Hybrid Estimation of Distribution Algorithm for solving Single Row Facility Layout Problem

Chao Ou-Yang [a,*], Amalia Utamima [a,b]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, Taipei City, 106, Taiwan, ROC
[b] Information System Department, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

ABSTRACT

The layout positioning problem of facilities on a straight line is known as Single Row Facility Layout Problem (SRFLP). The objective of SRFLP, categorized as NP Complete problem, is to arrange the layout so that the sum of distances between all facilities' pairs can be minimized.

Estimation of Distribution Algorithm (EDA) efficiently improves the solution quality in first few runs, but the diversity loss grows rapidly as more iterations are run. To maintain the diversity, hybridization with metaheuristic algorithms is needed. This research proposes Hybrid Estimation of Distribution Algorithm (EDAhybrid), an algorithm which consists of hybridization of EDA, Particle Swarm Optimization (PSO), and Tabu Search. Another hybridization algorithm, extended Artificial Chromosomes Genetic Algorithm (eACGA), is also built as benchmark. EDAhybrid's performance is tested in 15 benchmark problems of SRFLP and it successfully achieves optimum solution. Moreover, the mean error rates of EDAhybrid always get the lowest value compared to other algorithms.

SRFLP can be enhanced by considering more constraints, so it becomes enhanced SRFLP. Computational results show that EDAhybrid can also solve Enhanced SRFLP effectively. Therefore, we can conclude that EDAhybrid is a promising metaheuristic algorithm which can be used to solve the basic and enhanced SRFLP.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The Single Row Facility Layout Problem (SRFLP) is taken into account when multiple products with different production volumes and different process routings need to be manufactured. The objective of SRFLP is to set up the facilities so that sum of the distances between all facility pairs can be minimized (Amaral, 2006). Because SRFLP is proven to be a Nondeterministic Polynomial-time (NP) Complete problem, the exact methods applied to large instances of the problem are time consuming. Hence, heuristic methods are built to acquire a near optimal solution to the problem (Samarghandi & Eshghi, 2010).

Existing researches which applied metaheuristics have contributed to solve the SRFLP. Despite its contribution, each study, in fact, has particular benefits and limitations. By solving SRFLP effectively, it is hoped that an algorithm can also succeed in solving the different cases of Facility Layout Problem.

Estimation of Distribution Algorithms (EDAs) are stochastic optimization techniques that explore the space of potential solutions by exploiting the inter variable dependency and sampling probabilistic models of promising candidate solutions (Hauschild & Pelikan, 2011). Therefore, they could efficiently solve hard optimization problems (Zhang & Muhlenbein, 2004). Moreover, EDAs are predicted to potentially effective to solve SRFLP. EDAs may cause overfitting in the search space and cannot provide the general information (Santana, Larrañaga, & Lozano, 2008). EDAs efficiently improve the solution quality in the first few runs, but the diversity loss grows rapidly as more iteration is run. To maintain the diversity, hybridization with metaheuristic algorithm is needed. EDAs are used to characterize the parental solutions and to search around the current solution space. After that, metaheuristics might introduce new solutions into the population to maintain diversity, which can avoid the spremature convergence of EDAs (Chen, Chen, Chang, & Chen, 2012).

This study proposes Hybrid Estimation of Distribution Algorithm (EDAhybrid), an algorithm which consists of hybridization of Estimation of Distribution Algorithm (EDA), Particle Swarm Optimization (PSO), and Tabu Search algorithm to surmount the basic and enhanced SRFLP. PSO is utilized as metaheuristic algorithm for maintaining the diversity of EDA. Tabu Search explores the global best value achieved in every iteration. Other hybridization algorithm is built as benchmark; that is extended Artificial Chromosomes Genetic Algorithm (eACGA).

The objectives of this research are as follows:

a. To develop EDAhybrid, a new meta-heuristic algorithm which is the hybridization of Estimation of Distribution Algorithm, Particle Swarm Optimization, and Tabu Search Algorithm.

* Corresponding author. Tel.: +886 2 27376340.
E-mail address: ouyang@mail.ntust.edu.tw (C. Ou-Yang).

b. To design an enhanced Single Row Facility Layout Problem that considers more constraints which include not only flow, length, and clearance space, but also the installation cost and safety reason. The detail of this objective is explained in Section 2.3.

c. To apply EDAhybrid algorithm to solve the basic and enhanced Single Row Facility Layout Problems.

2. Literatures review

2.1. Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) are stochastic optimization techniques that explore the space of potential solutions by exploiting the inter variable dependency and sampling probabilistic models of promising candidate solutions (Hauschild & Pelikan, 2011). EDAs construct a probabilistic model to get the parental distribution and sample new solutions (Pelikan, Goldberg, & Lobo, 2002). EDAs make use of sampling from probabilistic models that avoid the disruption of partial dominant solutions (Santana et al., 2008) and they makes differentiation from Genetic Algorithms (GAs). EDAs might be a powerful method capable of capturing and manipulating the building blocks of chromosomes. Therefore, they could efficiently solve hard optimization problems (Zhang & Muhlenbein, 2004). The complete reviews of EDAs are presented by Lozano et al. (2006) and Hauschild and Pelikan (2011).

The Extended Artificial Chromosome Genetic Algorithm (eACGA) is derived from Artificial Chromosome with Genetic algorithm (ACGA), an algorithm that joins EDA and GA in effective manner (Chen et al., 2012). ACGA is able to interpret parental distribution by sampling new solutions from the univariate probabilistic models and also genetic operators. The main difference characteristic of probabilistic models in ACGA compared to most EDAs is ACGA samples new individuals periodically whereas EDAs generate new solutions entirely (Chang, Hsieh, Chen, Lin, & Huang, 2009). ACGA was used to chip resistor scheduling problem (Chang et al., 2009) in order to accelerate the convergence rate in GA.

A research has proposed eACGA as a solution for scheduling problems. In (Chen et al., 2012), ACGA is improved as eACGA and employed to solve permutation flowshop scheduling problems. eACGA collects not only the univariate probabilistic model, like ACGA just discussed, but also the bivariate probabilistic model. eACGA seems to be very powerful since it considers both univariate and bivariate statistic information. The use of variable interaction in bivariate probabilistic model can represent better individual information for EDA part in eACGA (Chen et al., 2012).

2.2. Metaheuristic algorithms for solving Single Row Facility Layout Problem

SRFLP is proven to be a NP Complete problem. Therefore, the exact methods applied to large instances of the problem are time consuming; hence heuristic methods have been built to acquire a near optimal solution of the problem (Samarghandi & Eshghi, 2010).

Recent papers have already tried to solve SRFLP with metaheuristic approaches. A simulated annealing method to handle SRFLP was developed by Heragu and Alfa (1992). Solimanpur, Vrat, and Shankar (2005) improved mixed integer model and an ant algorithm to overcome SRFLP. Samarghandi, Taabayan, and Jahantigh (2010) used the new factoradic based Particle Swarm Optimization for coding and encoding technique.

Samarghandi et al. presented and proved a theorem to find the optimal solution of a special case of SRFLP. The results obtained by the theorem is proven to be very useful in reducing the computational efforts when a new algorithm based on Tabu Search for the SRFLP (Samarghandi & Eshghi, 2010) is performed. Datta, Amaral,

and Figueira (2011) offer permutation based genetic algorithm with specially designed crossover and mutation operators for getting optimal solution of SRFLP.

2.3. Enhanced Single Row Facility Layout Problem

Minimizing material-handling costs and providing a safe workplace for employees are the main considerations in the design of manufacturing layouts (Heragu, 2008). In each case, a number of suitable locations might be available; however, the cost of assigning a machine might differ in each location. It could depend on such factors as the existing condition of the site and/or the modification needed to the foundation likewise as the environs (Sule, 2009).

According to Drira, Pierreval, and Hajri-Gabouj (2007), the objectives of research in facility layout problem at manufacturing industry are minimizing space cost (for unequal size facility), handling cost, rearrangement cost (for dynamic layout), backtracking and bypassing, traffic congestion (for cellular layout), and shape irregularities (for unequal size facility). In SRFLP, the objectives of related researches (Amaral, 2009; Samarghandi et al., 2010; Solimanpur et al., 2005) usually focus on minimizing the weighted sum of distances by considering length of facilities and total materials flowing between facilities so that material handling cost can be minimized.

Heragu (2008) presents case study about real world problem in manufacturing company that consider technological constraint in a layout positioning. The technological constraint here means two facilities cannot be adjacent by each other. This kind of constraint, which cannot put the specified two facilities by each other, in construction site layout problem is called safety constraint (Mawdesley & Al-Jibouri, 2003). In another book, Sule (2009) presents a fixed cost concept that already described earlier about the cost for installing a facility that might differ in each location.

Combining these two ideas, we can enhance the SRFLP objective function to be more comprehensive and more applicable in real case by considering fixed cost and safety constraints. This problem is labeled as Enhanced SRFLP.

3. Problem statement

3.1. Single Row Facility Layout Problem

This research considers Single Row Facility Layout Problem (SRFLP) with different sizes of facilities. The objective is to minimize Z which stands for sum of the distances between all facility pairs. The length l_i of each facility i and a $n \times n$ matrix $T = [T_{ij}]$ are given; T_{ij} refers to the traffic loads between facilities i and j. The distance between two facilities is supposed to be taken between their mid points. ABSMODEL, proposed by Heragu and Kusiak (1991), is a well known model for solving SRFLP. ABSMODEL is illustrated in Eq. (1). Heragu and Kusiak (1991) defined d_{ij} as Eq. (2). Note that D_{ij} is not necessarily equal to s_{ij}. If facility k is placed between facilities i and j with $s_{ij} = 0$ then $D_{ij} = l_k$.

$$\min \; z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} T_{ij} d_{ij}$$

$$\text{s.t.} : \; d_{ij} \geqslant \frac{1}{2}(l_i + l_j) + s_{ij}$$

$$d_{ij} \leqslant 0; \quad i = 1, 2, \ldots, n-1; \quad j = i+1, \ldots, n$$

(1)

T_{ij} is the traffic loads between facilities i and j, d_{ij} the distance between the centers of the facilities i and j, s_{ij} the necessary clearance or gap between the two facilities, l_i is the length of facility i.

$$d_{ij} = \frac{1}{2}(l_i + l_j) + D_{ij}$$

(2)

D_{ij} is the space between facilities i and j.

3.2. Enhanced Single Row Facility Layout Problem

Combining the safety constraint presented by Heragu (2008) and the fixed cost concept mentioned by Sule (2009), we could enhance the SRFLP objective function to be more extensive and more applicable in real case by considering the safety constraint and the installation cost. The problem is labeled as Enhanced SRFLP.

The Enhanced Single Row Facility Layout Problem (enhanced SRFLP) is an enhanced model of single row facility layout which considers length of facilities, traffic loads between facilities, installation cost of each facility and also safety constraint. Similar to SRFLP, the objective of enhanced SRFLP is to minimize Z which stands for sum of the distances between all facility pairs.

The objective function of Enhanced SRFLP is shown in the following equation:

$$\min \ z = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} T_{ij}d_{ij}B_{ij} + \sum_{i=1}^{n}\sum_{x=1}^{n}\delta_{ix}C_{ix}$$

$$\text{s.t.}: \ d_{ij} \geqslant \frac{1}{2}(l_i + l_j) + s_{ij} \qquad (3)$$

$$d_{ij} \leqslant 0; \quad i = 1, 2, \ldots, n-1; \quad j = i+1, \ldots, n$$

T_{ij} is the traffic loads between facilities i and j, d_{ij} the distance between the centers of the facilities i and j, s_{ij} the clearance space between the two facilities, δ_{ix} the permutation matrix variable (equals to 1 if facility i is assigned to location x, otherwise its equals to 0), C_{ix} the construction cost of assigning facility i to location x, and B_{ij} is the interactive cost of assigning facility i on the location neighboring facility j.

To calculate Z, the penalty cost will be given to a solution violating the safety constraint. B_{ij} is related to the safety constraint handling. B_{ij} equals to penalty cost if facility i cannot be adjacent to facility j due to safety constraint, otherwise it equals to 1.

4. Methodology

Before developing an EDAhybrid algorithm, we firstly build a modified eACGA to deal with SRFLP. The EDA and GA are also developed to become benchmarks. EDA developed here is an extended version of the basic EDA. EDA used in this research considers two probabilistic models suggested by Chen et al. (2012). eACGA can perform better than EDA and GA, but for large number of facilities, its performance is decreased.

Since finding shows that eACGA could not find optimal solution, PSO is chosen to become alternative algorithm to be hybridized with EDA. PSO is chosen because its computational time is faster than GA. This will be a big advantage in dealing with large number of facilities. The hybridization of EDA and PSO is created and gets running time faster than eACGA, but its performance still cannot get optimal solution in problems with large number of facilities. So, a local search based on Tabu Search is added to increase its performance. The algorithm then is named EDAhybrid.

In a problem containing n facilities labeled as F^1, F^2, ..., F^n, a sequence X contains permutation numbers of all the labels. X can be represented by assigning facility $X[i]$ to location i, where $i \in [1, n]$. Different sequences in X represent different layout solutions. The representation of chromosome is shown in Table 1. The number of chromosome's location is adjusted with the number of facilities in the benchmark problems. Table 1 shows the example of chromosome representation with 5 facilities. $X[2] = 3$ means assigning facility 3 to location 2.

Table 1
Chromosome illustration.

i	1	2	3	4	5
X[i]	5	3	2	1	4

4.1. General procedure of eACGA

The eACGA framework taken from Chen et al. (2012) is modified to be more suitable with our code to solve SRFLP. The eACGA procedure starts with initialization of all variables. A population consisting of a number of chromosomes is also initialized randomly. The main iteration starts with a decision for choosing between EDA or GA procedure to be performed. If g, which represents the current generation, can be divided by 2, then EDA procedure is executed, otherwise GA procedure is performed.

EDA procedure starts with a selection process that attempts to choose chromosomes with better fitness values. Chromosomes' fitness values are sorted first and a group of chromosomes with better fitness are selected. The probabilistic models form the ordinal (univariate model) and dependency (bivariate model) matrices from selected chromosomes. These probabilistic models will be explained in Section 4.2.1. After the two probabilistic models are established, a group of chromosomes for the next generation are generated with the sampling process.

On the other side, GA procedure consists of crossover and mutation processes. The crossover rate and mutation rate decide whether chromosomes mate and mutate, respectively. We use 0.7 for the crossover rate and 0.3 for the mutation rate. Roulette wheel selection chooses the parental chromosomes. Two point crossover operator mates two chromosomes with better fitness values as the parents. The mutation then probably occurs and chooses between three mutations operators that flip, swap, or slide, and that is to be undertaken.

The replacement step replaces the parental chromosomes with their offspring. This step is only done once in each generation. The fitness value of every chromosome then is calculated. The elitism strategy is also done in this step. The elitism rate is 10% of the population size. A group of new generation chromosomes are sent to be processed into the next iteration.

4.2. Procedure of EDAhybrid

The procedure of EDAhybrid is shown in Fig. 1, while the pseudocode of EDAhybrid is provided in Fig. 2. The procedure starts with initialization of all variables. A swarm consisting of a number of particles is also initialized randomly. Initialization of every particle uses random permutation from 1 until n facilities. The main iteration is preceded with initialization and setting of all parameters. Next, a decision to choose between EDA or PSO procedure is made. If g, which represents the current generation, can be divided by 2, EDA procedure is executed, otherwise PSO procedure is performed.

Population replacement replaces the parental chromosomes with their offspring. Particles' best fitness and theirs location are expressed as pBest. Global best fitness among particles and its location is called gBest. Fitness calculation followed by pBest and gBest updating are then done after.

Local search based on Tabu Search is carried out with gBest as the input. If Tabu Search can get better solution than gBest, then gBest is replaced with solution from Tabu Search. Next, if the EDA part is performed in the following generation, the elitism strategy is undertaken. The elitism rate is 10% of the population size. The new generation of particles is processed into next
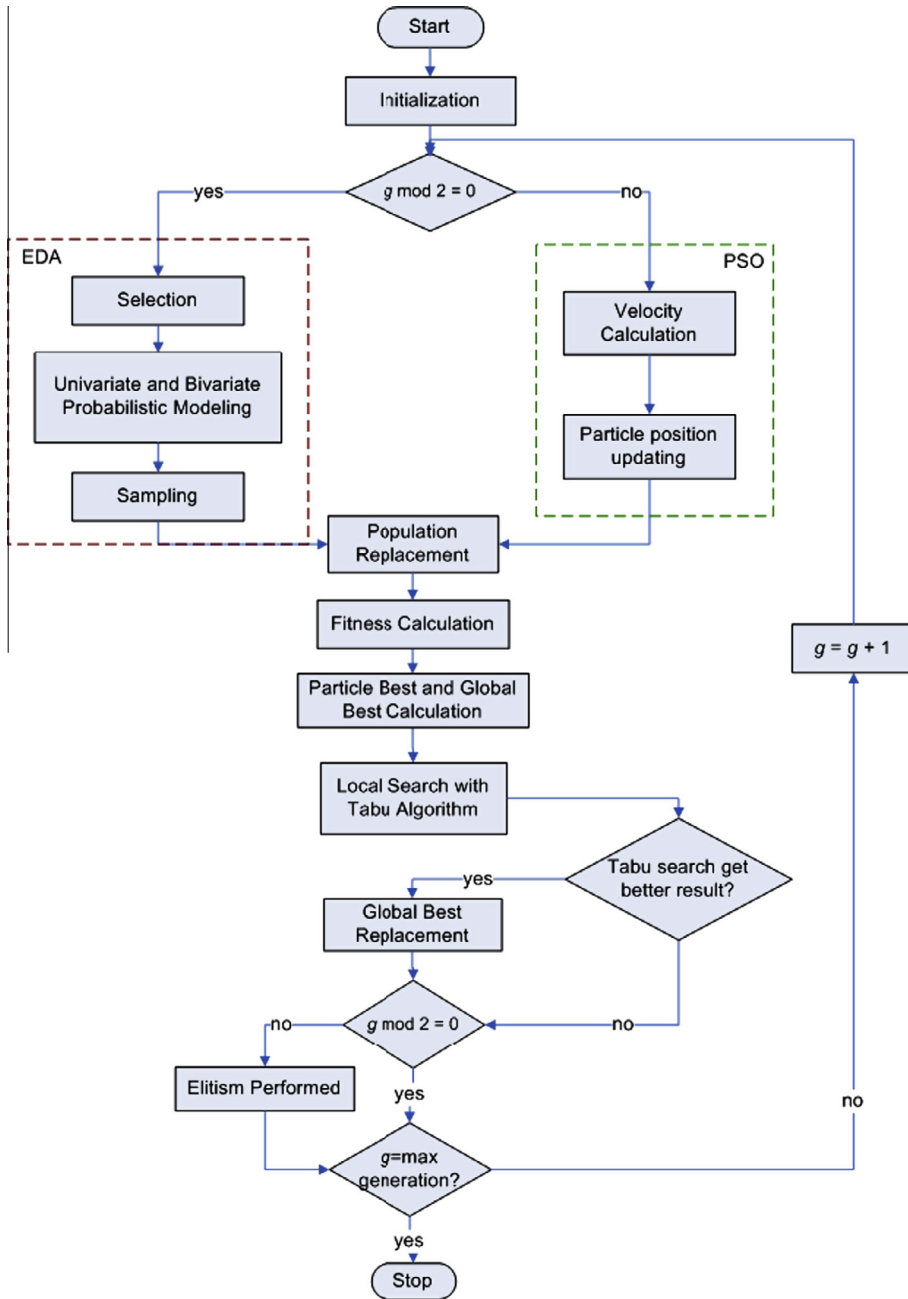
Fig. 1. EDAhybrid framework.

iteration. Detailed procedure of EDA, PSO, and Tabu Search will be explained in next sections.

4.2.1. Estimation of Distribution Algorithm part

In every generation g, a group of C particles which have better fitness are selected. The particles are labeled as X^1, X^2, ..., and X^C, where C is half of population size (Hauschild & Pelikan, 2011). The distribution of parental particles are interpreted by sampling new solutions from the univariate and bivariate probabilistic model (Chen et al., 2012).

We develop our univariate probabilistic model, while the bivariate probabilistic model is adopted from Chen et al. (2012). The univariate or ordinal probabilistic model, $\phi_{i[i]}$ in Eq. (4) shows the importance of facilities in the sequence. It represents how many times facility i is placed at position [i] at current generation. $A_{i[i]}^k$

is set to 1 if facility i is placed at position [i], otherwise it is set to 0.

$$\phi_{i[i]} = \sum_{k=1}^{C} A_{i[i]}^k \tag{4}$$

where $i = 1, ..., n$; $k = 1, ..., C$.

The bivariate or dependency probabilistic model ψ_{ij}, in Eq. (5), represents how many times facility j is placed immediately after facility i. B_{ij}^k is set to 1 if facility j is placed next to facility i, otherwise it is set to 0.

$$\psi_{ij} = \sum_{k=1}^{C} B_{ij}^k \tag{5}$$

where $i, j = 1, ..., n$; $k = 1, ..., C$.

P : a swarm of particles

S : a group of selected particles that have better fitness value than others

$pBest$: particles' best fitness and theirs location

$gBest$: global best fitness among particles and its location

$tabuSol$: solution of Tabu Search part

```
 1  Set up and initialize EDAhybrid parameters;
 2  Initialize swarm P randomly;
 3  Compute pBest and gBest;
 4  for g = 1 until maxGen do
 5      if g is divisible by 2
 6            Select S;
 7            Calculate Univariate probabilistic model(S);
 8            Calculate Bivariate probabilistic model(S);
 9            Sampling();
10      else
11            Calculate velocity of P();
12            Update P();
13      end if
14       Calculate fitness value;
15      Update pBest and gBest;
16      TabuSearch(gBest);
17      if  tabuSol <  gBest
18            gBest ← tabuSol;
19      end if
20       if g is not divisible by 2
21              Elitism();
22       end if
23  end for
```

Fig. 2. EDAhybrid pseudocode.

Let $P_{i[i]}$ be the probability value of assigning facility i at position $[i]$. Selecting facility i has better probability value than other facilities when both probabilistic models' statistical information is used, $\phi_{i[i]}$ is added by ψ_{ij}. They will summarize the probability values of all unassigned facilities that could be set at position $[i]$.

For every particles' offspring O^1, O^2, ..., and O^{2C}, some methods are used to assign facilities to a specified location. Selecting a facility at the first location, our approach picks randomly the first facility that appears in the group of selected C particles, that is $X[1]$. We think that this approach is more promising than the one proposed in Chen et al. (2012), which generates facility randomly.

For assigning the remaining facilities, we modify the formula used in Chen et al. (2012) which multiplying $\phi_{i[i]}$ with ψ_{ij}. We change the formula to become adding $\phi_{i[i]}$ with ψ_{ij}, that is Eq. (6). The reason is because add operation, not multiplication one, gives better result when tested in EDAhybrid.

$$P_{i[i]} = \frac{\phi_{i[i]} + \psi_{ij}}{\sum_{i \in \Omega}(\phi_{i[i]} + \psi_{ij})} \quad (6)$$

where $P_{i[i]}$: the probability value of assigning facility i at position $[i]$, $[i] = 2, 3, ..., n$; $i,j = 1, 2, ..., n$, Ω = set of unassigned facilities. The pseudocode shown in Fig. 3 demonstrates the assignment procedure for placing facilities into location 2 until n. This pseudocode is a modified version from the one presented in Chen et al. (2012).

4.2.2. Particle Swarm Optimization part

PSO procedure starts when current generation is odd and begins with initialization of all related parameters. Next, the velocities calculation of every particle is operated. If the velocity is larger, the particle is more likely to change to a new permutation sequence. The velocity update formula remains the same like provided in Haupt and Haupt (2004), while the particle update process is changed. EDAhybrid uses permutation based particle updating based on concept from Hu, Eberhart, and Shi (2003).

4.2.3. Tabu Search part

The best solution from individual generated by EDA or PSO part in every generation is the input for Tabu Search. Fig. 4 provides the pseudocode of Tabu Search in EDAhybrid. The Tabu Search procedure is performed for n (equal to the number of facilities) generations. Swaplist is constructed in every tabu generation; this process is shown in line 3 of Fig. 4. Swaplist consists of three columns, the first and second columns are the specified facility that will be swapped and the last column provides the cost of that *move*. When calculating a *move*'s cost, the algorithm checks whether the *move* is taboo or not by looking in *tabulist*. If the *move* is listed in *tabulist* and the *move*'s cost is greater than *tabuSolution*, a penalty cost is added; otherwise the *move* is accepted without a penalty cost.

The algorithm records the *move* with minimum cost in line 8. Tabulist is updated in line 9. If the currentSolution is better than tabuSolution, then tabuSolution is updated. The iterations continue to run until n generations.

At the end of this local search, the algorithm checks whether *tabuSolution*, as the output of Tabu Search, can find better solution than the current best solution. This part is shown in line 14 until 16. If *tabuSolution* succeeds to get lower cost, then the current best solution is replaced with *tabuSolution*.

Ω: set of unassigned facilities.

F: set of assigned facilities. *F* is empty in the beginning.

θ: a random probability is drawn from U(0,1)

i: a selected facility by proportional selection

k: the element index of the facility's position

n: number of facilities

```
1   Procedure Sampling()
2   F ← Φ
3   for k=2 until n do
4          θ ← U(0,1)
5          Select a facility i satisfy θ ≤ P_i[i], where i Є Ω
6          F(k) ← i
7          Ω ← Ω\i
8   end for
```

Fig. 3. Assignment of facilities pseudocode.

tabuSolution   : solution of Tabu Search

BestSolution   : current global best solution

Move   : the swap of 2 facilities

n   : number of facilities in the problem

```
1   tabuSolution ← BestSolution
2   while i ≤n do
3      Construct swaplist ()
4      Calculate cost for every move in swaplist()
5      if move is taboo and move's cost > tabuSolution
6          give move penalty cost()
7      end if
8      currentSolution ←min(cost)
9      Update tabulist()
10     if  currentSolution < tabuSolution
11          tabuSolution ← currentSolution
12     end if
13  end  while
14  if  tabuSolution < BestSolution
15      BestSolution ← tabuSolution
16  end if
```

Fig. 4. Tabu Search pseudocode.

5. Result analysis

5.1. Parameters setting

There are some parameters used in EDAhybrid. General parameters, which are population size and generation size, are based on the size of the problem. The larger the number of facilities, the more population and generation are needed to get an optimal solution. The number of facilities is represented by *n*. The population size and generation size, for example *4n*, *10n*, etc., are obtained by an extensive experiment to EDAhybrid. For problem with 4 until 15 facilities, the population size needed is *4n*. The generation needed for 4 up to 15 facilities are *10n*. For large number of facilities, the population and generation needed are varied. Problem with 20 facilities needs *5n* populations and 350 generations, problem with 25 facilities requires *5n* populations and 550 generations, while problem with 30 facilities needs *6n* populations and 800 generations.

For EDA part parameters, the *selection size* also depends on the number of population, which is a half of population size (Ceberio, Irurozki, Mendiburu, & Lozano, 2012). The elitism rate is assumed like the common setting (Chen et al., 2012) that is 10% of the generation. To determine the PSO part parameters, this research uses Eq. (7) (Haupt & Haupt, 2004) to adjust inertia weight.

$$w = (maxgen - currentgen)/maxgen \qquad (7)$$

Tabu Search here is used as the local search, so that the algorithm is set to become faster with *n* generations, swaplist size equal to $2(n-1)$ and tabu tenure is $\lceil n/5 \rceil$.

Many literatures use acceleration constants (c_1 and c_2) equal to 2.05. Based on Bratton and Kennedy (2007) and Navalertporn and Afzulpurkar (2011), the standard of acceleration constants are also 2.05. So, this research uses acceleration constants equal to 2.05.

Table 2
Computational results of all algorithms.

Problem	n	OFV	OFV (other papers)	EDA		GA		PSO		eACGA		EDAhybrid	
		Samarghandi et al. (2010)		Min	Error (%)	Min	Error (%)	Min	Error (%)	Min	Error (%)	Min	Error (%)
P4	4	**638**	(Amaral, 2006) **638.0**	**638**	0.00	**638**	0.00	**638**	0.00	**638**	0.00	**638**	**0.00**
LW5	5	**151**	(Datta et al., 2011) **151.0**	**151**	0.30	**151**	0.00	**151**	0.23	**151**	0.00	**151**	**0.00**
N6	6	**1.99**	(Solimanpur et al., 2005) **1.990**	**1.99**	1.61	**1.99**	0.00	**1.99**	2.76	**1.99**	0.00	**1.99**	**0.00**
S8	8	**801**	(Amaral, 2006) **801.0**	**801**	0.58	**801**	1.20	**801**	2.25	**801**	0.14	**801**	**0.04**
S8H	8	**2324.5**	(Datta et al., 2011) **2324.5**	**2324.5**	0.13	**2324.5**	0.47	**2324.5**	0.74	**2324.5**	0.08	**2324.5**	**0.07**
S9	9	**2469.5**	(Amaral, 2006) **2469.5**	**2469.5**	0.48	**2469.5**	2.46	**2469.5**	5.74	**2469.5**	0.33	**2469.5**	**0.06**
S9H	9	**4695.5**	(Amaral, 2006) **4695.5**	**4695.5**	0.06	**4695.5**	0.34	**4695.5**	0.55	**4695.5**	0.06	**4695.5**	**0.05**
S10	10	**2781.5**	(Datta et al., 2011) **2781.5**	**2781.5**	1.68	**2781.5**	4.72	**2781.5**	5.24	**2781.5**	1.61	**2781.5**	**0.74**
S11	11	**6933.5**	(Datta et al., 2011) **6933.5**	**6933.5**	1.81	6953.5	4.82	**6933.5**	4.04	**6933.5**	2.54	**6933.5**	**1.20**
LW11	11	**6933.5**	(Datta et al., 2011) **6933.5**	6943.5	2.08	6985.5	4.25	6973.5	3.91	**6933.5**	2.09	**6933.5**	**0.96**
N12	12	**23.365**	(Solimanpur et al., 2005) **23.365**	23.645	2.53	23.655	5.17	23.575	7.50	23.58	4.38	**23.365**	**1.11**
P15	15	**6305**	(Amaral, 2006) **6305.0**	6379	4.52	6610	9.71	6576	10.21	6432	5.99	**6305**	**0.81**
P20	20	**15549**	(Datta et al., 2011) **15549.0**	16583	7.85	16,255	7.85	16,560	12.48	16,314	9.09	**15,549**	**1.80**
P25	25	4619	(Anjos et al., 2008) **4618.0**	4837	9.93	4705	5.90	4899	11.47	4865	7.59	4618	**2.04**
P30	30	**44,965**	(Datta et al., 2011) **44965.0**	49,237	12.73	46,319	4.92	48,643	11.90	46,369	6.69	**44,965**	**2.13**
			Mean error		3.09		3.45		5.27		2.71		**0.73**

n: number of facilities.
OFV: Objective Function Value.
Min: minimum solution found in 20 runs.
Error (%): the mean value in 20 runs compared to the lowest Min found among all algorithms.

Table 3
Standard deviation comparison among all algorithms.

Problem	EDA	GA	PSO	eACGA	EDAhybrid
P4	**0**	**0**	**0**	**0**	**0**
LW5	1.61	**0**	**0**	**0**	**0**
N6	0.08	**0**	0.11	**0**	**0**
S8	12.89	6.4	18.23	2.36	**1.57**
S8H	6.98	10.64	11.09	5.75	**4.92**
S9	27.88	36.35	137.67	14.12	**6.48**
S9H	5.23	13.3	20.97	5.41	**4.42**
S10	54.04	71.55	101.34	54.69	**53.51**
S11	120.28	153.3	132.22	133.56	**113.06**
LW11	137.13	127.31	157.09	115.08	**82.06**
N12	0.25	0.46	0.73	0.46	**0.27**
P15	193.87	255.31	227.45	140.35	**115.71**
P20	235.97	250.63	556.95	340.01	**226.34**
P25	87.14	85.96	124.57	83.23	**79.59**
P30	**637.91**	695.65	1217.5	1019.4	717.66

The bold values indicate the minimum values found among all algorithms.

Table 4
Mean running time (*seconds*) comparison among all algorithms.

Algorithm	Mean runtime (s)
EDA	25.52
GA	12.594
PSO	**2.598**
eACGA	18.56
EDAhybrid	40.082

The bold values indicate the minimum values found among all algorithms.

5.2. EDAhybrid for SRFLP

This section provides the performance comparison between EDAhybrid and several algorithms in solving SRFLP. We also compare the EDAhybrid's minimum achieved value with Objective Function Value (OFV) of recent researches. The performance of EDAhybrid is tested by using 15 SRFLP benchmark problems given in the literatures. The inputs given in benchmark dataset are a matrix containing flow between facilities, length of each facilities, and clearance space between facilities. Problems coded P4, LW5, S8, S8H, S9, S9H, S10, S11, LW11, and P15 are from Amaral (2006), P6 and P12 are found in Neghabat (1974), P20 and P30 are

provided in Heragu and Kusiak (1991) and P25 is listed in Anjos and Vannelli (2008).

The experiments for testing the effectiveness of EDAhybrid are held by comparing EDAhybrid's performance to EDA, GA, PSO, and eACGA algorithms. Each algorithm is run for 20 times, and the minimum value, standard deviation, error rate (mean value compared to minimum value) and running time are recorded from the results. We use Matlab function std(X) to compute standard deviation. The formula used is presented in Eq. (8), $s = \text{std}(X)$ returns the standard deviation using Eq. (8). The result s is the square root of an unbiased estimator of the variance of the population from which X is drawn.

$$s = \left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{\frac{1}{2}} \tag{8}$$

Table 2 shows the computational experiment results of all algorithms. The records with bold numbers in Table 2 imply the minimum values found among all algorithms. The optimum solutions found so far based on recent researches are listed in the 3rd and 4th columns. In terms of best solution, EDA and PSO can achieve optimum solution from P4 until S11, while GA gets optimum solution until S10. On the other hand, eACGA can get the optimum solution from P4 until LW11. Our proposed algorithm EDAhybrid always achieves optimum solution in all benchmark problems. This achievement results in a 0% gap if compared to OFV of previous papers.

Table 2 also provides the error rates examination of all algorithms. The error rates percentages, as shown in Eq. (9), are obtained by comparing the mean value in 20 runs and the lowest minimum value found in all algorithms. EDAhybrid gains minimum error rates of all problems and reaches the lowest mean error rate, which is 0.732%, compared to other algorithms.

$$error(\%) = \frac{\text{mean} - \text{min}_{\text{all}}}{\text{min}_{\text{all}}} \times 100\% \tag{9}$$

Standard deviation comparison among all algorithms is described in Table 3. Among all algorithms, the results show that EDAhybrid has the lowest standard deviation from P4 up to P25. EDA gets lowest standard deviation in P30, but EDA cannot achieve optimum solution in this problem as shown in Table 2.

Table 5
The minimum value and error rates of all algorithms when solving enhanced SRFLP.

n	EDA		GA		PSO		eACGA		EDAhybrid	
	Min	Error (%)	Min	Error (%)	Min	Error (%)	Min	Error (%)	Min	Error (%)
5	**410.39**	1.82	**410.39**	0.73	**410.39**	1.70	**410.39**	0.55	**410.39**	**0.37**
11	9404.10	1.27	9718.20	6.14	9480.50	4.21	9404.10	1.57	**9398.10**	**0.61**
20	62368.00	4.56	61773.00	5.56	60882.00	7.47	61694.00	2.33	**60335.00**	**0.62**
Mean error		1.54		3.43		2.95		1.06		**0.53**

The bold values indicate the minimum values found among all algorithms.

Table 6
The standard deviation and running time (*seconds*) of all algorithms when solving enhanced SRFLP.

n	EDA		GA		PSO		eACGA		EDAhybrid	
	Stdev	Runtime	Stdev	Runtime	Stdev	Runtime	Stdev	Runtime	Stdev	Runtime
5	9.01	0.12	6.32	0.12	15.37	**0.02**	5.50	0.12	**4.62**	0.11
11	120.49	1.64	154.79	1.25	233.84	**0.15**	131.17	1.44	**98.04**	1.66
20	625.10	29.79	1379.40	17.38	1522.20	**2.76**	898.23	24.06	**157.45**	37.86

The bold values indicate the minimum values found among all algorithms.

The program code was run with Matlab, in Intel Core 2 Duo CPU and 2 GB of RAM. Table 4 provides the running time comparison of all algorithms. In terms of running time, PSO gets the fastest running time among all algorithms. EDAhybrid running time is slower than others because of the complexity of this algorithm.

5.3. EDAhybrid for enhanced SRFLP

After applying EDAhybrid to benchmark problems, the effectiveness of EDAhybrid in enhanced SRFLP is tested. Three new datasets with different number of facilities that are 5, 11 and 20 are generated. There are five kind of inputs, three of them are similar to basic SRFLP, the rest are a matrix containing installation cost and two facilities that cannot be adjacent to each other for safety reason. The first input is *length* derived from benchmark problem. *Clearance space* is assigned randomly and becomes the second input. *Flow between facilities* are generated randomly with Uniform distribution U(0,20). Similarly, *installation cost* are generated randomly with Uniform distribution U(0,10). The last input is *safety reason* which is assigned randomly.

All algorithms from the previous section are tested in solving enhanced SRFLP. We also use similar parameters setting, describe in Section 5.1. Each algorithm is run for 20 times, and the minimum, error rate (mean compared to minimum), standard deviation, and running time (in seconds) are extracted from the result.

Tables 5 and 6 show the results of all algorithms when solving enhanced SRFLP with 5, 11 and 20 facilities. The minimum OFV for those problems are attained by EDAhybrid. Compared to other algorithms in Table 5, EDAhybrid still acquires the lowest mean. EDAhybrid also always gets the lowest standard deviation compared to others in Table 6. Table 5 also provides the mean error rate for every algorithm in the bottom part. EDAhybrid gets the minimum percentage error in every case, so that it also achieves the lowest mean error rate compared to other algorithms. Hence, we can conclude that EDAhybrid can also perform very well in enhanced SRFLP.

6. Discussion and conclusion

This paper has proposed a new metaheuristic algorithm named Hybrid Estimation of Distribution Algorithm (EDAhybrid), which consists of hybridization of EDA, PSO and Tabu Search. To maintain the diversity of EDA, hybridization with meta-heuristic algorithm is needed, and this research chooses PSO as the meta-heuristic algorithm. EDAhybrid runs EDA and PSO alternately every two

generations, and then TS as a local search is added at the end of every iteration.

Based on computational results of 15 benchmark problems, the performance of EDAhybrid always achieves optimum solution in basic SRFLP. Compared to eACGA, EDA, PSO, and GA, the error rates of EDAhybrid always get the lowest value. EDAhybrid also mostly provides the lowest standard deviation than others. Compared with the OFV of recent researches in SRFLP, EDAhybrid always gets equal performance in achieving minimum objective function.

Instead of using one probabilistic model like in EDA, EDAhybrid uses two. EDA uses a univariate probabilistic model, while EDAhybrid applies univariate and bivariate probabilistic models. The use of variable interactions in bivariate probabilistic models could represent better individual information for the EDA part in EDAhybrid (Chen et al., 2012). It allows EDAhybrid to perform better than EDA. Artificial particles generated from global statistical information characterize the distribution of promising solutions in the search space. Therefore, the combination of artificial particles with PSO operators could improve the solution quality. This also makes EDAhybrid performs better than standard PSO and GA. EDAhybrid is able to perform better than eACGA because it adds Tabu Search as local search.

In conclusion, EDAhybrid is superior to other algorithms because it has EDA which runs alternately with PSO to maintain the diversity of EDA. Moreover, EDAhybrid adds Tabu Search as local search, so that it might improve the *global best* solution in every iteration.

Enhanced SRFLP problem is designed to make basic SRFLP closer to real case by adding more constraints. Compared to other algorithms, EDAhybrid can get the lowest minimum and error rates in enhanced SRFLP. We can conclude from this research that EDAhybrid can be used as a method to deal with basic and enhanced SRFLP and get high quality solution.

References

Amaral, A. R. S. (2006). On the exasct solution of a facility layout problem. *European Journal of Operational Research, 173*(2), 508–518. http://dx.doi.org/10.1016/j.ejor.2004.12.021.

Amaral, A. R. S. (2009). A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics, 157*(1), 183–190. http://dx.doi.org/10.1016/j.dam.2008.06.002.

Anjos, M. F., & Vannelli, A. (2008). Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS Journal on Computing, 20*(4), 611–617. http://dx.doi.org/10.1287/ijoc.1080.0270.

Bratton, D., & Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Proceeding of the IEEE Swarm Intelligence Symposium, SIS'2007* (pp. 120–127). Honolulu, Hawaii, New Jersey.

Ceberio, J., Irurozki, E., Mendiburu, A., & Lozano, J. A. (2012). A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence, 1*(1), 103–117. http://dx.doi.org/10.1007/s13748-011-0005-3.

Chang, P.-C., Hsieh, J.-C., Chen, S.-H., Lin, J.-L., & Huang, W.-H. (2009). Artificial chromosomes embedded in genetic algorithm for a chip resistor scheduling problem in minimizing the makespan. *Expert Systems with Applications, 36*(3), 7135–7141. http://dx.doi.org/10.1016/j.eswa.2008.08.047.

Chen, Y.-M., Chen, M.-C., Chang, P.-C., & Chen, S.-H. (2012). Extended artificial chromosomes genetic algorithm for permutation flowshop scheduling problems. *Computers and Industrial Engineering, 62*(2), 536–545. http://dx.doi.org/10.1016/j.cie.2011.11.002.

Datta, D., Amaral, A. R. S., & Figueira, J. R. (2011). Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research, 213*(2), 388–394. http://dx.doi.org/10.1016/j.ejor.2011.03.034.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control, 31*(2), 255–267. http://dx.doi.org/10.1016/j.arcontrol.2007.04.001.

Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms* (2nd ed.). Wiley Interscience.

Hauschild, M., & Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation, 1*(3), 111–128. http://dx.doi.org/10.1016/j.swevo.2011.08.003.

Heragu, S. S. (2008). *Facilities design* (3rd ed.). CRC Press, Taylor & Francis Group.

Heragu, S. S., & Alfa, A. S. (1992). Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research, 57*(2), 190–202. http://dx.doi.org/10.1016/0377-2217(92)90042-8.

Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal of Operational Research, 53*, 1–13.

Hu, X., Eberhart, R., & Shi, Y. (2003). Swarm intelligence for permutation optimization, a case study on N-queens problem.pdf. In *IEEE swarm intelligence symposium* (pp. 243–246). Indianapolis, USA.

Lozano, J. A. , Larranaga, P., Inza, I. and Bengoetxea, E. Edit (2006). Towards a new evolutionary computation: Advances in the Estimation of Distribution Algorithms. Springer.

Mawdesley, M. J., & Al-Jibouri, S. H. (2003). Proposed genetic algorithms for construction site layout. *Engineering Applications of Artificial Intelligence, 16*(5–6), 501–509.

Navalertporn, T., & Afzulpurkar, N. V. (2011). Optimization of tile manufacturing process using particle swarm optimization. *Swarm and Evolutionary Computation, 1*(2), 97–109. http://dx.doi.org/10.1016/j.swevo.2011.05.003.

Neghabat, F. (1974). An efficient equipment layout algorithm. *Operations Research, 22*, 622–628.

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications, 21*, 5–20.

Samarghandi, H., & Eshghi, K. (2010). An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research, 205*(1), 98–105. http://dx.doi.org/10.1016/j.ejor.2009.11.034.

Samarghandi, H., Taabayan, P., & Jahantigh, F. F. (2010). A particle swarm optimization for the single row facility layout problem. *Computers and Industrial Engineering, 58*(4), 529–534. http://dx.doi.org/10.1016/j.cie.2009.11.015.

Santana, R., Larrañaga, P., & Lozano, J. (2008). Combining variable neighborhood search and Estimation of Distribution Algorithms in the protein side chain placement problem. *Journal of Heuristics, 14*, 519–547.

Solimanpur, M., Vrat, P., & Shankar, R. (2005). An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers and Operations Research, 32*(3), 583–598. http://dx.doi.org/10.1016/j.cor.2003.08.005.

Sule, R. D. (2009). *Manufacturing facilities location, planning, and design* (3rd ed.). CRC Press, Taylor & Francis Group.

Zhang, Q., & Muhlenbein, H. (2004). On the convergence of a class of Estimation of Distribution Algorithms. *IEEE Transactions on Evolutionary Computation, 8*(2), 127–136.