.

#### 3.1 Observation of optimal layouts

We compared optimal layouts obtained from stochastic and deterministic IFLFAPs. Both were formulated as MILPs with a finite number of demand scenarios as described in section 2.3. The deterministic case had one single scenario. All cases were then solved to optimality by using CPLEX12.1. For the stochastic IFLFAPs, we used five scenarios to describe the uncertain demand. For the deterministic case, two cases were studied, namely maximal demand (equal to the maximal production capacity) and expected demand (average of all possible demands). Hence, we obtained three layouts by solving one stochastic and two deterministic cases. These layouts were then compared by calculating the total expected material handling costs using a continuous distribution for evaluation (obviously having the same mean as the stochastic case with five scenarios). The evaluation model for a given machine layout is found by solving the following linear program parametrically in the demand  $h$ .

$$f(h) = \min \sum_{i=1}^N \sum_{j=1}^N \sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} v_{n_i m_j} r_{n_i m_j} \quad (22)$$

$$\sum_{i=0}^N \sum_{n_i=0}^{N_i} v_{n_i m_j} t_{m_j} \leq c_{m_j} \quad m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (23)$$

$$\sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} v_{n_i m_j} = V_{ij}^h \quad i, j = 1, \dots, N \quad (24)$$

$$\sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{n_i m_j} = \sum_{q=1}^N \sum_{r_q=1}^{N_q} v_{m_j r_q} \quad m_j = 1, \dots, N_j \quad j = 1, \dots, N \quad (25)$$



$$v_{n_i m_j} \geq 0 \quad n_i = 1, \dots, N_i \quad m_j = 1, \dots, N_j \quad i, j = 1, \dots, N \quad (26)$$

Notation used here is the same as in the model in Section 2.2.2. Decision variable  $v_{n_i m_j}$  refers to the flow between machine pairs  $n_i$  and  $m_j$ , while  $r_{n_i m_j}$  is the unit flow cost between machines  $n_i$  and  $m_j$ . Since the machine layout is given, location variables  $x_{n_i k}$  and  $x_{m_j l}$  in the model in Section 2.2.2 are known. This is then used to calculate the unit flow cost by  $r_{n_i m_j} = d_{kl} * x_{n_i k} * x_{m_j l}$ . Constraint sets (23) to (25) correspond to constraints (6) to (8). In constraint (24) the right-hand side  $V_{ij}^h$  is the required flow amount from type  $i$  to type  $j$  decided by demand  $h$ . The  $h$  varies between low and high demands, following a given distribution. By using parametric optimization on the linear evaluation model, the function  $f(h)$  of minimum flow cost against demand  $h$  is obtained exactly. The total minimum expected flow cost was then obtained by the integration  $\int f(h)g(h)dh$  where  $g(h)$  is the density function of the probability distribution used to describe demand.

Since the MILP is very hard to solve, small-to-medium sized problems with 10 or fewer machines were studied. 200 experiments were performed for each case of 6, 7, 8, 9, and 10 machines. Results are shown in Tables 1 and 2. Table 1 compares expected flow costs of the three layouts in terms of how much worse the deterministic layouts were relative to the stochastic one. All numbers are measured in percent. The CPU times are compared in Table 2.

Layouts obtained from the stochastic formulation (with five scenarios) had the lowest expected material handling costs, while the worst layouts were obtained from the deterministic model using expected demand. A major reason for this is that only some of the machines were needed to meet the average demand. Flows through utilised machines indicated where to allocate them, whilst those which were not used at all were randomly assigned to the remaining locations (as there were no signals to the model as to where they should be placed). When demand varied from low to high, those randomly placed machines were likely to increase the total expected costs.

The deterministic model with maximal demand had better machine layouts than the model with average demand. Even more importantly, in most cases the layouts obtained from the deterministic model with maximal demand were the same as the ones from the stochastic model. The reason is that all the machines are used to meet the requirements of maximal demand, so no machines were randomly assigned locations. Also, the flow for maximal demand contains within itself feasible flows for lower demands, making the layout for maximal demand 'robust' to demand variation between the lowest and the highest. Layouts obtained from this deterministic model were close to the best ones using stochastic programming whilst the CPU times required were much lower since the deterministic model only had to solve the model in Section 2.2.2 with one scenario. Hence, instead of solving stochastic models, solutions with similar qualities can be obtained from solving a deterministic model using maximal demand. Based on this observation, a greedy heuristic is developed, since, as problem size grows, also the deterministic case with maximal demand becomes numerically unsolvable; These are very hard problems to solve.

### 3.2 Greedy heuristic

The general idea of our greedy heuristic (see Figure 1) is to transform the stochastic IFLFAP into a classic QAP by using optimal flows from the deterministic model with maximal demand to determine flows between all pairs of machine copies. But also this deterministic problem is very hard to solve, hence, the heuristic will approximate the solution to this

Table 1: Expected flow costs, measured as percentage deviations from the stochastic design.

number of machines	(Max-Sto)%		(Exp-Sto)%	
	avg	std	avg	std
6	0.37	0.12	0.88	0.16
7	0.55	0.18	1.30	0.19
8	0.62	0.13	1.39	0.20
9	0.61	0.13	1.49	0.21
10	0.45	0.19	2.21	0.27

Table 2: CPU times (in seconds) of the different models.

number of machines	stochastic		Max		Exp	
	avg	std	avg	std	avg	std
6	0.63	0.05	0.07	0.00	0.04	0.00
7	5.82	0.64	0.46	0.03	0.18	0.02
8	30.21	3.30	1.15	0.07	0.56	0.05
9	231.12	24.95	7.94	0.62	4.97	0.89
10	795.88	137.92	19.54	2.29	17.01	4.49

deterministic problem. So we approximate at two levels: We use a deterministic model with maximal demand rather than the true stochastic model, and we approximate the solution to the deterministic model. The numerical results will show that the resulting heuristic is very fast and very accurate.

All existing algorithms for the QAP can then be applied. This will imply that whenever the QAP can be solved (hard as it is), the much more realistic case with multiple machine copies and random demand can also be solved (with very high accuracy). From a practical perspective, this is a very important step forward.

The heuristic has three steps. To make it easier to understand the heuristic, we shall use an example with six machines and six locations to illustrate. Figure 2 shows the six locations. The unit flow cost between two locations is equal to the taxi distance. The job sequence is fixed at  $type1 \rightarrow type2 \rightarrow type3$ . Demand in each production period varies between 0 and the production capacity of 30. In this example, the uncertain demand is assumed to follow a  $\beta$  distribution with shape parameters (5,2). The objective is to design a machine layout to minimize the expected material handling costs. Information about each machine type (capacity of copies of each type, number of copies of each type, and total capacity of each type) is shown in Table 3.

Table 3: Information about machine types

	type 1	type 2	type 3
Capacity of copies	30	12	16
Number of copies	1 (m11)	3 (m21, m22, m23)	2 (m31,m32)
Total capacity	30	36	32

In Step 1 we shall find a flow map, that is, a set of flows covering all pairs of machine copies (rather than all pairs of machine types). With such a flow map, we are back to a

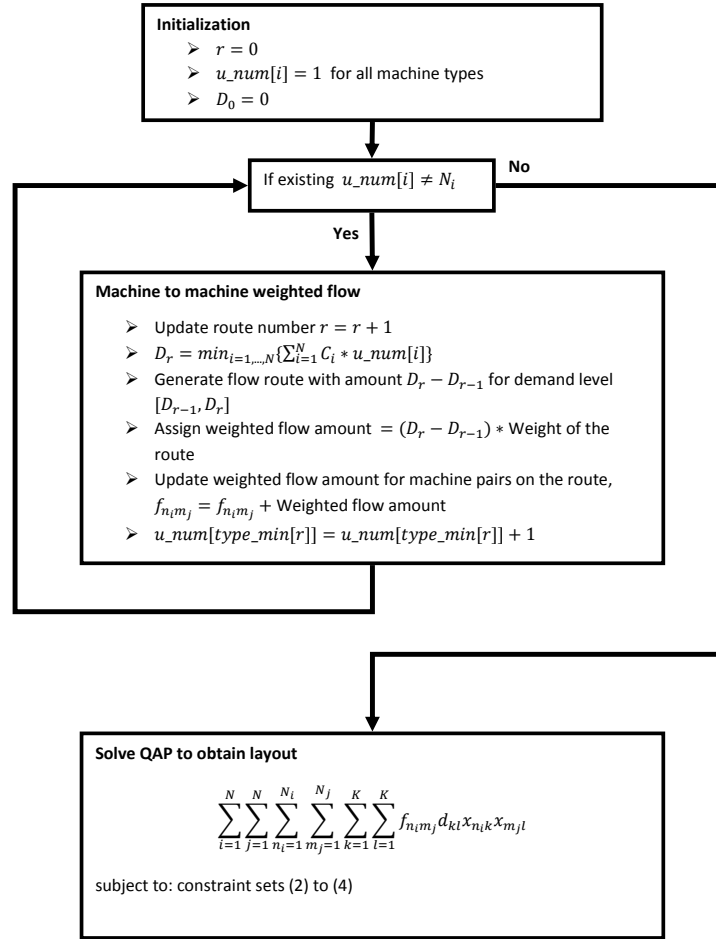


Figure 1: Flow chart of the Greedy Heuristic for the stochastic IFLFAP

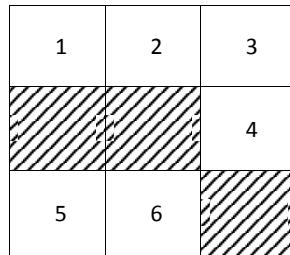


Figure 2: Six possible locations for machines

classical QAP. Unless the flow map is carefully selected, the resulting QAP will lead to a very bad layout. We know that the deterministic IFLFAP with maximal demand produces good solutions. It is also clear, from the problem structure, including the use of the taxi norm, that if the flow map is such that there exists a design with only neighboring machine copies having flow between them, then that flow map (and resulting design) must be very good (and often optimal). For this to happen, it is also intuitive that there should not be too many non-zeros in the flow map since in that case it will be impossible to place all machine copies with non-zero flows next to each other. The heuristic is based on these observations, but the quality of the heuristic is demonstrated in the numerical tests, and not in any way proven by these observations.

- *Step 1: Assign flows to pairs of machine copies*

Let us show the logic of this step by using our example from Table 3 and Figure 2. The results are shown in Figure 3.

All flow must pass through all machines types. So it must start in machine  $m11$  (the first and only copy of machine type 1), pass onto machine  $m21$  (the first copy of machine type 2), and finally onto  $m31$ . Since all copies of a type are the same, the determination of which copy is the "first" is arbitrary. We then send as much flow as we can on this path. The max flow is determined by machine  $m21$  as it has a capacity of only 12. This is illustrated as Route 1 in Figure 3. At this point it is necessary to move to the second copy of machine type 2, namely machine  $m22$ . This allows us to send another four units of flow, at which time machine  $m31$  is full (it has a capacity of 16). This is Route 2 in Figure 3. We then move onto the second copy of machine type 3, and get a path (Route 3) of capacity 8, filling up machine  $m22$ . Then finally we get Route 4 with a capacity of 6.

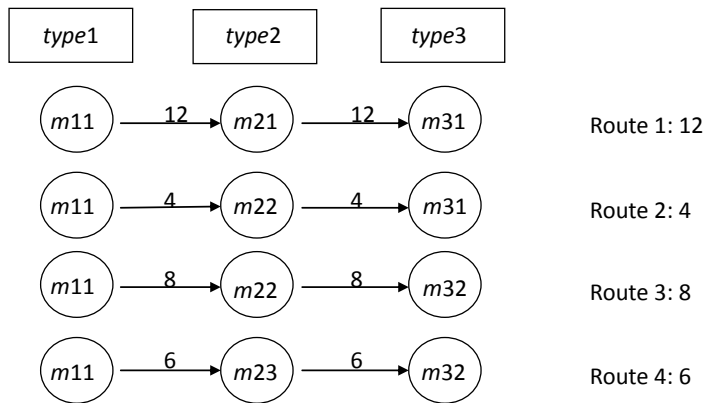


Figure 3: Four flow routes in the example

So, the approach divides the maximal demand into  $n$  ranges  $[0, D_1], \dots, [D_{n-1}, D_n]$ . The upper bound in the last range  $D_n$  equals maximal demand. In the  $r_{th}$  de-

mand range( $[D_{r-1}, D_r]$ ), the total capacity of machine type  $i$  is obtained by multiplying the number of utilized copies ( $u\_num[i]$ ) and a copy's capacity for that type ( $c_i$ ).  $type\_min[r]$  is the type which has the smallest total capacity in the  $r$ th range and the value of the upper bound  $D_r$  equals to the total capacity for  $type\_min[r]$ . Then the  $r$ 'th range updates to range  $(r+1)$ :  $[D_r, D_{r+1}]$ , by utilizing one more copy of  $type\_min[r]$  which results in the number of used copies being updated to  $(u\_num[type\_min[r]] + 1)$ . A new flow route is generated in range  $(r+1)$  by connecting the newly utilized copy of  $type\_min[r]$  with the existing available copies of the other types. The new route has flow amount of  $(D_{r+1} - D_r)$ .  $D_r$  is the lower bound which has the same value as the upper bound in the previous range. The upper bound  $D_{r+1}$  is the total capacity of  $type\_min[r+1]$  where  $type\_min[r+1]$  is the type with minimum total capacity in range  $(r+1)$ . One more copy of the type with minimum capacity is then utilized in the next demand range. The process continues until all machines are utilized and connected. Flow routes are obtained from each range of demand.

Results of Step 1 on the example are shown in Table 4 which includes range/route numbers ( $r$ ), the number of utilized copies of each type ( $u\_num[i]$ ), the type with minimum total capacity ( $type\_min[r]$ ), the maximal value on each range ( $D_r$ ) and flow amount of each route ( $D_r - D_{r-1}$ ).

Table 4: Step 1 of the example

range/route( $r$ )	$u\_num[1]$	$u\_num[2]$	$u\_num[3]$	$type\_min[r]$	$D_r$	$D_r - D_{r-1}$
1	1	1	1	2	12	12
2	1	1+1=2	1	3	16	4
3	1	2	1+1=2	2	24	8
4	1	2+1=3	2	1	30	6

- *Step 2: Add weights on each of the flow routes*

In this step, weights are added on each of the flow routes to show their importance. The idea is that the QAP needs to know that Route 1, somehow, is more important than Route 4 since it is used all the time, while Route 4 is used only when demand is very high. The priority of each route is determined by its chance of being used when demand varies between low and high. The flow route from the demand range  $[D_{r-1}, D_r]$  is used when demand is above  $D_{r-1}$ . Hence the flow route for the range  $[0, D_1]$  has the highest priority since it is used for all possible demands from 0 to  $D_n$ . The flow route for the last range  $[D_{n-1}, D_n]$  has the lowest priority since it is used only for the smallest range from  $D_{n-1}$  to  $D_n$ . We have adopted two weighting schemes. We call them the probabilistic weighting scheme and the simple weighting scheme. The probabilistic weighting scheme weights flow routes according to their probabilities of being used which are obtained from the distribution of demand. In the simple weighting scheme, a total number of  $n$  flow routes are weighted by  $n$  integers from 1 to  $n$  according to priorities of each route. Larger integers are associated to routes with higher priorities. After obtaining weights from each of the two weighting schemes, weighted flows are calculated for each route by the multiplication of flow and weight. For each pair of machines, the machine-to-machine weighted flow is then obtained by the sum of the pair's flows from all routes.

Table 5 shows weights and weighted flows of the four routes by applying the two weighting schemes on the example. The weights in the probability weighting scheme are obtained from the assumed  $\beta$  distribution of demand. Weighted flows between pairs of machines are then obtained and shown in Figure 4.

Table 5: Weighted flows

route	(Probability weighting scheme)		(Simple Weighting Scheme)	
	weight	weighted flow	weight	weighted flow
1	1	12	4	48
2	0.96	3.84	3	12
3	0.86	6.88	2	16
4	0.34	2.04	1	6

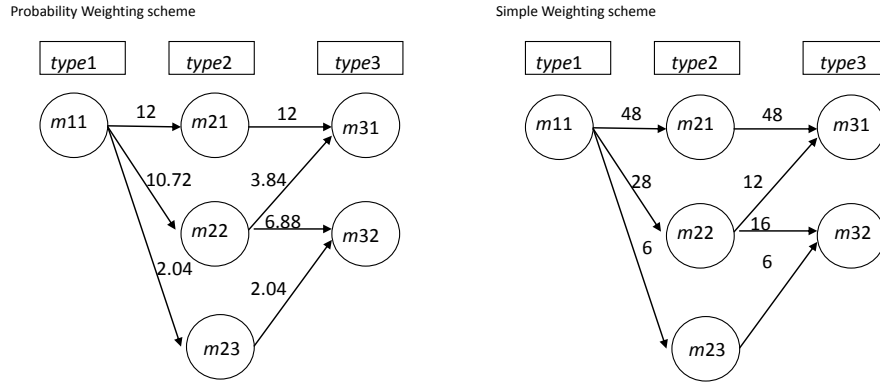


Figure 4: Weighted flows for two different weighting schemes

- *Step 3: solve QAP to obtain layout*

Since flows between all pairs of machines are known from Step 2, the original stochastic IFLFAP has been transformed to a classic QAP.

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{N_i=1}^{N_i} \sum_{m_j=1}^{M_j} \sum_{k=1}^K \sum_{l=1}^K f_{n_i m_j} d_{kl} x_{n_i k} x_{m_j l} \quad (27)$$

subject to: constraint sets (2) to (4).

Here  $f_{n_i m_j}$  is the weighted flow between machines  $n_i$  and  $m_j$ , obtained from Step 2. To solve this QAP, all existing methods, both exact and heuristic, can be applied. In our heuristic, we solve the QAP optimally by using CPLEX 12.1. For testing purposes, it is useful to solve the QAP to optimality whenever possible, since we then know that all errors stem from the new heuristic, and no noise is added by the solution of the QAP.

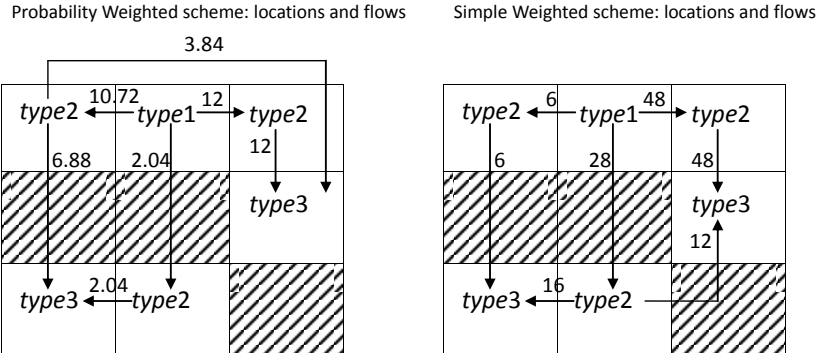


Figure 5: Results of locations and flows in the Example for two different weighting schemes

For the 6-machine example, machine locations and weighted flows are shown in Figure 5. Since all copies of a machine type are exactly the same, the locations of machines are shown by type. Note that although the flows are different, the designs are the same. Hence, in fact, the two schemes produced the same solution.

## 4 Experimental Results

In order to evaluate the proposed greedy heuristic, numerical experiments have been studied on problems with size from 6 to 30 machines. Data sets for these problems are randomly generated. Distances between possible locations are measured by the taxi norm. Parameters including beta distribution for demand, machine capacity, the number of duplicates of each machine type and the visiting sequence of machine types are randomly generated.

Our greedy heuristic and the random problem generator were written in C++ interfaced with CPLEX 12.1. All reported CPU times are from a personal computer with Intel Core 2 Duo 2.99GHZ CPU and 3.25GB of RAM.

200 data sets are randomly generated with the above procedure for problem sizes from 6 to 10. In these cases, optimal solutions of the stochastic MILP (presented in Section 2.3) are used as benchmark to evaluate the heuristic performance. Table 6 presents how much the heuristic results of our two weighting schemes, the probability weighting scheme (PWS) and the simple weighting scheme (SWS), are away from the benchmark result of the stochastic MILP (Sto). CPU times of the PWS, SWS and Sto are shown in Table 7. The numerical results show that the heuristic performs extremely well with results within 1% of the optimal solution, while the computing time is only a few seconds. For the case of 10 machines CPU-times are cut by a factor of over 500.

Table 6: Compare the cost difference between PWS and Sto, SWS and Sto for problem size from 6 to 10

number of machines	(PWS-Sto)%		(SWS-Sto)%	
	avg	std	avg	std
6	0.06	0.03	0.01	0.01
7	0.06	0.02	0.35	0.10
8	0.19	0.08	0.21	0.13
9	0.25	0.10	0.26	0.15
10	0.23	0.13	0.34	0.18

Table 7: CPU times(in seconds) of PWS, SWS and Sto

number of machines	PWS		SWS		Sto	
	avg	std	avg	std	avg	std
6	0.03	0.01	0.06	0.02	0.63	0.05
7	0.09	0.06	0.10	0.04	5.82	0.64
8	0.29	0.08	0.30	0.10	30.21	3.30
9	0.50	0.12	0.53	0.15	231.12	24.95
10	1.37	0.10	1.42	0.13	795.88	137.92

For larger problems, two data sets are generated for each of the problem sizes 15, 20, 25 and 30. Since optimal solutions of the stochastic MILPs are no longer available for these larger cases, another benchmark is required. The one chosen is the solution of the deterministic model with maximal demand (MAX) because Table 1 shows that this deterministic solution is very close to the stochastic solution but with less computing time. The numerical results show that the heuristic results are very close to the benchmark, but with considerable less computing time. For 25 machines, the largest case where we could solve the benchmark problem to optimality, CPU-times are cut by a factor of about 2500.

Table 8: Results of cost and CPU time (in seconds) of PWS, SWS and MAX for larger problem size

number of machines		PWS			SWS			MAX	
		cost	(PWS-MAX)%	CPU time	cost	(SWS-MAX)%	CPU time	cost	CPU time
15	case 01	556.10	-5.29	1.84	578.91	-1.41	1.78	587.16	13984.80
	case 02	290.25	0.09	2.39	290.19	0.07	2.99	290	30744.91
20	case 01	634.75	0.61	25.89	626.12	-0.76	21.25	630.89	41940.73
	case 02	739.71	-0.14	22.30	746.85	0.83	29.47	740.71	79416.82
25	case 01	331.44	0.00	34.42	331.43	0.00	32.75	331.44	90252.64
	case 02	1170.64	-0.42	77.40	1254.78	6.73	80.40	1175.63 †	
30	case 01	914.31	-11.02	1090.53	919.11	10.55	1716.01	1027.57 †	
	case 02	1222.58	-0.66	10800.01	1225.95	-0.38	7911.95	1230.65 †	

†: the best feasible solution obtained within time limit of 129600 seconds.



## 5 Conclusion

A distributed layout is obtained by simultaneously determining the facility layout problem and the flow assignment problem. This paper studies how to design a distributed layout which minimizes the total expected material handling cost in a stochastic environment where demand is uncertain. We developed a greedy heuristic which transforms the stochastic integrated layout problem into a classical QAP by assigning weighted flows to machine copies. Two weighting schemes were applied in order to show the difference of priorities of different flow routes. Compared with other heuristics for solving this integrated problem, the main benefit of this heuristic is that the time-consuming QAP only needs to be solved once. The numerical tests show that our heuristic can effectively find solutions within 1% of the best alternative, but these alternatives are orders of magnitude harder to solve. Hence, we can find very good solutions to a stochastic QAP by solving a deterministic QAP of the same dimension: stochastics is added at no computational cost. Further research will be the development of a greedy heuristic for the integrated layout problem in a stochastic environment where both demand and products (visiting sequence of machine types) are uncertain.

## References

- Balakrishnan J, Cheng C, Wong K (2003) Facopt: a user friendly facility layout optimization system. *Computers and Operations Research* 30(11):1625–1641
- Baykasoglu A, Cindy N (2001) A simulated annealing algorithm for dynamic plant layout problem. *Computers and Operations Research* 28(14):1403–1426
- Benjaafar S, Sheikhzadeh M (2000) Design of flexible plant layouts. *IIE Transactions* 32(4):309–322
- Bukard R, Bonniger T (1983) A heuristic for quadratic boolean programs with applications to quadratic assignment problems. *European Journal of Operational Research* 13(4):347–386
- Chiang W, Chiang C (1998) Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation. *European Journal of Operational Research* 106(2–3):457–488
- Francis R, White J (1974) *Facility layout and location: an analytical approach*. Prentice Hall, Englewood Cliffs, NJ
- Helm S, Hadley S (2000) Tabu search based heuristics for multi floor facility layout. *International Journal of Production Research* 38(2):365–383
- Heragu S, Alfa A (2003) Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research* 57(2):190–202
- Heragu S, Kusiak A (1988) Machine layout problem in flexible manufacturing systems. *Operational Research* 36(2):258–268
- Koopmans T, Beckman M (1957) Assignment problems and the location of economic activities. *Econometrica* 25(1):53–76
- Lee K, Han S, Roh M (2003) An improved genetic algorithm for facility layout problems having inner structure walls and passage. *Computers and Operations Research* 30(1):117–138
- Lium AG, Crainic TG, Wallace SW (2009) A study of demand stochasticity in stochastic network design. *Transportation Science* 43(2):144–157, DOI 10.1287/trsc.1090.0265
- Misevicius A (2003) A modified simulated annealing algorithm for quadratic assignment problem. *Informatica* 14(4):497–514

- Salimanpur M, Jafari A (2008) Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers and Industrial Engineering* 55(3):609–619
- Singh S, Sharma R (2006) A review of different approaches to the facility layout problems. *International Journal of Advanced Manufacturing Technology* 30(5–6):425–433
- Singh S, Singh V (2010) An improved heuristic approach for multi-objective facility layout problem. *International Journal of Production Research* 48(4):1171–1194
- Suzaki K (1985) Japanese manufacturing techniques: Their importance to u.s. manufacturers. *Journal of Business Strategy* 5(3):10–19
- Taghavi A, Murat A (2011) A heuristic procedure for the integrated facility layout design and flow assignment problem. *Computer and Industrial Engineering* 61(1):55–63
- Tavakkoli-Moghaddain R, Shanyan E (1998) Facility layout design by genetic algorithms. *Computers and Industrial Engineering* 35(3):527–530
- Thapalia BK, Crainic TG, Kaut M, Wallace SW (2012) Single source single-commodity stochastic network design. *Computational Management Science* 9(1):139–160, DOI 10.1007/s10287-010-0129-0, special issue on ‘Optimal decision making under uncertainty’
- Tomkins J, Bozer Y, Frazelle E, Tanchoco J, Trevino J (2003) *Facility Planning*. John Wiley and Sons, New York
- Urban T, Chiang W, Russell R (2000) The integrated machine allocation and layout problem. *International Journal of Production Research* 38(12):2911–2930