

Article

Hybrid Harmony Search-Simulated Annealing Algorithm for Location-Inventory-Routing Problem in Supply Chain Network Design with Defect and Non-Defect Items

Farahanim Misni ^{1,2}, Lai Soon Lee ^{1,3,*}  and Hsin-Vonn Seow ⁴ 

¹ Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; farahanim@ump.edu.my

² Centre for Mathematical Sciences, Universiti Malaysia Pahang, Lebuhraya Tun Razak, Gambang, Kuantan 26300, Pahang, Malaysia

³ Laboratory of Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia

⁴ Nottingham University Business School, University of Nottingham Malaysia, Semenyih 43500, Selangor, Malaysia; Hsin-Vonn.Seow@nottingham.edu.my

* Correspondence: lls@upm.edu.my

Received: 30 July 2020; Accepted: 19 September 2020; Published: 22 September 2020



Abstract: This paper considers a location-inventory-routing problem (LIRP) that integrates the strategic, tactical, and operational decision planning in supply chain network design. Both defect and non-defect items of returned products are considered in the cost of reverse logistics based on the economic production quantity model. The objective of the LIRP is to minimize the total cost of location-allocation of established depots, the cost of inventory, including production setup and holding cost, as well as the cost of travelled distance by the vehicles between the open depots and assigned customers. A Hybrid Harmony Search-Simulated Annealing (HS-SA) algorithm is proposed in this paper. This algorithm incorporates the dynamic values of harmony memory considering rate and pitch adjustment rate with the local optimization techniques to hybridize with the idea of probabilistic acceptance rule from simulated annealing, to avoid the local extreme points. Computational experiments on popular benchmark data sets show that the proposed hybrid HS-SA algorithm outperforms a standard HS and an improved HS for all cases.

Keywords: harmony search; simulated annealing; location-inventory-routing; reverse logistics; supply chain

1. Introduction

The need for designing networks in a supply chain has been attracting a lot of attention in recent years; a result of the booming development of industrial commerce. A practice in traditional commerce is that customers can return the unsatisfactory products they bought within a given certain period of time. This process is called the reverse logistics by which the products are returned for proper planning of their disposal [1]. Reverse logistics can also be defined as the process of moving goods for capturing value. Efficient reverse logistics can reduce cost especially transportation cost, shipping cost, and inventory cost. Besides that, it also improves the service by offering a fast response to the demands and returns from customers. By implementing reverse logistics in supply chain network design, the expectation of a gain in both profit and increased customer retention can be expected, benefiting both from the economic and social factor of the economy respectively [2].

Reverse logistics is necessary for various reasons; such as the increasing trend in customers' returns, product lifecycles, and demanding customers [3]. To maintain a high level of customer satisfaction and quickly respond to the complaints, some industries are willing to handle those returns on their own and thus strive to provide an efficient logistics system to accommodate this. Hence, one can argue that it is necessary for the reverse logistics network and logistics process to be integrated. Nowadays, a large number of organizations are experiencing high volumes of product returns. Managing product returns in reverse logistics depends on detailed planning, especially when the volume of return flow had increased considerably [4]. The returned products can be re-entered into the market to be resold either with or without some refurbish process depending on the condition of the products. Returned products can be categorized into two categories which are defect or non-defect products. The non-defect products will be re-sold in the next batch while the defect products will need to go through the process of refurbishing before re-entering the market [5].

In supply chain network design, besides facility location and vehicle routing, inventory control is also considered an important part of the whole process of optimization. Designing a distribution network consists of the three sub-problems which is essential to drive the key performance for the overall productivity and profitability. Looking at previous works from literature, most of the studies done focused on the integration of only two sub-problems such as location-routing problem (LRP), Inventory-Routing Problem (IRP), or Location-Inventory Problem (LIP). We found work on the integration of all three sub-problems, i.e., Location-Inventory-Routing Problem (LIRP) to be very limited especially those utilising the Economic Production Quantity (EPQ) model. There is a body of literature with regards to the EOQ model. The EOQ model determines the optimal order quantity a company should purchase from other companies to minimize inventory cost. On the other hand, the EPQ model determines the optimal production quantity for manufacturing. The main difference between these two models is that the EPQ model assumes the company will produce its own quantity, therefore the orders are available in an incremental manner while the products are being produced. While the EOQ model assumes the order quantity arrives complete and immediately after ordering, meaning that the parts are produced by another company and are ready to be shipped when the order is made.

Since the EPQ model is less researched despite being a practice of most companies in the industry, we propose the LIRP utilizes the EPQ model for defect and non-defect items in reverse logistics. The problem is solved by a hybrid harmony search-simulated annealing (HS-SA) algorithm. The proposed algorithm incorporates the dynamic values of harmony memory considering rate (HMCR) and pitch adjustment rate (PAR) with the local optimization techniques from a HS, to hybridize with the idea of probabilistic acceptance rule from the SA to deal with premature convergence. Computational experiments on the popular benchmark dataset of Perl, Gaskell, and Christofides show that the proposed algorithm outperformed a standard HS and an improved HS for all cases.

This study is the extension of the research from [6,7]. The idea of solving the vehicle routing problem (VRP) using a standard HS algorithm is first proposed in [6]. The problem is further extended into the LRP in [7]. The HS in [6] solved the VRP using four local search techniques with fixed values of HMCR and PAR. In [7], they introduced the dynamic values of HMCR and PAR to solve the LRP. In this paper, we extended the methodology in [7] to solve the LIRP by adding two popular local search methods: 2-opt and 3-opt, as the local optimization techniques. The proposed HS is further enhanced by hybridizing it with SA with the implementation of probabilistic acceptance rule during the improvisation procedure of the HS.

The remainders of the paper are organized as follows: the literature review of supply chain network design is presented in Section 2. The mathematical formulation of the LIRP with EPQ model is given in Section 3. Section 4 describes the framework and procedures of the proposed hybrid HS-SA algorithm. The computational experiments on three sets of popular benchmark datasets are performed, and the results are discussed in Section 5. Lastly, the conclusions and future direction are given in Section 6.

2. Literature Review

The warehouse LRP (WLRP) is first proposed by [8]. The problem is decomposed into three sub-problems: Multi-Depot Vehicle Routing Problem (MDVRP), Warehouse Location-Allocation Problem (WLAP), and Multi-Depot Routing Allocation Problem (MDRAP) and is solved using the heuristic approach. Some other studies divided the problem into two phases [9]. The first phase is a LAP and the VRP follows in the second phase. The problem is solved sequentially and iteratively by SA. Reference [10] discussed the capacitated LRP under disruption and implemented several approaches in problem-solving such as metaheuristic based on maximum-likelihood sampling method, route-allocation improvement, two-stage neighborhood search and SA.

A two-phase hybrid heuristic search approach for LRP is introduced in [11]. The problem is decomposed into LAP and VRP. For the construction of the location distribution, they proposed a Tabu Search (TS) to determine a good configuration of the facilities to be used. Meanwhile, an Ant Colony Algorithm (ACO) was used at the routing phase to obtain good routes for the given configuration. The combination of Particle Swarm Optimization (PSO) with several heuristic algorithms is called Hybrid PSO (HybPSO-LRP) and is introduced in [12]. They combined the Multiple Phase Neighborhood Search-Greedy Randomized Adaptive Search Procedure (MPNS-GRASP) algorithm, the Expanding Neighborhood Search (ENS) strategy, and a Path Relinking (PR) strategy into the PSO. With the same approach used in [12], reference [13] combined the same techniques with the Honey Bee Mating Optimization (HBMO). They solved the problem using a hybrid HBMO for LRP (HBMO-LRP).

We found several studies on the integrated IRP from literature. For example, reference [14] solved a multi-objective coordination in a supply chain with three objective functions: maximizing the firm's profit, minimizing vendor's routing, and maximizing the average service level of buyers. They determined the vehicle routing, the router assigned and the amount delivered for each buyer. A HS algorithm is proposed due to the complexity of the problem. Considering the environmental impact, authors in [15] proposed a green IRP to minimize the total cost of inventory and routing, vehicle fixed cost, and emissions that are determined by load, distance, speed, and vehicle characteristics. They employed an exact method which is a Branch-and-Cut Algorithm (BCA) to add user cuts to the model and used a commercial solver to solve the linear programming relaxation problem. Similar to [16], they analyzed the benefits of collaborative energy use (CO₂ emission) and perishability in IRP with uncertain demand. The authors minimized the cost of routing, based on fuel and wage cost, waste, driving time and inventory. The problem is solved by the ILOG-OPL development studio and CPLEX 12.6 optimization.

A new bi-objective mathematical model which takes into account the economic, social, and environmental issues in the distribution of perishable products with a specified expiration date in IRPs is considered in [17]. They focused on inventory and distribution costs for the first objective and then looked at social issues to shape the second objective by considering the rate of accidents for the vehicle and the number of expired products. The Torabi-Hassani method was used and included vehicle noise emission as a constraint. The integrated Production, Inventory, and Distribution Routing Problem (PIDRP) was developed and solved heuristically with two solutions approach [18]. In the first phase, they solved the Mixed-Integer Program (MIP) with CPLEX MIP solver and in the second phase, they solved an associated consolidation problem to handle the potential inefficiency of direct shipments involving routing and inventories by Extended Optimal Partitioning (EOP) procedure. This approach can simultaneously coordinate the production, inventory, and transportation operation in PIDRP. The two-step procedure is getting more attention in solving IRP, researchers in [19] performed a comparative analysis of a series of heuristics for manufacturing supply chain with a MIP formulation and two-step procedure development. The first step is to estimate the daily delivery quantities and followed by solving a VRP in the second step. A previously developed branch-and-price algorithm is used and the production decision and inventory flow balance were taken into account in the model.

Besides IRP, the integrated LIP is also gaining popularity. An optimization model for perishable products with fuzzy capacity and carbon emission constraints for integrated LIP is proposed by [20].

They formulated a Mixed-Integer Nonlinear Programming (MINP) and is solved by using two methods which are Hybrid Genetic Algorithm (HGA) and Hybrid HS (HHS). They found that HHS gave a better solution but needed higher computing time when compared to the HGA. The LIP under vendor-managed inventory was considered by [21]. They solved the multi-objective LIP model based on the Non-dominated Sorting Genetic Algorithm (NSGA-II). This method gave promising solutions when compared with the well-known multi-objective evolutionary algorithm. By using the same approach as in [21] but hybridized with the Multi-objective Particle Swarm Optimization (MOPSO), a bi-dynamic closed-loop LIP under facility disruption risk is proposed in [22]. They considered the effectiveness of returned products in a period and retailer demand in the next period. The bi-level programming problem where the upper level is for determining the appropriate location of 3rd checking sites and the lower level is to present the coordinated inventory replenishment is studied by [23]. They considered the location and inventory policies in the supply chain when product returns are allowed.

Other related study of LIP can be found in [24]. The authors focused on three decisions which are the determination of depot locations, the allocation of flows, and the shipment sizes. Nonlinear continuous programming is formulated and an iterative heuristic is developed to estimate the depot flows, to solve the linear program, and to improve the flow depot. Meanwhile, researchers in [25] explored the continuous non-linear formulation for LIP with uncertain demand that integrates location, allocation, and inventory decisions. They solved the linear program using a heuristic algorithm and used the solution to improve the variable estimators for the next iteration. A closed-loop LIP is considered by [26] that determines which depot and remanufacturing centre need to be opened. An exact two-phase Lagrangian relaxation algorithm is developed and a mixed-integer nonlinear location-allocation model is formulated. Extension to the study in [26], an equivalent formulation with fewer nonlinear terms in the objective function is proposed in [27]. They used piecewise linearization to transform and solve it using CPLEX. Then the solution in CPLEX is compared with two previously published Lagrangian relaxation-based heuristic.

With respect to the LIRP, some of the researchers focus only on the non-defect items in reverse logistics such as in [28]. They solved the LIRP of the e-supply chain environment by using a Hybrid Genetic Simulated Annealing Algorithm (HGSAA) and compared it to a standard GA. Reference [29] used a Pseudo-Parallel Genetic Algorithm Integrating Simulated Annealing (PPGSA) to solve stochastic LIRP with continuous review inventory policy. Reverse logistics considering both defect and non-defect items of returned products has also been studied by [30]. They considered both quality of returned products in the e-commerce supply chain system. They found that the Hybrid Ant Colony Optimization algorithm (HACO) was considerably efficient and effective compared to a standard ACO. New TS (NTS) has been proposed by [31] to simultaneously integrate the LIRP and considered the minimization of manufacturing goods for new products and remanufacturing goods for damaged products in the objective function.

Based on the literature, we believe that each of the decision planning stages in the supply chain has equal significance. Recently, many industries aim to minimize the cost of suppliers by producing their own products. As the industries nowadays are trying to be greener and at the same time handle the reverse logistics effectively, they prefer to deal directly with the customers from manufacturing stage until the process of recovering the returns products. The EPQ model is gaining attention to be practiced among industries, especially among manufacturing and remanufacturing companies. With this in mind, the LIRP with EPQ model in reverse logistics is considered in this study to give new insights and solutions to the manufacturing industries in optimizing the production and inventory.

3. Mathematical Formulation of LIRP with EPQ Model

Given a set of the potential depots with fixed locations and customers with known demands and returns, the LIRP with EPQ model in reverse logistics is designed to determine the optimal location and number of open depots, the production quantity produced at each open depot as well as the routing of the assigned customers with known demands and returns to the open depots. The depots

will function as a distributor as well as a collector. At the depot, the condition of the returned products will be inspected and classified into two categories: defect and non-defect items. The production setup cost will be affected where the non-defect items re-enters into the market, while the defect items will need to be counted in the process of production planning. The objective of the LIRP is to minimize the cost of establishing facilities, cost of inventory including production setup and holding, and cost of distance travelled by vehicles. Figure 1 shows an example of LIRP in the supply chain with three depots and nine customers. It can be seen from Figure 1 that customers 1, 2, and 3 are served by depot 1, while the remaining 6 customers are served by depot 2 using two vehicles. Depot 3 remains closed.

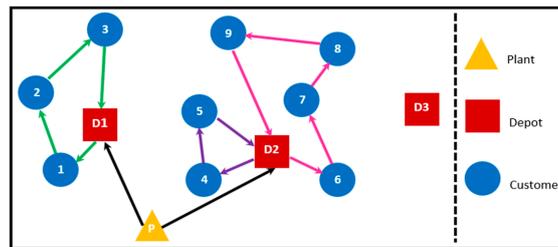


Figure 1. Example of a LIRP with 3 depots and 9 customers.

The mathematical formulation of LIRP with the EPQ model is given as follows. The objective and constraints are given below:

Sets:

- I = set of all depots ($i = 1, 2, \dots, I$).
- J = set of all customers ($j = 1, 2, \dots, J$).
- K = set of all vehicles ($k = 1, 2, \dots, K$).
- U = set of depots and customers ($u = 1, 2, \dots, I + J$).

Input Parameters:

- D_j = demand of customer j .
- r_j = non-defect items returned by customer j .
- s_j = defect items returned by customer j .
- R_j = returned products by customer j ($R_j = r_j + s_j$).
- d_{ij} = distance from i to j .
- VC_k = capacity of vehicle k .
- N = number of customers.
- F_i = fixed operating cost of depot i .
- DC = distance cost per mile.
- VD_i = maximum through at the depot i .
- KC = setup cost per production.
- h = holding cost per unit inventory.
- P = production rate.
- TL_{ijk} = total load of customer j at depot i by vehicle k .

Decision Variables:

$$z_i = \begin{cases} 1, & \text{if depot } i \text{ is open} \\ 0, & \text{otherwise} \end{cases},$$

$$y_{ij} = \begin{cases} 1, & \text{if depot } i \text{ is assigned to customer } j \\ 0, & \text{otherwise} \end{cases},$$

$$x_{ijk} = \begin{cases} 1, & \text{if arc } (i, j) \text{ is traveled by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

Q_i = optimal number of production quantity for each depot i ,

U_{lk} = auxiliary variable for sub-tour elimination constraint in vehicle k of customer l .

$$\min z = \sum_{i \in I} F_i z_i + \sum_{i \in U} \sum_{j \in U} \sum_{k \in K} DC \times d_{ij} x_{ijk} + KC \sum_{i \in I} \sum_{j \in J} \left[\frac{(D_j - r_j + s_j) y_{ij}}{Q_i} \right] + \frac{h}{2P} \sum_{i \in I} \left[Q_i \left(P - \sum_{j \in J} (D_j + r_j + s_j) y_{ij} \right) \right] \tag{1}$$

subject to:

$$\sum_{k \in K} \sum_{i \in U} x_{ijk} = 1, \forall j \in J \tag{2}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1, \forall k \in K \tag{3}$$

$$\sum_{i \in U} \sum_{j \in J} D_j x_{ijk} \leq VC_k, \forall k \in K \tag{4}$$

$$\sum_{i \in U} \sum_{j \in J} (r_j + s_j) x_{ijk} \leq VC_k, \forall k \in K \tag{5}$$

$$TL_{ijk} - (D_j - (r_j + s_j)) x_{ijk} \leq VC_k, \forall i \in I, \forall j \in J, \forall k \in K \tag{6}$$

$$\sum_{i \in U} x_{ijk} - \sum_{j \in U} x_{jik} = 0, \forall i \in I, \forall k \in K \tag{7}$$

$$\sum_{j \in J} D_j y_{ij} \leq VD_i z_i, \forall i \in I \tag{8}$$

$$U_{lk} - U_{jk} + N x_{ijk} \leq N - 1, \forall l, j \in J, \forall k \in K \tag{9}$$

$$\sum_{i \in U} x_{ilk} + x_{ujk} - y_{ij} \leq 1, \forall i \in I, \forall j \in J, \forall k \in K \tag{10}$$

$$y_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J \tag{11}$$

$$x_{ijk} \in \{0, 1\}, \forall i \in I, \forall j \in J, \forall k \in K \tag{12}$$

$$z_i \in \{0, 1\}, \forall i \in I \tag{13}$$

$$U_{lk} \geq 0, \forall l \in J, \forall k \in K \tag{14}$$

$$Q_i \geq 0, \forall i \in I. \tag{15}$$

The objective function of LIRP (Equation (1)) is to minimize the total fixed operating cost of depots, the total distance travelled cost by the vehicles, and the total cost of production setup and inventory holding. Constraints in Equations (2) and (3) indicate that each of the customers must be assigned in a single route and it can be served by only one vehicle. The total demands and total returns at each route cannot exceed the vehicle capacity limit and the total load on the vehicle at any arc must not exceed the vehicle capacity. These constraints are shown in Equations (4)–(6), respectively. Equation (7) states that the vehicle must start and end at the same depot. Besides the vehicle capacity limit, the capacity constraint for the depot is given in Equation (8). Equation (9) represents the new sub tour elimination constraint and Equation (10) specified that the customer will be assigned to the depot if there is a route from the depot. The binary values on the decision variable and the positive values for the auxiliary variable are defined in Equations (11)–(15), respectively.

Based on the objective function given in Equation (1), the total cost of inventory (TCI) consists of production setup cost and holding inventory cost. The optimal value of Q_i can be obtained by deriving the first derivative of the cost function in Equation (16) with respect to Q_i as follows:

TCI = Production setup cost + Inventory holding cost

$$TCI = KC \sum_{i \in I} \sum_{j \in J} \left[\frac{(D_j - r_j + s_j)y_{ij}}{Q_i} \right] + \frac{h}{2P} \sum_{i \in I} \left[Q_i \left(P - \sum_{j \in J} (D_j + r_j + s_j)y_{ij} \right) \right]. \tag{16}$$

$$\frac{\partial TCI}{\partial Q_i} = -KC \sum_{i \in I} \sum_{j \in J} \left[\frac{(D_j - r_j + s_j)y_{ij}}{Q_i^2} \right] + \frac{h}{2P} \sum_{i \in I} \left[\left(P - \sum_{j \in J} (D_j + r_j + s_j)y_{ij} \right) \right]. \tag{17}$$

Let $\frac{\partial TCI}{\partial Q_i} = 0$, hence

$$-KC \sum_{i \in I} \sum_{j \in J} \left[\frac{(D_j - r_j + s_j)y_{ij}}{Q_i^2} \right] + \frac{h}{2P} \sum_{i \in I} \left[\left(P - \sum_{j \in J} (D_j + r_j + s_j)y_{ij} \right) \right] = 0. \tag{18}$$

$$Q_i = \sqrt{\frac{2P(KC) \sum_{i \in I} \sum_{j \in J} (D_j - r_j + s_j)y_{ij}}{h \sum_{i \in I} (P - \sum_{j \in J} (D_j + r_j + s_j)y_{ij})}}. \tag{19}$$

The cost function of TCI is a convex function where the second derivative of the function is always non-negative (Equation (20)).

$$\frac{\partial^2 TCI}{\partial Q_i^2} = KC \sum_{i \in I} \sum_{j \in J} \left[\frac{(D_j - r_j + s_j)y_{ij}}{Q_i^3} \right] \geq 0. \tag{20}$$

From the objective function in Equation (1), we determine three decision planning as follows:

- (1) the number of open depots and assigned customers at each open depot.
- (2) the routing of vehicles delivering demands from the depot to the customers.
- (3) the optimal number of production quantity of each open depot.

We solve the problem iteratively where the allocation of customers to each depot has been assigned at the first stage with the possible number of open depots. At this stage, the vehicle capacity limit constraint has been relaxed. At the second stage, the process of minimizing location and routing with considering vehicle capacity constraint is done. To obtain the cost of inventory as stated in the objective function, the Q_i is calculated by the Equation (19) where the formulation is depending on the decision variable of y_{ij} and the total demand and returned from each customer at each depot. In other words, the inventory cost will be calculated when the decision variable for location and routing are found throughout the process of improvisation at each iteration. The process of minimizing the overall cost depends on the total operating depots, delivery routing, and inventory planning.

4. Hybrid Harmony Search-Simulated Annealing Algorithm

4.1. Harmony Search

Harmony Search (HS) is a population-based metaheuristic algorithm that mimics the music improvisation of a group orchestra. The musicians will improvise their harmonies with these three options: (1) select any pitch in the recorded harmony memory; (2) select and fix any previous pitch in memory; or (3) search a new music harmonisation within the range of the pitch. The process of obtaining the optimal solution is similar to this efficient search for a perfect state of harmony [32,33]. There are five main steps in the proposed HS:

Step 1: Initialisation

In HS, several solutions are generated to produce an initial population. The solutions can be generated either randomly or by the heuristics. A random initial population called harmony memory (HM) is created and sorted according to their fitness values. The number of solutions in the HM is the harmony memory size (HMS). An HM can be described in the following matrix:

$$HM = \left(\begin{array}{cccc|c} x_1^1 & x_2^1 & \dots & x_n^1 & f(x^1) \\ x_1^2 & x_2^2 & \dots & x_n^2 & f(x^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_n^{HMS} & f(x^{HMS}) \end{array} \right), \tag{21}$$

where,

x_i^j = decision variable of $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, HMS$,
 $f(x^j)$ = fitness function of x^j for $j = 1, 2, \dots, HMS$.

Step 2: Parameter Setting

Harmony memory considering rate (HMCR) and pitch adjustment rate (PAR) are two main parameters used in the HS algorithm. An appropriate value of HMCR will lead to the choices of good solutions as an element of new solutions. The convergence will be slow if the value of HMCR is too high. Some potential good solutions will not be well explored. But if it is too low, only a few good solutions are selected for the next iteration. Therefore, to use memory effectively, the value of HMCR should be between 0.7 and 0.95 [34]. The speed of convergence in a standard HS algorithm may be slow if the value of PAR is too small. Besides, a high value of PAR is also not recommended. It can cause solutions to be stuck around a few potential optimal and easily get trapped in local optima. Due to this, [34] stated that the value of PAR is suggested to be between 0.1 and 0.5. However, this range may be suitable and valid for some problems only.

To balance the exploration and exploitation, instead of using a fixed value, in this paper, the proposed HS uses a dynamic value of $HMCR_{it}$ and PAR_{it} introduced by [35]. The reason for reducing the $HMCR_{it}$ slowly is to increase the probability of exploring more solution space, not in the HM. Hence, the global optimal can be attained [36]. The dynamic value of $HMCR_{it}$ and PAR_{it} can avoid the solutions getting trapped in the local optimum quickly as well. With a slight modification, the proposed $HMCR_{it}$ and PAR_{it} are reduced gradually when there is no improvement found in the solution. In this paper, the range values of $[HMCR_{min}, HMCR_{max}]$ and $[PAR_{min}, PAR_{max}]$ are set to be $[0.7, 0.95]$ and $[0.3, 0.9]$, respectively. Equations (23) and (24) below are the formulation of dynamic $HMCR_{it}$ and PAR_{it} :

$$HMCR_{it} = HMCR_{max} - \frac{(HMCR_{max} - HMCR_{min}) \times it}{MaxIt} \tag{22}$$

$$PAR_{it} = PAR_{max} - \frac{(PAR_{max} - PAR_{min}) \times it}{MaxIt}, \tag{23}$$

where,

- it = current iteration,
- $MaxIt$ = maximum number of iterations,
- $HMCR_{max}$ = maximum value of HMCR,
- $HMCR_{min}$ = minimum value of HMCR,
- PAR_{max} = maximum value of PAR,
- PAR_{min} = minimum value of PAR.

Step 3: Improvisation

In the proposed HS, to increase the speed of convergence, multi solutions are generated at each iteration and called HM_{new} . Each newly generated solution is taken from the HM randomly with the probability of HMCR, otherwise, it will be generated randomly within the range of the harmony vector. To further enhance the intensification of the method, the proposed HS implements the local optimization in three different ways: within a depot, between two depots, and within the vehicle route. Hence, five multi-neighbourhood local search techniques are proposed:

- (1) **Swap**: exchange the position of two random customers within the same route or between two different routes. Swapping can be done within a depot or between the depots.
- (2) **Insertion**: insert a customer in between two other random customers within the same depot.
- (3) **Relocation**: relocate the customers from the current depot to a new depot.
- (4) **2-Opt**: swapping two customers in the same vehicle route and reverse the substring between the swapped customers.
- (5) **3-Opt**: deleting three edges of the customers in the same vehicle route and create another three sub tours in three possible ways: non-reversing substring, with one reversing substring, or with two reversing substring.

The local optimization is applied to the new solution with the probability of PAR_{it} and the choice of multi-neighbourhood local search technique is randomly selected with equal probability. As mentioned earlier, we extended the methodology in [6] by adding two popular local search methods: 2-opt and 3-opt, as the local optimization techniques. The examples of these five techniques are graphically shown in Figure 2a–f.

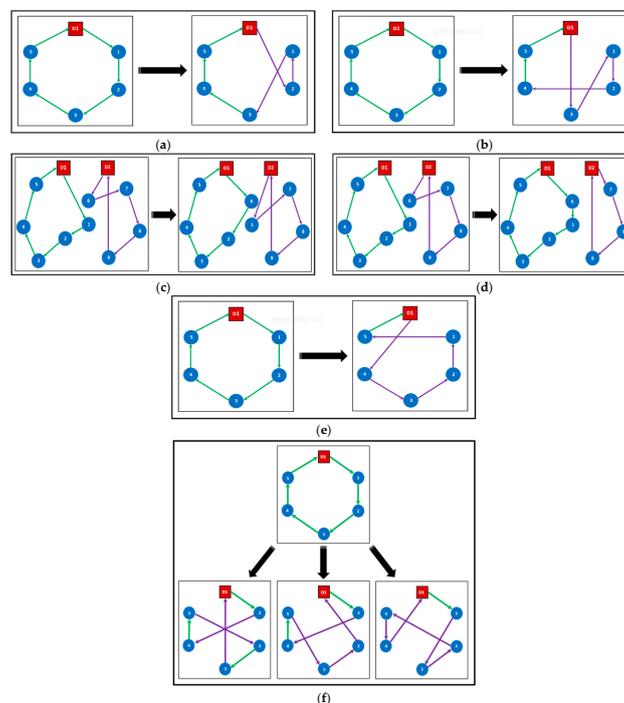


Figure 2. The figures show how each of the multi-neighbourhood search techniques finds their neighborhood: (a) Swap within depot: customer 1 and 2 are swapped within the same depot; (b) Insertion: customer 3 are inserted in between depot and customer 1 in the route sequence; (c) Swap between depots: customer 6 in depot 1 is swapped with customer 1 in depot 2; (d) Relocation: customer 6 in depot 2 is moved to depot 1; (e) 2-Opt: swap customer 1 and 4 in the same route and the route sequence between swapped customers are reversed; (f) 3-Opt: three possibilities of 3-Opt which are either non-reversing, one reversing or two reversing substring.

Step 4: Update HM

To update the solutions in HM, the HM_{new} created at each iteration is combined with current HM. The solutions are sorted according to the value of the objective function. The best solutions with the size of HMS will be kept for the next iteration.

Step 5: Termination

Repeat *Step 3* and *4*. The proposed algorithm will be terminated when either the *MaxIt* is reached or *t* consecutive non-improving iterations are performed.

4.2. Hybrid Harmony Search—Simulated Annealing

To prevent the possibility of the HM population been trapped into local optima due to premature convergency, the proposed HS algorithm is further enhanced by hybridizing a SA with the implementation of probabilistic acceptance rule based on the Boltzmann factor, $e^{-\frac{\Delta}{T_{it}}}$ during the improvisation procedure of the HS. In SA, the worst solution will be accepted within a probability and allows the search to proceed to its neighbourhood (Algorithm 1). By preserving the worst solution in a population, it provides practical randomness into the search to avoid the local extreme points. The probability of acceptance in SA is depending on the value of annealing temperature and the difference in the value of an objective function. At each iteration, the annealing temperature will be reduced according to the value of the cooling rate, which is a function of temperature. The higher value in the cooling rate, the slower the temperature reduction [37]. The procedures of probabilistic acceptance rule in SA described in Algorithm 1 are of a standard SA that has been adapted into the proposed hybrid HS-SA as shown in Algorithm 2.

Algorithm 1. SA Algorithm.

```

1: initialize parameter,  $T_0$  = initial temperature,  $\alpha$  = cooling rate;
2: generate initial solution,  $x$ ;
3: calculate objective function,  $f(x)$ ;
4:  $it = 0$ ;
5: while ( $it < MaxIt$ )
6:     generate new solution  $x'$  in the neighbourhood of  $x$ 
7:     if  $f(x') < f(x)$ 
8:          $x \leftarrow x'$ 
9:     else
10:         $\Delta = f(x') - f(x)$ 
11:        if  $rand < \exp\left(\frac{-\Delta}{T_{it}}\right)$ 
12:             $x \leftarrow x'$ 
14:         $T_{it+1} = \alpha \times T_{it}$ 
16:     $it = it + 1$ 
17: end while

```

Algorithm 2. Hybrid HS-SA Algorithm.

```

1: begin
2: define the parameter setting:  $HMCR_{max}$ ,  $HMCR_{min}$ ,  $PAR_{max}$ ,  $PAR_{min}$ ,  $HMS$ ,  $T_0$ ,  $\alpha$ ,  $MaxIt$ ,  $t$ 
3: generate solution vectors and calculate fitness function for the initial population of HM.
4: while (( $it < MaxIt$ ) OR ( $non\_improving < t$ )) do
5:     while (no. of harmony vectors  $< HM_{new}$ ) do
6:         do location-allocation and route allocation
7:         calculate  $HMCR_{it}$  and  $PAR_{it}$ 
8:         if  $rand < HMCR_{it}$ 
9:             select solution vectors from the HM randomly
10:            if  $rand < PAR_{it}$ 
11:                implement one of the multi-neighbourhood search techniques randomly to
                create new neighbourhood solution,  $x'$ 
12:                if  $f(x') < f(x)$ 
13:                     $x \leftarrow x'$ 
14:                else
15:                     $\Delta = f(x') - f(x)$ 
16:                    if  $rand < \exp\left(\frac{-\Delta}{T_{it}}\right)$ 
17:                         $x \leftarrow x'$ 
18:                     $T_{it+1} = \alpha \times T_{it}$ 
19:                else
20:                    generate the new solution from the vector harmony space.
21:                do inventory planning
22:                update HM: sort and combine HM and  $HM_{new}$ 
23:            end while
24:        update the best solution vectors of HM
25:    end while
26:    return the best solution

```

5. Computational Experiments and Discussion

In this section, computational experiments are performed to illustrate the performance of the proposed hybrid HS-SA compared with other approaches in the literature. The proposed algorithm is coded in MATLAB software R2017b on a laptop computer with 1.60 GHz Intel® Core™ i5-4200U CPU with 8 GB of RAM. The parameters configuration of the proposed algorithm is provided in Table 1. The test problem instances are three popular benchmark datasets of Perl, Gaskell, and Christofides with additional simulated data of returned products and production rate. The additional data were generated randomly by using a uniform distribution. The production rate is set to be greater than the total demand and returns, $P > \sum_{j \in I} D_j + r_j + s_j$ while the returned products, R_j are generated by the uniform distribution of $R_j \sim U(0, D_j)$. We set 70% of the returned products as the non-defects (r_j) items while the remaining are the defect items (s_j). The characteristics of the dataset are given in Table 2.

Table 1. Parameters Setting for SHS, PHS, and HS-SA.

Parameter	SHS	PHS	HS-SA
HMS	300	300	300
HMCR	0.85	[0.7, 0.95]	[0.7, 0.95]
PAR	0.55	[0.3, 0.9]	[0.3, 0.9]
MaxIt	10,000	5000	5000
t	500	100	100
T_0	–	–	30
α	–	–	0.98

Table 2. Characteristics of Perl, Gaskell, and Christofides dataset.

	No. of Depot	No. of Customer	Depot Capacity	Vehicle Capacity
Perl 1	2	12	280	140
Perl 2	15	55	550	120
Perl 3	7	85	850	160
Gaskell 1	5	21	15,000	6000
Gaskell 2	5	22	15,000	4500
Gaskell 3	5	29	15,000	4500
Gaskell 4	5	32	35,000	8000
Gaskell 5	5	36	15,000	250
Christofides 1	5	50	10,000	160
Christofides 2	10	75	10,000	140
Christofides 3	10	100	10,000	200

To solve the LIRP problem, the location and allocation of the depots and customers are solved first. In the location-allocation problem, each of the customers is initially assigned to the nearest depots based on the Euclidean distance formulation. The customers will be moved around to the possible depots during the process of improvisation. The vehicle capacity limit constraint has been relaxed to minimize the number of open depots and it is assumed that only one vehicle is being used at each depot. However, the depot capacity limit should not be violated during the process. Then, the allocation of customers at each depot needs to be sequenced and divided into vehicles according to the vehicle capacity limit. This process is performed among the open depots only. The process of local optimization is performed within the open depots. Both depot capacity limits and vehicle capacity limits should not be violated. Finally, the inventory part is included during the process of minimizing the cost.

Since there is no study on the LIRP that utilizes the EPQ model for defect and non-defect items in reverse logistics from the literature, in order to validate the performance of the proposed hybrid HS-SA algorithm, the LRP will be solved first and compared with other heuristics and metaheuristics found in the literature that used the same benchmark datasets. Tables 3–5 present the comparative results of the three benchmark datasets on the proposed hybrid HS-SA with the heuristic method (HR) of [8], simulated annealing in [9] (SA-W) and [10] (SA-Z), hybrid tabu search and ant colony (TACO) from [11], particle swarm optimization (PSO) with different variants of particle swarm optimization which are: standard PSO, PSO with MPNS-GRASP (PMG), PSO with MPNS-GRASP and ENS (PMGE) and hybrid PSO (HLRP) in [12] and honey bee mating optimization (HBMO) in [13]. A standard HS (SHS) and the proposed HS (PHS) from [7] are also included for comparison in all experiments. For each problem instances, the proposed algorithm as well as SHS and PHS are performed for 10 independent runs. In each table, note that the numerical results of the selected algorithms are extracted from the original papers found in the literature and the best result of each problem instances is highlighted in bold. Figures 3–5 illustrate the results of Tables 3–5 in a bar chart respectively.

Table 3. Comparative results of Perl dataset for LRP.

Algorithm	Perl 1		Perl 2		Perl 3	
	Mean	Best	Mean	Best	Mean	Best
HR	–	204.0	–	1146.9	–	1657.6
SA-W	–	204.0	–	1119.8	–	1656.7
SA-Z	–	204.0	–	1115.4	–	1642.0
TACO	–	204.0	–	1139.9	–	1642.6
PSO	–	204.0	–	1135.9	–	1656.9
PMG	–	204.0	–	1136.2	–	1656.9
PMGE	–	204.0	–	1135.9	–	1656.9
HLRP	–	204.0	–	1135.9	–	1657.1
HBMO	–	204.0	–	1127.1	–	1655.2
SHS	204.0	204.0	1161.4	1158.8	1846.3	1832.1
PHS	204.0	204.0	1115.6	1113.2	1666.0	1653.0
HS-SA	204.0	204.0	1114.7	1113.2	1658.4	1645.0

Table 4. Comparative results of Gaskell dataset for LRP.

Algorithm	Gaskell 1		Gaskell 2		Gaskell 3		Gaskell 4		Gaskell 5	
	Mean	Best								
SA-Z	–	429.6	–	585.1	–	523.2	–	577.6	–	471.3
PSO	–	437.1	–	592.1	–	572.1	–	574.1	–	470.7
PMG	–	435.9	–	591.8	–	572.1	–	571.7	–	470.7
PMGE	–	435.9	–	591.7	–	572.1	–	571.7	–	470.7
HLRP	–	432.9	–	588.5	–	572.1	–	570.8	–	470.7
HBMO	–	431.9	–	587.2	–	572.1	–	569.8	–	470.7
SHS	449.1	443.7	602.9	598.0	550.6	541.2	578.7	576.2	528.5	520.7
PHS	437.9	431.7	584.3	577.9	522.7	511.8	562.6	559.9	475.5	464.1
HS-SA	429.3	424.9	577.3	575.2	502.5	493.1	559.5	552.4	473.5	464.1

Table 5. Comparative results of Christofides dataset for LRP.

Algorithm	Christofides 1		Christofides 2		Christofides 3	
	Mean	Best	Mean	Best	Mean	Best
SA-Z	–	612.6	–	914.0	–	967.6
PSO	–	582.7	–	888.9	–	895.7
PMG	–	582.7	–	887.1	–	893.2
PMGE	–	582.7	–	886.9	–	891.5
HLRP	–	582.7	–	886.3	–	889.4
HBMO	–	582.7	–	886.3	–	889.4
SHS	606.0	602.9	961.2	953.3	981.6	977.6
PHS	591.2	587.7	889.3	886.7	894.6	891.6
HS-SA	577.8	573.6	888.2	885.0	891.9	885.5

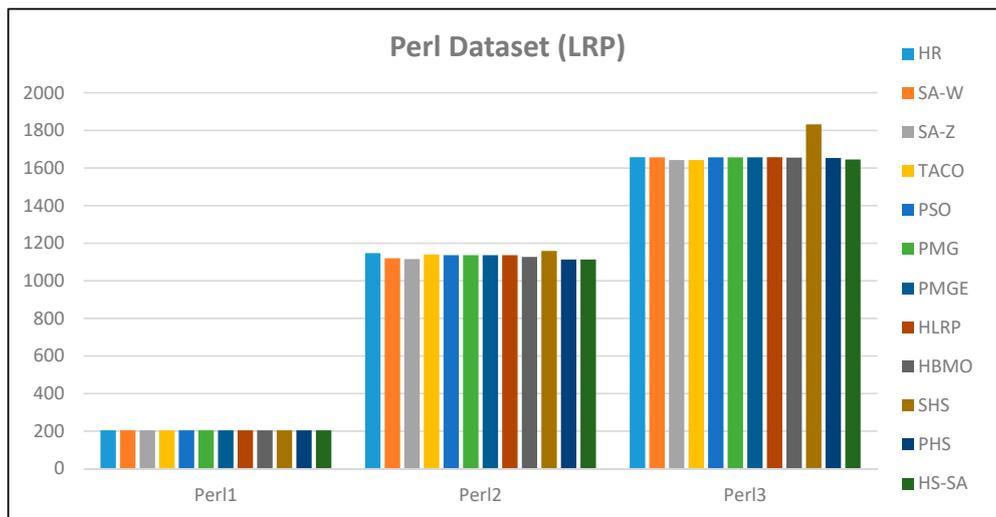


Figure 3. Best result of Perl dataset for LRP.

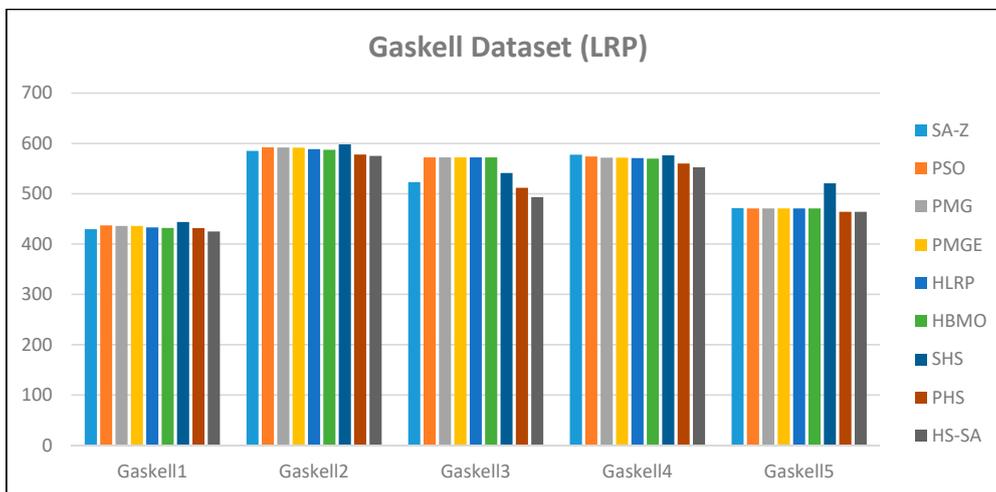


Figure 4. Best result of Gaskell dataset for LRP.

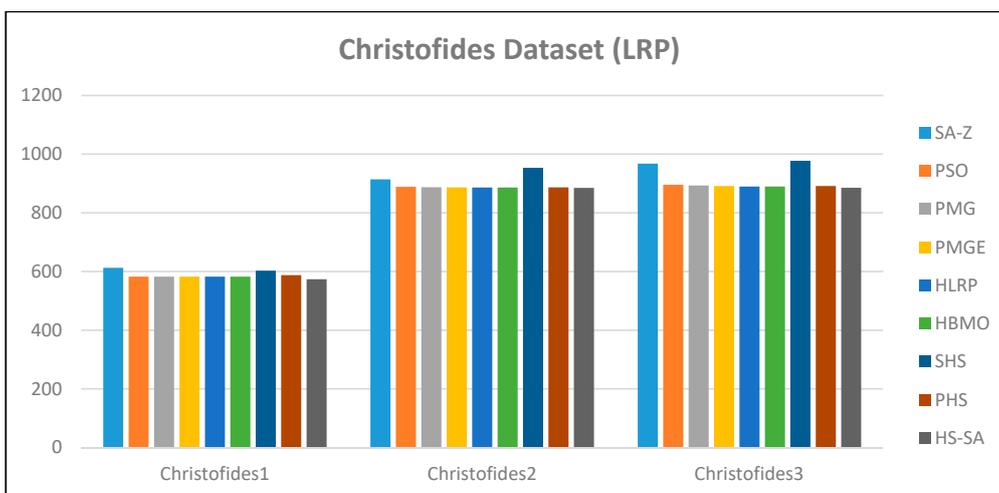


Figure 5. Best result of Christofides dataset for LRP.

As compared to the SHS and PHS in [7], the hybrid HS-SA performed significantly better for all instances in Perl, Gaskell, and Christofides, except Perl 1. However, for Perl 3, results in SA-Z [10] and TACO [11] are slightly better than the hybrid HS-SA algorithm. All algorithms managed to find the optimal solution in Perl 1 since the number of customers and depots are small. The SHS did not perform well for medium and large size. This indicates that the SHS should be improved and modified to get better solutions [7]. From the numerical experiments in LRP, hybrid HS-SA has successfully solved the problem with comparable results. Therefore, the proposed algorithm will be used to solve the LIRP that utilizes the EPQ model for defect and non-defect items in reverse logistics.

The computational results for all test instances in LIRP are shown in Tables 6–8. Since there is no benchmark for the EPQ model in LIRP, the comparison is assessed between the hybrid HS-SA algorithm with SHS and PHS only. All algorithms are performed for 10 independent runs and the mean, standard deviation, coefficient of variation (C.V.), and the best result of each algorithm are reported in column 3–6 respectively. The Wilcoxon signed-rank test with $\alpha = 0.05$ is performed to test the significance of the results. As the Wilcoxon signed-rank test does not assume normality in the data, it can be used when this assumption has been violated and the use of the dependent *t*-test is inappropriate. The *p*-value of the test is reported in the last column of the tables. The best result of each problem instances is highlighted in bold. In addition, the best result of the all problem instances are summarized as a bar chart in Figure 6.

Table 6. Comparative results of Perl dataset in LIRP for SHS, PHS, and HS-SA.

Instance	Algorithm	Mean	Std. Dev.	C.V.	Best	<i>p</i> -Value
Perl 1	SHS	263.27	5.99×10^{-14}	2.28×10^{-16}	263.3	1.000
	PHS	263.27	5.99×10^{-14}	2.28×10^{-16}	263.3	1.000
	HS-SA	263.27	5.99×10^{-14}	2.28×10^{-16}	263.3	–
Perl 2	SHS	1693.18	3.2646	0.0019	1689.5	0.005
	PHS	1646.25	2.1141	0.0013	1643.9	0.005
	HS-SA	1485.27	1.9309	0.0013	1482.4	–
Perl 3	SHS	2271.26	8.9040	0.0039	2257.7	0.005
	PHS	2099.91	5.5006	0.0026	2096.2	0.005
	HS-SA	2082.90	4.5094	0.0022	2078.0	–

Table 7. Comparative results of Gaskell dataset in LIRP for SHS, PHS, and HS-SA.

Instance	Algorithm	Mean	Std. Dev.	C.V.	Best	<i>p</i> -Value
Gaskell 1	SHS	1592.13	7.1444	0.0045	1586.2	0.005
	PHS	1585.23	4.9927	0.0031	1578.5	0.028
	HS-SA	1579.26	3.7438	0.0024	1575.7	–
Gaskell 2	SHS	1042.76	3.7271	0.0036	1039.0	0.005
	PHS	1009.80	1.9408	0.0019	1008.9	0.022
	HS-SA	1007.32	1.8396	0.0018	1006.1	–
Gaskell 3	SHS	829.58	5.9608	0.0072	820.2	0.005
	PHS	797.70	5.6214	0.0070	790.3	0.013
	HS-SA	789.51	5.2295	0.0066	780.9	–
Gaskell 4	SHS	1032.74	3.3067	0.0032	1030.6	0.005
	PHS	1021.37	2.6579	0.0026	1018.1	0.028
	HS-SA	1018.49	1.5920	0.0016	1017.0	–
Gaskell 5	SHS	610.80	6.4625	0.0106	604.7	0.005
	PHS	602.50	3.9302	0.0065	599.2	0.005
	HS-SA	571.79	3.6303	0.0063	566.9	–

Table 8. Comparative results of Gaskell dataset in LIRP for SHS, PHS, and HS-SA.

Instance	Algorithm	Mean	Std. Dev.	C.V.	Best	<i>p</i> -Value
Christofides 1	SHS	848.40	2.8983	0.0034	845.4	0.005
	PHS	834.99	2.4324	0.0029	833.6	0.005
	HS-SA	819.76	2.2342	0.0027	817.9	–
Christofides 2	SHS	1220.87	4.4823	0.0037	1214.3	0.005
	PHS	1151.87	2.3561	0.0020	1148.2	0.007
	HS-SA	1149.48	2.2828	0.0020	1146.1	–
Christofides 3	SHS	1384.60	3.2345	0.0023	1381.0	0.005
	PHS	1349.85	2.9744	0.0022	1347.2	0.005
	HS-SA	1312.49	2.6949	0.0020	1307.2	–

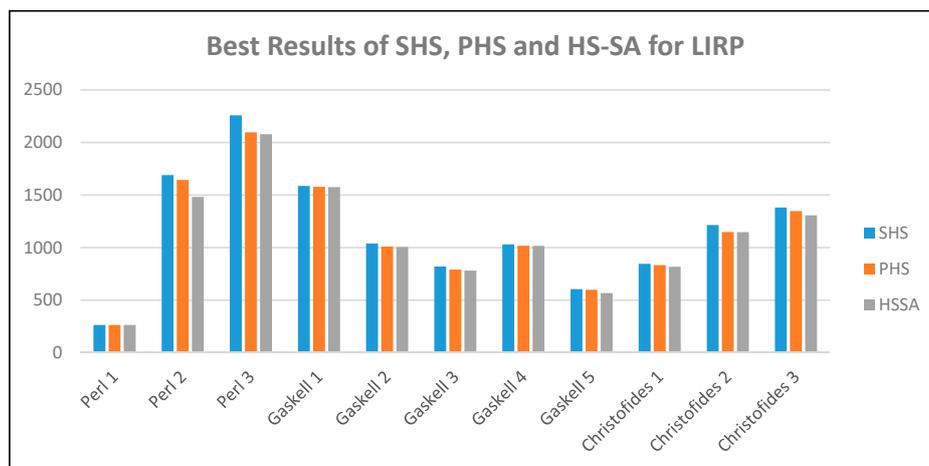


Figure 6. Best result of Perl, Gaskell, and Christofides dataset in LIRP for SHS, PHS, and HS-SA.

The Wilcoxon signed-rank test elicits a statistical significant difference in the performance of finding the minimum cost of the LIRP between the hybrid HS-SA and the SHS and PHS across all the datasets (*p*-value < 0.005) based on the difference in the median cost. These tests are used to show that the results are significant, supporting that the proposed algorithm will produce results that are different from the results of the SHS and PHS. Apart from the Perl 1 instance where the proposed algorithm produced the same results as SHS and PHS, all other datasets showed *p*-values of less than 0.05. This means the results generated by the hybrid HS-SA is significant and different from the values from the SHA and PHS. So, the results from the proposed algorithm is significant from the other works, but is it better? From Tables 6–8, the hybrid HS-SA outperforms the solutions in SHS and PHS for all cases of datasets of Perl, Gaskell, and Christofides. For this, we use the C.V. values. The lower the value of the C.V., the more precise the estimate. As reported in Tables 6–8, all the C.V. values are low which supports the high predictiveness of the proposed algorithm.

Tables 9–11 present the detailed results of all problem instances produced by the proposed hybrid HS-SA for LIRP. The total cost for each problem instance is highlighted in bold. These tables served as the benchmark for the LIRP. For instance, Gaskell 1 required two depots with a fixed cost of 50 each to serve all the 21 customers with four vehicles (see, Table 10). In depot 1, with the inventory cost of 573.05, two vehicles are deployed to serve the customers (19→21→20→17) and (18→15→12→14→16) with costs of 59.45 and 86.90 respectively. While in depot 2, with the inventory cost of 577.74, two vehicles are used to serve the customers (9→7→5→2→1→6) and (8→3→4→11→13→10) with costs of 83.01 and 95.55 respectively. In this result, the best solution obtained suggest to close Depot 3–5. The overall minimum cost for Gaskell 1 is 1575.7. Figure 7 illustrate the topological layout of the Gaskell 1 produced by the proposed hybrid HS-SA for LIRP.

Table 9. Distribution of routes and costs for Perl dataset generated by HS-SA for LIRP.

Instance	Depot	Vehicle	Route	Cost			
				Depot	Distance	Inventory	Total
Perl 1	D1	V1	D1→9→8→6→1→2→3→7→D1	100	44.34	59.36	263.27
		V2	D1→10→12→11→5→4→D1		59.63		
	D5	V1	D5→38→54→39→55→43→34→D5	240	48.85	123.90	
		V2	D5→23→28→12→17→22→30→D5		47.56		
		V3	D5→8→3→7→31→29→19→D5		28.78		
		V4	D5→33→14→27→16→32→45→D5		52.26		
Perl 2	D10	V5	D10→13→1→2→42→4→9→D10	240	18.45	123.18	1482.43
		V6	D10→6→10→21→51→37→47→D10		56.42		
		V7	D10→44→46→40→52→50→53→D10		62.37		
		V8	D10→5→11→D10		3.41		
	D12	V9	D12→36→35→24→26→18→15→D12	240	40.02	122.15	
		V10	D12→41→20→49→48→25→D12		35.08		
Perl 3	D2	V1	D2→54→59→58→39→56→57→40→55→D2	372	69.27	171.63	
		V2	D2→62→61→65→66→68→69→27→16→D2		44.90		
		V3	D2→22→17→28→75→23→29→19→30→D2		43.98		
		V4	D2→63→14→70→71→12→73→74→72→D2		44.75		
		V5	D2→32→38→2→45→33→D2		26.34		
	D4	V6	D4→4→42→15→31→7→3→8→5→D4	372	37.06	140.58	2077.97
		V7	D4→44→64→80→52→46→43→34→1→D4		63.27		
		V8	D4→47→53→50→81→83→67→82→11→D4		51.73		
	D6	V9	D6→49→76→51→79→13→78→21→20→D6	372	53.65	120.73	
		V10	D6→24→37→77→10→60→9→6→41→D6		39.54		
		V11	D6→25→36→85→18→84→26→35→48→D6		54.54		

Table 10. Distribution of routes and costs for Gaskell dataset generated by HS-SA for LIRP.

Instance	Depot	Vehicle	Route	Cost			
				Depot	Distance	Inventory	Total
Gaskell 1	D1	V1	D1→19→21→20→17→D1	50	59.45	573.05	1575.70
		V2	D1→18→15→12→14→16→D1		86.90		
	D2	V3	D2→9→7→5→2→1→6→D2	50	83.01	577.74	
		V4	D2→8→3→4→11→13→10→D2		95.55		
Gaskell 2	D1	V1	D1→9→D1	50	41.23	430.91	1006.14
		V2	D1→10→13→11→6→1→2→3→16→15→14→17→22→20→19→18→12→D1		334.73		
		V3	D1→7→8→5→4→21→D1		149.27		
Gaskell 3	D3	V1	D3→2→D3	50	42.76	287.76	780.87
		V2	D3→21→14→8→9→17→7→13→16→15→12→11→10→23→18→19→20→D3		219.24		
		V3	D3→22→26→28→27→29→25→24→6→1→5→4→3→D3		181.11		
Gaskell 4	D3	V1	D3→26→27→16→28→29→D3	50	111.77	464.65	1017.0
		V2	D3→14→17→25→24→23→22→20→21→18→19→15→D3		140.46		
		V3	D3→11→5→6→7→8→9→10→32→13→1→D3		90.59		
		V4	D3→31→12→2→4→3→30→D3		159.53		
Gaskell 5	D5	V1	D5→22→28→29→30→36→35→34→33→27→D5	50	108.88	102.78	566.93
		V2	D5→21→20→26→32→31→25→19→13→14→D5		101.70		
		V3	D5→15→9→8→7→1→2→3→4→10→D5		97.50		
		V4	D5→16→17→11→5→6→12→18→24→23→D5		106.07		

Table 11. Distribution of routes and costs for Christofides dataset generated by HS-SA for LIRP.

Instance	Depot	Vehicle	Route	Cost			
				Depot	Distance	Inventory	Total
Christofides 1	D2	V1	D2→4→17→37→15→33→45→44→42→19→40→41→13→D2	40	119.68	130.14	817.86
		V2	D2→6→48→23→7→43→24→14→25→D2		98.12		
		V3	D2→18→46→27→32→11→38→5→12→47→D2		78.65		
	D5	V4	D5→1→8→26→31→28→3→36→35→20→22→D5	40	95.21	114.14	
		V5	D5→2→16→50→9→49→10→39→30→34→21→29→D5	40	101.92		
Christofides 2	D1	V1	D1→17→3→44→24→49→16→51→6→D1	40	85.59	143.31	1146.09
		V2	D1→8→35→14→11→53→7→D1		43.90		
		V3	D1→32→50→18→55→25→9→40→D1		98.72		
		V4	D1→19→59→66→65→38→10→58→D1		97.16		
		V5	D1→4→27→52→46→34→67→D1		34.29		
		V6	D1→30→48→74→28→62→73→63→33→2→68→75→D1		91.12		
		V7	D1→54→13→57→15→5→29→45→D1		75.40		
		V8	D1→26→72→31→39→12→D1		77.34		
	D5	V9	D5→69→71→60→70→20→37→36→D5	40	61.19	117.74	
		V10	D5→61→64→42→41→56→23→43→1→22→21→47→D5	40	140.33		
Christofides 3	D3	V1	D3→69→1→50→33→81→51→9→35→71→65→66→20→30→70→D3	40	110.56	154.43	1307.24
		V2	D3→88→62→19→11→64→49→36→47→46→8→82→48→7→52→D3		127.18		
		V3	D3→31→10→63→90→32→D3		51.83		
	D5	V4	D5→89→18→60→83→45→17→84→5→59→94→53→28→27→D5	40	97.34	136.43	
		V5	D5→95→92→37→91→44→14→38→86→16→61→93→99→96→6→D5	40	92.79		
		V6	D5→13→87→97→98→85→100→42→43→15→57→2→40→58→D5	40	95.14		
	D8	V7	D8→12→80→68→76→77→3→79→78→34→29→24→55→25→54→D8	40	95.56	130.85	
		V8	D8→4→39→67→23→56→75→41→22→74→72→73→21→26→D8		95.13		

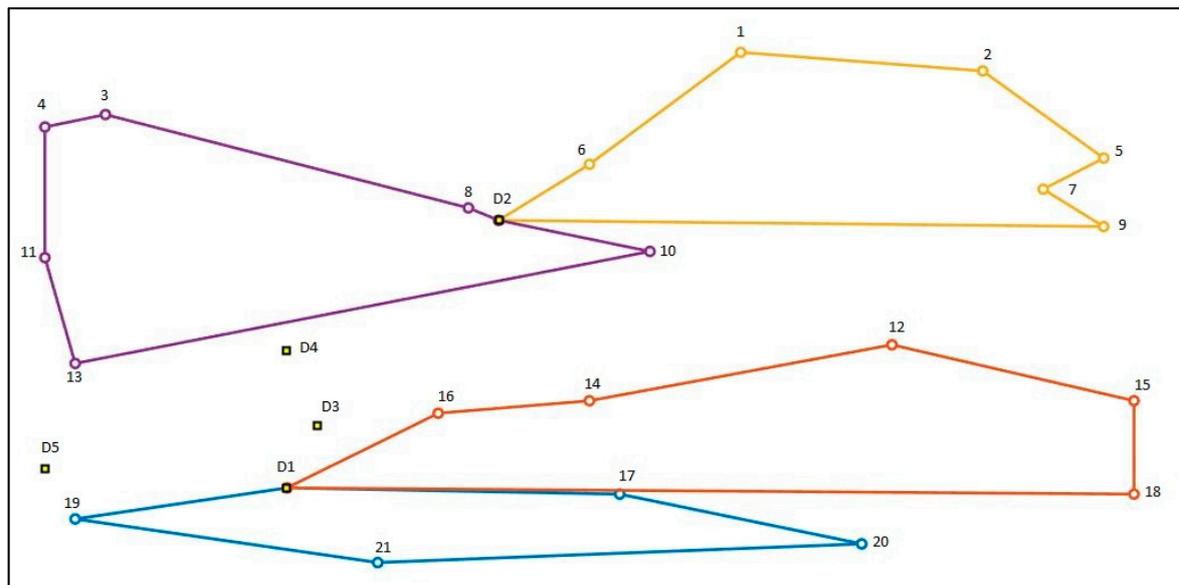


Figure 7. Topological layout of the Gaskell 1 produced by HS-SA for LIRP.

6. Conclusions and Future Work

The integration of facility location, inventory planning, and vehicle routing is one of the most challenging problems in the supply chain network design. In this study, the Economic Production Quantity (EPQ) model of reverse logistics that considers the returned products from the customers is included during the process of optimization. Since the problem is NP-hard, a hybrid population-based metaheuristic called hybrid Harmony Search-Simulated Annealing (HS-SA) algorithm is proposed. Computational experiments on three sets of popular benchmark instances show that the proposed hybrid HS-SA algorithm is successful in finding better solutions as compared to a standard HS, as well as other approaches from the literature of Location-Routing Problem (LRP) and Location-Inventory-Routing Problem (LIRP). For future work, the HS can be hybridized with other metaheuristic methods such as differential evolution, genetic algorithm, and tabu search to further enhanced the efficiency of the approach. Environmental issues such as the CO₂ emission and the use of electric vehicles can also be integrated into the model to be solved.

Author Contributions: Conceptualization, F.M. and L.S.L.; methodology, F.M. and L.S.L.; software, F.M.; validation, F.M., L.S.L., and H.-V.S.; formal analysis, F.M., L.S.L., and H.-V.S.; investigation, F.M. and L.S.L.; writing—original draft preparation, F.M. and L.S.L.; writing—review and editing, F.M., L.S.L., and H.-V.S.; supervision, L.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Universiti Putra Malaysia (UPM) through Geran Putra-Inisiatif Putra Siswazah (GP-IPS/2017/9579400).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prahinski, C.; Kocabasoglu, C. Empirical research opportunities in reverse supply chains. *Omega* **2006**, *34*, 519–532. [[CrossRef](#)]
2. Tibben-Lembke, R.S. The impact of reverse logistics on the total cost of ownership. *J. Mark. Theory Pract.* **1998**, *6*, 51–60. [[CrossRef](#)]
3. Guide, V.D.R., Jr.; Souza, G.C.; Van Wassenhove, L.N.; Blackburn, J.D. Time value of commercial product returns. *Manag. Sci.* **2006**, *52*, 1200–1214. [[CrossRef](#)]
4. Daugherty, P.J.; Richey, R.G.; Genchev, S.E.; Chen, H. Reverse logistics: Superior performance through focused resource commitments to information technology. *Transp. Res. Part E Logist. Transp. Rev.* **2005**, *41*, 77–92. [[CrossRef](#)]

5. Petersen, J.A.; Kumar, V. Perceived risk, product returns, and optimal resource allocation: Evidence from a field experiment. *J. Mark. Res.* **2015**, *52*, 268–285. [[CrossRef](#)]
6. Misni, F.; Lee, L.S. Harmony search for multi-depot vehicle routing problem. *Malays. J. Math. Sci.* **2019**, *13*, 311–328.
7. Misni, F.; Lee, L.S. Harmony search algorithm for location-routing problem in supply chain network design. *ASM Sci. J.* **2019**, *12*, 48–56.
8. Perl, J.; Daskin, M.S. A warehouse location-routing problem. *Transp. Res. Part B Methodol.* **1985**, *19*, 381–396. [[CrossRef](#)]
9. Wu, T.H.; Low, C.; Bai, J.W. Heuristic solutions to multi-depot location-routing problems. *Comput. Oper. Res.* **2002**, *29*, 1393–1415. [[CrossRef](#)]
10. Zhang, Y.; Qi, M.; Lin, W.H.; Miao, L. A metaheuristic approach to the reliable location routing problem under disruptions. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *83*, 90–110. [[CrossRef](#)]
11. Wang, X.; Sun, X.; Fang, Y. A two-phase hybrid heuristic search approach to the location-routing problem. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 12 October 2005; Volume 4, pp. 3338–3343.
12. Marinakis, Y.; Marinaki, M. A particle swarm optimization algorithm with path relinking for the location routing problem. *J. Math. Model. Algorithms* **2008**, *7*, 59–78. [[CrossRef](#)]
13. Marinakis, Y.; Marinaki, M.; Matsatsinis, N. Honey bees mating optimization for the location routing problem. In Proceedings of the 2008 IEEE International Engineering Management Conference, Estoril, Portugal, 28–30 June 2008; pp. 1–5.
14. Alaei, S.; Setak, M. Multi objective coordination of a supply chain with routing and service level consideration. *Int. J. Prod. Econ.* **2015**, *167*, 271–281. [[CrossRef](#)]
15. Cheng, C.; Yang, P.; Qi, M.; Rousseau, L.M. Modeling a green inventory routing problem with a heterogeneous fleet. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *97*, 97–112. [[CrossRef](#)]
16. Soysal, M.; Bloemhof-Ruwaard, J.M.; Haijema, R.; van der Vorst, J.G. Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Comput. Oper. Res.* **2018**, *89*, 168–182. [[CrossRef](#)]
17. Rahimi, M.; Baboli, A.; Rezik, Y. Sustainable inventory routing problem for perishable products by considering reverse logistic. *IFAC Pap. Online* **2016**, *49*, 949–954. [[CrossRef](#)]
18. Lei, L.; Liu, S.; Ruszczyński, A.; Park, S. On the integrated production, inventory, and distribution routing problem. *IIE Trans.* **2006**, *38*, 955–970. [[CrossRef](#)]
19. Bard, J.F.; Nananukul, N. Heuristics for a multiperiod inventory routing problem with production decisions. *Comput. Ind. Eng.* **2009**, *57*, 713–723. [[CrossRef](#)]
20. Dai, Z.; Aqlan, F.; Zheng, X.; Gao, K. A location-inventory supply chain network model using two heuristic algorithms for perishable products with fuzzy constraints. *Comput. Ind. Eng.* **2018**, *119*, 338–352. [[CrossRef](#)]
21. Liao, S.H.; Hsieh, C.L.; Lai, P.J. An evolutionary approach for multi-objective optimization of the integrated location-inventory distribution network problem in vendor-managed inventory. *Expert Syst. Appl.* **2011**, *38*, 6768–6776. [[CrossRef](#)]
22. Asl-Najafi, J.; Zahiri, B.; Bozorgi-Amiri, A.; Taheri-Moghaddam, A. A dynamic closed-loop location-inventory problem under disruption risk. *Comput. Ind. Eng.* **2015**, *90*, 414–428. [[CrossRef](#)]
23. Wang, Z.; Yao, D.Q.; Huang, P. A new location-inventory policy with reverse logistics applied to B2C e-markets of China. *Int. J. Prod. Econ.* **2007**, *107*, 350–363. [[CrossRef](#)]
24. Tancrez, J.S.; Lange, J.C.; Semal, P. A location-inventory model for large three-level supply chains. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 485–502. [[CrossRef](#)]
25. Puga, M.S.; Tancrez, J.S. A heuristic algorithm for solving large location-inventory problems with demand uncertainty. *Eur. J. Oper. Res.* **2017**, *259*, 413–423. [[CrossRef](#)]
26. Diabat, A.; Abdallah, T.; Henschel, A. A closed-loop location-inventory problem with spare parts consideration. *Comput. Oper. Res.* **2015**, *54*, 245–256. [[CrossRef](#)]
27. Diabat, A.; Theodorou, E. A location-inventory supply chain problem: Reformulation and piecewise linearization. *Comput. Ind. Eng.* **2015**, *90*, 381–389. [[CrossRef](#)]
28. Li, Y.; Guo, H.; Wang, L.; Fu, J. A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under E-supply chain environment. *Sci. World J.* **2013**, *2013*, 125893. [[CrossRef](#)]

29. Liu, B.; Chen, H.; Li, Y.; Liu, X. A pseudo-parallel genetic algorithm integrating simulated annealing for stochastic location-inventory-routing problem with consideration of returns in e-commerce. *Discret. Dyn. Nat. Soc.* **2015**, *2015*, 586581. [[CrossRef](#)]
30. Deng, S.; Li, Y.; Guo, H.; Liu, B. Solving a closed-loop location-inventory-routing problem with mixed quality defects returns in e-commerce by hybrid ant colony optimization algorithm. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 6467812. [[CrossRef](#)]
31. Yuchi, Q.; He, Z.; Yang, Z.; Wang, N. A location-inventory-routing problem in forward and reverse logistics network design. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 3475369. [[CrossRef](#)]
32. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
33. Lee, K.S.; Geem, Z.W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3902–3933. [[CrossRef](#)]
34. Yang, X.S. Harmony Search as a Metaheuristic Algorithm. In *Music-Inspired Harmony Search Algorithm: Theory and Applications*; Geem, Z.W., Ed.; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2009; Volume 191, pp. 1–14.
35. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [[CrossRef](#)]
36. Moh'd Alia, O.; Mandava, R. The variants of the harmony search algorithm: An overview. *Artif. Intell. Rev.* **2011**, *36*, 49–68. [[CrossRef](#)]
37. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).