Solving multi-product inventory ship routing with a heterogeneous fleet model using a hybrid cross entropy-genetic algorithm: a case study in Indonesia

Budi Santosa\textsuperscript{a}, Rita Damayanti\textsuperscript{a} and Biswajit Sarkar\textsuperscript{b}

\textsuperscript{a}Industrial Engineering, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo, Surabaya 60111, Indonesia; \textsuperscript{b}Department of Industrial & Management Engineering, Hanyang University, Erica Campus, Ansan, Republic of Korea

ABSTRACT
This paper presents a model and an algorithm for an inventory ship routing problem (ISRP). It consists of two main parts: a model development of the ship routing problem in a multi-product inventory with a heterogeneous fleet and an algorithm development to solve the problem. The problem is referred to as ISRP. ISRP considers several parameters including the deadweight tonnage (DWT), product compatibility, port setup, and compartment washing costs. Considering these parameters, the objective function is to minimize the total cost, which consists of traveling, port setup, ship charter, and compartment washing costs. From the resulting model, there are two major steps used to solve the problem. The first is to select the ships in order to satisfy the constraint that restricts the mooring rule. The second is to find the best route, product allocation, and shipped quantity. ISRP is an Non Polynomial-hard problem. Finding the solution of such problem needs a high computation time. A new hybrid metaheuristics, namely the cross entropy-genetic algorithm (CEGA), was proposed to solve ISRP. The results were then compared with those resulted from a hybrid Tabu Search to measure the hybrid CEGA performance. The results showed that CEGA provided better solutions than those produced by the hybrid Tabu Search.

1. Introduction
Product shipments by sea play an important role in global trade. It is estimated about 65–85\% of the global trade were delivered by sea (Andersson, Duesund, & Fagerholt, 2011). Additionally, Hvattum, Fagerholt, and Armentano (2009) also stated that sea transport is the main means of delivery in international trade. Profit of product delivery by sea can be evaluated in terms of quantity and price. Shipping by sea is able to send items in large quantities. It is also reported that the product delivery through this pathway requires the least expensive cost compared to other modes of transportation (Chopra & Meindl, 2007). But the shipping cost can be higher if the delivery is not managed properly. Therefore, we need a proper procedure such that the cost of shipping products by sea can be minimized.
One of the problems faced in the delivery of products by sea is an inventory ship routing problem (ISRP). ISRP arises from the concept of vendor-managed inventory (VMI) that makes a manufacturer or a supplier responsible for all decisions related to the product inventory on the buyer side. ISRP integrates the inventory management problems at each port and determines the route of the ship (Siswanto, Essam, & Sarker, 2011). Problems of inventory levels are limited by needs in each consumption port to ensure that there is no stock out during a specified planning period. Furthermore, a routing problem is a problem of determining the sequence of ships from a depot to the consumption ports, which results in minimum transportation costs. When the routing process has been performed, it should be followed by the selection of ships because not all ports can be visited by any ship. This condition applies because each port has a specific capacity. The capacity should be adjusted to the ship’s DWT (Al-Khayyal & Hwang, 2007). In addition, the product compatibility also needs to be considered during the loading on to the ship (Rusdiansyah & Nurminarsih, 2012) as all products typically cannot be placed in one ship. It must also be taken into account the products that have been loaded beforehand. If the product to be loaded is different from the previous ones, the compartment of the ship must be washed first (Hvattum et al., 2009). Thus, the loading activity should consider the washing cost while attempting to minimize total cost.

Multi-product ISRP with a heterogeneous fleet is an NP-hard problem. The problem-solving process using an exact method requires a long computation time, especially when the problem size is large. In such cases, better methods are required. In this paper, we proposed a model of single depot multi product ISRP with a heterogeneous fleet; considering the product compatibility, DWT of the ships, port setup, and compartment washing costs. To solve the problem, we proposed a hybrid cross entropy-genetic algorithm (CEGA). The case study was taken from an Indonesian oil state-owned company. Cross entropy (CE) was selected due to its proven capacity to solve NP-hard problems (de Boer, Kroese, Mannor, & Rubinstein, 2005) and considerably easy to apply in combinatorial problems (Laguna, Duarte, & Marti, 2007). However, CE needs longer computation time if it stands alone. Thus, we hybridized the CE with genetic algorithm (GA) to reduce the computation time (Santosa, Budiman, & Wiratno, 2011). Using a feature of GA, a new sample can be obtained quickly through a mutation mechanism. GA is a search method based on the principle of natural selection and genetics (Sastry, Goldberg, & Kendall, 2005). A study has shown that hybrid GA was well capable of solving inventory ship routing product (Ho, Ho, Ji, & Lau, 2008). To show how well the proposed hybrid works, we also presented a comparison of our solutions with those resulted from a hybrid Tabu Search.

The rest of the paper is organized as follows. Section 2 presents the problem definition, Section 3 describes the literature review, Section 4 discusses the mathematical model of the problem, and Section 5 explains the proposed algorithm, CEGA. Section 6 describes the experiments, results and analysis that are finally concluded in Section 7.

2. Problem definition

The problems we try to solve in this paper include:

1. Develop a model of the multi-product ISRP for a heterogeneous fleet with a single loading port that takes into account the weight of the ship mooring at the port,
compatibility of products, port setup cost, and washing compartment cost to produce a solution with a minimum cost. The total cost includes traveling, port setup, ship charter, and compartment washing costs;

(2) Develop a hybrid CEGA for solving the multi-product ISRP with a heterogeneous fleet and to compare the performance of this algorithm with a hybrid Tabu Search. In addition, when determining the ship to moor in the port, we consider DWT factors such that there is no stock out or excess inventory. From the model we expect to find solutions, which include the optimal route, ships selected, product allocation to the ships, and quantity shipped to the given demand points.

The novel idea we proposed in the model was to include the DWT of ship, port setup and compartment washing costs. In addition, the size of the problem is quite big so that it is not efficient to use an exact method. We also developed a new algorithm, CEGA, to solve the model.

3. Literature review

3.1. Inventory ship routing problem

One of the problems arising from the concept of VMI is the ISRP. With the VMI, a supplier is responsible for product inventory in the retailer/consumer (Chopra & Meindl, 2007). The supplier has the authority to decide the order quantity to be delivered to consumers acting as a buyer. This shipping decision is based on the information about stock levels on the buyer by following a predetermined replenishment time. The main characteristic of VMI strategy is shorter replenishment time, frequency and delivery on time so that it will lead to the optimization of production and transportation planning. VMI can benefit both supplier and buyer sides. Supplier side profits will increase because suppliers can quickly determine the need for raw materials and can decide for themselves when to send. While on the buyer side, buyers will not experience delays and always have the goods in the warehouse. ISRP is included in the inventory routing problem that integrates inventory management and vehicle routing problem. An inventory level is limited by the need for stock at each consumption port to ensure that there is no stock out.

Some studies have been performed widely in order to solve ISRP. Al-Khayyal and Hwang (2007) studied the multi-commodity liquid bulk routing and scheduling problem using a mixed integer linear programming (MILP). The objective is to minimize the operational total cost, which consists of travel and setup costs for loading/unloading with a dedicated compartment. The method they used can only solve small size problems.

In larger problems, Rusdiansyah and Nurminarsih (2012) solved the multi product tanker scheduling problem by considering product loading compatibility using a Tabu Search and a swap procedure to minimize the total cost that includes port charge, management fee, bunker consumption, bunker consumption discharge and intrinsic cost. Ho et al. (2008) developed a hybrid GA for the multi-depot Vehicle Routing Problem combined with the Clarke and Wright’s saving method and nearest neighbor to minimize the traveled distance and time to serve all customers. A ship inventory routing and scheduling problem with undedicated compartments was discussed in Siswanto et al. (2011) where the model aimed to find the minimum cost solution while satisfying a number of technical and physical
constraints within a given planning horizon. The problem was broken down into four sub-problems that need to be solved simultaneously: route selection, ship selection, loading, and unloading activity procedures. Siswanto et al. (2011) used a MILP to model the problem. In obtaining the solution, they used a one-step greedy heuristic, and then based on this heuristic, a set of heuristics was proposed. A number of problem instances were used to compare the solutions of the two approaches. Differently, Rusdiansyah and Nurminarsih (2012) considered product compatibility and assumed no setup and compartment washing costs. In a different case, a problem of a short sea fuel oil distribution, where the routing and scheduling of ships between ports such that the demand for various fuel oil products can be satisfied during the planning horizon was solved using hybrid heuristics (Agra, Christiansen, Delgado, & Hvattum, 2015). The approach employed three heuristic procedures: rolling horizon, local branching and feasibility pump to solve the problem of small size.

In a recent research, Lee and Kim (2015) introduced an industrial ship routing problem faced by one of the world-class steel manufacturing companies. The problem was to determine how to route a fleet of heterogeneous ships to carry the cargoes, given a set of cargoes with pickup, delivery ports, and time windows. The cargoes can be distributed across multiple ships if their time window is not violated. A ship can handle multiple cargoes within a route. The fleet of heterogeneous ships consists of two types of ships: company-owned ships and tramp ships. A company-owned ship can deliver cargoes from multiple supply ports to multiple delivery ports within a route, while a tramp ship can only deliver a cargo directly from a supply port to a delivery port within a route. To reduce operation cost, the two types of the ships should be well-coordinated and utilized. A mixed integer programming model and an adaptive large neighborhood search-based heuristic were proposed to solve the problem.

Hemmati, Stålhane, Hvattum, and Andersson (2015) discussed a VMI service application in tramp shipping. A two-phase heuristics was proposed to determine routes and schedules for a shipping company. The heuristic first converted inventories into cargoes, thus turning the problem into a classic ship routing and scheduling problem. It then used an adaptive large neighborhood search to solve the resulting cargo routing and scheduling problem. The heuristic iteratively changed the cargoes generated to handle the customer’s inventories, based on the information obtained from an initial solution. Computational results were presented, discussed, and compared with exact solutions on large realistic instances. The results revealed the potential savings from converting traditional contracts of affreightment to an integrated VMI service. The factors that influence the benefits obtainable through VMI were also analyzed. Figure 1 shows the summary of previous research, which directly influences this paper in terms of cases and methods.

### 3.2 Cross entropy-genetic algorithm

CE is a relatively new method. The method was inspired by a concept of modern information theory, namely the concept of Kullback–Leibler distance (Rubinstein & Kroese, 2004). CE was originally applied in the simulation of rare events, and then developed for several cases such as combinatorial optimization, continuous optimization, machine learning, and some other problems. CE is included in the family of Monte Carlo techniques that can be used to solve the case of estimation and optimization. In this estimation, the CE provided
an adaptive way to find the optimal sampling distribution for some fairly wide-ranging problems (Rubinstein & Kroese, 2004).

In further developments, it was found that a simple modification of the CE method can be used not only to estimate the probability of rare events, but also to solve complex combinatorial optimization problem by minimizing the CE. Combinatorial optimization is used to represent optimization problems whose solution is a combination of many possibilities. CE involves an iterative procedure, where each iteration can be split into two phases:

1. Generate random samples \((x)\) by using a mechanism or a specific distribution. In a combinatorial problem, \(x\) might be routes, trajectories or sequences of doing jobs. While in a continuous optimization problem, \(x\) could be a vector of solutions.

2. Update the parameters \((\nu)\) of the random mechanism based on elite sample data to produce better samples in the next iteration. In a combinatorial problem, \(\nu\) might be matrix of probability. In a continuous optimization problem, \(\nu\) might be the mean and standard deviation.

Elite samples are a percentage of the whole samples that we choose to update the parameter \(\nu\). CE in principle uses elite sample data to determine new parameters that will be used to generate a new set of population that is closer to a solution. An algorithm settlement with CE can be written as follows: (Rubinstein & Kroese, 2004; Santos & Willy, 2011)

1. Determine the values of \(N\) (number of samples), parameters \(\nu_0\).
2. Generate a sample of \(N\) with a specific mechanism, utilizing \(\nu_0\) parameters.
3. Evaluate this sample by inserting into the objective function. Then sort the objective function value. Select \(\rho N\) samples with smallest objective function values.
4. Renew \(\nu\) adaptively. For \(\nu\) that is already updated, use it to generate new \(x\) values. Updating the parameter vector \(\nu\) can be done by the following formula:

\[
\nu_i = \omega \nu + (1 - \alpha) \nu_{i-1}
\]
where
\( \omega_t \) average of parameters in elite samples in current iteration
\( \alpha \) a smoothing parameter (0 < \( \alpha \) < 1)
\( \nu_{t-1} \) parameter value in the previous iteration

Santosa and Hardiansyah (2010) used CE method for solving a generalized orienteering problem. CE solved the problem by finding the optimum route and maximizing the total score of visiting some predetermined places. CE produced better results compared to those of the Harmony Search and Artificial Neural Network. In an m-Machines No-Wait job-shop scheduling problem, a hybrid CEGA successfully solved the problem by minimizing the completion time of each job (Santosa et al., 2011).

4. Mathematical model

The development of the proposed model follows the models developed by Rusdiansyah and Nurminarsih (2012) and Siswanto et al. (2011). The main difference between this model and that of Rusdiansyah and Nurminarsih is that our model assumes a heterogeneous fleet and every fleet cannot lean back in all ports. We also consider a compartment washing port setup. Moreover, the difference of the proposed model with the model of Siswanto et al. (2011) is the use of single depot and product compatibility consideration where there are products that cannot be placed adjacent to other products.

The case study was taken from an Indonesian oil state-owned company. The model assumes that the products are always available at the loading port during the planning horizon. The consumption rate at each port and the speed of the ship are constant. The unloading process at the port can be done by more than one ship simultaneously and unloading process of second products and so on at the same port and ship can be done immediately after the previous process is completed without having to wait. The unloading process will continue until completed although the operations hour has ended.

The compartment is undedicated; ship can carry any type of product, but only single type at a time. The ships used are rent ships with time charters thus the system is not influenced by the amount of product the ships carry, but based on the time and duration of the charter. In one trip, the ship could visit more than one unloading port when the capacity of the ship is greater than the amount of cargo that must be delivered in a single port. When all the unloading ports have sufficient inventory until the end of the planning horizon, the ship will return to the loading port. Loading port can receive ships with any tonnage and open for 24 h. Unloading port consumes more than one product with a specified consumption rate. Unloading port has a limited operational time (time windows) called daylight. Both loading and unloading ports can serve multiple ships at the same time.

The proposed mathematical model can be explained as follows.

**Decision variables**

\( c_{ijv} \) transportation cost from node \( i \) to \( j \) by ship \( v \)

\( cp_{iv} \) setup cost in port \( i \) for ship \( v \)

\( crv_{v} \) ship \( v \) rent cost
cc \(_{vc}\) compartment (c) washing cost in ship (v)

\(x_{imnv}\) 1 if the ship moves from unloading port (i, m) to the unloading port (j, n)

\(x_{imdnv}\) 1 if the ship moves from unloading port (i, m) to the loading port (d, n)

\(y_{im}\) 1 if position (i, m) is not reachable

\(r_{ik}\) 1 if the stock level of all unloading port (i) was able to meet the consumption needs

\(u_{inv}\) 1 if the DWT of the ships (v) is smaller than unloading port (i) capacity

\(z_{vc}\) 1 if compartment(c) of ship (v) is washed before loading the next cargo

\(by_{dkncv}\) 1 if there is product \((k)\) in compartment \((c)\) of ship \((v)\) that loaded in loading port \((d, m)\)

\(x_{vimnk}\) 1 if product \((k)\) is brought by the ship \((v)\) in the m arrival in port (i)

\(I_{imvk}\) the amount of products \((k)\) that ship \((v)\) bring in compartment \((c)\) after leave unloading port \((i, m)\)

\(q_{imvkc}\) the number of products \((k)\) unloaded in port \((i, m)\) by ship \((v)\)

\(q_{dmvkc}\) the number of products \((k)\) loaded in port \((d, m)\) by ship \((v)\)

\(p_{inv}\) 1 if ship \((v)\) visit unloading port \((i, m)\) and do the port setup

\(p_{dmv}\) 1 if ship \((v)\) visit loading port \((d, m)\) and do the port setup

\(ta_{im}\) starting of processing time in unloading port \((i, m)\)

\(ta_{dm}\) the arrival time of the ship at loading port \((d, m)\)

\(te_{im}\) ending Service time in unloading port \((i, m)\)

\(te_{dm}\) ending Service time in loading port \((d, m)\)

\(s_{imk}\) product \((k)\) stock levels at unloading port \((i, m)\)

\(DWT_{v}\) ship \((v)\) DWT

\(DWT_{i}\) unloading capacity in port \((i)\)

\(TC_{vc}\) compartment \((c)\) washing time in ship \((v)\)

\(k_{ik}\) product \((k)\) that cannot be load in the same ship with another product

\(J_{ik}\) 1 if product \((k)\) being load/unload in port \((i)\)

\(QS_{ik}\) shipping quantity of product \((k)\) in port \((i)\)

\(CMax_{vc}\) maximum capacity of compartment \((c)\) in ship \((v)\)

\(PH\) planning horizon

\(TP\) port setup time

\(TQu_{ik}\) unloading speed of product \((k)\) in port \((i)\)

\(TQi_{dk}\) loading speed of product \((k)\) in port \((d)\)

\(TT\) traveling time

\([A_i, B_i]\) unloading time windows in port \((i)\)

\(H_{ik}\) initial inventory in loading port \((i)\) of product \((k)\)

\(SMin_{ik}\) minimum inventory level of product \((k)\) in port \((i)\)

\(SMax_{ik}\) maximum inventory level of product \((k)\) in port \((i)\)

\(CR_{ik}\) consumption rate of product \((k)\) in port \((i)\)

\(H_T\) set of all ports

\(H_{iv}\) set of all unloading ports, where \(H_{iv} \in H_T\)

\(N\) set of all \((i, m)\) where \(i \subseteq H_{iv}\) and \(m = 1, \ldots, M_i\)

\(U\) set of \((d, m)\) where \(d\) is loading port, \(d \in H_T\) and \(m = 1, \ldots, M_i\)

\(A_v\) set of all arcs that possible for the ship \((v)\)
Objective function

Minimize

\[
\sum_{v \in V} \sum_{(i,m,n) \in A_v} c_{ijv}x_{imjnv} \\
+ \sum_{v \in V} \sum_{(d,m,n) \in A_v} c_{dv}(x_{dmjnv} + x_{imdvn}) \\
+ \sum_{v \in V} c_{pv} \left( \sum_{(i,m) \in N} p_{inv} + \sum_{(d,m) \in U} p_{dmv} \right) \\
+ \sum_{v \in V} crv \sum_{(d,m) \in U} (ta_{dm} - ta_{d(m-1)}) \\
+ \sum_{v \in V} \sum_{c \in C} cc_{vc}z_{vc}
\]

Constraints

\[
\sum_{(d,m) \in U} \sum_{(j,n) \in N} x_{jmdnv} - \sum_{(d,m) \in U} \sum_{(j,n) \in N} x_{dmjnv} \cdot u_{iv} - r_{ik} = 0, u_{iv} = \begin{cases} 1, & DWT_v \leq DWT_i \\ 0, & DWT_v > DWT_i \end{cases} \\
\]

\[
r_{ik} = \begin{cases} 1, & s_{imk} + q_{invkc} \geq CR_{ik}(PH - te_{im}) \ , \forall v \in V, \forall k \in K, \forall (v,d,m) \in VxU, \forall (v,i,m, \ldots) \\ 0, & s_{imk} + q_{invkc} < CR_{ik}(PH - te_{im}) \ , \forall v \in V, \forall (v,d,m) \in VxU, \forall (v,i,m, \ldots) \end{cases}
\]

\[
\sum_{(i,m) \in N} \sum_{(d,n) \in U} x_{imdv} = \begin{cases} 1, & l_{invkc} = 0, \forall v \in V, \forall (v,d,n) \in VxU, \forall (v,i,m) \in VxN \\ 0, & l_{invkc} > 0, \forall v \in V, \forall (v,d,n) \in VxU, \forall (v,i,m) \in VxN \end{cases}
\]

\[
\sum_{(j,n) \in N} x_{jimnv} - \sum_{(j,n) \in N} x_{imjnv} \cdot u_{iv} = 0, \forall (v,i,m) \in VxN, i \neq j
\]

\[
\sum_{v \in V} \sum_{(j,n) \in N} x_{jimnv} + y_{im} = 1, \forall (i,m) \in N, i \neq j
\]

\[
y_{im} - y_{i(m-1)} \geq 0, \forall (i,m) \in N, \ m \neq 1
\]

\[
y_{dm} - y_{d(m-1)} \geq 0, \forall (d,m) \in U, \ m \neq 1
\]

\[
x_{dmjnv} \in \{0, 1\}, \forall v \in V, \forall (d,m,j,n) \in A_v
\]

\[
x_{imjnv} \in \{0, 1\}, \forall v \in V, \forall (i,m,j,n) \in A_v
\]

\[
u_{iv}, r_{ik} \in \{0, 1\}, \forall k \in K, \forall v \in V, \forall i \in H_{Tv}
\]
\[ y_{im} \in \{0, 1\}, \forall (i, m) \in N \]  

\[
z_v = \begin{cases} 
1, & by_{dmvkc} - by_{d(m+1)kc_v} = 0, \forall v \in V, \forall (d, m) \in U, \forall k_v \in K_v, \forall (k, c) \in K_v C_v, k_v \neq k \\
0, & by_{dmvkc} - by_{d(m+1)kc_v} \neq 0 
\end{cases}
\]  

\[
\sum_{k \in K} by_{dmvkc} \leq 1, \forall v \in V, \forall (d, m) \in U, \forall (k, c) \in K_v C_v
\]  

\[
\sum_{c \in C_v} by_{dmvkc} \leq |C_v| \left(1 - by_{dmvkc_k_v}\right), \forall v \in V, \forall k \in K_v, \forall (d, m) \in U, \forall c \in C_v, \forall k_v \in K_v, \forall (k, c) \in K_v
\]  

\[ z_v \in \{0, 1\}, \forall v \in V, \forall c \in C_v \]  

\[ by_{dmvkc} \in \{0, 1\}, \forall v \in V, \forall (d, m) \in U, \forall (k, c) \in K_v C_v \]  

\[
\sum_{v \in V} xy_{imkv} = 1, \forall k \in K_v, \forall (i, m) \in N
\]  

\[
x_{imv} \left(I_{inv} - J_{akq_{inv}c} - I_{jnvkc}\right) = 0, \forall v \in V, \forall (i, m, j, n) \in N, \forall (k, c) \in K_v C_v, i \neq j
\]  

\[
x_{dmv} \left(J_{akq_{dmv}c} - I_{jnvkc}\right) = 0, \forall v \in V, \forall (d, m) \in U, \forall (j, n) \in N, \forall (k, c) \in K_v C_v, j \neq d
\]  

\[
\sum_{c \in C_v} \sum_{k \in K_v} q_{dmvkc} = \sum_{(i,m) \in N} \sum_{k \in K_v} xy_{imkv} QS_{ik} + \left(\sum_{c \in C_v} CMax_{vc} - \sum_{(i,m) \in N} \sum_{k \in K_v} xy_{imkv} QS_{ik}\right), \forall v \in V, \forall (d, m) \in U
\]  

\[
q_{dmvkc} \leq CMax_{vc}, \forall v \in V, \forall (d, m) \in U, \forall (k, c) \in K_v C_v, i \neq d
\]  

\[
q_{imvkc} \leq CMax_{vc} \sum_{(i,m) \in N} x_{jnv}, \forall v \in V, \forall (i, m) \in N, \forall (k, c) \in K_v C_v, i \neq j
\]  

\[
I_{imvkc} \leq CMax_{vc} \sum_{(i,m) \in N} x_{jnv}, \forall v \in V, \forall (i, m) \in N, \forall (k, c) \in K_v C_v, i \neq j
\]  

\[
x_{jnv} - p_{imv} = 0, \forall v \in V, \forall (i, m, j, n) \in N
\]
\[ x_{jndmv} - P_{dmv} = 0, \forall v \in V, \forall (j, n) \in N, \forall (d, m) \in U \]  
(18)

\[ x'y_{invk} \in \{0, 1\}, \forall v \in V, \forall k \in K, \forall (i, m) \in N \]  
(19a)

\[ P_{inv}, P_{dmv} \in \{0, 1\}, \forall v \in V, \forall (i, m) \in N, \forall (d, m) \in U \]  
(19b)

\[ I_{invkc}, q_{invkc}, q_{dmvkc} \geq 0, \forall v \in V, \forall (i, m) \in N, \forall (d, m) \in U, \forall (k, c) \in K_c xC_v \]  
(20)

\[ t_{a(i+1)} - t_{aim} \geq 0, \forall (i, m) \in N \]  
(21a)

\[ t_{a(d+1)} - t_{adm} \geq 0, \forall (d, m) \in U \]  
(21b)

\[ ta_{im} + TP_{p_{inv}} \leq \left[ \frac{I_{ik} - SMin_{ij}}{CR_{ik}} \right], \forall k \in K, \forall (i, m) \in N \]  
(22)

\[ A_i \leq ta_{im} \leq B_i, \forall (i, m) \in N \]  
(23)

\[ ta_{im} + TP_{p_{inv}} + \sum_{k \in K_c} \sum_{c \in C_c} TQu_{dk} q_{invk} - te_{im} = 0, \forall v \in V, \forall (i, m) \in N, \forall (k, c) \in K_c xC_v \]  
(24a)

\[ ta_{dn} + TP_{p_{dmv}} + \sum_{k \in K_c} \sum_{c \in C_c} TQl_{dk} q_{dmvkc} + \sum_{c \in C_c} TC_{vc} z_{vc} - te_{dm} = 0, \forall v \in V, \forall (d, m) \in U \]  
(24b)

\[ x_{ijmnv}[te_{im} + TT_{ijv} - ta_{jn}] \leq 0, \forall v \in V, \forall (i, m, j, n) \in A_v \]  
(25a)

\[ x_{dmjnv}[te_{dm} + TT_{djv} - ta_{jn}] \leq 0, \forall v \in V, \forall (d, m, j, n) \in A_v \]  
(25b)

\[ x_{imdnv}[te_{im} + TT_{idv} - ta_{dn}] = 0, \forall v \in V, \forall (i, m, d, n) \in A_v \]  
(25c)

\[ ta_{im}, te_{im} \geq 0, \forall (i, m) \in N \]  
(26a)

\[ ta_{dm}, te_{dm} \geq 0, \forall (d, m) \in U \]  
(26b)

\[ s_{imk} = I_{ik} - CR_{ik} \left( ta_{im} + TP_{p_{inv}} \right) \]  
(27)
Equation (1) is an objective function that consists of the traveling, port setup, ship charter and compartment washing costs. Constraint (2) ensures that each ship should leave the loading port if the stock level at the unloading port cannot cover the consumption of product during a specified planning horizon. Equation (3) makes the empty ship need to be charged back to the loading port. Constraint (4) ensures that the ships coming into the port have to leave the port to the port \( j \). Constraints (5) and (6) relate to the ship that visits each port. Constraint (7) shows that the routing variables are binary. Constraint (8) ensures that the compartment washing should be done when the product to be loaded is different from the previous load. Constraints (9) and (10) are related to the compatibility of the product. Constraint (11) indicates that the variable is binary. Constraint (12) shows that the delivery cannot be split. Constraints (13)–(16) describe the number of products that can be loaded and unloaded, which should not exceed the capacity of the compartment. Constraints (17) and (18) show the port setup variable. Equations (19) and (20) restrict the type of the variable. Constraint (21) explains that the starting processing time and arrival time \((m + 1)\) should occur after \( m \). Constraint (22) ensures the unloading time cannot exceed the time by which inventory at each port can serve their consumption. Constraint (23) associates with the starting processing time that cannot exceed the time windows. Constraints (24) and (25) indicate the arrival and departure time. Constraint (26) shows that the time variable is a continuous variable. Equation (27) is used to check the stock level when the ship is ready to do the process and port setup. Constraints (28)–(31) ensure that stock levels cannot exceed the minimum and maximum limit of unloading port storage when unloaded products leave the port until the end of planning horizon. The last constraint explains that stock levels are continuous variables.

5. Proposed method

In this study, we proposed a hybrid CEGA. In this method, we applied different mutation types including swap, insertion, inversion, forward, and backward mutations. To accelerate the computing time without reducing the accuracy of the solutions, the crossover mechanism in GA was eliminated due to its complicated process and very often did not provide better results after rearranging the solution to avoid infeasibility. The steps of the algorithm are described as follows:

**Step 1.** Define inputs and outputs

Inputs used in this algorithm include planning horizon, cost parameters, product parameters, ship parameters, port parameters, shipping parameters and algorithm
parameters, whereas the outputs include total cost, ship routes, the allocation of products, total travel time, the selected sample, the number of iterations, and the computing time.

**Step 2.** Determine initial parameter values

Initial parameters consist of the number of samples generated in the population: \( N \), ratio of elite samples \( \rho \), smoothing coefficient \( \alpha \), mutation parameter \( P_m \) and termination criteria of iteration \( \varepsilon \).

**Step 3.** Generate the initial sample

The samples are randomly generated for the first iteration; while in the next iterations, samples are generated through a mutation step.

**Step 4.** Calculate the shipment quantity

Shipment quantity is calculated by the following formula:

\[
\text{Shipment quantity}(Q_{S_{ik}}) = \left( PH - \frac{II_{ik} - SM_{M_{ik}}}{CR_{ik}} \right) \times CR_{ik} + (ta_{im} + TP_i) \times CR_{ik} \tag{32}
\]

where

\[
C = TP_d + (DWT_i/TQ_d) + TT_{di} \tag{33}
\]

\[
ta_{im} = c \text{ if } c - \left( \frac{c}{24} \times 24 \right) \geq A_i \text{ and } c - \left( \frac{c}{24} \times 24 \right) < B_i \tag{34}
\]

\[
ta_{im} = \left( \frac{c}{24} \times 24 \right) + A_i \text{ if } c - \left( \frac{c}{24} \times 24 \right) \geq A_i \text{ and } c - \left( \frac{c}{24} \times 24 \right) > B_i \tag{35}
\]

\[
ta_{im} = A_i \text{ if } c - A_i < 0 \tag{36}
\]

**Step 5.** Loading products in the compartment

When performing a loading activity, the product compatibility and compartment washing cost should be considered.

**Step 6.** Update the shipment quantity

This step ensures that the ship departs with full of load conditions. This updating depends on the consumption rate at each port.

**Step 7.** Ship routing

This step ensures that the start time should be in the range of time windows.

**Step 8.** Calculate the objective function

The objective function consists of the travel, port setup, ship charter, and compartment washing costs.

**Step 9.** Select the elite sample

Choose elite samples of size \( \lceil \rho N \rceil \) individuals with smallest objective function values.

**Step 10.** Elitism

Elitism includes one sample at each iteration and the elite sample does not mutate.
Step 11. Mutation parameter update

Mutation parameters are updated with the following formula:

\[ A_{it} = (1 - \alpha) \times u + (A_{it-1} \times \alpha) \]  \hspace{1cm} (37)

where \( u \) can be calculated by the following formula:

\[ u = \frac{\bar{ze}}{2 \times z_{best}} \]  \hspace{1cm} (38)

\( \bar{ze} \) is the average of elite sample and \( z_{best} \) is the best objective function at each iteration. Mutation parameter value is determined as follows:

\[ Pm_{it} = \frac{A_{it}}{2} \]  \hspace{1cm} (39)

Step 12. Select the type of mutation

The mutation used in this research is swap, insertion, inversion, forward, and backward mutation. The selection mechanism is done randomly.

Step 13. Mutation

Mutations are performed on all samples except the elite sample.

---

### Table 1. Transportation costs.

| (IDR) | Depot | Port 1 | Port 2 | Port 3 |
|-------|-------|--------|--------|--------|
| Depot | 0     | 21,163,450 | 42,326,900 | 70,544,833 |
| Port 1| 21,163,450 | 0 | 49,381,383 | 63,490,350 |
| Port 2| 42,326,900 | 49,381,383 | 0 | 47,617,763 |
| Port 3| 70,544,833 | 63,490,350 | 47,617,763 | 0 |

### Table 2. Port setup costs.

| Port     | Port setup cost (IDR) |
|----------|-----------------------|
| Depot    | 2,500,000             |
| Port 1   | 1,250,000             |
| Port 2   | 1,750,000             |
| Port 3   | 1,750,000             |

### Table 3. Ship rental and washing costs.

| Ship     | Rental cost (IDR/day) | Compartment washing cost (IDR/compartment) |
|----------|-----------------------|--------------------------------------------|
| Ship 1   | 80,000,000            | 500,000                                    |
| Ship 2   | 120,000,000           | 928,572                                    |

### Table 4. Product compatibility.

| First product | Second product | Different ships |
|---------------|----------------|-----------------|
| Product 1     | Product 2      | Yes             |
| Product 1     | Product 3      | Yes             |
| Product 1     | Product 4      | Yes             |
| Product 2     | Product 3      | No              |
| Product 2     | Product 4      | No              |
| Product 3     | Product 4      | No              |
6. Experiments, results and analysis

Experiments were performed through four steps: first, check the validity through a small size problem; second, tune the parameters in; third, evaluate the ability of the algorithm to find best solutions with different conditions, and fourth, evaluate the performance of the proposed algorithm. Before we applied the algorithm for evaluating, we tested on a small problem to assure that our algorithm was valid. We tested on a data-set with 1 depot, 3 ports, 2 ships, and 4 products. From the enumeration we obtained routes for Ship 1 and Ship 2 as follows
Table 10. The endurance and quantity supplied at unloading port.

| Port   | Product 1 | Product 2 | Product 3 | Product 4 |
|--------|-----------|-----------|-----------|-----------|
| Port 1 | 5.43      | 4.83      | 6.00      | 6.73      |
| Port 2 | 7.99      | 8.10      | 7.92      | 6.66      |
| Port 3 | 7.04      | 8.00      | 8.20      | 8.35      |

Table 11. Quantity of shipped product at each unloading port.

| Port   | Product 1 | Product 2 | Product 3 | Product 4 |
|--------|-----------|-----------|-----------|-----------|
| Port 1 | 588.13    | 550.44    | 368.43    | 353.38    |
| Port 2 | 1039.78   | 905.82    | 624.16    | 781.30    |
| Port 3 | 1816.34   | 1282.75   | 679.17    | 1044.54   |

Table 12. Different scenarios.

| Experiment | Scenarios                                                      |
|------------|----------------------------------------------------------------|
| 1          | 1a 8 ports-3 ships-4 products                                |
|            | 1b 8 ports-3 ships-3 products                                |
|            | 1c 8 ports-3 ships-2 products                                |
| 2          | 2a 10 ports-6 ships-3 products                               |
|            | 2b 10 ports-5 ships-3 products                               |
|            | 2c 10 ports-4 ships-3 products                               |
| 3          | 3a 5 ports-5 ships-4 products                                |
|            | 3b 7 ports-5 ships-4 products                                |
|            | 3c 10 ports-5 ships-4 products                               |
| 4          | 4a 7 ports-5 ships-4 products and ships 3, 4, 5 cannot moor at Port 1 |
|            | 4b 7 ports-5 ships-4 products and ships 3, 4, 5 cannot moor at Port 2 |
|            | 4c 7 ports-5 ships-4 products and ships 3, 4, 5 cannot moor at Port 1 and ship 5 cannot moor at Port 2 and 3 |

- Ship 1: Depot-Port 1-depot, transport products 3, 4 and 2 when leaving the depot for the first time.
- Depot-Port 2-Port 3-Port 1-depot, transport Product 1 when leaving the depot for the second time
- Ship 2: Depot-Port 2-Port 3-depot, transport products 3, 4 and 2, when leaving the depot for the first time

The proposed algorithm produced the same results as those of the enumeration. This confirmed the validity of our algorithm. Then, we proceeded with experiments on larger scale problems.

The next experiments were done using data shown in Tables 1–12. To measure the performance of our proposed algorithm, we also solved the model using a hybrid Tabu Search for comparison.

There was one depot, and varied number of ports, number of products and number of ships. We simulated some scenarios to find the best solution for different conditions. The problem included selecting ship, finding route, allocating products and determining the
Table 13. Optimal solution (3 unloading ports, 4 products, and 2 ships).

| No. | Sample     | Route   | Compartment allocation | Quantity – unloaded (KL) | Departure time at loading port (h) |
|-----|------------|---------|-------------------------|--------------------------|------------------------------------|
| 1   | 2-3-1-2-1  | 1       | 3-4-2                   | 4200                     | 2625.77 3374.23                    | 1.02 1.03                            |
|     | 2-3-1      | 1-1-1   | 1-1-1                   | 1358.99                  | 2.157,11 683.90                    | 19.58                               |
Table 14. Optimal solution (3 unloading ports, 4 products, and 2 ships).

| No. | Sample | Port 1  | Port 2  | Port 3  | Process starting time (h) |
|-----|--------|---------|---------|---------|--------------------------|
|     |        | Ship 1  | Ship 2  | Ship 1  | Ship 2                    |
|     |        | Port 1  | Port 2  | Port 3  | Port 1 | Port 2 | Port 3 | Port 1 | Port 2 | Port 3 | Port 1 | Port 2 | Port 3 |
| 1   | 2-3-1-2-1 | 7.02    | 13.03   | 27.55   | 7.02  | 13.03 | 30.00 | 31.58 | 54.00 | 78.00 | 31.58 | 54.00 | 78.00 |
Table 15. Optimal solution (3 unloading ports, 4 products, and 2 ships).

| No. | Sample | Port 1 | Port 2 | Port 3 | End time (h) | Time arrive at depot (h) | Total time (h) | Total cost (IDR) |
|-----|--------|--------|--------|--------|--------------|--------------------------|----------------|-----------------|
| 1   | 2-3-1-2-1 | 8.06   | 14.05  | 31.03  | 14.06        | 51.03                    | 136.04         | 934,430,490     |

Cost (IDR)

| Ship 1 | Ship 2 | Total cost (IDR) |
|--------|--------|------------------|
| 92,988,812 | 421,650,094 | 934,430,490 |
| 419,841,584 |          |                  |
quantity shipped. Although the problem was a real case, due to the company’s confidential reason the data used here were not real but still a representative of the case.

- Parameters related to costs
- Products’ parameters related with products
- Parameters related with ships
- Parameters related to ports
- Parameter related to service

From the above data, the depot refers to the loading port, while Port 1, 2 and 3 are unloading port. It can also be seen that Ship 2 is not allowed to visit Port 1 since the ship capacity (DWT) ≥ port capacity. In addition, we can also see that when a ship is transporting Product 1 then the ship cannot transport other products. Before doing computation, the endurance of each product should be known first. The endurances are shown in Table 10, while the quantity of shipped product is shown in Table 11.

We ran a set of experiments with different conditions to see how the algorithm works. After running the experiments, the best solution was shown in Table 13–15. A solution sample of the problem consists of two parts: first, the sequence of visiting unloading port, and second the sequence of ship required to supply unloading port.

Tables 13–15 indicate that the optimal solution is 2-3-1-2-1. From this solution, the route can be explained as follows. For Ship 1, the route is Depot-Port 1-depot with carrying products 3, 4, and 2 when leaving depot for the first time. Then for the second route is Depot-Port 2-Port 3-Port 1-depot with carrying Product 1 when leaving depot for the second time. While, for Ship 2, the route is Depot-Port 2-Port 3-depot by carrying Products 3, 4 and 2 when leaving the depot for the first time. The total shipping time for both ships is 136.04 h, and the total cost paid for this solution is IDR 934,430,490 with the details as shown in Table 16.

We presented the results of the experiments in the Figures 2–4. In addition, results were also shown in Tables 15 and 16. From the solution, we can see that the Ship 2 does not transport to Port 1 since the capacity of dock in Port 1 cannot contain the ship. In addition, Product 1 cannot be contained all together with other products in one ship. The stock level of each unloading port during planning period was shown in Figure 2. In Figure 2, we can see that stock level in each port did not experience stock out or exceed the maximum capacity. As an example, Port 1 received Product 1 from the ship in Day 3. So in that day the stock increased to 1000 KL and this port never received the product again during a predetermined time horizon.

While the routing process of each vessel is indicated by Figure 3 and 4, each node presents a pair of \((i, m)\) where \(i\) is the port of unloading and \(m\) is the number of ship visits in the port as well as \((d, m)\) where \(d\) is the port of loading (depot) and \(m\) is the number of vessels in

| Table 16. Details of cost contributing to objective function. |
|-------------------------------------------------------------|
| **Cost type** | **Ship 1** | **Ship 2** |
| Transportation cost | 216,925,363 | 160,489,496 |
| Port setup cost | 11,000,000 | 6,000,000 |
| Ship rental cost | 283,355,034 | 255,160,598 |
| Compartment washing cost | 1,500,000 | – |
| **Total** | **934,430,490** | |
Figure 2. Position of stock level for each port during time horizon.
the port visit. As an example the route of Ship 1 can be described as follows: depot → Port 1 → depot → Port 2 → Port 3 → Port 1 → depot.

From the first experiment, we obtained the best parameters to be used in the algorithm which were $N = 1000$, $\rho = 0.9$ and $\alpha = 0.2$. $N$ value influences the number of samples that should be evaluated at each iteration. As the value of $N$ increases, the solution gets better. The parameters $\alpha$ and $\rho$ affected the number of iterations. However, determining the values of those three parameters needs to consider the time consumed to reach the best solution. From the second experiment, it can be concluded that the algorithm was able to produce consistent solutions regarding the changing of some conditions and constraints.

The last experiment was conducted using a hybrid Tabu Search to obtain results for comparison. Table 17 showed the difference between the best solution of the hybrid CEGA
Table 17. Comparison of best solutions between CEGA and HTS.

| Experiment | CEGA (IDR)     | Hybrid TS (IDR) | GAP     | %GAP  |
|------------|----------------|----------------|---------|-------|
| 1a         | 3,655,100,000  | 3,684,500,000  | −29,400,000 | −0.80 |
| 1b         | 2,149,300,000  | 2,168,800,000  | −19,500,000 | −0.90 |
| 1c         | 1,562,900,000  | 1,739,200,000  | −176,300,000 | −10.14 |
| 2a         | 2,594,400,000  | 2,699,900,000  | −105,500,000 | −3.91 |
| 2b         | 2,692,800,000  | 2,807,300,000  | −114,500,000 | −4.08 |
| 2c         | 2,982,000,000  | 3,020,800,000  | −38,800,000  | −1.28 |
| 3a         | 1,607,900,000  | 1,711,300,000  | −103,400,000 | −6.04 |
| 3b         | 2,021,300,000  | 2,310,500,000  | −289,200,000 | −12.52 |
| 3c         | 3,818,900,000  | 4,117,500,000  | −298,600,000 | −7.25 |
| 4a         | 2,288,700,000  | 2,384,500,000  | −95,800,000  | −4.02 |
| 4b         | 2,579,900,000  | 2,621,300,000  | −41,400,000  | −1.58 |
| 4c         | 2,664,700,000  | 2,834,600,000  | −169,900,000 | −5.99 |

and hybrid Tabu Search. Table 18 showed the average results of five replications for both methods. The GAP and % GAP were calculated using the following formula:

\[
\text{GAP} = \text{solution CEGA} - \text{solution HTS}
\]

\[
\% \text{GAP} = \frac{\text{solution CEGA} - \text{solution HTS}}{\text{solution HTS}} \times 100\%
\]

Table 18. Comparison between average solution of hybrid CEGA and hybrid TS.

| Experiment | Hybrid CEGA (IDR) | Hybrid TS (IDR) | GAP    | %GAP  |
|------------|-------------------|-----------------|--------|-------|
| 1a         | 3,736,380,000     | 3,799,140,000   | −62,760,000 | −1.65 |
| 1b         | 2,157,300,000     | 2,198,300,000   | −41,000,000  | −1.87 |
| 1c         | 1,577,320,000     | 1,795,540,000   | −218,220,000 | −12.15 |
| 2a         | 2,619,040,000     | 2,778,180,000   | −159,140,000 | −5.73 |
| 2b         | 2,699,640,000     | 2,834,820,000   | −135,180,000 | −4.77 |
| 2c         | 2,987,740,000     | 3,034,420,000   | −46,680,000  | −1.54 |
| 3a         | 1,608,600,000     | 1,786,520,000   | −177,920,000 | −9.96 |
| 3b         | 2,104,620,000     | 2,371,780,000   | −267,160,000 | −11.26 |
| 3c         | 3,860,200,000     | 4,154,140,000   | −293,940,000 | −7.08 |
| 4a         | 2,357,700,000     | 2,521,400,000   | −163,700,000 | −6.49 |
| 4b         | 2,597,620,000     | 2,725,400,000   | −127,780,000 | −4.69 |
| 4c         | 2,731,960,000     | 2,889,440,000   | −157,480,000 | −5.45 |
From the 12 experiments that have been done as shown in Tables 16 and 17, the hybrid CEGA produced better solutions than the hybrid Tabu Search for both best and average values. The results produced by the hybrid CEGA were smaller than those of the hybrid Tabu Search. Using a hybrid CEGA we can obtain less total cost for some different conditions on number of ships, products, and ports. Decision-makers can use the benefit of these results to take good decision with less cost without violating the constraints. Decision-makers may utilize the results to determine number of ships, to select route, select ports to visit, product allocation, and quantity to ship.

One of the reasons to cause the difference is that CEGA is a population-based technique, whereas a hybrid Tabu Search is an individual-based technique. In the first category, the possibility to obtain better results is higher than those of individual based-techniques. CEGA evaluates more samples than a hybrid Tabu Search in each iteration. In addition, CEGA have mutation step that consists of swap, insertion, inversion, forward and backward mutations. This mechanism makes diversification its solution and spreads the results to a larger solution space. The advantage of this step is that there are more solutions can be examined. In a hybrid Tabu Search, the reason that makes the solution worse is the application of taboo list which only works for a combination producing solutions that violate the constraint, while the solution that is worse than the previous solution is not included in this list. So there is a possibility of repeating the same solutions and producing non-optimal solutions.

7. Conclusions

This study proposed a model of the multi-product ISRP with a heterogeneous fleet that considers the weight of the ship moored in port, product compatibility, port setup, and compartment washing costs. The new features included in the model were the deadweight of ship consideration, port setup, and compartment washing costs. The resulting mathematical model is characterized as NP-hard problem. To solve the proposed model, a hybrid CEGA was developed. To test the validity of our algorithm, we applied on small size problem and compared it to the enumeration results. The algorithm was then applied to solve some larger scale problems. The algorithm has successfully found the number of required ships, quantity of shipped products, product allocation, and the best route that minimized the total cost. The algorithm was able to solve the problems consistently on several different conditions. The different conditions involved the changes in the number of products, number of ships, number of ports to be visited, and the number of ports that can be visited by the ship. Before we tested the algorithm on some different conditions, we tuned the parameters in to find the best parameters’ values. To evaluate the performance of the algorithm, the results were compared with those of the hybrid Tabu Search based on the total cost value. The comparison of the hybrid CEGA with the hybrid Tabu Search showed that the hybrid CEGA provides better results for all conditions in terms of total cost. The proposed algorithm produced lower total costs compared to the hybrid Tabu Search. The proposed algorithm provided better solutions for all data-sets in terms of solution quality. For the future research, model can be modified by removing some assumptions used in this paper to better represent the real case condition. The model can be extended to contain multi-objective optimization, not all products available when ship coming to the port, consumpton rate on unloading port is not constant, etc.
Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Higher Education Ministry of Indonesia.

References

Agra, A., Christiansen, M., Delgado, A., & Hvattum, L. M. (2015). A maritime inventory routing problem with stochastic sailing and port times. *Computers and Operations Research, 61*, 18–30.

Al-Khayyal, F., & Hwang, S. J. (2007). Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part I: Applications and model. *Middle East, 176*, 106–130.

Andersson, H., Duesund, J. M., & Fagerholt, K. (2011). Ship routing and scheduling with cargo coupling and synchronization constraints. *Computers & Industrial Engineering, 61*, 1107–1116.

Chopra, S., & Meindl, P. (2007). *Supply Chain Management Strategy, Planning and Operation*. New Jersey: Prentice Hall.

de Boer, P. T. D., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research, 134*, 19–67.

Hemmati, A., Stålhane, M., Hvattum, L. M., & Andersson, H. (2015). An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. *Computers & Operations Research, 64*, 274–282.

Ho, W., Ho, G. T. S., Ji, P., & Lau, H. C. W. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence, 21*, 548–557.

Hvattum, L. M., Fagerholt, K., & Armentano, V. A. (2009). Tank allocation problems in maritime bulk shipping. *Computers & Industrial Engineering, 36*, 3051–3060.

Laguna, M., Duarte, A., & Marti, R. (2007). Hybridizing the cross-entropy method: An application to the max-cut problem. *Computers & Operations Research, 36*, 487–498.

Lee, J., & Kim, B.-I. (2015). Industrial ship routing problem with split delivery and two types of vessels. *Expert Systems with Applications, 42*, 1–12.

Rubinstein, R. Y., & Kroese, D. P. (2004). *The cross-entropy method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. New York, NY: Springer Science+Business Media.

Rusdiansyah, A., & Nurminarsih, S. (2012). Hybrid tabu search for solving multi product tanker scheduling problem (m-TSP) considering product loading compatibility constraint. In V. Kachitvichyanukul, H. T. Luong, & R. Pitakaso (Eds.), *Proceedings of the Asia Pacific industrial engineering & management systems conference*, Phuket, Thailand, APIEMS (pp. 1085–1092).

Santosa, B., Budiman, M. A., & Wiratno, S. E. (2011). A cross entropy-genetic algorithm for m-machines no-wait job-shop scheduling problem. *Journal of Intelligent Learning Systems and Applications, 3*, 171–180.

Santosa, B., & Hardiansyah, N. (2010). Cross entropy method for solving generalized orienteering problem. *iBusiness, 2*, 342–347.

Santosa, B., & Willy, P. (2011). *Metoda metaheuristik, Konsep dan Implementasi* [Metaheuristics method: Concept and implementation]. Surabaya: Guna Widya.

Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic Algorithms. In E. K. Burke & G. Kendall (Eds.), *Search methodologies*. New York, NY: Springer.

Siswanto, N., Essam, D., & Sarker, R. (2011). Solving the ship inventory routing and scheduling problem with undedicated compartments. *Computers & Industrial Engineering, 61*, 289–299.