Applied Soft Computing 31 (2015) 30–47

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems

Ali Mohtashami[a], Madjid Tavana[b,c,*], Francisco J. Santos-Arteaga[d], Ali Fallahian-Najafabadi[e]

[a] Department of Industrial Management, Faculty of Management and Accounting, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Business Systems and Analytics Department, Lindback Distinguished Chair of Information Systems and Decision Sciences, La Salle University, Philadelphia, PA 19141, USA
[c] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn D-33098, Germany
[d] Departamento de Economía Aplicada II, Facultad de Económicas, Universidad Complutense de Madrid, Campus de Somosaguas, Pozuelo 28223, Spain
[e] Faculty of Management and Accounting, Qazvin Branch, Islamic Azad University, Qazvin, Iran

ARTICLE INFO

Article history:
Received 19 October 2014
Received in revised form 10 February 2015
Accepted 10 February 2015
Available online 28 February 2015

Keywords:
Supply chain
Cross-docking
Non-dominated sorting genetic algorithm
Particle swarm optimization
Scheduling
Transportation planning

ABSTRACT

Cross-docking is a material handling and distribution technique in which products are transferred directly from the receiving dock to the shipping dock, reducing the need for a warehouse or distribution center. This process minimizes the storage and order-picking functions in a warehouse. In this paper, we consider cross-docking in a supply chain and propose a multi-objective mathematical model for minimizing the make-span, transportation cost and the number of truck trips in the supply chain. The proposed model allows a truck to travel from a supplier to the cross-dock facility and from the supplier directly to the customers. We propose two meta-heuristic algorithms, the non-dominated sorting genetic algorithm (NSGA-II) and the multi-objective particle swarm optimization (MOPSO), to solve the multi-objective mathematical model. We demonstrate the applicability of the proposed method and exhibit the efficacy of the procedure with a numerical example. The numerical results show the relative superiority of the NSGA-II method over the MOPSO method.

1. Introduction

Cross-docking is a material handling and distribution method for moving goods directly from receiving to shipping. The primary objective in cross-docking is to eliminate storage and excessive handling [1]. Cross-docking can reduce inventories, increase operational efficiency, shorten product delivery time, increase customer satisfaction and responsiveness, and tighten control in the distribution operation. Cross-docking has become a widely used distribution method as companies streamline their distribution functions. In supply chain management, cross-docking is a logistics method for minimizing the need for warehouse storage and coordinating the distribution activities related to loading goods between delivery vehicles and shipping vehicles. In general, five activities are conducted in the distribution centers: reception, organizing, storing, selection, and transfer. Improving these activities can have a direct impact on the productivity in a supply chain.

Handling freight at cross-docking terminals is a complex planning task which comprises several activities including unloading shipments delivered by inbound trucks, sorting according to designated destinations, and loading onto outbound trucks delivery. Cross-docking operations need to be carefully synchronized due to the absence of storage facilities inside a cross-dock [2–5]. This synchronization requires coordination between the performance of inbound and outbound trucks so that these trucks can be effectively scheduled for delivery.

Lee et al. [6] argue that previous studies have dealt mostly with the conceptual advantages of cross-docking from the strategic viewpoint. They consider both cross-docking and vehicle routing scheduling and propose a heuristic algorithm based on a tabu search to solve this NP hard problem. Waller et al. [7] developed models to predict the changes in the retailer's system-wide inventory levels as a result of cross-docking and examined

* Corresponding author at: Business Systems and Analytics Department, Lindback Distinguished Chair of Information Systems and Decision Sciences, La Salle University, Philadelphia, PA 19141, USA. Tel.: +1 215 951 1129; fax: +1 267 295 2854.

E-mail addresses: mohtashami@qiau.ac.ir (A. Mohtashami), tavana@lasalle.edu (M. Tavana), fransant@ucm.es (F.J. Santos-Arteaga), alifallahian@yahoo.com (A. Fallahian-Najafabadi).

URL: http://tavana.us/ (M. Tavana).

http://dx.doi.org/10.1016/j.asoc.2015.02.030
1568-4946/© 2015 Elsevier B.V. All rights reserved.

the impact of a number of relevant parameters on the cross-docking performance. Yan and Tang [8] and Tang and Yan [5] constructed analytical models for distribution strategies for two different types of cross-docking: pre-distribution cross-docking and post-distribution cross-docking. They developed three models and showed that pre-distribution cross-docking is preferred for environments with shorter supply lead time and lower uncertainty in demand. Chen and Lee [9] studied the scheduling problem in cross-docking and showed that the problem is a NP-hard problem. They developed a polynomial approximation algorithm with an error-bound analysis. Chen and Song [10] also studied the scheduling problem in cross-docking and used a mixed integer programming model to minimize the total operational time (makespan).

Vahdani and Zandieh [11] applied five meta-heuristic algorithms: the genetic algorithm, tabu search, simulated annealing, an electromagnetism-like algorithm and a variable neighborhood search to schedule the trucks in a cross-docking system by minimizing total operation time when a storage facility is used to temporarily hold goods at the shipping dock. They showed that the variable neighborhood search outperforms the other four methods in this study. Liao et al. [12] considered a model for integrating cross-docking into the vehicle routing problem. The objective of their model was to determine the number of vehicles and a set of vehicle schedules to minimize the sum of the operational and transportation costs. They proposed a new tabu search algorithm and obtained a good feasible solution for the problem.

Boloori Arabani et al. [13,14] studied three famous multi-objective algorithms including the non-dominated sorting genetic algorithm-II (NSGA-II), the strong Pareto evolutionary algorithm-II (SPEA-II), and the sub-population genetic algorithm-II (SPGA-II) to solve the cross-docking scheduling problems. They considered the makespan and the total lateness of all outbound trailers as objective functions and used four criteria to evaluate each algorithm. Boloori Arabani et al. [15] also studied a scheduling problem of inbound and outbound trucks in a cross-docking system. Again they considered minimization of the total makespan and minimization of the total lateness of outbound trucks and developed three multi-objective algorithms based on the sub-population concept of evolutionary algorithms. Sadykov [16] proposed a dynamic programming algorithm to optimize the schedule for both inbound and outbound trucks when products are unloaded in the inbound door but the corresponding outbound truck is not immediately available in the outbound door. Kuo [17] studied the efficiency of multi-door cross docking by optimizing both inbound and outbound truck sequencing and both inbound and outbound truck dock assignment. A variable neighborhood search algorithm was proposed to minimize the makespan and optimize the sequence of all inbound and outbound trucks. Four simulated annealing algorithms were adopted for comparison. The experimental results showed that the proposed variable neighborhood search algorithm was efficient and effective in solving cross-docking problems.

Larbi et al. [18] argue that little attention has been given to the transshipment operations inside a cross-dock. They studied the transshipment scheduling problem in a single receiving and a single shipping door cross-dock under various conditions including complete information on the order of arrivals and the contents of all inbound trucks, and the availability of partial and no information on the sequence of upcoming trucks. They proposed an optimal graph based model for the full information case and two heuristics for the other two cases. Liao et al. [19] considered minimizing total makespan or equivalently maximizing the throughput of a cross-docking system and studied inbound and outbound truck sequencing for cross-docking operations. They proposed two new hybrid differential evolution algorithms which performed better than the pure differential evolution algorithms in the literature. The also used 30 problems to test the effectiveness of the proposed algorithms. Liao et al. [20] also studied the simultaneous dock assignment and sequencing of inbound trucks for a multi-door cross docking operation with the objective of minimizing the total weighted tardiness. They solved the problem with six different meta-heuristic algorithms including simulated annealing, tabu search, ant colony optimization, differential evolution, and two hybrid differential-evolution algorithms.

Meysam Mousavi and Tavakkoli-Moghaddam [21] formulated a two-stage mixed-integer programming problem for cross-docking location planning and solved this problem with a two-stage hybrid simulated annealing and tabu search model. They randomly generated several problems and showed that the proposed hybrid simulated annealing performs efficiently and converges rapidly to reasonable solutions. Bellanger et al. [22] studied the optimization of a cross-docking system by modeling the system as a three-stage (i.e., receiving dock, sorting stations, and shipping docks) hybrid flow shop, in which shipments and orders were represented as batches. Their goal was to find a schedule that minimizes the completion time of the latest batch. They developed several heuristic schemes which produced good quality feasible solutions. They then proposed a branch-and-bound algorithm that took the three stage problem decomposition into consideration. They also proposed a lower bound to evaluate the heuristics and to reduce the tree size during the branch-and-bound computation. They conducted several computational experiments and showed that their algorithm provided good quality feasible solutions.

Meysam Mousavi et al. [23] introduced two novel deterministic mixed-integer linear programming models for determining the location of cross-docking centers and scheduling of the vehicle routing problem. They also proposed a hybrid fuzzy possibilistic–stochastic programming solution to incorporate two kinds of uncertainties into their mathematical programming models. They generated several test problems and appraised the applicability and suitability of the proposed model. Agustina et al. [24] studied cross-docking operations for just-in-time food delivery with minimum cost of delivery (i.e., inventory holding and transportation costs) and penalty costs for early or tardy deliveries. They integrated vehicle scheduling and routing, which have traditionally been modeled separately, in one comprehensive model. They formulated the problem as a mixed-integer linear program. They also introduced the concept of customer zones to reduce the size of the solution space. Van Belle et al. [25] have presented an extensive review of the cross-docking literature and concepts. They have also discussed guidelines for the successful use and implementation of cross-docking strategies.

In this study, we focus on the cross-docking transportation operations in a supply chain. As shown in Fig. 1, the incoming trucks with varying capacities ship the products directly from suppliers to customers or offload them in the cross-dock. The products are then shipped by the outgoing trucks with different capacities from the cross-dock to the customers. In other words, each truck can ship products from one or more suppliers. In addition, a given truck can ship the products to one or more customers. The figure shows that the trucks which are used in transportation have different capacities.

We propose a multi-objective mathematical model for minimizing the make-span, transportation cost and the number of truck trips in the supply chain. The proposed model allows a truck to travel from a supplier to the cross-dock facility and from the supplier directly to the customers. We propose two meta-heuristic algorithms, the NSGA-II and the multi-objective particle swarm optimization (MOPSO), to solve the multi-objective mathematical model. We also demonstrate the applicability of the proposed method and exhibit the efficacy of the procedure with a numerical example.

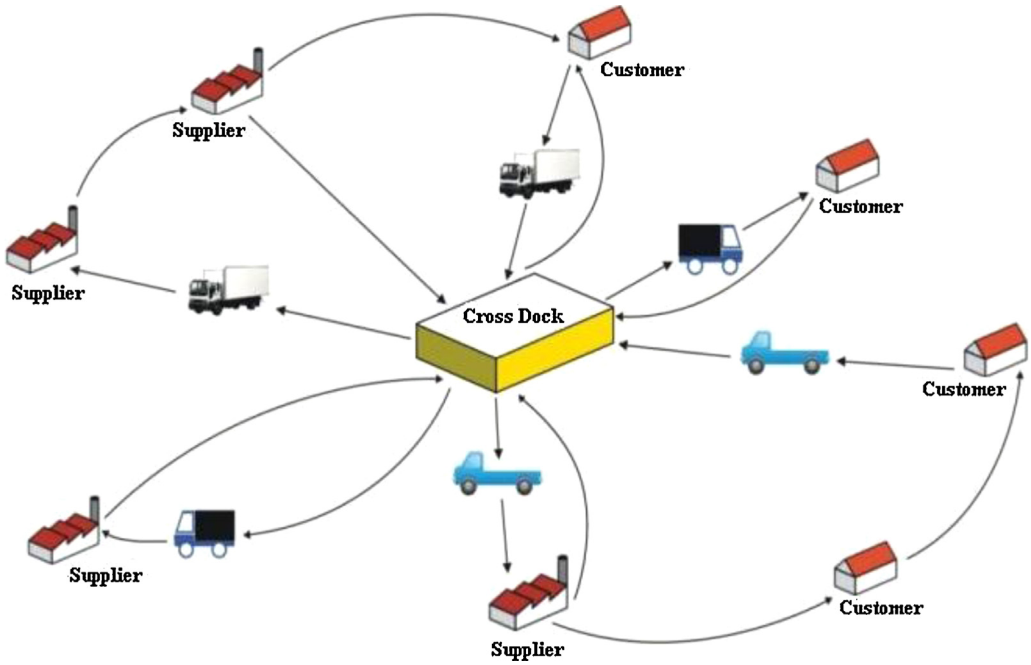


Fig. 1. Cross-docking transportation operations in a supply chain.

The remainder of this paper is organized as follows. In Section 2, we review multi-objective evolutionary algorithms including the NSGA-II method and the MOPSO method. In Section 3, we present the multi-objective cross-docking model proposed in this study. In Section 4, we demonstrate the applicability of the proposed method and exhibit the efficacy of the procedure with a numerical example. In Section 5, we perform a comparative analysis between the NSGA-II method and the MOPSO method. Finally, in Section 6, we present our conclusions and future research directions.

2. Multi-objective evolutionary algorithms

Evolutionary algorithms are stochastic search methods that are designed to emulate the language of natural biological evolution. These search methods apply the principle of survival of the fittest to a population of alternative solutions in order to produce better solutions to a problem. The most important advantage of evolutionary algorithms in multi-objective optimization problems is their capacity to achieve a set of non-dominated solutions without assigning weights to the target functions which are a function of the decision makers' views. The notion of searching a solution space is a key characteristic of evolutionary algorithms which makes them suitable for multi-objective optimization [26]. Evolutionary algorithms used for solving multi-objective optimization problems are known as multi-objective evolutionary algorithms (MOEAs). We apply two known multi-objective meta-heuristic algorithms, the NSGA-II and MOPSO, to achieve an optimum solution in the cross-docking problem introduced in this paper.

2.1. Non-dominated sorting genetic algorithm

As the number of goals have increased in the optimization problems, more applications of multi-objective genetic algorithms (which have been an outgrowth of single-objective algorithms) have been considered. Deb [27] proposed a non-dominated sorting genetic algorithm (NSGA-II) which has been transformed into a multi-objective algorithm by the addition of two necessary operators to a single-objective genetic algorithm. The two operators are: (1) the operator which assigns a superiority criterion (ranking) to the members of the population based on non-dominated sorting; (2) the operator which maintains the variety of solutions among the ones with similar ranks. This method can find the set of best solutions, known as the optimal front, instead of a single best solution. In this algorithm, it is necessary to explain the concepts of dominance, non-dominated sorting, and swarm distance.

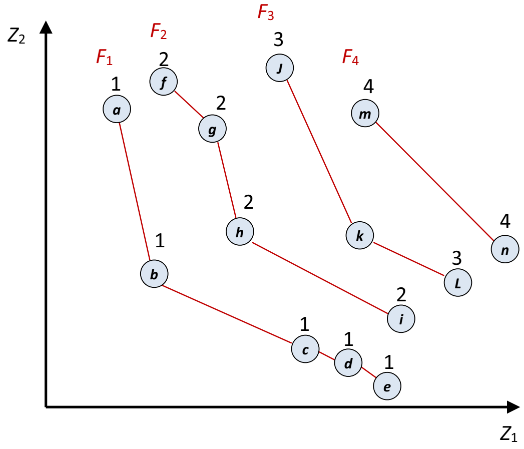


Fig. 2. The point sorting in the NSGA-II method.

2.1.1. Concept of dominance

In the NSGA-II method, the members of populations are sorted according to the concept of non-dominance. In the non-dominant sorting method, the non-dominated members of the population are assigned a ranking of 1 and they are put in the first front set. They are then removed from the population and among the remaining members, the non-dominated members are selected and a ranking of 2 is assigned to them. These members are put in the second front and they are removed from the population temporarily. This process continues until all the members of the population are put in different fronts (Fig. 2). The dominance of one point with respect

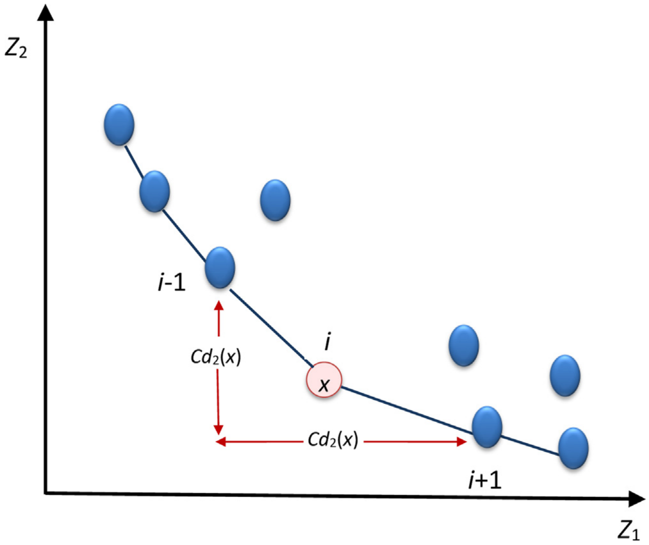


Fig. 3. Swarm distance calculation.

to another point has been shown in Eq. (1). This shows that point b is dominated by point a. f_i and f_j are target functions i and j.

$$\forall i \in \{1, \ldots, n\} : f_i(a) \geq f_i(b) \wedge \exists j \in \{1, \ldots, n\} : f_j(a) > f_j(b) \tag{1}$$

2.1.2. Concept of non-dominated sorting

When a single-objective algorithm is discussed, the superiority of solutions relative to each other is very simple and clear, because there is only one target function. For minimization problems, the minimum value of the target function is the best solution and it is superior to the other solutions. However, a multi-objective algorithm requires at least two target functions. In most cases, some points are found which are not superior to each other and that cannot be compared with each other by the concept of dominance. Therefore, they should be sorted according to a criterion in order to obtain the best solutions. In this algorithm, each solution is assigned a rank based on the number of dominations relative to other points. At the end of the algorithm, those points which have the best rank (rank 1) are identified as the solution set or Pareto front points.

2.1.3. Concept of maintaining the variety of answers (swarm distance)

After arranging the points in the fronts, another criterion is used to evaluate the points in a front which does not have swarm distance. The swarm distance is a substitute for *shared* in the NSGA method. First, the results are sorted for every target function Z. Those points which have the maximum and minimum values of the target function are assigned extreme distance values. The swarm distance for the members of the population in each front is calculated according to Eqs. (2)–(4). As shown in Fig. 3, the distances are considered among the members that are in the same front.

$$cd_1(x) = \frac{\left|z_1^{i+1} - z_1^{i-1}\right|}{\left|z_1^{\max} - z_1^{\min}\right|} \tag{2}$$

$$cd_2(x) = \frac{\left|z_2^{i+1} - z_2^{i-1}\right|}{\left|z_2^{\max} - z_2^{\min}\right|} \tag{3}$$

$$cd(x) = cd_1(x) + cd_2(x) \tag{4}$$

The reason behind the creation of these two operators in a multi-objective genetic algorithm is related to the selection stage. In this stage, a number of chromosomes should be selected from the chromosomes of the parents and children in order to start the next stage and other solutions should be deleted. Fig. 4 shows one of the iterations in the NSGA-II algorithm.

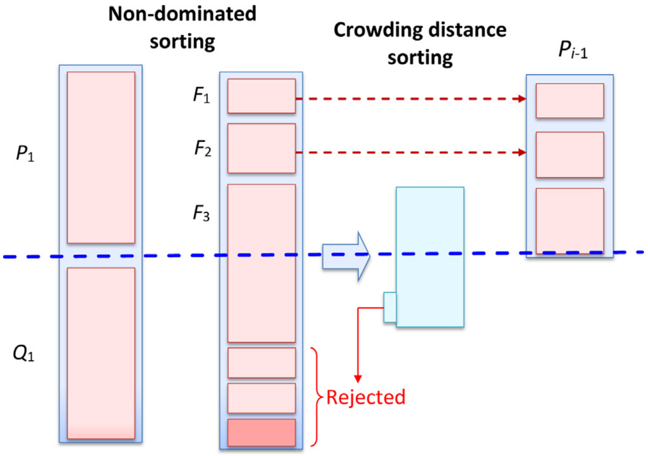


Fig. 4. The iteration process in NSGA-II.

2.2. Multiple objective particle swarm optimization

PSO is a social search algorithm which has been copied from the behavior of birds' flocks. This algorithm was introduced for the first time by Eberhart and Kennedy [28]. The PSO was initially used to discover the patterns of birds' flight and the sudden changes that happen in their course of flight as well as the optimum changes in the shape of the flock. The PSO algorithm begins with a group of random answers (particles) and then it seeks to find the optimum answer by sorting (finding new positions) the particles in each repetition. When decision variables and particle positions are equal to zero and one, velocity and position vectors of each particle in each iteration of the algorithm are calculated by Eqs. (5)–(8).

$$V_{it} = w.V_{it-1} + c_1.r_{1.}(pBest_i - x_{it}) + c_2.r_2(nBest_i - x_{it}) \tag{5}$$

$$-V_{\max} \leq V_{it} \leq V_{\max} \tag{6}$$

$$s_i = \frac{1}{1 + \exp(-V_{it})} \tag{7}$$

$$x_{it} = \begin{cases} 1 & \rho \leq s_i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

According to Eq. (5), the new velocity vector of each particle is calculated by the previous velocity of the same particle (V_{it-1}), the optimum position which has been achieved by the particle ($p\text{Best}_i$), and the position of the best particle which is near to that particle ($n\text{Best}_i$). When the neighborhood of the particle includes all the particles of the group, $n\text{Best}_i$ is the position of the best particle in the group which is denoted by Gbest. r_1 and r_2 are two random numbers generated independently by a normal distribution in [0,1]. c_1 and c_2, which are called the learning coefficients, control the impact of pBest and nBest on the search process. w is the inertia weight coefficient. $V_{\max}$ is the maximum of particles' velocity vector. $V_{\max}$ is a constraint that controls the capability of the particles' universal search. Using Eq. (7), we can transfer each particle's velocity vector to change the probability vector. In this equation, s_i indicates the probability of $x_{it} = 1$. The new position of each particle is found by Eq. (8). The new positions of the particles are taken as the new inputs of the simulation model of the particles' movement in the second iteration. Then, the target function is obtained once again. This process continues until the maximum number of iterations is reached. In this equation, ρ is assumed to be a random variable

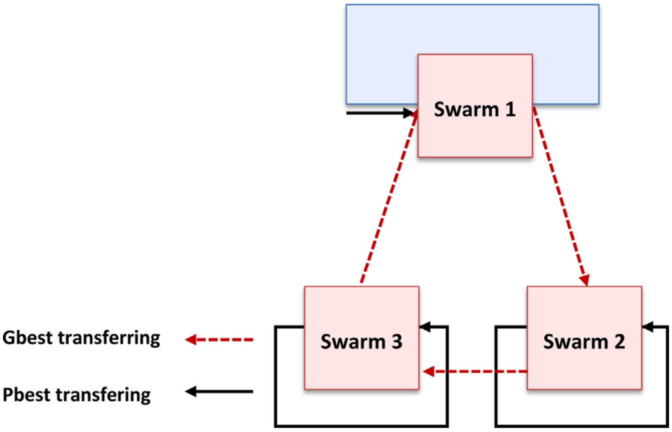


Fig. 5. Schematic view of the Pbest and the *G*best exchange in a triple-objective problem.

normally distributed between zero and one. The position of each particle is determined by its relative position to the other particles.

Moreover, the movement toward the optimum point is different in the single-objective PSO and the MOPSO.

In the single-objective PSO algorithm, each particle follows its target function. In the MOPSO algorithm, we have more than one target function. Therefore, the algorithm should be compatible with all the target functions. The vector evaluation PSO is a multi-objective algorithm which uses one set for every target function. Consequently, in each set, each particle has its specific *P*best. But, in each set, *G*best is replaced by the *G*best of other sets for the next iterations. Fig. 5 presents an example of *P*best and *G*best exchange in a triple-objective problem [29].

In the proposed method, a dynamic archive for the exchange of produced particles is created in each iteration. The particles are compared to each other, the non-dominated solutions are maintained, and other solutions are deleted at the end of each iteration. In this process, we have access to the members of the external archive in each iteration. Also, the size of the external archive is changed dynamically.

3. Multi-objective cross-docking model

We consider three minimization goals in the cross-docking model proposed in this study: the time of transportation operations, the expenses of transportation by the trucks, and the number of transportations by the trucks between sources and targets. Also, the number and capacity of the trucks, the distances which are covered by the trucks between sources and targets, the time of shipment and offloading, and the time of products' movement in the cross-docking are modeled using uniform distribution functions with different domains. Table 1 presents the random data spans for each problem.

Table 1
Random data spans for each problem.

Distribution functions	Parameters
~Uniform (5,15)	Number of trucks
~ Uniform (50,150)	Capacity of trucks
~ Uniform (50,500)	The distance which is covered by the trucks
~ Uniform (5,25)	The time of shipment
~ Uniform (10,30)	The time of products' offloading
~ Uniform (5,30)	The time of products' movement in cross-docking

3.1. Model hypotheses

The following hypotheses are applicable to the model proposed in Section 3.2:

- All incoming and out-going trucks are accessible at zero time.
- All incoming trucks can act as an out-going truck.
- All products in cross-docking should be transferred to the customers and are not allowed to be stored for a long period of time.
- The capacity of cross-docking is unlimited.
- The number and type of supplied products are fixed.
- The number and type of products which are demanded by customers are fixed.
- The number of suppliers, customers, and cross-dockings are fixed.
- In a sequence of transportation, a truck can carry the products of more than one supplier and also it can offload the products for more than one customer.
- In a sequence of transportation, based on the allowed capacity, a truck can transport the whole or part of a customer's demands.
- The demands of a given customer might be satisfied by one or more suppliers.
- The incoming and out-going trucks cannot carry more products than their capacities.
- The type and amount of products which are carried by incoming trucks should be equal to the customers' demands, and
- if the amount of products which are offloaded by incoming trucks in the cross-docking is very high, these products can be stored temporarily until the time of transportation to the targets.

The following variables and parameters are used in the model proposed in this study:

Indices

N	the number of suppliers
M	the number of customers
R	the number of incoming trucks (reception)
H	the number of out-going trucks (dispatch)
U	the number of incoming and out-going trucks
K	the number of shipment and offloading conditions
Q	the capacity of the trucks
E	a large number
$P_{c\alpha}$	the number of products unit in α condition needed by the customer (c)
$P_{s\alpha}$	the number of products unit in α condition needed by the supplier (s)

Time

$T^i_{cds\alpha}$	the time of transportation by the incoming truck i from cross-docking (cd) to supplier (s), while it is shipping type α order
$T^i_{sc\alpha}$	the time of transportation by the incoming truck i from supplier (s) to customer (c), while it has shipped type α order
$T^i_{scd\alpha}$	the time of transportation by the incoming truck i from supplier (s) to cross-docking (cd), while it has shipped type α order
T^k_{rsh}	the time of transportation of a k type product from the entrance of cross-docking (r) to its exit (sh)
$T^j_{cdc\alpha}$	the time of transportation by the out-going truck j from cross-docking (cd) to customer, while it has shipped type α order
$T^i_{s_m s_n \alpha}$	the time of transportation by the incoming truck i from supplier in the source (s_m) to the supplier in the target (s_n), while it is shipping type α order

$T^{i}_{c_m c_n \alpha}$ the time of transportation by the incoming truck i from the customer in the source (c_m) to the customer in the target (c_n), while it is offloading type α order

$T^{j}_{c_m c_n \alpha}$ the time of transportation by the outgoing truck j from the customer in the source (c_m) to the customer in the target (c_n), while it is offloading type α order

$T^{i}_{ccd\alpha}$ the time of transportation by the incoming truck i from customer (c) to cross-docking (cd), while it has offloaded type α order

$T^{j}_{ccd\alpha}$ the time of transportation by the outgoing truck j from customer (c) to cross-docking (cd), while it has offloaded type α order

$T^{g}_{c_n cd\alpha}$ the time required for the last truck g to return from the last customer in the source (c_n) and arrive at cross-docking (cd), while it has offloaded type α order

$L^{i}_{cds\alpha}$ the time required for type α order to be shipped in the incoming truck i by supplier (s)

$UL^{i}_{scd\alpha}$ the time required for type α order to be offloaded from incoming truck i into cross-docking (cd)

$UL^{i}_{sc\alpha}$ the time required for type α order to be offloaded from incoming truck i to the customer (c)

$L^{j}_{cdc\alpha}$ the time required for type α order to be shipped from cross-docking (cd) by the out-going truck j

$UL^{j}_{cdc\alpha}$ the time required for type α order to be offloaded from out-going truck j to the customer (c)

Expenses

$C^{i}_{cds\alpha}$ the expenses related to the transportation by incoming truck i from cross-docking (cd) to supplier (s), while type α order is shipped

$C^{i}_{sc\alpha}$ the expenses related to the transportation by incoming truck i from supplier (s) to customer (c), while it has shipped type α order

$C^{i}_{scd\alpha}$ the expenses related to the transportation by incoming truck i from supplier (s) to cross-docking (cd), while it has shipped type α order

C^{k}_{rsh} the expenses related to the transportation of type k products from the entrance of cross-docking (r) to its exit (sh)

$C^{j}_{cdc\alpha}$ the expenses related to the out-going truck j from cross-docking to customer (c), while it has shipped type α order

$C^{i}_{s_m s_n \alpha}$ the expenses related to the transportation by incoming truck i from the source supplier (s_m) to target supplier (s_n), while type α order is shipped

$C^{i}_{c_m c_n \alpha}$ the expenses related to transportation by incoming truck i from source customer (c_m) to target customer (c_n), while it offloads type α order

$C^{j}_{c_m c_n \alpha}$ the expenses related the transportation by out-going truck j from source customer (c_m) to target customer (c_n), while it offloads type α order

$C^{i}_{ccd\alpha}$ the expenses related to the transportation by incoming truck i from customer (c) to cross-docking (cd), while it has offloaded type α order

$C^{j}_{ccd\alpha}$ the expenses related to the transportation by out-going truck j from customer (c) to cross-docking (cd), while it has offloaded type α order

Integer variables

$q_{\alpha s}$ the number of product units α which are shipped by incoming trucks i from suppliers

$q_{\alpha cd}$ the number of product units α which are shipped by out-going truck j from cross-docking

F the number of transportations by trucks between sources and targets

Continuous variables

T the time of transportation

C the expenses of transportation

Zero and one variables

$$X^{i}_{cds\alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from cross docking } (cd) \text{ toward supplier } (s) \text{ and it ships type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{i}_{s_m s_n \alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from source supplier } (s_m) \text{ toward targetsupplier } (s_n)\text{and it ships type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{i}_{scd\alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from supplier } (s) \text{ toward cross docking } (cd) \text{ and it has shipped type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{i}_{sc\alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from supplier } (s) \text{ toward customer } (c)\text{and it has shipped type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{i}_{c_m c_n \alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from source customer } (c_m) \text{ toward target customer } (c_n) \text{ and it has shipped type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{j}_{c_m c_n \alpha} = \begin{cases} 1 & \text{when the out-going truck } j \text{ moves from source customer } (c_m) \text{ to target customer } (c_n)\text{and it has shipped type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^{j}_{cdc\alpha} = \begin{cases} 1 & \text{when the out-going truck } j \text{ moves from cross docking } (cd) \text{ toward customer } (c) \text{ and it has shipped type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^i_{ccd\alpha} = \begin{cases} 1 & \text{when the incoming truck } i \text{ moves from customer } (c) \text{ toward cross docking } (cd) \text{ and it has offloaded type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$X^j_{ccd\alpha} = \begin{cases} 1 & \text{when the out-going truck } j \text{ moves from customer } (c) \text{ toward cross docking } (cd) \text{ and it has offloaded type } \alpha \text{ order} \\ 0 & \text{in other cases} \end{cases}$$

$$p_{ij} = \begin{cases} 1 & \text{when incoming truck } i \text{ takes precedence over incoming truck } j \\ 0 & \text{in other cases} \end{cases}$$

$$q_{g_1 g_2} = \begin{cases} 1 & \text{when truck } g_1 \text{ takes precedence over truck } g_2 \\ 0 & \text{in other cases} \end{cases}$$

The set of variables and parameters described above define a supply chain environment that integrates a standard cross-docking process within its logistic structure. In particular, the logistic framework defined by these variables is presented in Figs. 6 and 7. These figures can be directly compared with Fig. 1, which represents the standard textbook case of a cross-docking process integrated within a supply chain.

We have separated the transportation duties of incoming and outgoing trucks to make the model more tractable. Note how, incoming trucks do not connect the cross-dock directly with the customer, a task reserved exclusively for outgoing trucks. At the same time, incoming trucks are in charge of collecting orders from the supplier and either bringing them back to the cross-dock or delivering them to the customers. Thus, the model proposed extends the main research environments formalized in the current cross-docking literature, which range from different variants of the cross-docking scheduling problem [13,14,30] to vehicle routing [6], and integrates them within a supply chain structure.

As we will emphasize in the following section, the decision variables of the mathematical model define a binary decision vector to which the NSGA-II and MOPSO algorithms can be applied. Moreover, we design the multi-objective optimization problem in a way such that the binary decision vector determining the entire transportation structure extends the sequential transportation structure defined by Boloori Arabani et al. [13,14] within a more complex supply chain setting. The main contribution of the current paper is the formalization of this logistic structure together with the definition of the multi-objective optimization problem in a way such that it can be analyzed and solved using standard evolutionary algorithms.

3.2. Mathematical model

The mathematical model proposed in this study is based on three different target functions. As emphasized above, the model has been designed so that the transportation sequences determining the logistic structure of the model are encoded in a binary decision vector. If $X = [X^i, X^j]$ is the decision vector of the problem, i.e., $X = [X^i_{cds\alpha}, X^i_{s_m s_n \alpha}, X^i_{scd\alpha}, X^i_{sc\alpha}, X^i_{c_m c_n \alpha}, X^i_{ccd\alpha}, X^j_{cdc\alpha}, X^j_{c_m c_n \alpha}, X^j_{ccd\alpha}], \forall \alpha \in K, i \in R, j \in H,$ and $z = [T(X^i, X^j), C(X^i, X^j), F(X^i, X^j)]$ is the target vector, the target functions and the range of the decision variables of the model can be defined as follows:

$$\begin{aligned} T: Min Z_1 = \Bigg[& \left(\sum_{i=1}^{R} \sum_{s=1}^{N} \left(T^i_{cds\alpha} + L^i_{cds\alpha}\right) \times X^i_{cds\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} T^i_{s_m s_n \alpha} \times X^i_{s_m s_n \alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} \sum_{k=1}^{K} \left(T^i_{scd\alpha} + UL^i_{scd\alpha} + T^k_{rsh}\right) \times X^i_{scd\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} \sum_{c=1}^{M} \left(T^i_{sc\alpha} + UL^i_{sc\alpha}\right) \times X^i_{sc\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{c=1}^{M} T^i_{c_m c_n \alpha} \times X^i_{c_m c_n \alpha} \right) \\ & + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} (T^j_{cdc\alpha} + L^j_{cdc\alpha} + UL^j_{cdc\alpha}) \times X^j_{cdc\alpha} \right) \\ & + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} T^j_{c_m c_n \alpha} \times X^j_{c_m c_n \alpha} \right) + \left(\sum_{i=1}^{R} \sum_{c=1}^{N} T^i_{ccd\alpha} \times X^i_{ccd\alpha} \right) \\ & + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} T^j_{ccd\alpha} \times X^j_{ccd\alpha} \right) \Bigg] \end{aligned} \tag{9}$$

$$\begin{aligned} C: Min Z_2 = \Bigg[& \left(\sum_{i=1}^{R} \sum_{s=1}^{N} C^i_{cds\alpha} \times X^i_{cds\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} C^i_{s_m s_n \alpha} \times X^i_{s_m s_n \alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} \sum_{k=1}^{K} \left(C^i_{scd\alpha} + C^k_{rsh}\right) \times X^i_{scd\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{s=1}^{N} \sum_{k=1}^{K} C^i_{sc\alpha} \times X^i_{sc\alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{c=1}^{M} C^i_{c_m c_n \alpha} \times X^i_{c_m c_n \alpha} \right) \\ & + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} C^j_{cdc\alpha} \times X^j_{cdc\alpha} \right) \\ & + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} C^j_{c_m c_n \alpha} \times X^j_{c_m c_n \alpha} \right) \\ & + \left(\sum_{i=1}^{R} \sum_{c=1}^{M} C^i_{ccd\alpha} \times X^i_{ccd\alpha} \right) + \left(\sum_{j=1}^{H} \sum_{c=1}^{M} C^j_{ccd\alpha} \times X^j_{ccd\alpha} \right) \end{aligned} \tag{10}$$

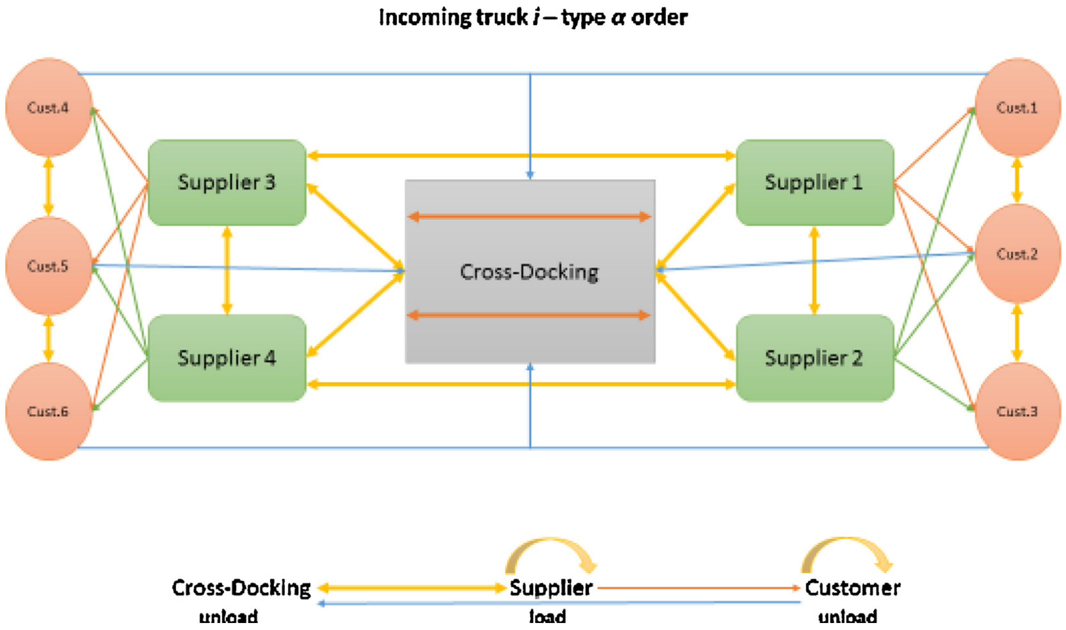


Fig. 6. Incoming truck *i*-type *α* order: potential transportation schedule.

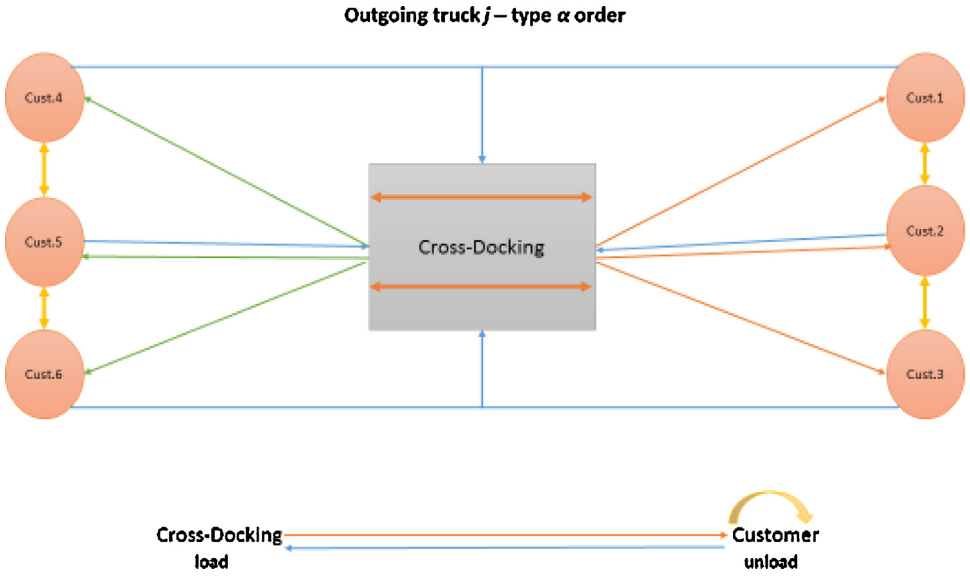


Fig. 7. Outgoing truck *j*-type *α* order: potential transportation schedule.

$$F: Min\,Z_3 = \sum_{i=1}^{R}\sum_{j=1}^{H}\sum_{c=1}^{M}\sum_{s=1}^{N}(X^i_{cds\alpha} + X^i_{s_m s_n \alpha} + X^i_{scd\alpha} + X^i_{sc\alpha} + X^i_{c_m a_n \alpha} + X^j_{cdc\alpha} + X^i_{cd\alpha} + X^j_{ccd\alpha}) \tag{11}$$

s.t.:

$$T \geq T^g_{c_n cd\alpha} \times X^g_{ccd\alpha} \qquad \forall g \in \{i\, or\, j\} \tag{12}$$

$$\sum_{i=1}^{R}\sum_{s=1}^{N} q_{\alpha s} \times X^i_{cds\alpha} = P_{c\alpha} \quad \forall \alpha \in K\, \& \forall c \in M \tag{13}$$

$$\sum_{j=1}^{H}\sum_{c=1}^{M} q_{\alpha cd} \times X^j_{cdc\alpha} + \sum_{i=1}^{R}\sum_{s=1}^{N}\sum_{c=1}^{M} q_{\alpha s} \times X^i_{sc\alpha} = P_{s\alpha} \quad \forall \alpha \in K\, \& \forall s \in N \tag{14}$$

$$\sum_{i=1}^{R}\sum_{s=1}^{N} X^i_{cds\alpha} \geq 1 \quad \forall \alpha \in K, \forall s \in N \& \forall i \in R \tag{15}$$

$$X^i_{cds\alpha} = \sum_{i=1}^{R}\sum_{s=1}^{N} X^i_{ss_n\alpha} + \sum_{i=1}^{R}\sum_{s=1}^{N} X^i_{scd\alpha} + \sum_{i=1}^{R}\sum_{s=1}^{N}\sum_{c=1}^{M} X^i_{sc\alpha} \quad \forall \alpha \in K \& \forall i \in R \tag{16}$$

$$X^i_{s_m s_n \alpha} = \sum_{i=1}^{R}\sum_{s=1}^{N} X^i_{s_n s_k \alpha} + \sum_{i=1}^{R}\sum_{s=1}^{N} X^i_{s_n cd\alpha} + \sum_{i=1}^{R}\sum_{s=1}^{N}\sum_{c=1}^{M} X^i_{s_n c\alpha} \quad \forall \alpha \in K \& \forall i \in R \tag{17}$$

$$X^{j}_{cdc\alpha} = \sum_{j=1}^{H}\sum_{c=1}^{M} X^{j}_{c_m c_n \alpha} + \sum_{j=1}^{H}\sum_{c=1}^{M} X^{j}_{ccd\alpha} \quad \forall\alpha \in K \& \forall j \in H \tag{18}$$

$$X^{i}_{c_m c_n \alpha} = \sum_{i=1}^{R}\sum_{c=1}^{M} X^{i}_{c_n c_k \alpha} + \sum_{i=1}^{R}\sum_{c=1}^{M} X^{i}_{ccd\alpha} \quad \forall\alpha \in K \& \forall i \in R \tag{19}$$

$$X^{j}_{c_m c_n \alpha} = \sum_{j=1}^{H}\sum_{c=1}^{M} X^{j}_{c_n c_k \alpha} + \sum_{j=1}^{H}\sum_{c=1}^{M} X^{j}_{ccd\alpha} \quad \forall\alpha \in K \& \forall j \in H \tag{20}$$

$$X^{i}_{sc\alpha} = \sum_{i=1}^{R}\sum_{c=1}^{M} X^{i}_{c_m c_n \alpha} + \sum_{i=1}^{R}\sum_{c=1}^{M} X^{i}_{ccd\alpha} \quad \forall\alpha \in K \& \forall i \in R \tag{21}$$

$$\sum_{i=1}^{R}\sum_{s=1}^{N} q_{\alpha s} \times X^{i}_{scd\alpha} = \sum_{j=1}^{H}\sum_{c=1}^{M} q_{\alpha cd} \times X^{j}_{cdc\alpha} \quad \forall\alpha \in K \tag{22}$$

$$\begin{aligned} T^{j}_{cds\alpha} \times X^{j}_{cds\alpha} \geq {} & (B^{i}_{s_m s_n \alpha} \times X^{i}_{s_m s_n \alpha} + B^{i}_{sc\alpha} \times X^{i}_{sc\alpha} + B^{i}_{scd\alpha} \times X^{i}_{scd\alpha}) \\ & - E(1 - P_{ij}) \quad \forall\alpha \in K, \forall i \in R, \forall j \in H, \forall s \in N \& \forall c \in M \end{aligned} \tag{23}$$

$$\begin{aligned} T^{i}_{cds\alpha} \times X^{i}_{cds\alpha} \geq {} & (B^{j}_{s_m s_n \alpha} \times X^{j}_{s_m s_n \alpha} + B^{j}_{sc\alpha} \times X^{j}_{sc\alpha} + B^{j}_{scd\alpha} \times X^{j}_{scd\alpha}) \\ & - (E \times P_{ij}) \quad \forall\alpha \in K, \forall i \in R, \forall j \in H, \forall s \in N \& \forall c \in M \end{aligned} \tag{24}$$

$$\begin{aligned} & T^{g_2}_{cdc\alpha} \times X^{g_2}_{cdc\alpha} + T^{g_2}_{sc\alpha} \times X^{g_2}_{sc\alpha} \geq (A^{g_1}_{ccd\alpha} \times X^{g_1}_{ccd\alpha} + A^{g_1}_{c_m c_n \alpha} \times X^{g_1}_{c_m c_n \alpha}) \\ & - E(1 - q_{g_1 g_2}) \quad \forall\alpha \in K, \forall s \in N \& \forall c \in M, \forall g_1 \& g_2 \in \{i \, or \, j\} \end{aligned} \tag{25}$$

$$\begin{aligned} & T^{g_1}_{cdc\alpha} \times X^{g_1}_{cdc\alpha} + T^{g_1}_{sc\alpha} \times X^{g_1}_{sc\alpha} \geq (A^{g_2}_{ccd\alpha} \times X^{g_2}_{ccd\alpha} + A^{g_2}_{c_m c_n \alpha} \times X^{g_2}_{c_m c_n \alpha}) \\ & - (E \times q_{g_1 g_2}) \quad \forall\alpha \in K, \forall s \in N \& \forall c \in M, \forall g_1 \& g_2 \in \{i \, or \, j\} \end{aligned} \tag{26}$$

$$\begin{aligned} & B^{i}_{s_m s_n \alpha} \times X^{i}_{s_m s_n \alpha} + B^{i}_{sc\alpha} \times X^{i}_{sc\alpha} + B^{i}_{scd\alpha} \\ & \times X^{i}_{scd\alpha} \geq \sum_{s=1}^{N} (T^{i}_{cds\alpha} + L^{i}_{cds\alpha}) \times X^{i}_{cds\alpha} + \sum_{s=1}^{N} T^{i}_{s_m s_n \alpha} \times X^{i}_{s_m s_n \alpha} \end{aligned} \tag{27}$$

$$\begin{aligned} & A^{g}_{c_m c_n \alpha} \times X^{g}_{c_m c_n \alpha} + A^{g}_{ccd\alpha} \times X^{g}_{ccd\alpha} \geq \sum_{s=1}^{N}\sum_{c=1}^{M} (T^{i}_{sc\alpha} + UL^{i}_{sc\alpha}) \times X^{i}_{sc\alpha} \\ & + \sum_{c=1}^{M} (T^{j}_{cdc\alpha} + L^{j}_{cdc\alpha} + UL^{j}_{cdc\alpha}) \times X^{j}_{cdc\alpha} + \sum_{c=1}^{M} T^{g}_{c_m c_n \alpha} \times X^{g}_{c_m c_n \alpha} \end{aligned} \tag{28}$$

$$\sum_{s=1}^{N} X^{i}_{cds\alpha} \times q_{\alpha s} \leq Q^{i} \quad \forall\alpha \in K \& \forall i \in R \tag{29}$$

$$\sum_{c=1}^{M} X^{j}_{cdc\alpha} \times q_{\alpha cd} \leq Q^{j} \quad \forall\alpha \in K \& \forall j \in H \tag{30}$$

$$(X^{i}_{cds\alpha}, X^{i}_{s_m s_n \alpha}, X^{i}_{scd\alpha}, X^{i}_{sc\alpha}, X^{i}_{c_m c_n \alpha}, X^{j}_{cdc\alpha}, X^{j}_{c_m c_n \alpha}, X^{i}_{ccd\alpha}, X^{j}_{ccd\alpha}) \in \{0, 1\}$$

$$\begin{aligned} & i = 1, 2, 3, \ldots, R, \quad j = 1, 2, 3, \ldots, H, \quad s = 1, 2, 3, \ldots, N, \\ & c = 1, 2, 3, \ldots, M, \quad \alpha = 1, 2, 3, \ldots, K \end{aligned}$$

All variables ≥ 0, $\forall i, j, s, c$ and α

The first target function is related to the minimization of the entire operation time in the supply chain. This is equal to the sum of the time periods which are spent transporting products from suppliers to customers, products' shipment from suppliers and cross-docking into the trucks, products' offloading from trucks to customers and cross-docking, and also the movement of products from cross-docking's entrance to its exit. Note that this target function as well as the next one represents the set of transportation processes defining the logistic structure of the model, whose sequential properties are determined using the set of constraints given in Eqs. (12)–(30). In order to illustrate this point, we have disaggregated Eq. (9) into the different transportation processes represented by each one of its terms. The resulting description is presented in Fig. 8. Note that the transportation processes described in Fig. 8 correspond to those presented in Figs. 6 and 7, when defining the cross-docking structure integrated within a supply chain.

The second target function is related to the minimization of transportation expenses in the supply chain, which is equal to the sum of expenses of incoming and out-going trucks from suppliers to customers. The third target function minimizes the number of transportations. When a truck moves from one source or one target to another source or another target, it is considered as one transport.

Constraint (12) guarantees that the whole time of transportation operations is longer than or equal to the time that the last incoming or out-going truck spends to move from customer to cross-docking. Constraint (13) guarantees that the sum of the products which are shipped by suppliers into incoming trucks *i* is equal to the demands of the customers. Constraint (14) guarantees that the sum of customers' demands is exactly equal to the products which are produced by suppliers. Constraint (15) guarantees that at least one of the incoming trucks *i* move toward the supplier in order to ship type α order. Constraint (16) guarantees that when the incoming truck *i* moves from cross-docking and arrives at the supplier's (s) place, it selects supplier (s_n), customer (c), or cross-docking (cd) as its next target. Constraint (17) guarantees that when incoming truck *i* moves from source supplier (s_m) and arrives at target supplier (s_n), it selects supplier (s_k), customer, or cross-docking as its next target. Constraint (18) guarantees that when out-going truck *j* moves from cross-docking and arrives at the customer's place, it selects customer (c_n) or cross-docking as its next target. Constraint (19) guarantees that when the incoming truck *i* moves from source customer (c_m) and arrives at the target customer's (c_n) place, it selects the customer (c_k) or cross-docking (cd) as its next target. Constraint (20) is similar to constraint (19), but it is applied to out-going truck *j*. Constraint (21) guarantees that when incoming truck *i* moves from a supplier and arrives at the customer's (c) place, it selects the customer (c_n) or cross ducking as its next target. Constraint (22) guarantees that the number of product units that are offloaded from incoming trucks *i* into cross-docking is exactly equal to the number of product units which are shipped by out-going trucks *j* toward customer (c). In fact, this constraint guarantees that the products are not held in cross-docking for a long time and all products are sent to the customers after a short period of time.

Constraint (23) guarantees that the time of entrance (incoming truck *j*) from cross-docking to supplier (s) is longer than or equal to the time which is spent by incoming truck *i* to exit from

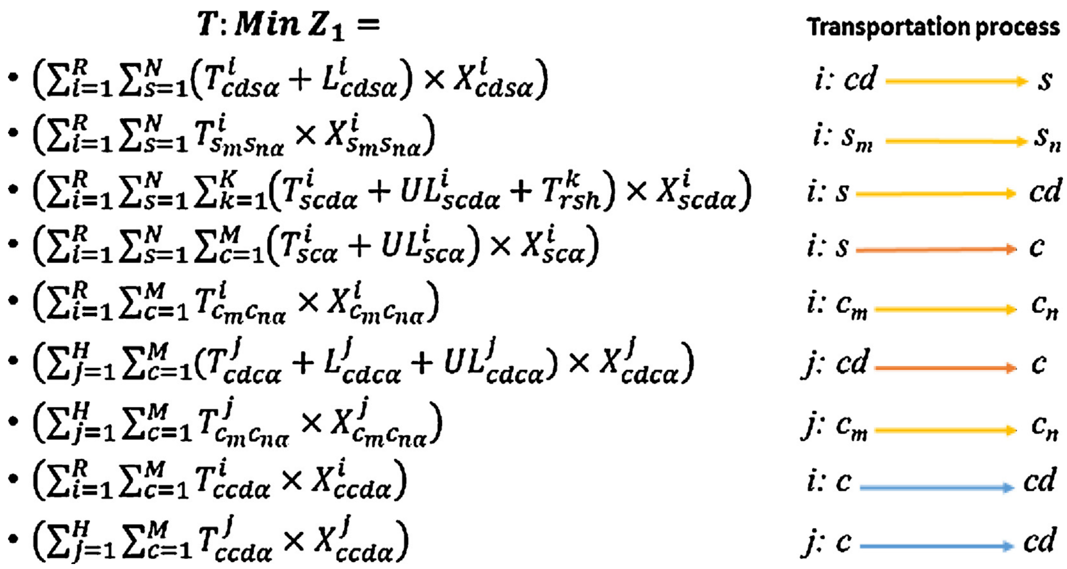

$T: Min\, Z_1 =$
Transportation process
• $\left(\sum_{i=1}^{R}\sum_{s=1}^{N}\left(T_{cdsa}^{i}+L_{cdsa}^{i}\right)\times X_{cdsa}^{i}\right)$
$i: cd \rightarrow s$
• $\left(\sum_{i=1}^{R}\sum_{s=1}^{N}T_{s_m s_n a}^{i}\times X_{s_m s_n a}^{i}\right)$
$i: s_m \rightarrow s_n$
• $\left(\sum_{i=1}^{R}\sum_{s=1}^{N}\sum_{k=1}^{K}\left(T_{scda}^{i}+UL_{scda}^{i}+T_{rsh}^{k}\right)\times X_{scda}^{i}\right)$
$i: s \rightarrow cd$
• $\left(\sum_{i=1}^{R}\sum_{s=1}^{N}\sum_{c=1}^{M}\left(T_{sca}^{i}+UL_{sca}^{i}\right)\times X_{sca}^{i}\right)$
$i: s \rightarrow c$
• $\left(\sum_{i=1}^{R}\sum_{c=1}^{M}T_{c_m c_n a}^{i}\times X_{c_m c_n a}^{i}\right)$
$i: c_m \rightarrow c_n$
• $\left(\sum_{j=1}^{H}\sum_{c=1}^{M}(T_{cdca}^{j}+L_{cdca}^{j}+UL_{cdca}^{j})\times X_{cdca}^{j}\right)$
$j: cd \rightarrow c$
• $\left(\sum_{j=1}^{H}\sum_{c=1}^{M}T_{c_m c_n a}^{j}\times X_{c_m c_n a}^{j}\right)$
$j: c_m \rightarrow c_n$
• $\left(\sum_{i=1}^{R}\sum_{c=1}^{M}T_{ccda}^{i}\times X_{ccda}^{i}\right)$
$i: c \rightarrow cd$
• $\left(\sum_{j=1}^{H}\sum_{c=1}^{M}T_{ccda}^{j}\times X_{ccda}^{j}\right)$
$j: c \rightarrow cd$

Fig. 8. Transportation processes represented by each term within Eq. (9).

supplier's (s) place. This constraint can be applied when the incoming truck i takes precedence over the incoming truck j. Constraint (24) is similar to constraint (23). This constraint can be applied in the case when the incoming truck i takes no precedence over incoming truck j. Constraint (25) guarantees that the time of entrance (incoming or out-going truck $g2$) from cross-docking (cd) or supplier (s) to the customer (c) is greater than or equal to the time which is spent by an incoming or out-going truck $g1$ to exit from customer's (c) place. This constraint can be applied when truck $g1$ takes precedence over truck $g2$. Constraint (26) is similar to constraint (25). This constraint can be applied when truck $g1$ takes no precedence over truck $g2$. Constraint (27) guarantees that the time of exit of incoming truck i from supplier (s) is longer than or equal to the sum of three time periods: the time of truck i's entrance into supplier's (s) place, the time of products' shipment from suppliers by incoming truck i, and the time of transportation by incoming truck i among suppliers. Constraint (28) guarantees that the exit time of incoming or out-going truck g from customer (c) should be longer than or equal to the sum of three time periods: the entrance time of incoming or out-going truck g to customer's (c) place, the time of products offloading from incoming or out-going truck g among the customers, and the time of transportations by incoming or out-going truck g among customers. Constraints (29) and (30) are related to the capacities of incoming and out-going trucks. Constraint (29) and (30) guarantee that the amount of products being shipped from suppliers and cross-docking does not exceed the allowed capacity of incoming and out-going trucks.

4. Numerical results

In the current section, we illustrate how the NSGA-II and MOPSO algorithms described in Section 2 are applied to solve the multi-objective optimization problem presented in Section 3. The considerable importance of evolutionary algorithms when solving complex real world optimization problems in different industrial fields has been illustrated repeatedly. For example, Oduguwa et al. [31] presented a survey of evolutionary computing applications used for solving real world problems in the manufacturing industry, while Precup and Preitl [32] developed structurally stable fuzzy control systems and illustrated their application to mechatronics. Recent applications involving the two algorithms considered in the current paper are given by Zăvoianu et al. [33], who embedded artificial neural networks within a NSGA-II algorithm to optimize the performance of electrical drives, and El-Hefnawy [34], who introduced a modified PSO algorithm to solve fuzzy bi-level single and multi-objective problems.

4.1. Data production

We use 10 sample problems to demonstrate the applicability of the two algorithms proposed in this study. We consider a realistic problem with four suppliers, five customers, one cross-docking, and 10 different products. In all 10 sample problems, the numbers of suppliers, customers, cross-dockings, and products have been assumed to be fixed. Also, the type of products has been assumed to be fixed. For all 10 sample problems, the supplies of suppliers have been presented in Table 2 and the demands of the customers have been provided in Table 3.

In all 10 sample problems, the number of trucks and their capacity, along with the selected time of product shipments from the source, the time of product offloading in the targets, the time of product movement from unloading shipments delivered by inbound trucks to loading onto outbound trucks delivery, and the distances covered by the trucks between sources and targets, have been chosen randomly based on the uniform distribution functions

Table 2
Suppliers' supplies for 10 sample problems.

Supplier	Supply									
	A	B	C	D	E	F	G	H	I	J
Supplier 1	200	0	300	0	400	50	0	0	350	0
Supplier 2	0	300	0	200	0	150	0	200	0	300
Supplier 3	0	50	150	0	100	100	400	0	0	50
Supplier 4	150	0	0	200	0	300	0	100	0	0

Table 3
Customer demands for 10 sample problems.

Customer	Demand									
	A	B	C	D	E	F	G	H	I	J
Customer 1	0	0	0	0	0	150	0	0	0	0
Customer 2	50	0	100	0	250	150	0	0	150	0
Customer 3	0	200	50	0	0	100	50	200	0	350
Customer 4	200	0	300	0	250	50	50	100	0	0
Customer 5	100	150	0	400	0	150	300	0	200	0

Table 4
Random values for the number of trucks and capacities for 10 sample problems.

Problem sample	Number of trucks	Truck capacities														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11	120	88	185	54	166	89	169	165	94	137	62	–	–	–	–
2	13	186	69	187	145	65	92	132	194	195	74	196	194	123	–	–
3	15	108	57	82	144	137	93	157	76	83	102	121	127	74	167	130
4	10	61	58	130	167	190	69	135	120	52	101	–	–	–	–	–
5	8	84	158	198	139	140	167	110	92	84	158	198	–	–	–	–
6	12	67	195	100	74	160	108	97	197	74	95	192	127	–	–	–
7	14	61	193	156	57	110	184	188	189	88	91	126	93	81	164	–
8	5	146	122	59	199	194	–	–	–	–	–	–	–	–	–	–
9	6	65	62	127	180	111	179	–	–	–	–	–	–	–	–	–
10	11	74	140	119	68	174	180	77	158	75	82	199	–	–	–	–

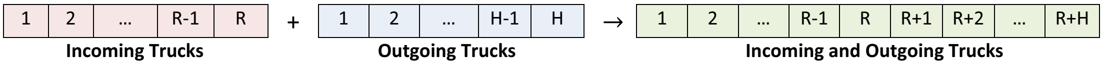


Fig. 9. The total set of incoming and outgoing trucks.

Table 5
Variables composing the transportation sequence of trucks $i \in R$ and $j \in H$ for a given type $\alpha \in K$ order.

		$X^i_{cds\alpha}$	$X^i_{s_m s_n \alpha}$			$X^i_{scd\alpha}$		$X^i_{sc\alpha}$			$X^i_{c_m c_n \alpha}$			$X^i_{ccd\alpha}$	$X^j_{cdc\alpha}$	$X^j_{c_m c_n \alpha}$			$X^j_{ccd\alpha}$
			$\mathbf{s_1}$	$\mathbf{s_i}$	$\mathbf{s_N}$			$\mathbf{s_1}$	$\mathbf{s_i}$	$\mathbf{s_N}$	$\mathbf{c_1}$	$\mathbf{c_i}$	$\mathbf{c_M}$			$\mathbf{c_1}$	$\mathbf{c_i}$	$\mathbf{c_M}$	
	$\mathbf{s_1}$	$X^i_{cds_1\alpha}$	$X^i_{s_1 s_1 \alpha}$	$X^i_{s_i s_1 \alpha}$	$X^i_{s_N s_1 \alpha}$	$X^i_{s_1 cd\alpha}$	$\mathbf{c_1}$	$X^i_{s_1 c_1 \alpha}$	$X^i_{s_i c_1 \alpha}$	$X^i_{s_N c_1 \alpha}$	$X^i_{c_1 c_1 \alpha}$	$X^i_{c_i c_1 \alpha}$	$X^i_{c_M c_1 \alpha}$	$X^i_{c_1 cd\alpha}$	$X^j_{cdc_1\alpha}$	$X^j_{c_1 c_1 \alpha}$	$X^j_{c_i c_1 \alpha}$	$X^j_{c_M c_1 \alpha}$	$X^j_{c_1 cd\alpha}$
	$\mathbf{s_2}$	.	.	.	.	.	$\mathbf{c_2}$	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$X=$	$\mathbf{s_i}$	$X^i_{cds_i\alpha}$	$X^i_{s_1 s_i \alpha}$	$X^i_{s_i s_i \alpha}$	$X^i_{s_N s_i \alpha}$	$X^i_{s_i cd\alpha}$	$\mathbf{c_i}$	$X^i_{s_1 c_i \alpha}$	$X^i_{s_i c_i \alpha}$	$X^i_{s_N c_i \alpha}$	$X^i_{c_1 c_i \alpha}$	$X^i_{c_i c_i \alpha}$	$X^i_{c_M c_i \alpha}$	$X^i_{c_i cd\alpha}$	$X^j_{cdc_i\alpha}$	$X^j_{c_1 c_i \alpha}$	$X^j_{c_i c_i \alpha}$	$X^j_{c_M c_i \alpha}$	$X^j_{c_i cd\alpha}$
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	$\mathbf{s_{N-1}}$	.	.	.	.	.	$\mathbf{c_{M-1}}$	.	.	.	.	.	.	.	.	.	.	.	.
	$\mathbf{s_N}$	$X^i_{cds_N\alpha}$	$X^i_{s_1 s_N \alpha}$	$X^i_{s_i s_N \alpha}$	$X^i_{s_N s_N \alpha}$	$X^i_{s_N cd\alpha}$	$\mathbf{c_M}$	$X^i_{s_1 c_M \alpha}$	$X^i_{s_i c_M \alpha}$	$X^i_{s_N c_M \alpha}$	$X^i_{c_1 c_M \alpha}$	$X^i_{c_i c_M \alpha}$	$X^i_{c_M c_M \alpha}$	$X^i_{c_M cd\alpha}$	$X^j_{cdc_M\alpha}$	$X^j_{c_1 c_M \alpha}$	$X^j_{c_i c_M \alpha}$	$X^j_{c_M c_M \alpha}$	$X^j_{c_M cd\alpha}$

shown in Table 1. The values obtained for the 10 problems have been given in Table 4.

4.2. Algorithms' parameters

We used the NSGA-II and MOPSO algorithms to solve this problem. For the NSGA-II algorithm, the size of population is 100. In each execution, the number of iterations is 200. Also, the probability of intersection (crossover) is assumed to be 0.9 and the probability of mutation is assumed to be 0.05. For the MOPSO algorithm, the size of population is equal to 100. In each execution, the number of iterations is 200. $c1$, $c2$, and w are 1.494, 1.494, and 0.729, respectively. It must be noted that all parameters of the algorithm have been selected empirically using a trial and error method and all problems are solved on a personal computer (Intel® Core™ DUE CPU processor Ram 4 GB) with the MATLAB 7.12.0 program.

We elaborate further on the stochastic structure of both algorithms below.

The NSGA-II and MOPSO algorithms are applied to an initial population of chromosomes (sequences of trucks) that defines a binary transportation sequence based on the set of available trucks. This population, which constitutes also the decision vector of the multi-objective cross-docking model ($X = [X^i_{cds\alpha}, X^i_{s_m s_n \alpha}, X^i_{scd\alpha}, X^i_{sc\alpha}, X^i_{c_m c_n \alpha}, X^i_{ccd\alpha}, X^j_{cdc\alpha}, X^j_{c_m c_n \alpha}, X^j_{ccd\alpha}], \forall \alpha \in K, i \in R, j \in H$) is described in detail in Fig. 9 and Table 5.

Fig. 9 illustrates how incoming and outgoing trucks are integrated within the set of total trucks considered in the multi-objective cross-docking model. Table 5 adapts a section of the decision vector to provide a more detailed description of the variables composing the transportation sequence of an incoming truck $i \in R$ and an outgoing truck $j \in H$ for a given order of type $\alpha \in K$. Note that, in order to simplify the presentation, we have assumed that $N = M$ when describing Table 5. For illustrative purposes, we have provided a particular set of transportation sequences for incoming truck i and outgoing truck j in Table 6, whose binary coding is described in Table 7.

Table 6
Transportation sequences of trucks $i \in R$ and $j \in H$ for a given type $\alpha \in K$ order.

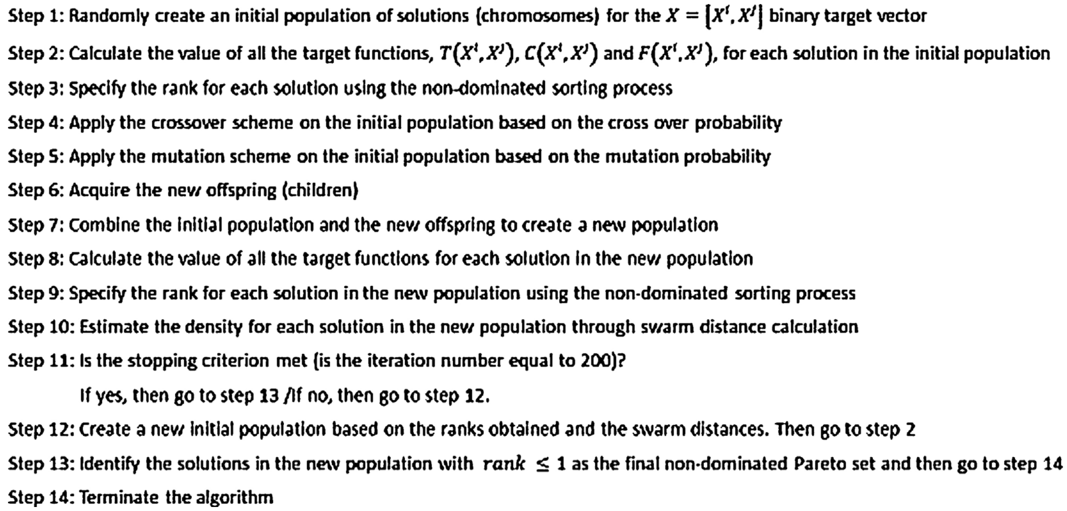
cd	*s*	*s′*	cd	*c*	*c′*	*cd*
i	1	3	$i \rightarrow j$	5	10	*j*
i	5			1	8	*i*
i	10	8	$i \rightarrow j$			

The transportation sequences of incoming truck i indicate that this truck performs three different sequences of trips for a given order of type $\alpha \in K$. Consider the first sequence of trips described in Table 6. In its initial trip, truck i moves from cross-docking and arrives at the place of supplier 1. The incoming truck i moves then from source supplier 1 and arrives at target supplier 3 before returning to cross-docking. Similarly, the outgoing truck j moves initially from cross-docking to customer 5. The outgoing truck j moves then from source customer 5 and arrives at target customer 8. After the delivery, it returns to cross-docking as its next target.

The pseudo-code of the NSGA-II algorithm applied to solve the multi-objective cross-docking model is presented in Fig. 10. The crossover and mutation schemes implemented to the binary

Table 7
Binary coding of the transportation sequences of trucks $i \in R$ and $j \in H$ described in Table 6.

		$X^i_{cds\alpha}$	$X^i_{s_m s_n \alpha}$			$X^i_{scd\alpha}$		$X^i_{sc\alpha}$			$X^i_{c_m c_n \alpha}$			$X^i_{ccd\alpha}$	$X^j_{cdc\alpha}$	$X^j_{c_m c_n \alpha}$			$X^j_{ccd\alpha}$
			s_1	s_5	s_{10}			s_1	s_5	s_{10}	c_1	c_5	c_{10}			c_1	c_5	c_{10}	
	s_1	**1**	0	0	0	0	c_1	0	**1**	0	0	0	0	0	0	0	0	0	0
	s_2	0	0	0	0	0	c_2	0	0	0	0	0	0	0	0	0	0	0	0
	s_3	0	**1**	0	0	**1**	c_3	0	0	0	0	0	0	0	0	0	0	0	0
	s_4	0	0	0	0	0	c_4	0	0	0	0	0	0	0	0	0	0	0	0
	s_5	**1**	0	0	0	0	c_5	0	0	0	0	0	0	0	**1**	0	0	0	0
$X=$	s_6	0	0	0	0	0	c_6	0	0	0	0	0	0	0	0	0	0	0	0
	s_7	0	0	0	0	0	c_7	0	0	0	0	0	0	0	0	0	0	0	0
	s_8	0	0	0	**1**	**1**	c_8	0	0	0	**1**	0	0	**1**	0	0	0	0	0
	s_9	0	0	0	0	0	c_9	0	0	0	0	0	0	0	0	0	0	0	0
	s_{10}	**1**	0	0	0	0	c_{10}	0	0	0	0	0	0	0	0	0	**1**	0	**1**



```
Step 1: Randomly create an initial population of solutions (chromosomes) for the $X = [X^i, X^j]$ binary target vector
Step 2: Calculate the value of all the target functions, $T(X^i,X^j)$, $C(X^i,X^j)$ and $F(X^i,X^j)$, for each solution in the initial population
Step 3: Specify the rank for each solution using the non-dominated sorting process
Step 4: Apply the crossover scheme on the initial population based on the cross over probability
Step 5: Apply the mutation scheme on the initial population based on the mutation probability
Step 6: Acquire the new offspring (children)
Step 7: Combine the initial population and the new offspring to create a new population
Step 8: Calculate the value of all the target functions for each solution in the new population
Step 9: Specify the rank for each solution in the new population using the non-dominated sorting process
Step 10: Estimate the density for each solution in the new population through swarm distance calculation
Step 11: Is the stopping criterion met (is the iteration number equal to 200)?
          If yes, then go to step 13 /If no, then go to step 12.
Step 12: Create a new initial population based on the ranks obtained and the swarm distances. Then go to step 2
Step 13: Identify the solutions in the new population with $rank \leq 1$ as the final non-dominated Pareto set and then go to step 14
Step 14: Terminate the algorithm
```

Fig. 10. Pseudo-code of the NSGA-II algorithm.

transportation sequence of the trucks are described below in more detail. It should be noted that the crossover and mutation schemes applied are standard. As was the case in, for example, Fallah-Mehdipour et al. [35], we have placed the emphasis of the paper on the solvability of the problem instead of the design of new variants of the algorithms. In this regard, the current model can be solved using any of the multi-objective genetic-based algorithms proposed in the cross-docking literature, see Boloori Arabani et al. [13,14], where a binary decision matrix defines the optimal transportation sequence of the trucks. A comparison among all the potential algorithms is outside the scope of the current paper, but can be performed in future extensions.

The crossover scheme implemented is a simple derivative of the standard 1-point crossover and is described in Fig. 11. Given the assumed probability of intersection (crossover), we determine randomly the number of genes that will be affected from each parent chromosome. Then, the gene strings located before the point will be copied from the opposite parent, while the gene strings located after the point will remain unchanged for each corresponding child.

The mutation scheme implemented is a derivative of the standard inversion operation, where a set of genes (trucks) is randomly selected from a subset of parent chromosomes and their values transformed into ones (i.e. a random subset of trucks is assigned a random transportation duty).

The application of the MOPSO algorithm to solve the multi-objective cross-docking model is also standard and an example based on a cross-docking system can be found in Boloori Arabani et al. [13]. However, in the current setting, we include a mutation scheme within the main loop of the MOPSO algorithm. This mutation modifies the sequence of zeros and ones determining the transportation structure of the trucks in the multi-objective model. The probability of mutation will be defined as follows

$$pm = \left(1 - \frac{(it-1)}{(Maxit-1)}\right)^{1/\mu} \tag{31}$$

where it represents the current iteration, $Maxit$ is the maximum number of iterations considered, 200 in the current setting, and $\mu = 0.1$ is the mutation rate. Note that the probability of mutation decreases in the number of iterations.

As already explained, the variable x_{it} in Eq. (5) represents the binary decision vector defining the current position of the corresponding particle (truck) within a given iteration. The mutation scheme implemented modifies the transportation sequences of a subset of randomly selected trucks. The mutation process consist of selecting a random set of trucks from the population and modifying their respective s_i probabilities (and, therefore, their relative position). This modification is based on a random evaluation process determined by their previous s_i values and the current mutation probability. In particular, the mutation probability defines an interval domain on which the previous s_i may be uniformly redefined. The resulting s_i values obtained are then used in Eq. (8) to provide a reordering of the transportation structure of the trucks based on the new x_{it} position values of each particle.

4.3. Analyzing the results

The random values for the number of trucks and capacities for the 10 sample problems and the transportation sequence for truck 1 in problem 4 using the NSGA-II algorithm are given in Tables 8 and 9, respectively.

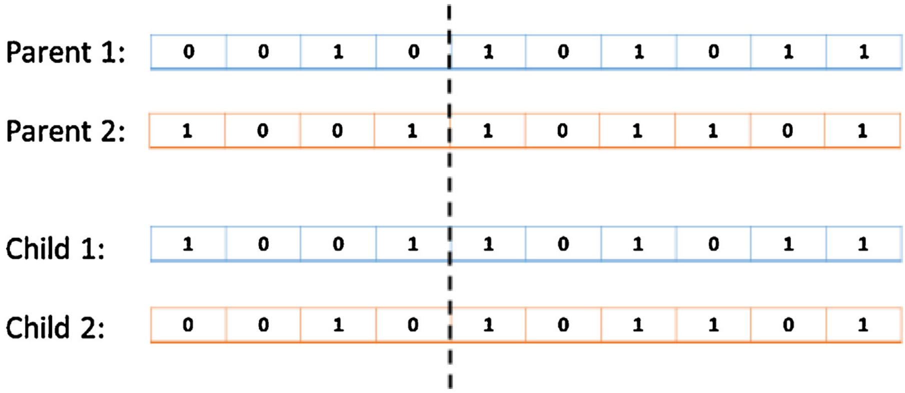

Parent 1: 0 0 1 0 1 0 1 0 1 1
Parent 2: 1 0 0 1 1 0 1 1 0 1
Child 1: 1 0 0 1 1 0 1 0 1 1
Child 2: 0 0 1 0 1 0 1 1 0 1

Fig. 11. Crossover scheme implemented within the NSGA-II algorithm.

Table 8
The transportation sequence for truck 1 in problem 4 (NSGA-II algorithm).

Target	Source	Number of products	Type of product	Transportation sequence
Cross-docking	Supplier 1	31	C	1
Cross-docking	Supplier 4	30	F	
Customer 1	Supplier 2	30	F	2
Customer 3	Supplier 2	30	H	
Customer 5	Supplier 4	61	D	3
Cross-docking	Supplier 2	30	B	4
Cross-docking	Supplier 2	30	J	
Customer 5	Supplier 2	31	D	5
Customer 3	Supplier 3	24	J	
Customer 5	Supplier 3	61	G	6
Customer 3	Supplier 2	9	B	7
Customer 5	Supplier 4	21	D	
Customer 3	Supplier 4	21	H	

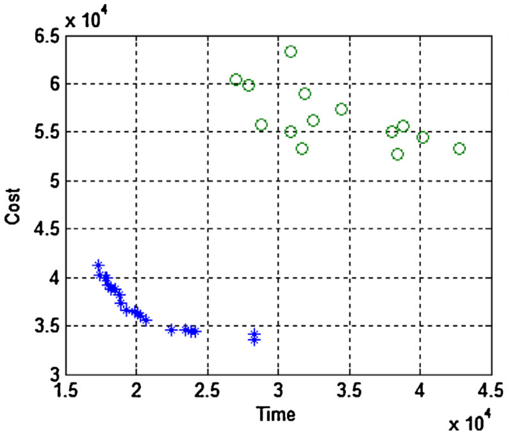


Fig. 12. Combinatory diagram of operation times and transportation costs (MOPSO, NSGA-II).

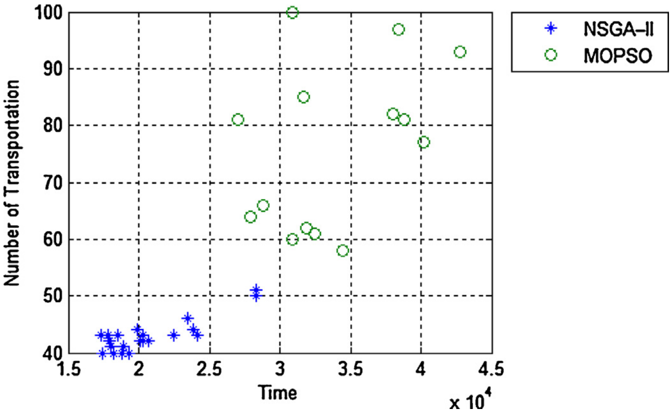


Fig. 13. Combinatory diagram of operation times and number of transportations (MOPSO, NSGA-II).

In problem 4, we consider 10 trucks with various capacities (101, 52, 120, 135, 69,190, 167, 130, 58, and 61). The trip sequence for Truck 1 (in problem 4) is given in Table 8 for the NSGA-II and in Table 9 for the MOPSO method. For example, according to Table 8, Truck 1 has 7 trip sequences. In each sequence, the type and number of products as well as the source and target have been given. In the first trip sequence for Truck 1 in problem 4 (NSGA-II algorithm), product C (31 items) is shipped from Supplier 1 and product F (30 items) is shipped from Supplier 4. Both products are sent to the cross-docking facility. Similarly, while taking into account the allowed capacities of the trucks, these trip sequences continue for the other nine trucks until all the products have been sent from the suppliers to the customers.

In Figs. 12–15, the Pareto solutions for Problem 4 are obtained by the NSGA-II and the MOPSO algorithms. Fig. 12 shows the solutions obtained by the NSGA-II and MOPSO algorithms in a two-dimensional space for the target functions of operation time and transportation expenses. As can be seen in this figure, transportation expenses decrease as the operation time increases. We can derive three results from the proposed model. Note that in two-dimensional diagrams, some solutions dominate other solutions. Fig. 13 shows the solutions obtained by the NSGA-II and MOPSO algorithms in a two-dimensional space for the target functions of operation time and the number of truck trips. It can be seen that as the number of truck trips increases, the operation time increases accordingly. Fig. 14 shows the solutions obtained by the NSGA-II and MOPSO algorithms in a two-dimensional space for transportation expenses and the number of truck trips. Fig. 14 shows that as the number of truck trips decreases, the transportations expenses increase. Finally, Fig. 15 shows the Pareto optimum

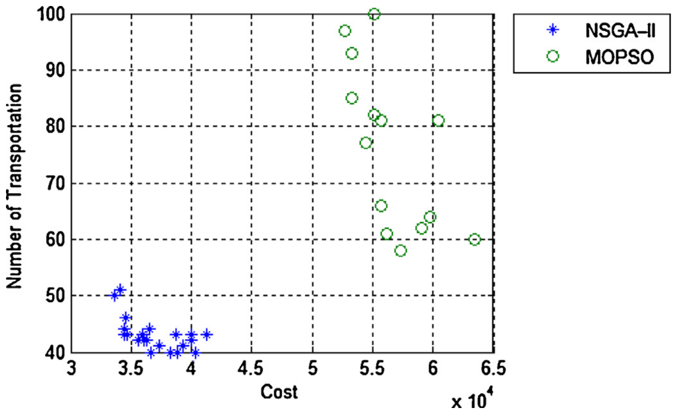


Fig. 14. Combinatory diagram of transportation costs and number of transportations (MOPSO, NSGA-II).

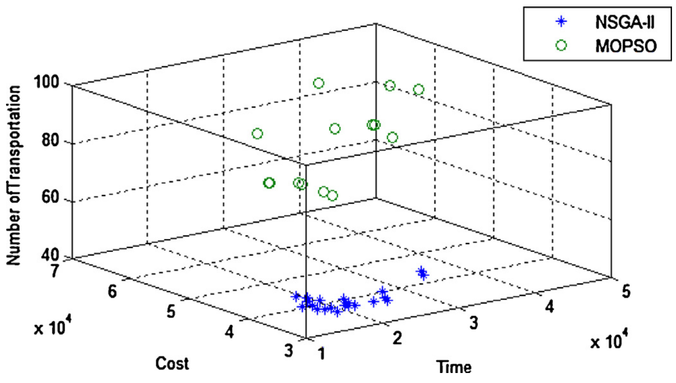


Fig. 15. Combinatory diagram of operation times, transportations costs, and number of transportations (MOPSO, NSGA-II).

solutions obtained by the NSGA-II and MOPSO algorithms, while considering all three target functions. It can be seen that as the number of truck trips decreases, the transportation costs by them decrease and the operation time increases. In a three-dimensional space, these solutions are non-dominated and no solution dominates another one. Therefore, the decision-maker can select one of the solutions based on his/her preferences.

Before comparing the performance of both algorithms, consider the applicability of the results obtained. The current paper

Table 9
The transportation sequence for truck 1 in problem 4 (MOPSO algorithm).

Target	Source	Number of products	Type of product	Transportation sequence
Customer 5	Supplier 2	9	B	1
Customer 4	Supplier 1	9	C	
Customer 5	Supplier 2	12	D	
Customer 4	Supplier 1	12	E	
Customer 5	Supplier 1	19	I	
Cross-docking	Supplier 1	3	A	2
Cross-docking	Supplier 2	3	B	
Cross-docking	Supplier 1	3	C	
Cross-docking	Supplier 2	3	D	
Cross-docking	Supplier 1	3	E	
Cross-docking	Supplier 1	3	F	
Cross-docking	Supplier 3	3	G	
Cross-docking	Supplier 2	3	H	
Cross-docking	Supplier 4	8	A	3
Cross-docking	Supplier 1	7	C	
Cross-docking	Supplier 1	5	E	
Cross-docking	Supplier 3	4	G	
Cross-docking	Supplier 3	3	J	
Customer 4	Supplier 1	6	A	4
Customer 4	Supplier 1	6	C	
Customer 5	Supplier 4	6	D	
Customer 4	Supplier 1	6	E	
Customer 5	Supplier 4	6	F	
Customer 4	Supplier 4	6	H	
Customer 5	Supplier 1	6	I	
Cross-docking	Supplier 1	9	A	5
Cross-docking	Supplier 3	10	C	
Cross-docking	Supplier 2	9	D	
Cross-docking	Supplier 2	9	H	
Cross-docking	Supplier 1	9	I	
Customer 4	Cross-docking	10	A	6
Customer 4	Cross-docking	10	C	
Customer 3	Cross-docking	13	J	
Customer 3	Cross-docking	7	G	
Customer 3	Supplier 2	8	B	7
Customer 5	Supplier 4	8	F	
Customer 3	Supplier 4	8	H	
Cross-docking	Supplier 4	11	A	8
Cross-docking	Supplier 3	10	C	
Cross-docking	Supplier 1	8	I	
Cross-docking	Supplier 2	10	J	
Customer 4	Supplier 1	21	C	9
Customer 2	Supplier 4	21	F	
Customer 5	Supplier 4	18	D	10
Customer 3	Supplier 2	14	H	
Customer 5	Supplier 1	20	I	
Customer 3	Supplier 2	5	B	11
Customer 3	Supplier 4	31	F	

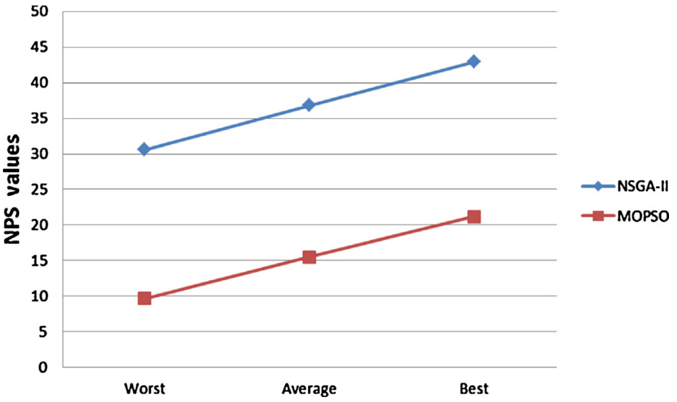


Fig. 16. Mean NPS values for all problems.

integrates cross-docking as a fundamental part of the supply chain within a unique multi-objective model that can be solved numerically using standard evolutionary algorithms. The literature generally concentrates either on the logistic processes of the chain absent cross-docking, Griffis et al. [36], or on the cross-docking process absent its supply side integration, Van Belle et al. [25]. Merging both lines of research provides a realistic approach to the modelization of cross-docking as an integrated and essential part of the supply chain. The current integrated approach relates directly to the logistic processes implemented, for example, by Sears [37] and UPS [38]. In this regard, it is widely known that many major retailers, such as Wal-Mart, utilize cross-docking services within their own distribution channels [39]. In particular, Fig. 1 represents the standard cross-docking strategy implemented by a retailer such as Wal-Mart, where the shipments from various suppliers are combined at the cross dock to create shipments for the final delivery destinations.

5. Comparative analysis

Convergence to optimum Pareto solutions and the creation of density and variety among the obtained solutions are the primary goals of every multi-objective evolutionary algorithm. Because these three goals are conflicting, there is no absolute criterion for deciding on the application of the algorithms. If we had access to such a criterion, it would have been possible to decide on the superiority of one algorithm over another one. In order to compare the quality of the solutions obtained by the NSGA-II and MOPSO algorithms, four criteria have been taken into account, including the number of Pareto solutions, the mean ideal distance measure (MIDM) (distance from the ideal point), the spacing measure (SM), and the width measure (WM) (maximum expansion). In order to compare the algorithms and to identify the most efficient one, each problem has been carried out 10 times. In other words, each algorithm has been applied 100 times for these problems. Overall, the NSGA-II and MOPSO algorithms have been applied 200 times for these 10 problems.

5.1. Number of Pareto solutions (NPSs)

Algorithms with more non-dominated solutions in the Pareto archive can provide more options to the decision maker(s) and, therefore, are more successful in determining the real Pareto optimum level [35]. The standard values of NPS for each problem (minimum, average, and maximum) have been presented in Table 10. The means of the values in Table 10 have been calculated in Fig. 16. According to Fig. 16, the NPS index is more suitable for the NSGA-II algorithm than the MOPSO algorithm.

Table 10
Minimum, average, and maximum NPS index for all sample problems.

Max	Average	Min	Algorithm	Problem set
36	30	27	NSGA-II	1
23	18	11	MOPSO	
30	26	21	NSGA-II	2
19	14	9	MOPSO	
47	38	34	NSGA-II	3
22	16	13	MOPSO	
28	23	18	NSGA-II	4
24	17	14	MOPSO	
35	28	21	NSGA-II	5
19	14	6	MOPSO	
48	39	33	NSGA-II	6
26	18	10	MOPSO	
61	53	42	NSGA-II	7
19	13	7	MOPSO	
44	41	35	NSGA-II	8
23	18	10	MOPSO	
49	43	35	NSGA-II	9
22	15	9	MOPSO	
51	47	40	NSGA-II	10
15	12	8	MOPSO	

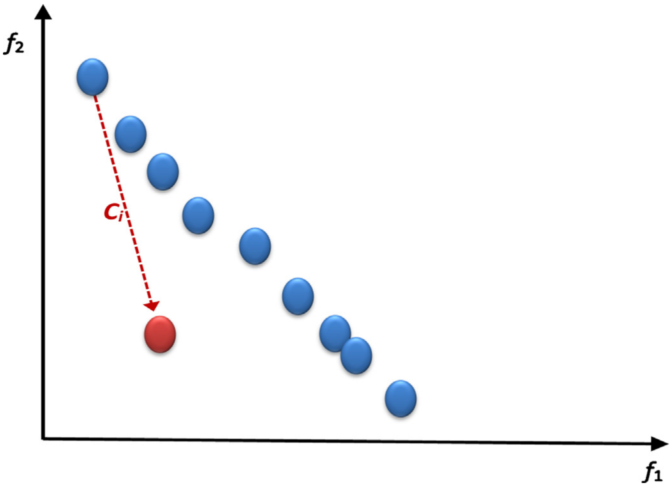


Fig. 17. Index of the mean ideal distance measure.

5.2. Mean ideal distance measure (MIDM)

This index (measure), which is used for measuring the degree of closeness to the real Pareto optimum level, is calculated by Eq. (32):

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{32}$$

In this equation, n is the number of solutions in the Pareto optimum set and c_i is the Euclidean distances of the Pareto set members from the ideal point, which is calculated by Eq. (33):

$$c_i = \sqrt{(f_{1i} - f_1^*)^2 + (f_{2i} - f_2^*)^2 + \cdots + (f_{mi} - f_m^*)^2} \tag{33}$$

In Eq. (33), f_{mi} is the value of the target function m for solution i, while f_m^* represents the ideal point (real Pareto optimum value) for the target function m. It is clear that in the comparison of Pareto optimum sets, the sets with a lower value for this index are better positioned (closer to) with respect to the real Pareto optimum level [40]. Fig. 17 shows a schematic view of the MIDM (distance from the ideal point).

The standard values of the MIDM for each problem (minimum, average, and maximum) have been presented in Table 11. The means of the values in Table 11 have been calculated and presented

Table 11
Minimum, average, and maximum MIDM index for all sample problems.

Max	Average	Min	Algorithm	Problem set
4632.1	4114.89	3596.24	NSGA-II	1
1405.18	10,965.53	7601.47	MOPSO	
2541.63	1728.19	997.17	NSGA-II	2
4927.25	4101.4	3582.82	MOPSO	
1254.46	796.91	237.86	NSGA-II	3
18,021.74	13,432.95	9684.3	MOPSO	
1498.6	1058.84	524.12	NSGA-II	4
15,749.63	9239.72	4562.27	MOPSO	
8214.97	4507.89	1742.34	NSGA-II	5
17,696.81	10,641.15	4456.1	MOPSO	
4072.64	2858.25	1418.38	NSGA-II	6
14,874.85	861.22	2145.9	MOPSO	
8952.49	4316.95	1389.71	NSGA-II	7
18,812.3	10,675.78	3464.28	MOPSO	
6978.94	4161.04	2104.56	NSGA-II	8
18,292.27	12,162.71	6843.14	MOPSO	
2245.43	1276.95	376.06	NSGA-II	9
10,202.61	5927.07	1473.41	MOPSO	
6147.82	3106.63	1520.32	NSGA-II	10
7962.17	4221.42	1287.29	MOPSO	

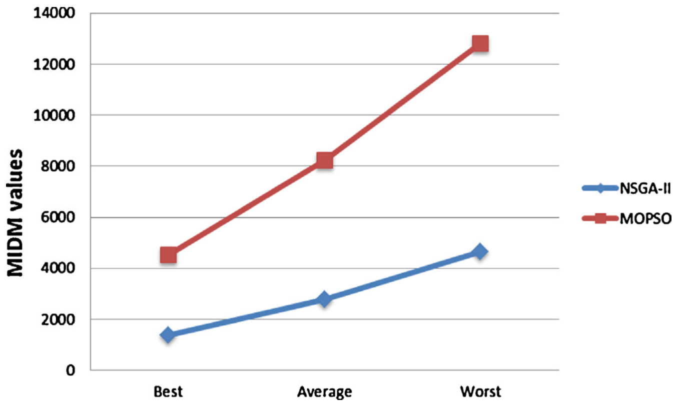


Fig. 18. Mean MIDM values for all problems.

in Fig. 18. According to Fig. 18, the MIDM index is more suitable for the NSGA-II algorithm than for the MOPSO algorithm.

5.3. Spacing measure (SM)

This index (measure), which was introduced by Schott [41], shows the relative distance between successive solutions by Eq. (34):

$$s = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \tag{34}$$

In this equation, $|Q|$ is the size of Pareto archive. d_i and d can be calculated by Eqs. (35) and (36):

$$d_i = \min_{k\in Q\wedge k\neq i}\sum_{m=1}^{M}\left|f_m^i - f_{m-1}^k\right| \tag{35}$$

$$\bar{d} = \sum_{i=1}^{|Q|}\frac{d_i}{|Q|} \tag{36}$$

In fact, the SM is the standard deviation of the d_i values. When the solutions are arranged normally beside each other, the value of the spacing (S) is small. Therefore, those algorithms whose final non-dominated solutions have a small spacing value are more favorable. The standard values of the SM for each problem (minimum, average, and maximum) have been given in Table 12. In Fig. 19, the means of these values (minimum, average, and maximum) have been presented for both algorithms. According to Fig. 19, the NSGA-II algorithm performs better than the MOPSO algorithm.

Table 12
Minimum, average, and maximum SM index for all sample problems.

Max	Average	Min	Algorithm	Problem set
162.34	78.51	0	NSGA-II	1
3214.81	1627.76	0	MOPSO	
501.2	246.36	0	NSGA-II	2
1889.67	939.66	0	MOPSO	
43.12	20.42	0	NSGA-II	3
6353.71	2275.64	0	MOPSO	
638.91	321.5	0	NSGA-II	4
2834.76	1417.54	0	MOPSO	
611.16	304.73	0	NSGA-II	5
4120.32	2022.39	0	MOPSO	
324.41	160.83	0	NSGA-II	6
34,330.47	17,164.58	0	MOPSO	
837.1	416.33	0	NSGA-II	7
37,638.28	1634.11	0	MOPSO	
321.44	167.39	0	NSGA-II	8
3772.92	1880.9	0	MOPSO	
300.78	147.71	0	NSGA-II	9
1311.57	648.22	0	MOPSO	
551.2	273.13	0	NSGA-II	10
2087.69	1038.29	0	MOPSO	

Table 13
Minimum, average, and maximum WM index for all sample problems.

Max	Average	Min	Algorithm	Problem set
1682.14	12,027.11	536.26	NSGA-II	1
2917.51	21,204.82	1420.36	MOPSO	
5792.38	4553.72	3152.12	NSGA-II	2
10,913.28	8540	6751.21	MOPSO	
2510.61	1751.13	1076.9	NSGA-II	3
2984.54	21,351.3	1427.75	MOPSO	
4450.81	2762.68	1176.44	NSGA-II	4
25,745.99	16,075.71	109,947.7	MOPSO	
14,218.76	9104.03	5271.11	NSGA-II	5
21,315.64	16,565.06	12,372	MOPSO	
9927.21	6505.4	2584.19	NSGA-II	6
19,142.38	16,357.35	14,098.74	MOPSO	
12,759.42	7791.61	2584.49	NSGA-II	7
25,376.77	17,827.27	8675.61	MOPSO	
13,914.33	8332.97	4142.27	NSGA-II	8
32,417.78	21,011.66	10,163.5	MOPSO	
3087.91	1967.04	729.54	NSGA-II	9
13,357.82	9918.859	4979.21	MOPSO	
7248.11	4954.584	2379.57	NSGA-II	10
9910.65	6738.142	3073.04	MOPSO	

5.4. Width measure (WM)

The WM (maximum expansion) measure, which was introduced by Zitzler et al. [42], is the diameter length of the cube that is created by the final values of the non-dominated solutions set in the target space. Therefore, as this index increases, the Pareto archive solutions are expanded. This index is calculated by Eq. (37) as follows:

$$D = \sqrt{\sum_{i=1}^{M}(\max_{i=1:|Q|}f_m^i - \min_{i=1:|Q|}f_m^i)^2} \tag{37}$$

The index (measure) of maximum width shows the Pareto archive solutions. The standard values of the measure for each sample problem (minimum, average, and maximum) have been given in Table 13. In Fig. 20, the means of these values (minimum,

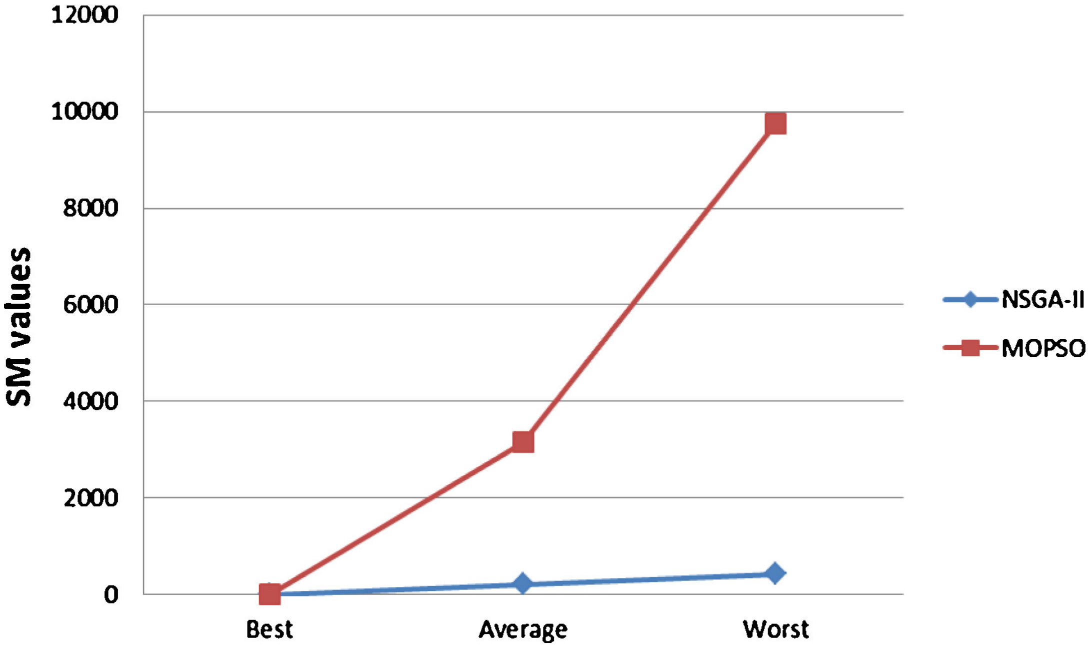


Fig. 19. Mean SM values for all problems.

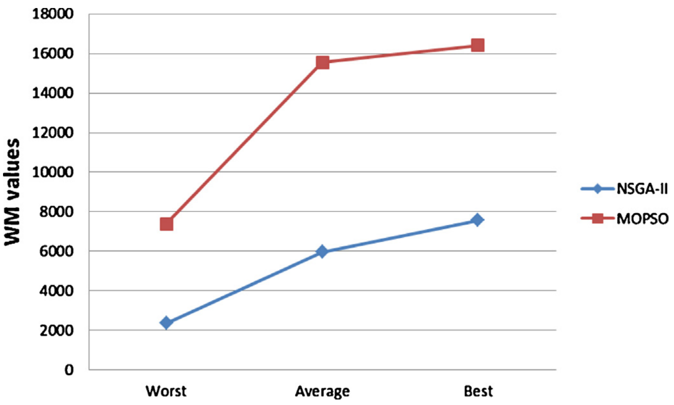


Fig. 20. Mean WM values for all problems.

average, and maximum) have been presented for both algorithms. Based on the WM index, the MOPSO algorithm performs better than the NSGA-II algorithm.

6. Conclusion and future research directions

One of the most important factors when implementing successful supply chain management is the ability to efficiently control the physical flow of materials and goods in the supply chain. Various methods have been proposed to increase customer satisfaction and reduce costs in supply chains. Cross-docking is an innovative storing strategy which has great potential for controlling distribution expenses in supply chains and satisfying customer needs. Cross-docking can also facilitate on-time distribution, increase supply-demand compatibility and utilize transportation assets more efficiently.

In this study, a multi-objective mathematical model was presented whose aim is to minimize the operation time, the transportation costs, and the number of truck trips in a supply chain and cross-docking facility. Two evolutionary meta-heuristic algorithms were used to solve the model: the NSGA-II and the MOPSO algorithms. In addition, the NPS, the MIDM, the SM, and the WM measures were used to compare and to select the best Pareto solutions. The results obtained by these algorithms show that using the Pareto archive size index (measure), the NSGA-II algorithm is superior to the MOPSO algorithm. Using the MIDM, which measures the closeness to the real Pareto optimum solution, the NSGA-II algorithm is superior to the MOPSO algorithm. Using the SM, which measures the Pareto archive solutions' density, the NSGA-II algorithm performs better than the MOPSO algorithm. Using the WM, the MOPSO algorithm provides wider solutions than the NSGA-II algorithm.

It should be noted that we have implicitly imposed several simplifying assumptions when defining the multi-objective optimization problem presented in the paper in order to guarantee its numerical tractability. For example, the cross-docking process modeled in the paper has been assumed to be simpler than the ones generally considered in the literature [13,14,30]. These simplifications have been imposed in order to allow us to concentrate on the supply chain structure of the process, while considering a cross dock with, for example, unlimited capacity. Moreover, optimal decisions depending on the stochastic properties of demand together with the allocation of different cross docks in order to optimize the distribution of products among consumers have also been omitted. Note that the resulting problems would be considerably harder to deal with, independently of the evolutionary algorithms applied to solve them, due to the increasingly complex formal structure required to define the corresponding solvable multi-objective optimization problems. Considering the limitations described above, the main extensions of the current paper are defined as follows.

1. In this study, the capacity of cross-docking was assumed to be unlimited, in spite of the fact that the capacity of cross-docking is limited in real applications. Therefore, future research can assume the limitation of the cross-docking capacity in the mathematical model.
2. In this study, only one cross-docking facility and one warehouse were taken into account in the proposed model. Future research can consider more than one cross-docking facility and warehouse.

3. In this study, an optimal location for establishing a cross-docking facility was not investigated. Optimal location planning for cross-docking facilities within the supply chain can be the subject of future research.
4. In future research, target functions can be included in the mathematical model to minimize the delays and to shorten the time that trucks spend in sources and targets.

Acknowledgement

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References

[1] M. Rohrer, Simulation and cross docking, in: Proceeding of the 1995 Winter Simulation Conference, 1995, pp. 846–849.

[2] C.E. Witt, Crossdocing: concepts demand choice, Mater. Handl. Eng. 53 (7) (1998) 44–49.

[3] I.F.A. Vis, K.J. Roodbergen, Positioning of goods in a cross-docking environment, Comput. Ind. Eng. 54 (3) (2008) 677–689.

[4] Z. Miao, A. Lim, H. Ma, Trailer dock assignment problem with operational time constraint within crossdocks, Eur. J. Oper. Res. 192 (1) (2009) 105–115.

[5] S.L. Tang, H. Yan, Pre-distribution vs. post-distribution for cross-docking with transshipments, Omega 38 (3–4) (2010) 192–202.

[6] Y.H. Lee, J.W. Jung, K.M. Lee, Vehicle routing scheduling for cross-docking in the supply chain, Comput. Ind. Eng. 51 (2) (2006) 247–256.

[7] M.A. Waller, C.R. Cassady, J. Ozment, Impact of cross-docking on inventory in a decentralized retail supply chain, Transp. Res. E: Logist. Transp. Rev. 42 (5) (2006) 359–382.

[8] H. Yan, S.L. Tang, Pre-distribution and post-distribution cross-docking operations, Transp. Res. E: Logist. Transp. Rev. 45 (6) (2009) 843–859.

[9] F. Chen, C.Y. Lee, Minimizing the makespan in a two-machine cross-docking flow shop problem, Eur. J. Oper. Res. 193 (1) (2009) 59–72.

[10] F. Chen, K. Song, Minimizing makespan in two-stage hybrid cross docking scheduling problem, Comput. Oper. Res. 36 (6) (2009) 2066–2073.

[11] B. Vahdani, M. Zandieh, Scheduling trucks in cross-docking systems: robust meta-heuristics, Comput. Ind. Eng. 58 (1) (2010) 12–24.

[12] C.J. Liao, Y. Lin, S.C. Shih, Vehicle routing with cross-docking in the supply chain, Expert Syst. Appl. 37 (10) (2010) 6868–6873.

[13] A.R. Boloori Arabani, S.M.T. Fatemi Ghomi, M. Zandieh, Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage, Expert Syst. Appl. 38 (3) (2011) 1964–1979.

[14] A.R. Boloori Arabani, M. Zandieh, S.M.T. Fatemi Ghomi, Multi-objective genetic-based algorithms for a cross-docking scheduling problem, Appl. Soft Comput. 11 (8) (2011) 4954–4970.

[15] A.R. Boloori Arabani, M. Zandieh, S.M.T. Fatemi Ghomi, A cross-docking scheduling problem with sub-population multi-objective algorithms, Int. J. Adv. Manuf. Technol. 58 (2012) 741–761.

[16] R. Sadykov, Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost, Ann. Oper. Res. 201 (2012) 423–440.

[17] Y. Kuo, Optimizing truck sequencing and truck dock assignment in a cross docking system, Expert Syst. Appl. 40 (14) (2013) 5532–5541.

[18] R. Larbi, G. Alpan, P. Baptiste, B. Penz, Scheduling cross docking operations under full, partial and no information on inbound arrivals, Comput. Oper. Res. 38 (6) (2011) 889–900.

[19] T.W. Liao, P.J. Egbelu, P.C. Chang, Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations, Appl. Soft Comput. 12 (11) (2012) 3683–3697.

[20] T.W. Liao, P.J. Egbelu, P.C. Chang, Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations, Int. J. Prod. Econ. 141 (1) (2013) 212–229.

[21] S. Meysam Mousavi, R. Tavakkoli-Moghaddam, A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain, J. Manuf. Syst. 32 (2) (2013) 335–347.

[22] A. Bellanger, S. Hanafi, C. Wilbaut, Three-stage hybrid-flowshop model for cross-docking, Comput. Oper. Res. 40 (4) (2013) 1109–1121.

[23] S. Meysam Mousavi, B. Vahdani, R. Tavakkoli-Moghaddam, H. Hashemi, Location of cross-docking centers and vehicle routing scheduling under uncertainty: a fuzzy possibilistic–stochastic programming model, Appl. Math. Model. 38 (7–8) (2014) 2249–2264.

[24] D. Agustina, C.K.M. Lee, R. Piplani, Vehicle scheduling and routing at a cross docking center for food supply chains, Int. J. Prod. Econ. 152 (2014) 29–41.

[25] J. Van Belle, P. Valckenaers, D. Cattrysse, Cross-docking: state of the art, Omega 40 (6) (2012) 827–846.

[26] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley and Sons, New York, 2001.

[27] K. Deb, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.

[28] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.

[29] K.E. Parsopoulos, M.N. Vrahatis, Particle swarm optimization method in multiobjective problems, in: Proceedings of the ACM 2002 Symposium on Applied Computing (ACM SAC 2002), Madrid, Spain, 2002, pp. 603–607.

[30] R. Chen, B. Fan, G. Tang, Scheduling problems in cross docking, in: D.-Z. Du, X. Hu, P.M. Pardalos (Eds.), Combinatorial Optimization and Applications, Lecture Notes in Computer Science, vol. 5573, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 421–429.

[31] V. Oduguwa, A. Tiwari, R. Roy, Evolutionary computing in manufacturing industry: an overview of recent applications, Appl. Soft Comput. 5 (3) (2005) 281–299.

[32] R.-E. Precup, S. Preitl, Stability and sensitivity analysis of fuzzy control systems. Mechatronics applications, Acta Polytech. Hung. 3 (1) (2006) 61–76.

[33] A.-C. Zăvoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, E.P. Klement, Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives, Eng. Appl. Artif. Intell. 26 (8) (2013) 1781–1794.

[34] N.A. El-Hefnawy, Solving bi-level problems using modified particle swarm optimization algorithm, Int. J. Artif. Intell. 12 (2) (2014) 88–101.

[35] E. Fallah-Mehdipour, O. Bozorg Haddad, M. Rezapour Tabari, A. Mario, Extraction of decision alternatives in construction management projects: application and adaptation of NSGA-II and MOPSO, Expert Syst. Appl. 39 (2012) 2794–2803.

[36] S.E. Griffis, J.E. Bell, D.J. Closs, Metaheuristics in logistics and supply chain management, J. Bus. Logist. 33 (2) (2012) 90–106.

[37] R. Kulwiec, Crossdocking as a supply chain strategy, Target 20 (3) (2004) 28–35.

[38] G. Forger, Ups starts world's premiere cross-docking operation, Mod. Mater. Handl. 50 (13) (1995) 36–38.

[39] G. Stalk, P. Evans, L.E. Shulman, Competing on capabilities: the new rules of corporate strategy, Harv. Bus. Rev. 70 (2) (1992) 57–69.

[40] J. Behnamian, S.M.T. Fatemi Ghomi, M. Zandieh, A multi-phase covering Pareto optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid meta-heuristic, Expert Syst. Appl. 36 (8) (2009) 11057–11069.

[41] J.R. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithms Optimization (Master's thesis), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[42] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. J. 8 (2) (2000) 125–148.