A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem

Mahmoud M. El-Sherbiny a,*, Rashid M. Alhamali b

a Operations Research Dept., Institute of Statistical Studies and Research (ISSR), Cairo University, Egypt
bDept. of Quantitative Analysis, College of Business Administration (CBA), King Saud University, Saudi Arabia

ARTICLE INFO

Article history:
Received 8 August 2011
Received in revised form 14 May 2012
Accepted 2 December 2012
Available online 20 December 2012

Keywords:
Fixed charge transportation
Convergence
Particle swarm
Genetic algorithm
Artificial immune system

ABSTRACT

Fixed Charge Transportation Problem (FCTP) is an NP-hard problem with many applications in both traditional and modern industrial situations. This paper introduces a Hybrid Particle Swarm algorithm with artificial Immune Learning (HPSIL) for solving fixed FCTPs. The HPSIL algorithm is an efficient and expensive computation. Generally, the methods of solving FCTP are classified as either exact or heuristic. Exact methods for solving FCTP include the cutting plane method (Rousseau, 1973), the vertex ranking method (McKeeown, 1975), and the branch-and-bound method (Palekar, Karwan, & Zionts, 1990), amongst others. However, exact methods are not very useful when a problem reaches a certain level, because they do not make the most use of the special network structure of the FCTP. Therefore, heuristic methods have been proposed, such as the adjacent extreme point search method (Balinski, 1961; Sun et al., 1993), and the Lagrangian relaxation method (Wright et al., 1989, 1991). Although these methods are usually computationally efficient, the major disadvantage of heuristic methods is the possibility of terminating at a local optimum that is far distant from the global optimum. Recently, some meta-heuristic methods have been employed in solving FCTP, such as the Tabu search method for solving FCTPs (Sun, Aronson, & Mckeown, 1998), the hybrid Genetic Algorithm (GA) based on a spanning tree with Prüfer number (Gen, Ida, & Li, 1998), and GAs based on a matrix permutation representation (Gottlieb, Julstrom, Rothlauf, & Raidl, 2001; Raidl & Julstrom, 2003) which have improved the effective coding of the spanning tree method based on edge sets. The GA creates a sorted set of edges to encode the spanning tree, which is more efficient compared to evolution strategies (ES) at a certain level (Su & Zhan, 2006). Moreover, to improve solution quality a lot of evolutionary algorithms use a random procedure to generate a Prüfer number with sum of problem dimensions (the number of suppliers and customers) less two digits in range of unity and the sum of problem dimensions. The generated Prüfer number, may not be translated to a feasible spanning tree to represent the solution of the FCTP. In order to overcome this problem, Gen & Cheng (2000) developed a criterion for checking the feasibility of the Prüfer number. Jo, Li, and Gen (2007) discovered that this technique fails to generate the feasible solution when the difference between the number of suppliers and the number of customers is very large and developed another feasibility criterion to check the feasibility of the Prüfer number and then used a repairing procedure for making it feasible. Jo et al. (2007) applied the spanning tree-based genetic algorithm which developed based on the solution structure
for the linear transportation problem to solve the nonlinear FCTP. Comments based on the calculations of this work were presented in Kannan et al. (2008). Actually, Kannan corrected (recalculated) the costs of the two examples presented in Jo et al. (2007). Since the repairing procedure may take long time to repair, Hajighaee, Keshteli, Molla-Alizadeh-Zavardehi, and Tavakkoli-Moghaddam (2010) addressed a nonlinear FCTP using a spanning tree based genetic algorithm and proposed a method to generate Prüfer number at random which does not need a repairing procedure. Othman, Delavar, Behnam, and LessaniBahrani (2011) used the fuzzy logic controllers to adopt the crossover parameters of the GA for solving FCTP and reached the local optimum remarkably faster. Molla-Alizadeh-Zavardehi et al. (2011) proposed an artificial immune algorithm (AIA) and a GA based on the spanning tree and Prüfer number representation for solving capacitated fixed-charge transportation problem in a two-stage supply chain network and selecting the suitable values of the algorithms’ parameters that gives the best performances of such algorithms. In addition, they investigated the impact of increasing the problem size on the performance. Based on this work comments on the mathematical model, the transportation graph and the total cost for the example are presented by El-Sherbiny (2012). Nevertheless, the quality of solutions attained largely depends on the randomness. On the other hand, the Particle Swarm Optimization (PSO) technique developed by Eberhart and Kennedy (1995) is a simple evolutionary algorithm that differs from other evolutionary computation techniques in that it is motivated by the simulation of social behavior. PSO has features of both GAs and ESs (Tandon, El-Mounayri, & Kishawy, 2002) and exhibits good performance in finding solutions of static optimization problems (Paropoulos and Vrahatis, 2001). Several modifications to the original PSO were developed to improve the solution quality, such as a combined PSO based on the previous global best and the current global best positions (El-Sherbiny, 2007), a modified algorithm for PSO with a constriction coefficient (El-Sherbiny, 2009), and a particle swarm inspired optimization algorithm without the velocity equation (El-Sherbiny, 2011).

To improve the quality of the solution, this paper introduces a Hybrid Particle Swarm algorithm with Artificial Immune Learning (HPSIL) for solving the FCTPs and studies the effect of various factors on the performance thereof. In addition, the results of experimentally comparing its performance with that of the genetic algorithms proposed in (Hajighaee et al., 2010; Othman et al., 2011) as well as the solution found by LINGO are presented.

The rest of the paper is organized as follows. In Section 2, the FCTP model is described, while in Section 3 some basic concepts of PSO and Artificial Immune System (AIS) are presented. The proposed HPSIL algorithm is introduced in Section 4. The HPSIL performance study based on the experimental design and analysis are presented in Section 5. In Section 6, numerical experiments with the HPSIL algorithm are discussed. Finally, the conclusion and future work are reported in Section 7.

2. FCTP model

FCTP can be described as a distribution problem, with \( m \) suppliers (warehouses, factories or plants) and \( n \) customers (destinations or demand points). Each of the \( m \) suppliers can ship to any of the \( n \) customers at a unit shipping cost \( c_{ij} \) (unit cost to ship from supplier \( i \) to customer \( j \)) plus a fixed cost \( f_{ij} \) assumed for opening the route. Each supplier \( i = 1, 2, \ldots, m \) has \( s_i \) units of supply and each customer \( j = 1, 2, \ldots, n \) demands \( d_j \) units. The objective is to determine which routes should be opened and the size of the shipment, so that the total cost of satisfying the demand, given the supply constraints, is minimized. The standard mathematical model of the FCTP can be expressed as follows:

\[
\begin{align*}
\text{Min } z &= \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}x_{ij} + f_{ij}y_{ij}) \\
\text{s.t. } &\sum_{i=1}^{m} x_{ij} \geq d_j \text{ for } j = 1, \ldots, n \\
&\sum_{j=1}^{n} x_{ij} \geq s_i \text{ for } i = 1, \ldots, m \\
x_{ij} &\geq 0, \text{ } \forall i, j \\
y_{ij} &\geq 0, \text{ } \forall i, j
\end{align*}
\]

where \( x_{ij} \) is the unknown quantity to be transported through the route \((i,j)\), that is, from supplier \( i \) to customer \( j \).

3. Basic concepts of PSO and AIS

PSO has some advantages over other similar optimization techniques. It is not largely affected by the problem size, can converge to the optimal solution in many problems where most analytical methods fail to converge (Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008). PSO is more efficient in maintaining the diversity of the swarm (Engelbrecht, 2006), since all the particles use the information related to the most successful particle in order to improve themselves and comparatively faster than some other optimization algorithm. Therefore, it can be effectively applied to different optimization problems (Deb, Basanta, & Sarkar, 2011).

On the other hand, AIS is self-organizing, learning capability, and not many system parameters are required, hence AIS offers powerful and robust information processing capabilities for solving complex optimization problems. The proposed algorithm tries to use the advantages of both PSO and AIS for solving FCTP. The following subsections present the basic concepts related to both PSO and AIS.

3.1. Particle swarm optimization

PSO is a population-based stochastic optimization technique, developed by Eberhart and Kennedy (1995). PSO simulates the social behavior of organisms, such as birds in a flock or fish in a school. This behavior can be described as an automatically and interactively updating system. In PSO, each single candidate solution can be considered to be a particle in the search space. Each particle makes use of its own memory as well as knowledge gained by the swarm as a whole to find the best solution. All the particles have fitness values which are evaluated by the objective function of the FCTP. During movement, each particle adjusts its position by changing its velocity according to its own experience and that of a neighboring particle, thus making use of the best position encountered by itself and its neighbor. Particles move through the problem space by following a current of optimum particles. The process is then iterated a fixed number of times or until a predetermined minimum error is achieved (Kennedy, 2003).

Suppose the search space is \( L \)-dimensional and \( K \) particles form the colony. The \( k \)th particle represents a \( L \)-dimensional vector \( \bar{p}_k \) where \( k = 1, 2, \ldots, K \). This means that the \( k \)th particle is located at \( \bar{p}_k = (p_{k1}, p_{k2}, \ldots, p_{KL}) \in S \) of the search space (solution space). The position of each particle is a potential result (potential solution) of the problem under study. We could calculate the particle’s fitness by putting its position into a designated objective function. The \( k \)th particle’s “flying” velocity at the \( t \)th iteration is denoted as \( \bar{v}_k(t) \), the local best position of the \( k \)th particle is denoted as \( \bar{g}_k(t) \) and the global best position of the swarm is denoted as \( \bar{g}(t) \). Each
particle updates its position using (3) and (4) where \( b_1 \) and \( b_2 \) are positive constants, called the acceleration constants, and \( r_1, r_2 \in [0, 1] \) are uniform random numbers.

\[
\begin{align*}
\bar{\nu}_k(t+1) &= \bar{\nu}_k(t) + b_1 r_1 (\bar{h}_k(t) - \bar{\nu}_k(t)) + b_2 r_2 (\bar{g}_k(t) - \bar{\nu}_k(t)) \quad (3) \\
\bar{\nu}_i(t+1) &= \bar{\nu}_i(t) + \bar{v}_i(t + 1) \quad (4)
\end{align*}
\]

The termination criterion for the iterations is determined by whether the max generation or a designated value of the particle fitness is reached. Compared to \( \bar{\nu}_k(t) \) and \( \bar{\nu}_i(t + 1) \) is the summation of two components \( b_1 r_1 (\bar{h}_k(t) - \bar{\nu}_k(t)) \) and \( b_2 r_2 (\bar{g}_k(t) - \bar{\nu}_k(t)) \) as shown in (3).

### 3.2. Artificial immune system

A human immune system is made up of numerous \( B \) and \( T \) cells, which are constantly being produced in the bone marrow and thymus, respectively. The level of \( B \) cell simulation depends not only on the success of the match to the antigen, but also on how well it matches with other \( B \) cells in the immune system. If the stimulation of \( B \) cells reaches a certain threshold, the \( B \) cell is transformed into a blast that begins to differentiate rapidly, producing clones that turn on a mutation mechanism that generates mutations in the gene coding for the antibody molecule, which is called somatic hypermutation. However, if the stimulation level falls below the threshold, the \( B \) cell does not replicate and ultimately dies.

Various special AISs have been developed to solve complex optimization problems. One of these is aiNet (De Castro & Timmis, 2002, 2002b), which was inspired by biological immune systems. Opt-aiNet (Timmis, Knight, Catro, & Hart, 2004), an application of aiNet for function optimization, considers the optimized objective function as an antigen and the candidate solutions as antibodies. The candidate antibodies evolve according to the matching degree of fitness between the antibodies and antigen. The better the match between them, the smaller is the mutation degree of the candidate antibody, and vice versa.

The opt-aiNet algorithm consists of seven components: initialization, affinity value, clone, mutation, selection, cell interaction, and recruitment. The mutation operation of candidate antibodies is executed by the following equation:

\[
\text{newAB} = \text{oldAB} + \frac{1}{\beta} e^{-\text{affinity}} \quad (5)
\]

where \( \text{oldAB} \) is the previous candidate antibody before mutation, \( \text{newAB} \) is the new antibody obtained after mutation, \( \text{fitness} \) is the objective function value related to \( \text{oldAB} \), \( \beta \) is a control parameter for mutation, and \( r \in [0, 1] \) is a uniform random number. Affinity is measured based on the Euclidean distance between two antibodies or between an antibody and an antigen (De Castro and Von Zuben, 2001). The affinity measure results from an adapted distance function, such that the affinity is maximum when the distance is minimum. It is clear that the larger the fitness value of the antibody, the smaller is the level of mutation.

### 4. HPSIL algorithm

While most of the previous works are based on using the Prüfer numbers with a spanning tree, the proposed algorithm uses a new approach considering the basic properties of the particle swarm optimization and the classical methods for solving transportation problems. In the HPSIL, a flexible chromosome structure with length \((m + n)\) is used (see Fig. 1), together with a decoding and an allocation procedures as illustrated in Figs. 3 and 4, respectively. Usage of such a chromosome structure allows more than one gene to have the same value. Thus, particle swarm optimization with integer solutions can be used to solve the FCTP. In this section, a hybrid particle swarm algorithm with artificial immune learning is proposed to solve the FCTP. The new coding scheme and the coding procedure are explained in Section 4.1. The allocation procedure to find a corresponding feasible solution for each generated particle is described in Section 4.2. In Section 4.3, six versions of the mutation equations for candidate antibodies used in hybrid immune networks with swarm learning to generate particles are presented. The general steps of the HPSIL algorithm, as illustrated in Fig. 2, are explained below:

Step 1: Set \( t = 0 \) and Generate an initial population of \( \bar{\nu}_k(t) \) particles. Each particle in \( \bar{\nu}_k(t) \) has an order structure of \((m + n)\) dimensional vector of integers. This order structure is used to represent the solution space (particles) as represented in Fig. 1.

Step 2: Apply the decoding procedure, as explained in Section 4.1, to find the suppliers sequence \( S \) and the customers sequence \( C \) based on population \( \bar{\nu}_k(t) \) resulting from step 1.

Step 3: Apply the allocation procedure, as explained in Section 4.2, to find the corresponding transportation allocation based on the suppliers sequence \( S \) and the customers sequence.
resulted from Step 2. The results of this procedure are the transported quantities from each supplier \(S\) to each customer \(D\).

**Step 4:** Calculate the value of the objective function (fitness value) of each particle in population \(\tilde{p}_k(t)\).

**Step 5:** If the termination criterion is met, then the best solution is reached for the FCTP. Otherwise, continue.

**Step 6:** Update the population positions \(\tilde{p}_k(t + 1)\) by executing one of the selected mutation equations and go to Step 2.

### 4.1. The main schema

One of the most important issues when designing a PSO algorithm is concerning its solution (particle) representation. To construct a direct relationship between the problem domain and the HPSIL, the proposed coding scheme consists of a set of integer numbers in the interval \([1, q]\), with the length of the scheme equal to \(m + n\), where \(q = \max(m, n)\), \(m\) is the number of suppliers, and \(n\) is the number of customers. Therefore, the length of each particle in the \(\tilde{p}_k(t)\) is equal to the sum of the problem dimensions. i.e. the particle \(k\) is donated as \(\tilde{p}_k = [p_{k1}, p_{k2}, \ldots, p_{kL}]\) where \(L = m + n\) and the value of each genome \(p_{kl}\), \(l = 1, 2, \ldots, L\). Fig. 1 depicts a sample particle which is used to code a \(4 \times 5\) FCTP or any FCTP with \(m + n = 9\). As shown in Fig. 1, the value of each genome \(p_{kl}\) is between 1 and \(q = 5\). In fact, each particle shows a group of numbers between 1 and \(q\) and each genome \(p_{kl}\) points to the index of an array converted to a supplier or customer number. Moreover, it can be noted that any number can be repeated and this repetition makes the particle structure more flexible in applying any mutation equation. Thus based on the flexibility of the particle structure, combined with the decoding procedure, any meta-heuristic techniques such as PSO, GA, Tabu search, and others can be applied.

In this procedure, \(\tilde{p}_k\) is an input particle (chromosome) that must be decoded to show two lists \(S\) and \(D\), where the list \(S\) and \(D\) represent the suppliers sequence and the customers sequence respectively. As an example, Fig. 3 illustrates the results.

---

**Input:**

\[
\tilde{p}_k(t) = \begin{bmatrix} 2 & 5 & 3 & 2 & 3 & 3 & 3 & 5 & 4 \end{bmatrix}_{j=1}
\]

**Processing:**

By applying the second “for loop” \(n=4\) times we will get the following:

| QS | \(j = \text{Mod}(p_{k1}, |QS|) + 1\) | QS(j) | S |
|----|---------------------------------|-------|---|
| at \(i=1\) | \[1, 2, 3, 4\] | 3 = 2 Mod 4 + 1 | 3 |
| at \(i=2\) | \[1, 2, 4\] | 3 = 5 Mod 3 + 1 | 4 |
| at \(i=3\) | \[1, 2\] | 2 = Mod 2 + 1 | 2 |
| at \(i=4\) | \[1\] | 1 = 2 Mod 1 + 1 | 1 |

By applying the first “for loop” \(m=5\) times we will get the following:

| QD | \(j = \text{Mod}(p_{k1}, |QD|) + 1\) | QD(j) | D |
|----|---------------------------------|-------|---|
| at \(i=5\) | \[1, 2, 3, 4, 5\] | 4 = 3 Mod 5 + 1 | 4 |
| at \(i=6\) | \[1, 2, 3, 5\] | 4 = 3 Mod 4 + 1 | 5 |
| at \(i=7\) | \[1, 2, 3\] | 1 = 3 Mod 3 + 1 | 1 |
| at \(i=8\) | \[2, 3\] | 2 = 5 Mod 2 + 1 | 3 |
| at \(i=9\) | \[2\] | 1 = 4 Mod 1 + 1 | 2 |

**Output:**

The final result of applying the decoding algorithm on the particle \(\tilde{p}_k(t)\) is

\[
\begin{align*}
S & = [3, 4, 2, 1] \quad \text{The suppliers sequence} \\
D & = [4, 5, 1, 3, 2] \quad \text{The customers sequence}
\end{align*}
\]

Fig. 3. An illustrative example of applying the decoding procedure.

\(D\) resulted from Step 2. The results of this procedure are the transported quantities from each supplier \(S\) to each customer \(D\). Step 4: Calculate the value of the objective function (fitness value) of each particle in population \(\tilde{p}_k(t)\). Step 5: If the termination criterion is met, then the best solution is reached for the FCTP. Otherwise, continue. Step 6: Update the population positions \(\tilde{p}_k(t + 1)\) by executing one of the selected mutation equations and go to Step 2.
of applying the decoding procedure to the particle presented in Fig. 1. The steps for the decoding procedure are presented below:

**Input:** the particle $\bar{p}_k$.

**Output:** the list $S$ and the list $D$.

**Step 1:** Create a collection list $QS$ that includes the number of suppliers $\{1, 2, \ldots, m\}$, and a collection list $QD$ that includes the number of customers $\{1, 2, \ldots, n\}$.

**Step 2:** Set $i = 1$.

**Step 3:** Set $j = \text{Mod}(p_{ki}, |QS|) + 1$.

**Step 4:** Add the item $QS(j)$ to the list $S$.

**Step 5:** Remove the item $j$ from the collection list $QS$.

**Step 6:** Increment $i$ with 1.

**Step 7:** Repeat the steps from 3 to 6 until $i = n + 1$.

**Step 8:** Set $j = \text{Mod}(p_{ki}, |QD|) + 1$.

**Step 9:** Add the item $QD(j)$ to the list $D$.

**Step 10:** Remove the item $j$ from the collection list $QD$.

**Step 11:** Increment $i$ with 1.

**Step 12:** Repeat the steps from 8 to 12 until $i = m + n + 1$.

where $|QS|$ and $|QD|$ are equal to the length of $QS$ and $QD$ respectively.

### 4.2. Allocation procedure

This procedure allocates the transported units based on the list $S$ and the list $D$ resulted from the decoding procedure. In other words, this procedure finds a feasible solution for FCTP based on the outputs of the decoding algorithm. The inputs of the allocation procedure are the two lists $S$ and $D$ (the output of the decoding procedure). Based on the two lists ($S$ and $D$) the allocation procedure allocates $X_{ij}$ (feasible solution) units for FCTP. The steps for the allocation procedure are presented below:

**Inputs:** the lists $S$ and $D$.

**Output:** the feasible solution $X_{ij}$.

**Step 1:** Set $k$ equal to 1.

**Step 2:** Set $i = S(1)$ and $j = D(1)$.

**Step 3:** If $s_i = d_j$ then {set $x_{ij} = s_i$, remove $S(1)$, and remove $D(1)$}

If $s_i > d_j$ then {set $x_{ij} = d_j$, set $s_i = s_i - d_j$, and remove $D(1)$}.

If $s_i < d_j$ then {set $x_{ij} = s_i$, set $d_j = d_j - s_i$, and remove $S(1)$}.

**Step 4:** Set $\text{Sol}(k, 1)$ equal to $i$, $\text{Sol}(k, 2) = j$, and $\text{Sol}(k, 3) = x_{ij}$ where $\text{Sol}$ is the solution array with $(n + m - 1, 3)$ dimensions.

**Step 5:** Update $k = k + 1$.

**Step 6:** Repeat from step 2 to step 5 until $|S| = 0$ or $|D| = 0$.

**Step 7:** Return the $\text{Sol}$ array.
Fig. 4 represents an illustrative example of applying this algorithm. This procedure guarantees the validity of both constraints (1) and (2) in the mathematical model and guarantees the feasibility of all the generated solutions. Step 6 of the procedures ensures that either the procedure continues until the length of supplier’s order $|S|$ or the length of customer’s order $|D|$ becomes zero, i.e., the procedure terminates when either of the supplier or demand quantities are shipped and the surplus/slack quantity is automatically deemed to have been allocated to a dummy supplier/customer. Hence, this procedure applies to both balanced and unbalanced transportation problems without introducing a dummy supplier or a dummy customer.

4.3. Mutation equations

While the mutation operation for candidate antibodies used in hybrid immune networks with swarm learning (Fu, Li, & Tan, 2007) are applied as the mutation equations with PSO as in (5) instead of (3) and (4), the proposed mutation operation of the HPSIL is defined by the following equation:

$$\bar{p}_k(t+1) = \bar{p}_k(t) + c_1 r_1 e^{\text{Affinity}} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t))$$

(6)

where $\bar{p}_k(t+1)$ and $\bar{p}_k(t)$ are the new and previous particles, respectively. $\bar{g}(t)$ is the position of the particle with the best global value function, $\beta$ is a control parameter for mutation, $r_1$ and $r_2$ are uniform random numbers, and $c_1$ and $c_2$ are the acceleration coefficients. Affinity is measured based on the Euclidean distance between two antibodies or between an antibody and an antigen (De Castro and Von Zuben, 2001). Using this Euclidean distance as a measure of affinity consumes more time than affinity function based on fitness value. One of the formulations based on makespan values of the schedules was presented by Costa, Vargas, Von Zuben, and Franca (2002) and also used by Orhan and Alper (2004). In the FCTP, particle $\bar{p}_k$ has a value function that refers to the affinity value of that particle. Affinity value of each particle is calculated from the value function. Hence the affinity function for the HPSIL is defined as in (7) where the fitness of the particle $\bar{p}_k$ is equal to the value of its objective function.

$$\text{Affinity} (\bar{p}_k) = \frac{1}{\text{Fitness}(\bar{p}_k)}$$

(7)

The aim of (6) is to guide the particle to fly towards its best position, and the affinity value emphasizes swarm learning whereby the particle always learns from the best of the swarm and has a tendency to fly toward the best. As a result it has the potential to compete for the best new particle. Carrying out the mutation guides the particle to fly toward its best position depending on several factors, including the affinity value, the position of the previous particle $\bar{h}_k(t)$, and the position of the global best particle $\bar{g}(t)$. Therefore, several candidate modifications can be carried out on the structure of the mutation Eq. (6). The proposed modifications are explained in the following paragraphs.

The first modification: The first term of the equation, which is the position of the previous particle $\bar{p}_k(t)$, is replaced with the best local particle position $\bar{h}_k(t)$.

The second modification: Since many affinity functions have been used in the literature, three different modification to affinity function (7) are proposed. The proposed affinity functions are as follows.

The first affinity function is the function value (fitness) of the best local particle position, and denoted as $\nu(\bar{h}_k(t))$. This affinity function is represented in the following equation:

$$\text{Affinity}_1 = \frac{1}{\nu(\bar{h}_k(t))}$$

(8)

The second affinity function is the difference between the value of the best local particle $\nu(\bar{h}_k(t))$ and the best global particle $\nu(\bar{g}(t))$ divided by the value of the best local particle $\nu(\bar{h}_k(t))$ as given in the following equation:

$$\text{Affinity}_2 = \frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{h}_k(t))}$$

(9)

The third affinity function is the difference between the value of the best local particle $\nu(\bar{h}_k(t))$ and the best global particle $\nu(\bar{g}(t))$ divided by the value of the best global particle $\nu(\bar{g}(t))$ as given in (10). Since the FCTP is a minimization problem, it can be observed that $\nu(\bar{h}_k(t)) > \nu(\bar{g}(t))$ in all cases.

$$\text{Affinity}_3 = \frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}$$

(10)

As a result of the three affinity functions and the two particle positions (the current or previous), six versions of mutation Eq. (6) are constructed. While in the original PSO algorithm the structure of the particle consists of real numbers, the FCTP requires a particle structure with integer values to represent the number of supplier and number of demand. Hence, the PSO is modified to generate an integer value in the search space. Simply stated, in the HPSIL algorithm real numbers generated through PSO within the defined search space are truncated into integers using the $\text{int} [ ]$ function, where $\text{int} [ ]$ represents the truncation function. Hence, the recommended mutation equations used by the HPSIL algorithm to generate a new population is one of (11)–(16).

$$\bar{p}_k(t+1) = \text{int} [\bar{p}_k(t) + c_1 r_1 e^{\text{Affinity}} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t))]$$

(11)

$$\bar{p}_k(t+1) = \text{int} [\bar{p}_k(t) + c_1 r_1 e^{\left(\frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}\right)} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t))]$$

(12)

$$\bar{p}_k(t+1) = \text{int} \left[ \bar{p}_k(t) + c_1 r_1 e^{\left(\frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}\right)} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t)) \right]$$

(13)

$$\bar{p}_k(t+1) = \text{int} \left[ \bar{h}_k(t) + c_1 r_1 e^{\left(\frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}\right)} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t)) \right]$$

(14)

$$\bar{p}_k(t+1) = \text{int} \left[ \bar{h}_k(t) + c_1 r_1 e^{\left(\frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}\right)} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t)) \right]$$

(15)

$$\bar{p}_k(t+1) = \text{int} \left[ \bar{h}_k(t) + c_1 r_1 e^{\left(\frac{\nu(\bar{h}_k(t)) - \nu(\bar{g}(t))}{\nu(\bar{g}(t))}\right)} + c_2 r_2 (\bar{g}(t) - \bar{p}_k(t)) \right]$$

(16)

5. Performance study

The aim of this section is to define the factors that affect the performance of the HPSIL algorithm and selecting the level of each factor which provide its best performance. In order to achieve this aim, two measures considered are detailed. In addition the characteristics of the problems used for analyzing the performance are presented. The analysis of experimental design presents the comparison of results based on the main factor effects and interaction effects for finding the best factor levels.

5.1. Experimental design

It can be observed that a number of factors affect the performance of the HPSIL algorithm. The factors include the particle position (PP) used in the first part of swarm equation (best local particle position or previous particle position), the population size (PSize), affinity functions (AFs), and the acceleration coefficients.
Table 1
Factors and their levels.

<table>
<thead>
<tr>
<th>Factors</th>
<th>Symbols</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle position</td>
<td>PP</td>
<td>Local best particle, Previous particle</td>
</tr>
<tr>
<td>Affinity function</td>
<td>AF</td>
<td>1, ( \frac{n(t)}{n(t-1) + n(t+1)} ), ( \frac{n(t)}{n(t-1) + n(t+1)} )</td>
</tr>
<tr>
<td>Population size</td>
<td>PSize</td>
<td>40, 60, 80</td>
</tr>
<tr>
<td>Acceleration coefficients</td>
<td>(C1, C2)</td>
<td>0.60, 0.40, 0.75, 0.40, 0.73, 0.37, 0.70, 0.30</td>
</tr>
</tbody>
</table>

Table 2
Characteristics of the FCT test problems.

<table>
<thead>
<tr>
<th>Problem size</th>
<th>Total supply</th>
<th>Rang of variable costs</th>
<th>Rang of fixed costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower limit</td>
<td>Upper limit</td>
<td>Lower limit</td>
</tr>
<tr>
<td>10 x 10</td>
<td>10,000</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>10 x 20</td>
<td>15,000</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>15 x 15</td>
<td>15,000</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>10 x 30</td>
<td>15,000</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>50 x 50</td>
<td>50,000</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 3
Mean RPD responses.

<table>
<thead>
<tr>
<th>Level</th>
<th>PP</th>
<th>AF</th>
<th>PSize</th>
<th>C1 and C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0312</td>
<td>0.0267</td>
<td>0.0361</td>
<td>0.0288</td>
</tr>
<tr>
<td>2</td>
<td>0.0268</td>
<td>0.0310</td>
<td>0.0294</td>
<td>0.0286</td>
</tr>
<tr>
<td>3</td>
<td>0.0293</td>
<td>0.0215</td>
<td>0.0278</td>
<td>0.0309</td>
</tr>
<tr>
<td>4</td>
<td>Delta 0.0044</td>
<td>0.0042</td>
<td>0.0146</td>
<td>0.0031</td>
</tr>
<tr>
<td>Rank</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

5.2. Analysis of experimental design

The algorithm is coded in Visual Basic and executed on a PC with a 1.83 GHz Intel Core 2 Duo processor and 4 GB of RAM. The factorial design and analysis are carried out using MINITAB Release 14.1 statistical software. For the sake of fairness, the same number of iterations are set for all trials or combinations of operator and parameter levels, that is, 1000 generations.

The mean RPD values of the trials are averaged at each level, as presented in Table 3. These values are plotted in Fig. 5. The response tables show the average of each response characteristic (mean RPD) for each level of each factor. The table includes ranks based on Delta statistics, which compare the relative magnitude of effects. The ranks based on Delta values; rank 1 to the highest Delta value, rank 2 to the second highest, and so on are presented in Table 3. The ranks indicate the relative importance of each factor to the response. It can be observed that the relative impact rankings on the performance of the proposed algorithm are the population size (PSize), the particle position (PP), the affinity function (AF), and the acceleration coefficients (C1 and C2) respectively. This means the population size (PSize), followed by the particle position (PP) have the highest impact on the performance of the proposed algorithm. Similarly the S/N ratios are averaged at each level and are presented in Table 4. Further, the mean S/N ratio results for
each parameter level are shown in Fig. 6. As can be seen, the S/N ratios illustrate the best parameters for the factors, which confirm the same results as the RPD values.

Referring to Figs. 5 and 6, we can observe that the performance of the algorithm is gradually improving as the PSize is increased. Based on the slope of the plot, there seems to be significant improvements in the performance at each level. Therefore, the higher the PSize, the better is the performance of the algorithm.

Also, the performance of the algorithm is better at level two of the PP, as compared to the first level, i.e., using the previous particle position gives a better performance than using the local best particle position in the mutation equations presented in (11)–(16) of the proposed algorithm. Similarly, the first AF considered at level 1 provides the best performance than the other two affinity functions, considered as levels 2 and 3 respectively. In other words, using the AF as in (8) will result in the algorithm performing better than using the affinity functions as in (9) and (10). In addition to that, the performance of the algorithm differs marginally at the four different levels of the (C1 and C2) considered. However, even though the relative best performance is achieved at the third level, which are at 0.73, 0.37 respectively, it seems that there is no significant difference in the performance among the first three levels of the (C1 and C2) considered.

An interaction between the factors can magnify or diminish the main effects. Hence, evaluating the effect of interactions is important. The pairwise interaction among the factors affecting RPD and $S/N$ ratio are plotted in Figs. 7 and 8, respectively. These plots show the impact of changing the setting of one factor on another. From the parts A–C of Fig. 7 it can be observed that the performance of the algorithm is uniformly better at level two of the particle position (PP) at all levels of the other three factors, viz., affinity function (AF), population size (PSize) and acceleration coefficients (C1 and C2). This conclusion can be arrived at since the interaction plots of the PP with the other three factors result in lines which consistently result in better performance for level two of the PP. Similarly, Parts G–I of Fig. 7 show that the performance of the algorithm is uniformly better for level three of the PSize. There seems to be considerable effect of interaction between the AF used and the (C1 and C2). It can be observed from Parts D–F of Fig. 7 that the affinity function 1 provides a better solution for both the levels of PPs, for higher values of PSize and when we use levels 1, 3 and 4 of the (C1 and C2). However, the affinity functions 2 and 3 will provide better solution for the second level of the acceleration coefficients. The interactions presented in Parts J–L of Fig. 7 indicate that the second level of the PP yields a better performance for all levels.

<table>
<thead>
<tr>
<th>Level</th>
<th>PP</th>
<th>AF</th>
<th>PSize</th>
<th>C1 and C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>−12.71</td>
<td>−12.70</td>
<td>−12.71</td>
<td>−12.70</td>
</tr>
<tr>
<td>2</td>
<td>−12.70</td>
<td>−12.71</td>
<td>−12.71</td>
<td>−12.70</td>
</tr>
<tr>
<td>3</td>
<td>−12.71</td>
<td>−12.69</td>
<td>−12.70</td>
<td>−12.71</td>
</tr>
<tr>
<td>Delta</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>Rank</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Fig. 6. Main effects plot (fitted mean) for $S/N$ ratio at each level of the factors.

Fig. 7. Main interaction plot (fitted mean) for RPD.
of the \((C_1\text{ and } C_2)\). Further, the level 1 of the acceleration coefficients yields equally good results for levels 1 and 3 of the affinity function. However, levels 2 and 3 of the \((C_1\text{ and } C_2)\) provide good results for level 2 of the AF. Part L of the Fig. 7 confirms that level 3 of the Psize provides uniformly better solutions for all levels of the \((C_1\text{ and } C_2)\). The same results will be arrived at by interpreting the interaction plots for \(S/N\) ratio presented in Fig. 8.

These results suggest that to obtain the minimum solutions for our FCTP, the setting for the particle position should be at the level 2 and the population size at its third level which is the highest value considered. Prior to considering the interactions, the analysis suggests that the affinity function at its first level, and the acceleration coefficients at their third level provide the best solution. The analysis of interaction effects also suggest that the acceleration coefficients at level 3 with affinity function at level 1 provide minimum solution.

Based on the above analysis, the best setup of the HPSIL algorithm is setting: the population size at its third level which is equal to 80 particles or more, the particle position at the level 2, the affinity function at its first level, and the acceleration coefficients at their third level which are at 0.73, 0.37 respectively. The same conclusion can be arrived at using Table 4 and Fig. 6 for \(S/N\) ratio.

6. Numerical experiments

To evaluate the performance of the proposed algorithm, two problems of different sizes, previously addressed by Hajiaghaei et al. (2010) and Othman et al. (2011) were solved, and compared with the solutions presented by the original researchers and with the solution from LINGO. The sizes of the problems are 4 \(\times\) 5 and 5 \(\times\) 10, respectively. The variable and fixed costs for the first problem are given in Table 5, while those for the second problem are given in Table 6. The parameters used in the HPSIL algorithm for these problems are optimally tuned parameters and operators based on experimental results.

Regarding the first problem as illustrated in Table 5, the supply and demand values from each plant (1–4) for each customer (1–5) are as follows: \(s_1 = 57, s_2 = 93, s_3 = 50, s_4 = 75, d_1 = 88, d_2 = 57,\)

<table>
<thead>
<tr>
<th>Plants</th>
<th>Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipping costs (c_{ij})</td>
<td>Fixed costs (f_{ij})</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5
Unit variable cost in 4 \(\times\) 5 problem.

<table>
<thead>
<tr>
<th>Plants</th>
<th>Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipping costs (c_{ij})</td>
<td>Fixed costs (f_{ij})</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 6
Unit variable cost in 5 \(\times\) 10 problem.
Concerning the first problem, the local optimum solution found by Hajaghaye-Keshtei et al. (2010) seems to be a global optimum solution of this problem, therefore the HPSIL algorithm cannot find a better solution. While, the solution found for the second problem by Hajaghaye-Keshtei et al. (2010) is not an optimal solution, the HPSIL algorithm resulted in a better solution.

7. Conclusion

In this paper, a HPSIL for solving the FCTP has been proposed. To investigate the influence of the parameters on the performance of the algorithm, experimental designs have been carried out. In the proposed HPSIL algorithm a flexible particle structure combined with decoding and allocation procedures are used instead of a Prüfer number and a spanning tree used with a genetic algorithm. One of the major contributions is that the HPSIL generates the feasibility of all generated solutions and can be used for solving both balanced and unbalanced FCTPs. Moreover, the chromosome (particle) structure combined with the decoding procedure can be used with any meta-heuristic techniques such as the Tabu search, genetic algorithms, ant colonies and artificial immune systems, among others. The comparison of the HPSIL algorithm with the GAS presented by Othman et al. (2011), Hajaghaye et al. (2010), and LINGO shows that the HPSIL algorithm provides an equal or better solution compared to the others. The performance of the HPSIL algorithm and the solution quality prove that HPSIL is highly competitive and can be considered as a viable alternative for solving FCTPs.

Future work includes further experimentation with the parameters for the HPSIL algorithm, such as studying the relation between the population size and the problem dimension, analyzing the effect of acceleration coefficients (C₁ and C₂) on the problem solution. Further, the HPSIL algorithm will be tested on other real life problems and investigate using of other meta-heuristic techniques combined with the proposed decoding and allocation procedures for solving the FCTP.

Acknowledgement

The authors appreciate and acknowledge the comments of the reviewers, which greatly helped us in improving the contents of the paper. This paper is supported by the Research Center at the College of Business Administration and the Deanship of Scientific Research at King Saud University, Riyadh.

References


Nagoya, Japan.


### Table 7

Transportation allocation matrix found LINGO for 4 × 5 problem.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>38</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>88</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td>42</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8

Transportation allocation matrix found by Hajaghaye et al. (2010) for 4 × 5 problem.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>69</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>225</td>
<td>71</td>
<td>15</td>
<td>88</td>
<td>47</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>215</td>
<td>64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 9

Transportation allocation matrix found by LINGO for 5 × 10 problem.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
<th>D8</th>
<th>D9</th>
<th>D10</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>106</td>
</tr>
<tr>
<td>S2</td>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>167</td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>133</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10

Transportation allocation matrix found by Hajaghaye et al. (2010) for 5 × 10 problem.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
<th>D8</th>
<th>D9</th>
<th>D10</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>124</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
<td></td>
<td></td>
<td></td>
<td>37</td>
<td></td>
</tr>
</tbody>
</table>

### Table 11

Transportation allocation matrix found by the HPSIL for 5 × 10 problem.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
<th>D8</th>
<th>D9</th>
<th>D10</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>124</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td></td>
<td></td>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>241</td>
<td>69</td>
</tr>
</tbody>
</table>


