An Integrated Model for Lot Sizing with Supplier Selection Considering Quantity Discounts, Expiry Dates, and Budget Availability

Teguh E.N. Sitepu*1, Andi Cakravastia*2

1Supply Chain Management Department, Institut Teknologi Harapan Bangsa
Jl. Dipati Ukur 80-84, Bandung, Indonesia
teguh_sitepu@ithb.ac.id

2Industrial Engineering Department, Institut Teknologi Bandung
Jl. Ganasa 10, Bandung, Indonesia
andi@mail.ti.itb.ac.id

Abstract—In this paper, a dynamic multi-product multi-period lot sizing with supplier selection problem (DLSSP) with quantity discount, expiry dates, and budget availability is presented. Demand of products for each period are independent and known. The cost consists of ordering, purchasing, transportation, expiry, holding, and interest charge. The objective is to find the optimal order quantity of all items in each period to minimize inventory cost. A mixed integer nonlinear model programming (MINLP) is first developed to model the problem. Since model is hard to solve using exact method, Genetic Algorithm (GA) and Simulated Annealing (SA) is applied, in which design parameters are set using Taguchi method. Computational results demonstrate the applicability of the proposed model and comparing the results show efficiency of both algorithms as well. The results show that, while both algorithms have statistically similar performances, proposed SA is the better algorithm in all problems.

Keywords—lot size, supplier selection, quantity discount, perishable product, simulated annealing

1. Introduction

Procurement is the process of obtaining goods and services within a supply chain. Managers must structure procurement with a goal of increasing supply chain surplus [1]. Ref. [2] claimed that purchased goods and services are one of the largest elements of cost for many firms. Determination of right procurement strategy can increase a firm’s profit [3].

Dynamic lot sizing problem (DLSP) is one of the basic procurement strategy, introduced by Wagner & Within [4]. They considered only one item and

one supplier. Ref. [5] developed DLSP model into dynamic multi-product multi-period lot sizing model with supplier selection problem (DLSSP). Since then, many developed models of DLSSP had been done by considering condition in real life. This paper considered quantity discount, expiry dates, and budget availability.

Quantity discount is an effective strategy for many suppliers to promote their product. On the other hand, buyer can buy product with cheaper price when buy product over certain amount. DLSSP model considered quantity discount had been done by [6], [7], and [8]. Their models considered unlimited lifetime item.

In real life, some products have limited lifetime. For example: food, pharmaceutical, and fashion product. Limited lifetime product will decay if it’s not used until expired. So, inventory pick-up policy is important for limited lifetime product [9]. Ref. [10] and [11] had been developed DLSP model considered limited lifetime product.

Based on literature review, majority of developed DLSSP models considered unlimited budget. In real life, a firm has limited budget, so cash is the heart of all businesses [12]. If procurement cost less than budget availability, the firm will buy with cash. But if procurement cost greater than budget availability, the firm must buy with credit and pay interest charge to the bank.

In this paper, DLSSP model for limited lifetime product developed by considering quantity discount and budget availability. Ref. [13] stated this problem is NP-hard problem. So, this paper proposed MINLP model to solve problem and comparing exact method with metaheuristic algorithm. Genetic Algorithm (GA) and Simulated Annealing (SA) presented.

The remaining of this paper is organized as follows. In section 2, some of related researches are reviewed. In section 3, problem is described. In
section 4, the formulation of proposed model. In section 5, construction of metaheuristic algorithms described. In section 6, experiments are carried out. In section 7, conclusion is presented and future directions discussed.

2. Related Works

Lot size problem can be categorized by its characteristic. One of the category is lot size problem with discrete time, deterministic demand, and finite planning horizon. This category called dynamic lot size problem (DLSP) and first introduced by Wagner & Within [4]. Basic DLSP model assumed product can buy only from one supplier. In real life, some products aren’t monopolized so the firm can select supplier whose offer cheaper price. That condition made DLSP model developed into dynamic lot sizing with supplier selection problem (DLSSP). Ref. [5] was first introduced DLSSP considering multi-product. Ref. [14] developed DLSSP model considering limited supplier capacity. Proposed model is mixed integer nonlinear programming (MINLP) and solved using LINGO software. Ref. [13] developed new method to solve Ref [5] model. Ref. [13] presented Reduce and Optimize Approach (ROA).

In procurement policy, supplier can offer discount quantity based on purchasing amount. This strategy can make supplier keep leading in price competitiveness. On the other hand, buyer can buy product in certain quantity to take advantage from supplier offering. Ref. [6] and [15] developed DLSP model considering quantity discount and limited warehouse capacity. Ref. [6] implemented GA and Memetic Algorithm (MA). While Ref. [15] developed heuristic algorithm based on GA. Ref. [16] developed GA to solve DLSSP model considering quantity discount. Proposed model has objective function to minimize ordering cost, purchasing cost, transportation cost, and holding cost. Ref. [8] added third dimension to Fordyce Webster Algorithm (FWA) to solved DLSSP model.

Based on literature review, majority of lot sizing models considered unlimited budget. Ref. [17] developed probabilistic inventory model considering budget availability and limited warehouse capacity. Proposed model was nonlinear and solved using LINGO software. Ref. [18] developed DLSP model considering budget availability and quantity discount. Solution procedure was using GA and PSO. Furthermore, Ref. [19] developed Ref. [18] into bi-objective mixed integer nonlinear programming (MINLP). Proposed model has objective function to minimize total inventory cost and storage area. Ref. [12] developed DLSP model to maximize profit. The firm must pay interest-bearing loan if budget availability not enough to buy product with cash. On the other hand, the firm will gain interest-bearing deposit if budget availability extant at one period.

All developed DLSP models above considering unlimited lifetime product. In real life case, some products have limited lifetime, so it will decay if it’s not used until expired. Ref. [10] developed dynamic lot sizing and scheduling problem considering limited lifetime product. Consideration of limited lifetime part and product increased planning complexity. Ref. [11] developed DLSSP considering pick-up policy by First-Expired-First-Out (FEFO). Proposed model was mixed integer nonlinear programming (MINLP) through linearization and then solved by GA and PSO.

3. Problem Description

We assume a retailer who sell multi products to customer. The retailer buys products from multi suppliers who offer different ordering, purchasing, and transportation cost. For purchasing cost, every supplier offers different all-unit discount policy shown in Fig 1. Every period, the retailer has limited budget to buy products. If budget isn’t enough to buy with cash, the retailer must borrow some money from bank so interest charge cost increasing.
to recycle or disposal activity. The retailer must optimize procurement policy to minimize total inventory cost along planning horizon. Total inventory cost consists of ordering, purchasing, transportation, expiry, holding, and interest charge.

![Figure 2. Inventory system of the proposed model](image)

### 4. Mathematical Model

#### 4.1 Assumptions

Assumptions in this problem are defined as follows:

a. Demand for each period is independent and known.
b. Initial and ending inventory are zero.
c. Each product has the same space at vehicle.
d. Shortage is not allowed.
e. Lead time is known and delivered at once in the beginning of a period.
f. First Expired-First Out (FEFO) policy is considered for picking up inventory.
g. The capacity of supplier and warehouse is unlimited.
h. The price is dependent on the order quantity.
i. Unused budget cannot be carried over to subsequent periods
j. Each product has a finite lifetime that greater than planning period

#### 4.2 Notations

##### 4.2.1 Indices

Indices in this problem are defined as follows:

- $m$: products
- $n$: remaining lifetime
- $s$: suppliers
- $b$: breakpoint
- $t$: periods

##### 4.2.2 Parameters

Parameters in this problem are defined as follows:

- $B_{U_mlt}$: budget availability for product type $m$ in period $t$
- $\beta_t$: interest rate in period $t$
- $W_m$: lifetime of product type $m$
- $H_m$: holding cost of product type $m$, per unit per period
- $J_m$: expiry cost of product type $m$, per unit per period
- $C_{mst}$: ordering cost of product type $m$ from supplier $s$
- $K_s$: transportation cost per vehicle from supplier $s$
- $D_{mt}$: demand of product type $m$ in period $t$
- $L_{mst}$: the lower bound quantity of product type $m$ from supplier $s$ with price break $b$
- $U_{mst}$: the upper bound quantity of product type $m$ from supplier $s$ with price break $b$
- $P_{mst}$: unit purchase cost of product type $m$ from supplier $s$ with price break $b$
- $N$: the maximum vehicle batch size
- $V$: a large number

##### 4.2.3 Decision Variables

Decision variables in this problem are defined as follows:

- $X_{mt}$: expired quantity of product $m$ in period $t$
- $I_{nmt}$: inventory level of product type $m$ with $n$ periods of lifetime remaining in the end of period $t$
- $R_{st}$: number of vehicle from supplier $s$ in period $t$
- $Q_{mt}$: total purchase quantity of product type $m$ in period $t$
- $P(Q_{mst})$: purchase cost for a unit product type $m$ based on the discount of supplier $s$ in period $t$
- $a_{mt}$: the maximum number of product type $m$ bought with cash in period $t$
- $\gamma_{mt}$: number of product type $m$ that bought with credit in period $t$
- $q_{mst}$: purchase quantity of product type $m$ from supplier $s$ in period $t$
- $CA_{mt}$: method of payment product type $m$ in period $t$
\[ Y_{mt} \] a binary variable, set equal to 1 if a purchase is made, and 0 if no purchase is made, of product type \( m \) in period \( t \).

\[ F_{mnt} \] a binary variable, set equal to 1 if a purchase is made, and 0 if no purchase is made, of product type \( m \) from supplier \( s \) in period \( t \).

\[ f_{msbt} \] a binary variable, set equal to 1 if a purchase is made, and 0 if no purchase is made, of product type \( m \) from supplier \( s \) with price break \( b \) in period \( t \).

### 4.3 Developed Mathematical Model

Due to the budget availability, one of the following conditions is satisfied for each period:

**Condition 1** : \( Q_{mt} = 0 \) \( \forall m, t \) \hspace{1cm} (1)

**Condition 2** : \( 0 < Q_{mt} \leq \alpha_{mt} \) \( \forall m, t \) \hspace{1cm} (2)

**Condition 3** : \( Q_{mt} > \alpha_{mt} \) \( \forall m, t \) \hspace{1cm} (3)

Expression (1) means that no product type \( m \) purchased in period \( t \). Expression (2) means that product type \( m \) in period \( t \) \( (Q_{mt}) \) purchased with cash because the quantity is equal or less than the maximum number of product type \( m \) can be bought with cash in period \( t \) \( (\alpha_{mt}) \). Expression (3) means that product type \( m \) in period \( t \) \( (Q_{mt}) \) purchased with credit because the quantity is greater than \( \alpha_{mt} \).

The nonlinear budget availability constraint sets are converted to linear expressions with the help of a new set of binary variables \( (CA_{mtn}) \).

\[ CA_{mnt} \begin{cases} 1, \text{if condition } \sigma \text{ is sustained} \\ 0, \text{otherwise} \end{cases} \forall \sigma, m, t \hspace{1cm} (4)

This way, the mathematical model of the problem is stated as follows:

\[
\min \sum_{m} \left( \sum_{s} \left( F_{msbt} C_{ms} + \left( P(Q_{mst}) F_{msbt} Q_{mst} \right) \right) + R_{s} K_{s} \right) + \sum_{m} \left( X_{mt} M_{m} + \sum_{n=1}^{N} \alpha_{nt} H_{n} + V_{mt}(1 + \beta) \sum_{s \in S} P(Q_{mst}) F_{msbt} \right)
\]

subject to

\[ I_{nmo} = I_{nmt} = 0, \quad \forall n, m \]

\[ Q_{mt} = \sum_{s \in S} Q_{msbt}, \quad \forall m, t \]

\[ R_{st} \geq \sum_{m=1}^{M} \frac{f_{msbt} \alpha_{mt}}{N}, \quad \forall s, t \]

\[
\begin{align*}
D_{mt} & \leq I_{m(t-1)} + V. (1 - z_{1mt}) \quad \forall m, t \\
X_{mt} & \geq I_{m(t-1)} - D_{mt} - V. (1 - z_{1mt}) \\
X_{mt} & \leq I_{m(t-1)} - D_{mt} + V. (1 - z_{1mt}) \\
I_{1mt} & \geq I_{m(t-1)} + V. (1 - z_{1mt}) \\
I_{1mt} & \leq I_{m(t-1)} + V. (1 - z_{1mt}) \\
I_{m(t-1)} & - V. (1 - z_{2mt}) - D_{mt} \quad \forall m, t \\
I_{2mt} & \geq I_{m(t-1)} + I_{2mt} - D_{mt} - V. (1 - z_{2mt}) \\
I_{2mt} & \leq I_{m(t-1)} + I_{2mt} - D_{mt} + V. (1 - z_{2mt}) \forall m, t \quad (9)
\end{align*}
\]

\[
\begin{align*}
I_{(n+1)m(t-1)} & \geq Q_{mt} - V. (1 - z_{1mt}) \\
I_{(n+1)m(t-1)} & \leq Q_{mt} + V. (1 - z_{1mt}) \\
I_{(n+1)m(t-1)} & - V. (1 - z_{2mt}) \quad \forall m, t \\
I_{(n+1)m(t-1)} & \geq I_{(n+1)m(t-1)} - D_{mt} - V. (1 - z_{2mt}) \\
I_{(n+1)m(t-1)} & \leq I_{(n+1)m(t-1)} - D_{mt} + V. (1 - z_{2mt}) \forall m, t \quad (10)
\end{align*}
\]

\[
\begin{align*}
I_{(n+1)m(t-1)} & \geq Q_{mt} - V. (1 - z_{2mt}) \\
I_{(n+1)m(t-1)} & \leq Q_{mt} + V. (1 - z_{2mt}) \\
I_{(n+1)m(t-1)} & - V. (1 - z_{2mt}) \quad \forall m, t \\
I_{(n+1)m(t-1)} & \geq I_{(n+1)m(t-1)} - D_{mt} - V. (1 - z_{2mt}) \\
I_{(n+1)m(t-1)} & \leq I_{(n+1)m(t-1)} - D_{mt} + V. (1 - z_{2mt}) \forall m, t \quad (11)
\end{align*}
\]

\[
\begin{align*}
\sum_{m} \left( \sum_{s \in S} \left( f_{msbt} C_{ms} + \left( P(Q_{mst}) f_{msbt} Q_{mst} \right) \right) \right) \\
\sum_{m} \left( X_{mt} M_{m} + \sum_{n=1}^{N} \alpha_{nt} H_{n} + V_{mt}(1 + \beta) \sum_{s \in S} P(Q_{mst}) f_{msbt} \right)
\]

\[
\begin{align*}
\sum_{m} Q_{m(t-1)} + \sum_{m} q_{m(t-1)} & - V. (1 - z_{1mt}) - \sum_{m} Q_{m(t-1)} + \sum_{m} q_{m(t-1)} + V. (1 - z_{1mt}) \geq 0 \\
\sum_{m} X_{mt} + \sum_{m} x_{mt} & \geq - V. (1 - z_{2mt}) \\
\sum_{m} X_{mt} + \sum_{m} x_{mt} & \leq - V. (1 - z_{2mt}) \\
\sum_{m} I_{m(t-1)} + \sum_{m} i_{m(t-1)} & - D_{mt} - V. (1 - z_{2mt}) \geq 0 \\
\sum_{m} I_{m(t-1)} + \sum_{m} i_{m(t-1)} + \sum_{m} i_{m(t-1)} - D_{mt} - V. (1 - z_{2mt}) \leq 0 \\
\sum_{m} I_{m(t-1)} + \sum_{m} i_{m(t-1)} - D_{mt} + V. (1 - z_{2mt}) \leq 0 \\
\sum_{m} I_{m(t-1)} + \sum_{m} i_{m(t-1)} + \sum_{m} i_{m(t-1)} - D_{mt} + V. (1 - z_{2mt}) \geq 0 \\
\sum_{m} z_{m(t-1)} = 1, \quad \forall m, t \\
q_{mst} & \geq U_{msbt} - V. (1 - f_{msbt}), \quad \forall m, s, b, t \\
q_{mst} & \leq U_{msbt} + V. (1 - f_{msbt}), \quad \forall m, s, b, t \\
\sum_{b \in B} f_{msbt} & = 1, \quad \forall m, s, t \\
q_{mst} & \leq V. F_{msbt}, \quad \forall m, s, t \\
P(Q_{mst}) & = \sum_{b \in B} f_{msbt} P_{msbt}, \quad \forall m, s, t
\end{align*}
\]
\[
\alpha_{mt} \geq \frac{\text{BUD}_{mt}}{\sum_{s \in \text{supp}} P(q_{ms})f_{ms} + CA_{mt}}, \quad \forall \ m, t \quad (20)
\]
\[
Q_{mt} \geq -(1 - CA_{mt}) \quad \forall \ m, t \quad (21)
\]
\[
Q_{mt} \leq V \cdot (1 - CA_{mt}) \quad \forall \ m, t \quad (22)
\]
\[
Q_{mt} \geq 1 - V \cdot (1 - CA_{mt}) \quad \forall \ m, t \quad (23)
\]
\[
Q_{mt} \leq \alpha_{mt} + V \cdot (1 - CA_{mt}) \quad \forall \ m, t \quad (24)
\]
\[
\sum_{s \in \text{supp}} CA_{ms} = 1 \quad \forall \ m, t \quad (25)
\]
\[
\gamma_{mt} = (Q_{mt} - \alpha_{mt}) \cdot CA_{mt}, \quad \forall \ m, t \quad (26)
\]
\[
\alpha_{mt} \in \{0,1\}, \quad \forall \ m, t \quad (27)
\]
\[
F_{mst} \in \{0,1\}, \quad \forall \ m, s, t \quad (28)
\]
\[
f_{mst} \in \{0,1\}, \quad \forall \ m, s, b, t \quad (29)
\]

The problem is to find the quantity of each product to be ordered in each period from the suppliers \( q_{ms} \) while minimizing the total cost, as formulated in Eq. (5). The first term of the objective function indicates total ordering cost, while the second term is used to calculate total purchasing cost. The next term represents the total transportation cost, while the next three terms are used to calculate expiry cost, holding cost, and interest charge.

Eq. (6) is to set no initial and ending inventory. Eq. (7) is to set total purchase quantity of product type \( m \) in period \( t \). In Eq. (8), the number of vehicle from supplier \( s \) in period \( t \), \( R_{mt} \), is equal to biggest integer greater than or equal total purchase quantity of all products from supplier \( s \) in period \( t \), \( \sum_{m \in \text{supp}} q_{ms} \), divide vehicle batch size, \( N \).

Eqs. (9)-(14) is linear expressions of product perishability model based on Ref. [11]. Eq. (9) mean the condition when the demand for product type \( m \) in period \( t \), \( D_{mt} \), is less than the inventory of product type \( m \) with one period of lifetime remaining in that period \( (D_{mt} \leq I_{1m(t-1)}) \). Eq. (10) mean the condition when the demand for product type \( m \) in period \( t \), \( D_{mt} \), is greater than the inventory of product type \( m \) with one period of lifetime remaining \( (D_{mt} \geq I_{1m(t-1)}) \), and less than the sum of the inventories of product type \( m \) with one and two periods of lifetime remaining \( (D_{mt} \leq I_{1m(t-1)} + I_{2m(t-1)}) \), and so on. Eq. (14) limit only one of the defined conditions can be occurred in a period time.

Eqs. (15)-(19) are expressions of dynamic inventory problem. Eq. (15) set the lower bound quantity while Eq. (16) set the upper bound quantity in a price break \( b \) for supplier \( s \). Eq. (17) makes sure that a quantity of product type \( m \) can only be purchased with one single price break \( b \) from supplier \( s \) in period \( t \). Eq. (18) is to set the purchase quantity of product type \( m \) from supplier \( s \) in period \( t \), and \( F_{mst} \) is a binary variable. Eq. (19) determines the purchase cost of product type \( m \) per unit, \( P(q_{ms}) \), based on the quantity purchased from supplier \( s \) in period \( t \).

Eqs. (20)-(25) are expressions of budget availability model. Eq. (20) is to set the maximum number of product type \( m \) can bought with cash in period \( t \). Eqs. (21)-(23) is linear expressions of condition from Eqs. (1)-(3). Eq. (24) is added to ensure that only one of the budget availability constraint sets can be satisfied. Eq. (25) determines the number of product type \( m \) that bought with credit in period \( t \).

5. Solution Procedure

Ref. [13] stated that DLSP is NP-hard problem. Because of that, exact method isn’t practical to solve DLSP. In this paper, GA and SA implemented to solve dynamic lot sizing problem with supplier selection considering quantity discount, expiry dates, and budget availability. Fitness function for each solution defined as minimum total cost, as formulated in Eq. (5).

5.1 Genetic Algorithm

Ref. [16] developed GA to solve DLSSP model. Procedure of algorithm in detail is explained as follows:

Step 1. Define parameter.
Define population number (pop), crossover probability (\( \rho_c \)), mutation probability (\( \rho_m \)), and generation number (\( k_{\text{max}} \)).

Step 2. Define coding scheme.
Every gen in chromosome at Ref. [16] represented purchasing status from supplier \( s \) in period \( t \). Because this paper considered multi-material, chromosome at Ref. [16] developed as shown in Fig 3. Chromosome decoded into three-dimensional matrix \( m \times s \times t \), where \( 1 \) if a purchase of product type \( m \) is made from supplier \( s \) in period \( t \), and \( 0 \) otherwise.
5.2 Simulated Annealing

Ref. [16] developed GA to solve lot sizing problem based on cumulative purchase quantity in Eq. (33). This procedure is irrelevant to developed model. Quantity discount condition make optimal solution can occur when a retailer buy product more than it needed. This paper proposed an algorithm based on Simulated Annealing (SA). Additional notations used in proposed SA are defined as follows:

- \( T_0 \): initial temperature
- \( T_a \): final temperature
- \( T \): current temperature
- \( \theta_0 \): initial solution
- \( \theta_b \): new solution
- \( \theta \): current solution
- \( \theta_a \): final solution
- \( \infty \): cooling rate
- \( k_{\text{max}} \): maximum iteration for every temperature increases
- \( K \): iteration
- \( f(\theta_0) \): objective function of initial solution
- \( f(\theta_b) \): objective function of new solution
- \( f(\theta) \): objective function of current solution
- \( f(\theta_a) \): objective function of final solution

Proposed algorithm shown in Fig 5. Procedure of proposed algorithm in detail is explained as follows:

Step 1. Define control parameters. Define initial temperature \( (T_0) \), final temperature \( (T_a) \), cooling rate \( (\infty) \), and maximum iteration \( (k_{\text{max}}) \).


Step 3. Set current temperature \( T = T_0 \). set \( k = 1 \) and \( \theta = \theta_0 \).

Step 4. Determine operator modification randomly. Generate new solution \( \theta_B \) using selected operator modification.

Step 5. Check the feasibility of new solution. If new solution is feasible go to Step 6, otherwise go to Step 10. Flowchart of new solution feasibility examination shown in Fig 4.

Step 6. Calculate objective function of new solution \( f(\theta_B) \).

Step 7. Calculate \( \Delta = f(\theta_B) - f(\theta) \).

If \( \Delta < 0 \) set \( \theta = \theta_B \), and go to Step 10. Otherwise, go to Step 8.

\[ \sum_{t\in S} F_{\text{mst}} = 1, \quad \forall \ m, t = 1 \]  \hspace{1cm} (30)
\[ \sum_{t\in S} F_{\text{mst}} \leq 1, \quad \forall \ m, t \geq 2 \]  \hspace{1cm} (31)

Step 4. Calculate fitness function.

Step 4.1. Determine purchasing status \( (Y_{mt}) \).

Define \( F_{\text{mst}} \) in purchasing status of product type \( m \) in period \( t \) \( (Y_{mt}) \) using Eq. (32).

\[ Y_{mt} = \sum_{t\in S} F_{\text{mst}}, \quad \forall m, t \]  \hspace{1cm} (32)

Step 4.2. Determine total purchase quantity \( (Q_{mt}) \).

Define \( Y_{mt} \) in purchase quantity of product type \( m \) in period \( t \) \( (Q_{mt}) \) using Eq. (33).

\[ Q_{mt} = Y_{mt} D_{\text{mst}} + \left( \frac{1 - Y_{mt+1}}{1 - Y_{mt+2}} \right) \left( \frac{1 - Y_{mt+3}}{1 - Y_{mt+4}} \right) \ldots \left( \frac{1 - Y_{mt+n}}{1 - Y_{mt+n+1}} \right), \forall m, t \]  \hspace{1cm} (33)

Step 4.3. Determine quantity allocation \( (q_{\text{mst}}) \).

Define \( Q_{\text{mst}} \) into quantity allocation for each supplier \( (q_{\text{mst}}) \) using Eq. (34).

\[ Q_{\text{mst}} = \sum_{t\in S} q_{\text{mst}} F_{\text{mst}}, \quad \forall m, t \]  \hspace{1cm} (34)

Step 4.4. Calculate total cost

Calculate total cost every chromosome based on quantity allocation \( (q_{\text{mst}}) \). Best chromosome is chromosome with minimum total cost on population.

Step 5. Crossover operator.

Select a pair of chromosome randomly, then applied two-cut point-crossover to produce two offspring.

Step 6. Mutation operator.

Mutation is used to avoid premature convergence. In this paper, mutation operator is applied by change gen randomly \( (0 \text{ to } 1 \text{ or } 1 \text{ to } 0) \).

Step 7. Selection.

After crossover and mutation applied, select chromosome for next generation \( (k = k + 1) \). Chromosome was ranked due to minimum fitness function and put into mating pool.

Step 8. Termination.

Repeat Steps 4-8 until \( k = k_{\text{max}} \).
Step 8. Generate random number \( r \) \((0,1)\).

Step 9. Calculate \( e^{(-\Delta/T)} \).
If \( r < e^{(-\Delta/T)} \), set solution \( \theta = \theta_b \) and go to Step 10. Otherwise, go directly to Step 10.

Step 10. Check \( k \) value.
If \( k = k_{\text{max}} \), go to Step 11. Otherwise, set \( k = k + 1 \) and go to Step 4.

Step 11. Set \( T = \alpha T \).

Step 12. Set \( T \) value.
If \( T < T_a \), go to Step 13. Otherwise, set \( k = 1 \) and go to Step 4.

Step 13. Set \( \theta_a = \theta \), and stop.

Figure 4. Flowchart of feasibility examination
Figure 5. Proposed algorithm
Proposed algorithm used three modification operators, as follows:

1. Quantity modified

Quantity modified operator used to modify quantity of product type $m$ purchased in period $t$ ($Q_{mt}$) randomly. It explores solution with the cheaper purchase cost. Illustration of this operator shown in Fig 6.

2. Quantity moved

Quantity moved is an operator suggested by Ref. [20]. This operator modifies quantity of product type $m$ purchased in period $t$ ($Q_{mt}$) and $t+1$ ($Q_{mt+1}$). Illustration of this operator shown in Fig 7.

Procedure of quantity moved in detail is explained as follows:

Procedure of quantity moved in detail is explained as follows:

- **do**
  - decide to add or reduce quantity purchased randomly;
  - **if** reduce, **then**
    - choose product type $m$ in period $t$ randomly;
    - move quantity as much $\Delta_1$ from period $t$ to period $t+1$
  - **if** add, **then**
    - choose product type $m$ in period $t$ randomly;
    - move quantity as much $\Delta_2$ from period $t+1$ to period $t$

where:

\[
\Delta_1 = r(0,1) \times \sum_{m=1}^{W_m} f_{umt}
\]

\[
\Delta_2 = r(0,1) \times Q_{mt+1}
\]

3. Supplier exchanged

Supplier exchanged is an operator to exchange selected supplier with unselected supplier. Illustration of this operator shown in Fig 8.

Procedure of supplier exchanged in detail is explained as follows:

- Choose a group of $\sum_{s \in S} F_{mst}$ which has value 1 randomly.
- Choose one of $F_{mst}$ which has value 0 from that group randomly.
- Modify chosen $F_{mst}$ to 1, others to 0.

6. Experiments and Discussion

6.1 Parameter Tuning

First step in this paper’s experiment is to determine optimal parameter to both metaheuristic algorithms. One of the methods to determine algorithm parameter is using Taguchi method. Taguchi method is fractional experiment method and can be used as alternative to full experiment method [21]. Some response types in Taguchi method is nominal is the best, smaller is better, and larger is better. In this paper, smaller is better used because fitness function is to minimize total cost. Formula for smaller is better:

\[
S/N = -10 \times \log \left( \frac{S(\bar{Y})}{n} \right)
\]

where:

- $Y$ = response
- $n$ = orthogonal array number

To demonstrate Taguchi method, first determine level each parameter for both algorithms shown in Table 1. There are 3 levels for each parameter both algorithms. Next, determine fractional experimental design using Minitab 17. Best combination parameter for both algorithms shown in Figs 9-10.

This paper generated numerical example randomly to verified proposed model. Parameter distribution function shown in Table 2. To find near optimal solution, GA and SA coded using MATLAB 17 in PC Intel Core i5 @2.25 GHz RAM 4 GB Operating System Windows 8.1 64-bit. Table 3 show fitness function and CPU time for exact method (LINGO), GA, and SA.
Table 1. Level and parameter both algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Low (1)</th>
<th>Mid (2)</th>
<th>High (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>Population (A)</td>
<td>50</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Prob_Crossover (B)</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>Prob_Mutation (C)</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Generation (D)</td>
<td>50</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>SA</td>
<td>Init_Temp. (A)</td>
<td>10</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Fin_Temp. (B)</td>
<td>0.01</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Cooling_Rate (C)</td>
<td>0.1</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>Max_Iteration (D)</td>
<td>50</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2. Parameters of numerical examples

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_{int}</td>
<td>N(50,10)</td>
<td>K_{x}</td>
<td>Uniform (25,35)</td>
</tr>
<tr>
<td>B/U_{int}</td>
<td>Uniform (200,800)</td>
<td>\beta_{t}</td>
<td>Uniform (0.1,0.12)</td>
</tr>
<tr>
<td>L_{min}</td>
<td>Uniform (0,50)</td>
<td>W_{m}</td>
<td>Uniform (2,4)</td>
</tr>
<tr>
<td>P_{med}</td>
<td>Uniform (3,4)</td>
<td>H_{m}</td>
<td>Uniform (0,3,0,6)</td>
</tr>
<tr>
<td>C_{med}</td>
<td>Uniform (15,25)</td>
<td>J_{m}</td>
<td>Uniform (0,5,0,8)</td>
</tr>
</tbody>
</table>

Table 3. Results from proposed model

<table>
<thead>
<tr>
<th>No</th>
<th>LINGO Z</th>
<th>CPU Time</th>
<th>GA Z</th>
<th>CPU Time</th>
<th>SA Z</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>0:01:38</td>
<td>230</td>
<td>0:00:23</td>
<td>230</td>
<td>0:00:05</td>
</tr>
<tr>
<td>2</td>
<td>201</td>
<td>0:03:22</td>
<td>238</td>
<td>0:00:08</td>
<td>231</td>
<td>0:00:04</td>
</tr>
<tr>
<td>3</td>
<td>360</td>
<td>0:04:54</td>
<td>387</td>
<td>0:00:11</td>
<td>368</td>
<td>0:00:04</td>
</tr>
<tr>
<td>4</td>
<td>672</td>
<td>0:08:46</td>
<td>702</td>
<td>0:00:25</td>
<td>702</td>
<td>0:00:05</td>
</tr>
<tr>
<td>5</td>
<td>201</td>
<td>0:11:08</td>
<td>238</td>
<td>0:00:12</td>
<td>231</td>
<td>0:00:04</td>
</tr>
<tr>
<td>6</td>
<td>469</td>
<td>0:27:08</td>
<td>489</td>
<td>0:00:12</td>
<td>472</td>
<td>0:00:05</td>
</tr>
<tr>
<td>7</td>
<td>355</td>
<td>0:29:45</td>
<td>383</td>
<td>0:00:20</td>
<td>357</td>
<td>0:00:05</td>
</tr>
<tr>
<td>8</td>
<td>468</td>
<td>1:09:42</td>
<td>471</td>
<td>0:00:30</td>
<td>471</td>
<td>0:00:06</td>
</tr>
<tr>
<td>9</td>
<td>746</td>
<td>6:32:23</td>
<td>785</td>
<td>0:00:17</td>
<td>785</td>
<td>0:00:04</td>
</tr>
<tr>
<td>Av</td>
<td>408</td>
<td>1:58:42</td>
<td>436</td>
<td>0:00:18</td>
<td>428</td>
<td>0:00:05</td>
</tr>
</tbody>
</table>

Running time is reported in the format of h:min:s

6.2 Analysis of Results

Fig. 11. show trend of fitness value trend for exact method, GA, and proposed SA. In all problems, proposed SA produce fitness value better than GA. To compare exact method, GA, and proposed SA performances statistically, we use one-way analysis of variance (ANOVA) using MINITAB 17. Table 4 show fitness value LINGO, GA, and proposed SA isn’t different statistically.

Table 4. Analysis of variance results of fitness value

<table>
<thead>
<tr>
<th>Source</th>
<th>Methodologies</th>
<th>Error</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF</td>
<td>2</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>SS</td>
<td>3,656</td>
<td>978,910</td>
<td>982,566</td>
</tr>
<tr>
<td>MS</td>
<td>1,828</td>
<td>40,788</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P value</td>
<td>0.95</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 12 show trend of CPU time for exact method, GA, and proposed SA. In all problems, both algorithms produce CPU time much smaller than exact method. Table 5 show CPU time of both algorithms different statistically with exact method.
Acknowledgments

The authors would like to thank to Supply Chain Management Department, Institut Teknologi Harapan Bangsa.

References


