Advances in Industrial Engineering, June 2026, 60(1): 1-29
DOI: 10.22059/aie.2025.395242.1946
RESEARCH PAPER

Integrated Multi-Agent Problem of Vehicle Routing and Cross-Dock Scheduling Considering Group Purchasing Strategies, Perishability of the Commodities and Requirements of the Customers

Hadi Sarmadi[1] and Mohammad Mahdi Nasiri[2 *]

[1]*Ph.D. Candidate, Department of Industrial Engineering, Kish International Campus, University of Tehran, Tehran, Iran.*
[2]*Professor, School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.*

Received: 13 May 2025, Revised: 05 September 2025, Accepted: 08 September 2025

Abstract

During the recent years, the companies in a wide range of industries have to design their activities in such a way to reduce the costs. A most popular way to reduce the costs in logistics is cross-docking. It is a strategy which is used to serve different purposes including the fast consolidation of received volume of commodities from suppliers, improving the responsiveness by shortening delivery lead time, reducing the inventory holding costs, eliminating spoilage costs of commodities, reducing transportation costs by employing full truck loading policy etc. The objective of this paper is to develop a mixed integer linear programming (MILP) model considering supplier selection and order allocation, perishability of commodities, group purchasing strategy and multi-agent scheduling into the well-known vehicle routing problem with cross-docking. Some small-sized test instances are applied to validate the new proposed model. A weighted-sum method is applied to solve small-sized instances. Then sensitivity analysis of the new proposed model is performed on the key parameters of the objective functions so that the supply decisions are evaluated while the parameters of the distribution costs are changed. Due to NP-hardness of the new proposed problem, two meta-heuristic algorithms including NSGA-II and MOPSO are applied to solve a wide range of instances. The obtained results by applying statistical hypothesis tests are compared through six different criteria. Also, an ordering technique that is called Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is applied to rank the meta-heuristic approaches.

Keywords:
Cross-Dock Scheduling,
Group Purchasing Strategy,
Multi-Agent Scheduling,
Perishable Commodities,
Vehicle Routing Problem.

Introduction

1.1. Cross-docking

Nowadays, cross-docking is a useful logistics strategy which is used by many companies in a wide range of industries to serve different goals including the rapidly consolidation of shipments from suppliers, shortening delivery lead time, achieving the scale economy of outgoing delivery means employing full truck loading policy instead of less than truck loading

* Corresponding author: (Mohammad Mahdi Nasiri)
Email: mmnasiri@ut.ac.ir

policy and at the end the reduction of total costs (Amini and Tavakkoli-Moghaddam (2016)). Unlike traditional warehouses, in cross-docking only receiving and shipping activities occur so that, when the customer order comes, the products will be picked and shipped to the customer (Agustina et al. (2010)). Two review papers which presented by Agustina et al. (2010) and Van Belle et al. (2012) show increasingly trend to subject of cross-docking. But on the other hand, applying successfully a cross-docking is dependent on a few crucial factors including the demand rate, shortage costs and the volume of the commodities distributed in the supply-chain (Apte and Viswanathan (2000)). As a matter of fact, cross-docking is a better choice when demand rates are sufficiently predictable and stable, shortage costs are low and large volume of commodities is supplied in the supply-chain (Van Belle et al. (2012)). For more review on cross-docking the researchers are referred to Nasiri et al. (2018); Boysen, and Flinder (2010).

As Van Belle et al. (2012) believe that the most important problems studied for cross-docking are including location of cross-dock, design of layout, networks of cross-docking, vehicle routing, assignment of dock-doors, scheduling of trucks. In addition, Stephan and Boysen (2011) presented another categorization of considered fields at cross-docking studies including location planning, layout planning, destination assignment, truck scheduling and inner transport scheduling. By the way, in this section we concentrate on reviewing of papers in which vehicle routing problem with cross-docking problems (VRPCD) have studied. According to aspects of scheduling, Nasiri et al. (2018) categorize the papers studying the VRPCD into the three main categories. At the first category of the reviewed papers, the authors have concentrated on the routing and assigning aspects and have ignored the scheduling viewpoints (for detailed study, see Alinaghian et al. (2016); Enderer et al. (2017); Nikolopoulou et al. (2017)). Also, Nasiri et al. (2018) state at their paper that the second category of the papers which studied by them, considered the aspect of the scheduling of the transportation process of the VRPCD but ignored some of scheduling aspects of the operations at the cross-docking processes such as sequencing and scheduling of the dock-door operations and the presented problem of vehicle routing and scheduling with cross-docking called as VRSPCD (for more study look at Yu et al. (2016); Yin and Chuang (2016); Maknoon and Laporte (2017); Grangeir et al. (2017)). At the third category of the studied papers at the research which provided by Nasiri et al. (2018), the authors have considered the aspects of the scheduling of both the transportation process and the operations of the cross-docking processes of the VRPCD (for detailed review the reader is recommended to study Yin et al. (2016); Rahbari and Nasiri (2017); Grangier et al. (2021); Wang et al. (2017)).

There is a study in which the authors (Nasiri et al. (2018)) have considered both assignment problem of orders and suppliers and supplier selection and order allocation problem simultaneously. But they did not consider some of the scheduling aspects which affect the problem.

1.2. Group purchasing

Ahmadi et al. (2018) state at their study that over the last years, many researchers have interested to apply the concept of group purchasing in their studies at a wide research area of industries. They also emphasis that the concept of group purchasing arises from health-care industry where a group purchasing organization (GPO) as an entity aggregates the purchasing order volumes of health-care providers in order to reduce the costs by profiting quantity discounts from manufacturers and suppliers. In this paper, the cross-dock owner is considered as a Group Purchasing Organization (GPO) to reduce supply costs through quantity discounts. Yan et al. (2017) compare the group buying versus individual purchasing and bring many evidences for the benefits of group purchasing. Sazvar et al. (2014) presented a mathematic modeling of quantity discount.

1.3. Perishability of the commodities

One of the most significant challenges in today's food industry is managing perishability and controlling the quality of the products throughout the supply chain (Rong et al. (2011)). From a customer standpoint, buyers are anxious about having as much data as possible, particularly with commodities which may have an impact on health, such as fresh and food products that are highly perishable. In addition, goods freshness is highly associated to consumer satisfaction and a better managing of products' perishability may enhance effectiveness and include some strong competitive advantages (Amorim et al. (2013)).

1.4. Multi-agent scheduling

It may happen that in a complicated problem some operations have to be measured by a different performance criterion and the others have to be evaluated by another measure. This is the concept of multi-agent approach. There are at least two agents in a multi-agent problem and each of the agents is interested in a set or subset of objects and has its own performance metric. However, all of the objects have common resources to reach the best compromise solution in order to enhance the responsiveness (Agnetis et al. (2014)). Several fields such as Cross-docking distribution; Rescheduling problems; Aircraft landings; Project scheduling; Railway scheduling and Communication networks are applications of multi-agent approach (Agnetis et al. (2014)). One of the most important applications of the multi-agent scheduling can be in optimizing the operations at a cross-dock. Agnetis et al. (2014) state that if there are two or more customers, one or a set of them can be as an agent and the others or all of them can be as another agent.

1.5. Problem solving methodologies

Multi-agent scheduling problems while each of the agents has a separate cost function could be solved in a multi-objective context applying appropriate approaches (Agnetis et al. (2014)). Mostly, due to the NP-hard nature of these problems exact solving of real-sizes is impossible or takes a very long time. Thus, usually approximated solution methods like meta-heuristic approaches are applied to tackle them. According to the existing studies on literature of similar problems at supply chain environment, several researchers are interested to use two meta-heuristic algorithms including multi-objective particle swarm optimization (MOPSO) and non-dominated sorting genetic algorithm (NSGA-II) more than the others to solve the multi-objective problems. For example, Farmand et al. (2021) state that for solving multi-objective form of problem which they studied through 10 papers, at nine studies (it means 90 percent) NSGA-II is only or one of the problem's solving approaches and at six studies (60 percent) both of two mentioned algorithms applied to solve the problems. To the best of the authors' knowledge, because of simplicity at calculations of MOPSO and NSGA-II algorithms, great number of researchers are interested to use them for solving the multi-objective problems.

1.6. Paper scope and outline

The most important criterion in many of supply chain papers which used to optimize the decisions, is costs particularly costs of supplying and purchasing. On the other hand, responsiveness and rapid distribution of the products that may be in conflict with the costs, is the other most interesting criterion in the supply chain (Nasiri et al. (2018)). On one another hand, managing perishability of products may have a significant role to minimize penalty costs and to improve competitive advantages by causing good relations and better integration between the supply chain network of organizations (Amorim et al. (2013)).

The aim of presented paper is to consider the benefits of group purchasing strategies and managing perishability of products into the integrated problem of supplier selection and order allocation and vehicle routing problem with cross-docking through an approach of multi-agent scheduling. There is a major challenging benefit from this modeling; the profits of integrated decision making about the supply-side and distribution-side of supply chain. So, the biggest contribution of this article is to propose a MILP model for an integrated problem of supplier selection considering group purchasing and multi-agent scheduling of distribution the products and managing perishability through a cross-dock in order to reduce the costs and also to enhance the effectiveness and responsiveness of the supply chain simultaneously.

In the proposed model, applying group purchasing strategies leads to a reduction in supply costs through quantity discounts. Additionally, managing perishability results in reduced spoilage costs of commodities at the cross-dock and enhances the effectiveness of cross-docking operations. On the other hand, applying multi-agent scheduling creates a trade-off between minimizing costs and enhancing the responsiveness of the supply chain by considering urgent deliveries.

The remainder of the paper on hand is summarized as follows. The next section describes the new proposed problem, notations and mathematical model and analyzes the model by applying weighted-sum method. With regards to the complexity of the problem, in section 3 we provide two meta-heuristic algorithms to solve real-scale problems. Section 4 reports numerical results and compares the results of the proposed solution approaches and finally in section 5 we conclude about the study.

2. Problem illustration and modeling

In this paper the following points for the first time are added to the vehicle routing problem with cross-docking in order to minimize the costs and to improve the responsiveness of the model.

1. Supplier selection and allocation of order quantity of commodities to them with regards to the price of commodities and also, regards to the purchasing levels based on the group purchasing strategies. This point implies the new suggested model to present a pre-distributed model which gains the profits of post-distribution. In other words, providing degrees of freedom in ordering each part of the demands from different suppliers shifts the interchangeability of the commodities from the cross-dock to the supply side. this can intrinsically include the profits of post-distribution;
2. Considering two agents which the first agent includes the customers who are interested in the soonest delivery. And, the second agent includes customers who are concern about the time-window of delivery. The main aim of categorizing the problem into the two agents is improvement of responsiveness of the model by reducing the tardiness and earliness costs and decreasing delivery time for customers who are interested in urgent deliveries;
3. Finally, the other purpose of the new suggested model is to minimize the costs associated to the agents. They are including fixed ordering costs of a seller and purchasing price, fixed and variable costs of using a vehicle, holding costs of commodities at cross-dock and spoilage costs of perishable commodities.

2.1. Problem statement

The presented new model consists of a three-echelon supply chain (Ahmadizar et al. (2015)) which includes suppliers (sellers), a distributor and customers. In this model the distributor is responsible for purchasing, transporting and dispatching services as a group purchasing organization and also as a 3PL company. Also, the distributor as decision-maker of the proposed supply chain owns (1) a cross-dock center at a predefined location with a set of strip doors and another set of stack doors; (2) a vehicle parking yard at a pre-determined location which is equipped by maintenance equipment and an expert working group; and (3) a fleet of heterogeneous vehicles which are different with each other in terms of capacity, fixed and variable costs and all of them are equipped with cooling systems. When the customers make the orders, determining the commodity type and the quantity as well as the delivery time interval, the owner of the supply chain plans the operations in order to minimize the total costs, including purchasing, transporting, consolidating at cross-dock, spoilage of commodities, inventory holding and earliness/tardiness costs. The key points of problem which should be decided about them are: (1) selection of suppliers and allocation of order quantity to them; (2) vehicle assignment to the selected suppliers as incoming vehicles; (3) scheduling of incoming vehicles in the pickup of commodities from suppliers; (4) incoming vehicle assignment to the strip doors of cross-dock; (5) incoming vehicle sequencing and scheduling for unloading operation at strip dock-doors; (6) vehicle assignment to the customers as outgoing vehicles; (7) outgoing vehicle assignment to the stack doors of cross-dock; (8) outgoing vehicle sequencing and scheduling for loading operation at stack dock-doors; (9) outgoing vehicle routing and scheduling in the delivery of commodities to the customers.

So, the new proposed problem consists of four sub-problems including (1) purchase or supply which is associated to the key point of decision 1; (2) pickup which is related to key points of decisions 2 and 3; (3) consolidation at cross-dock which is related to key points of decisions 4, 5, 7 and 8; and (4) delivery to the customers which is related to key points of decisions 6 and 9.

2.2. Assumptions

The following basic assumptions are considered at modeling of the new proposed problem: (1) The area of vehicle parking yard and also the space of temporary storage which is located at front of outgoing-side of cross-dock are not limited; (2) There is no limit on availability of equipment to load the commodities onto the vehicles at each of the suppliers and so, there is no queue at the suppliers; (3) The price of each commodity which are offered by a supplier is dependent on order quantity of that commodity and may differ from a supplier to another one; (4) There is a capacity limitation of supplying each of the commodities for every supplier; (5) The demand of a customer can be split to more than one supplier but the splitting the demand of a customer for a commodity type is not permitted and it should be supplied by only one supplier and also should be transported from the supplier to the cross-dock by only one incoming vehicle (like Nasiri et al. (2018)); (6) There is a heterogeneous fleet of vehicles equipped with cooling systems and they are used at both pickup and delivery side. All of them are available at the beginning of the day at the vehicle parking yard and also can be used at most one time during a planning horizon. Each of the vehicles may differ with another one in terms of capacity, fixed and variable costs; (7) Preemption at loading or unloading process on a vehicle is not permitted; (8) At cross-dock there are two incoming and outgoing sides and also there are different loading or unloading equipment and the time of loading or unloading of the same commodity at dock-doors may be different; (9) At the cross-dock some value adding works (i.e. sorting or labeling) on commodities is provided; (10) The time needed to changeover the vehicles in front of the dock-doors for all vehicles is the same; (11) The moving operations of a freight inside of the cross-dock can be started only after completely unloading of the incoming vehicle; (12) An incoming vehicle starts the pickup route from the vehicle parking yard and visit a supplier and then goes to the cross-dock and at the end immediately after unloading the freight returns to the vehicle parking yard. But an outgoing vehicle starts the delivery routes from the vehicle parking yard and then visit the cross-dock and after loading the freight visit one or more customer(s) and at the end returns to the vehicle parking yard. Also, at delivery side the demand of a customer should be delivered by only one outgoing vehicle; (13) The time needed for completing whole process should be limited to the planning time horizon. It is notable that considering variable horizons needs to add more constraints into the model and leads to more complexity at calculations.

2.3. Notation of new proposed model

The materials which applied to formulate the new proposed model are as followings:

2.3.1. Sets:

S	set of sellers (suppliers)
B	set of customers (buyers)
BB	a subset of B includes a few numbers of customers who are concerning intensely to receive their order as soon as possible
C	set of commodity types
CC	a subset of C includes perishable commodity types
N	set of real vehicles
I	set of strip doors
J	set of stack doors
E	set of purchasing levels

2.3.2. Indices:

s	index of sellers, $s \in S$
b	index of customers, $b \in B$
c	index of commodity types, $c \in C$
n	index of vehicles, $n \in \{0, 1, 2, \dots, \lvert N\rvert, \lvert N\rvert + 1\}$ where $n = 0$ and $n = \lvert N\rvert + 1$ are considered as dummy vehicles
i	index of strip doors, $i \in I$
j	index of stack doors, $j \in J$
e	index of purchasing level, $e \in E$

2.3.3. Parameters:

P_{sc}^{e}	unit price of commodity type c from seller s at purchasing level e
PC_{bc}	unit spoilage cost of commodity type c at cross-dock
V_c	unit volume of commodity type c
FCA_s	fixed ordering costs of seller s
CAS_{sc}	capacity of seller s to supply commodity type c
CAV_n	capacity of vehicle n
FCV_n	fixed costs of using vehicle n
VCV_n	unit variable costs of vehicle n including transportation, loading, waiting, and unloading operations [$ per min]
H_c	unit inventory holding costs of commodity type c at cross-dock
TNT	time needed to transfer a commodity from the strip door to the stack door at cross-dock [min]
TNC	time needed to changeover a vehicle [min]
T	duration of planning time horizon [min]
TRT_s	time needed to travel between the vehicle yard and seller s [min]
TRT_0	time needed to travel between the vehicle yard and cross-dock [min]
TRT_b	time needed to travel between the vehicle yard and customer b [min]
$TRT_{bb'}$	time needed to travel from customer b to b' [min]
TRT_{0s}	time needed to travel between cross-dock and seller s [min]
TRT_{0b}	time needed to travel between cross-dock and customer b [min]
L_{sc}	loading time of a unit of commodity type c at seller s [min]
U_{ic}	unloading time of a unit of commodity type c at strip door i of cross-dock [min]
L_{jc}	loading time of a unit of commodity type c at stack door j of cross-dock [min]
U_{bc}	unloading time of a unit of commodity type c at customer b [min]
D_{bc}	demand size of customer b for commodity type c [unit]
EPC_b	penalty cost of unit earliness for customer b [$ per unit per min]
TPC_b	penalty cost of unit tardiness for customer b [$ per unit per min]
LD_b	lower bound on the due delivery time for customer b
UD_b	upper bound on the due delivery time for customer b
DD_c	upper bound on the processing duration that can be tolerated by any commodity type $c \in CC$ at cross-dock[min]
Q_c^e	lower/upper bound on the order quantity of commodity type c at purchasing level e

2.3.4. Decision variables (binary):

x_{isbcn}	is equal to 1 when commodity type c demanded by customer b is supplied by seller s and transported by vehicle n to the strip door i of cross-dock

α_{sc}^{e}	is equal to 1 when the quantity of commodity type c ordered from seller s be at purchasing level e
a_s	is equal to 1 when supplier s is used
g_{sn}	is equal to 1 when vehicle n is assigned to supplier s
er_i	is equal to 1 when the strip door i of cross-dock is used
pr_{in}	is equal to 1 when vehicle n is unloaded at the strip door i of cross-dock
$qr_{nn'}$	is equal to 1 when the incoming vehicles n & n' are unloaded at the same strip door of cross-dock and n immediately precedes n'; $n \in \{0, 1, 2, \dots, \lvert N\rvert\}$ and $n' \in \{1, 2, \dots, \lvert N\rvert, \lvert N\rvert + 1\}$
eo_j	is equal to 1 when the stack door i of cross-dock is used
po_{jn}	is equal to 1 when vehicle n is loaded at the stack door j of cross-dock
$qo_{nn'}$	is equal to 1 when the outgoing vehicles n & n' are loaded at the same stack door of cross-dock and n immediately precedes n'; $n \in \{0, 1, 2, \dots, \lvert N\rvert\}$ and $n' \in \{1, 2, \dots, \lvert N\rvert, \lvert N\rvert + 1\}$
y_{jbn}	is equal to 1 when vehicle n delivered the demand of customer b through the stack door j of cross-dock
z_{bn}	is equal to 1 when vehicle n passed from customer b at first
$zz_{bb'n}$	is equal to 1 when vehicle n passed from customer b' immediately after customer b
zl_{bn}	is equal to 1 when vehicle n passed from customer b at the end
w_{bn}	is equal to 1 when incoming vehicle n picked up any commodity ordered by customer b
k_{bc}	is equal to 1 when processing duration of commodity type c at cross-dock exceeds DD_c
δ_b	is equal to 1 when $\sum_c k_{bc} > 0$ for the customer b

2.3.5. Decision variables (positive):

dtp_n	the moment when vehicle n departs from the vehicle yard
atp_n	the moment when vehicle n arrives at the vehicle yard
ata_{sn}	the moment when incoming vehicle n arrives at seller s
atr_n	the moment when incoming vehicle n arrives at the cross-dock
ato_n	the moment when outgoing vehicle n arrives at the cross-dock
dto_n	the moment when outgoing vehicle n departs from the cross-dock
atd_{bn}	the moment when outgoing vehicle n arrives at customer b
dtc_b	the moment when order of customer b departs from cross-dock
atc_{bc}	the moment when commodity type c ordered by customer b arrives at cross-dock
eat_b	earliness time at customer b
tat_b	tardiness time at customer b

2.4. Mathematic

Cost functions associated to each of the agents and the constraints of the new proposed model are presented as followings. Considering these objective functions at the new proposed model, makes a trade-off between minimizing costs for first agent (urgent deliveries) and second agent (time-window-sensitive deliveries).

2.4.1. Cost functions:

Minimize $of_1 = of_{11} + of_{12} + of_{13} + of_{14} + of_{15} + of_{16} + of_{17}$

$$of_{11} = \sum_{s,c}\sum_{e}\sum_{i,b\in BB,n} P_{sc}^{e} D_{bc}\, x_{isbcn}\alpha_{sc}^{e}$$

$$of_{12} = \sum_{i,s,b\in BB,c,n} x_{isbcn}\left(FCA_s + FCV_n + VCV_n\left(atp_n - dtp_n\right)\right) / \sum_{i,s,b,c,n} x_{isbcn}$$

$$of_{13} = \sum_{j,b\in BB,n} y_{jbn}(FCV_n + VCV_n\left(atp_n - dtp_n\right)) \Big/ \sum_{j,b,n} y_{jbn}$$

$$of_{14} = \sum_{b\in BB,c} H_c D_{bc}\left(dtc_b - atc_{bc}\right)$$

$$of_{15} = \sum_{b\in BB,c} TPC_b D_{bc} tat_b$$

$$of_{16} = \sum_{b\in BB,c\in CC} PC_{bc}\, k_{bc}$$

$$of_{17} = \sum_{b\in BB,n} atd_{bn}$$

The first cost function which is named of_1 is related to the customers who are concerning intensely to receive their order as soon as possible (as first agent). In other words, this function calculates the costs that related to the customers who are a member of subset $\{BB\}$.

The cost components of of_1 are described as follows:

The first term (of_{11}) calculates purchasing costs. The components of second term (of_{12}) calculate costs of supplier selection and also compute the fixed and variable costs of using the incoming vehicles to transfer the orders from suppliers to cross-dock. The variable costs of using the incoming vehicles are including the total time when they depart from the vehicle yard until they go back to there. The third term (of_{13}) measures the fixed and variable costs of outgoing vehicles to transfer the demands of customers to deliver them. Also, at this term the variable costs of using the outgoing vehicles are including the total time when they depart from the vehicle yard until they go back to there. The fourth term (of_{14}) computes the holding costs of commodities at the temporary storage which is located at front of stack doors of cross-dock. The fifth term (of_{15}) of first cost function describes tardiness penalty costs of delivery of the demands to the customers. The sixth term (of_{16}) measures the spoilage costs of commodities which may be spoiled at cross-dock. And the last term (of_{17}) calculates the moment of delivering the demand of the customers who are interested in soonest delivery.

$$\text{Minimize } of_2 = of_{21} + of_{22} + of_{23} + of_{24} + of_{25} + of_{26} + of_{27}$$

$$of_{21} = \sum_{s,c} \sum_{e} \sum_{i,b\epsilon B\backslash BB,n} P^e_{sc} D_{bc}\, x_{isbcn} \alpha^e_{sc}$$

$$of_{22} = \sum_{i,s,b\epsilon B\backslash BB,c,n} x_{isbcn} \left(FCA_s + FCV_n + VCV_n\left(atp_n - dtp_n\right)\right) / \sum_{i,s,b,c,n} x_{isbcn}$$

$$of_{23} = \sum_{j,b\epsilon B\backslash BB,n} y_{jbn}(FCV_n + VCV_n(atp_n - dtp_n)) \Big/ \sum_{j,b,n} y_{jbn}$$

$$of_{24} = \sum_{b\epsilon B\backslash BB,c} H_c D_{bc} (dtc_b - atc_{bc})$$

$$of_{25} = \sum_{b\epsilon B\backslash BB,c} TPC_b D_{bc} tat_b$$

$$of_{26} = \sum_{b\epsilon B\backslash BB,c\epsilon CC} PC_{bc}\, k_{bc}$$

$$of_{27} = \sum_{b\epsilon\{B\backslash BB\},c} EPC_b D_{bc} eat_b\, (1 - k_{bc})$$

The second cost function which is named of_2 is related to the customers who are concerned about lower-bound of delivery time and also are a member of subset $\{B\backslash BB\}$ (as second agent).

The cost components of of_1 are described as follows:

The first term (of_{21}) computes purchasing price. The second term (of_{22}) calculates costs of supplier selection and also computes the fixed and variable costs of using the incoming vehicles. The third term (of_{23}) measures the fixed and variable costs of outgoing vehicles to deliver the demands of customers. The fourth term (of_{24}) computes the holding costs of commodities at the temporary storage which is located at front of stack doors of cross-dock. The fifth term of first cost function (of_{25}) calculates tardiness penalty costs of delivery of the demands to the customers. The sixth term (of_{26}) computes the spoilage costs of commodities which may be spoiled at cross-dock. And the last term (of_{27}) describes earliness penalty costs of demand delivery to the included customers.

2.4.2. Constraints on supply-side of commodities:

$$\sum_{i,s,n} x_{isbcn} = 1 \qquad \forall\, b, c \quad if \quad D_{bc} \neq 0 \tag{1}$$

$$\sum_{i,b,n} D_{bc} x_{isbcn} \leq CAS_{sc} a_s \qquad \forall\, s,c \qquad (2)$$

$$oq_{sc} = \sum_{i,b,n} D_{bc} x_{isbcn} \qquad \forall\, s,c \qquad (3)$$

$$a_s \leq \sum_{i,b,c,n} x_{isbcn} \qquad \forall\, s \qquad (4)$$

Constraint (1) indicates that each of the demanded commodity types must be supplied only by one supplier and picked up only by one vehicle and splitting the demand of a commodity type is not permitted to avoid operational complexity and also quality issues. Equation (2) describes the capacity limitation of each supplier for each commodity type. Constraint (3) determines the order quantity of each commodity type from each supplier and equations (2) and (4) state if a supplier is used.

2.4.3. Constraints on pickup of commodities:

$$\sum_{i,s,b,c} V_c D_{bc} x_{isbcn} \leq CAV_n \sum_{s} g_{sn} \qquad \forall\, n \qquad (5)$$

$$\sum_{i,b,c} x_{isbcn} \leq M \times g_{sn} \qquad \forall\, s,n \qquad (6)$$

$$g_{sn} \leq \sum_{i,b,c} x_{isbcn} \qquad \forall\, s,n \qquad (7)$$

$$\sum_{s} g_{sn} + \sum_{b} z_{bn} \leq 1 \qquad \forall\, n \qquad (8)$$

$$\sum_{n} g_{sn} \leq 1 \qquad \forall\, s \qquad (9)$$

$$ata_{sn} \leq M \times g_{sn} \qquad \forall\, s,n \qquad (10)$$

$$ata_{sn} \geq dtp_n + TRT_s - M(1 - g_{sn}) \qquad \forall\, s,n \qquad (11)$$

Constraint (5) limits the loading of the incoming vehicles to their capacity. Equations (6) and (7) and (9) guarantee that only an individual incoming vehicle can pick up the commodities from a supplier only if the vehicle visits that supplier. Constraint (8) determines that each vehicle can be assigned at most to one supplier or customer at the beginning of its tour. Equation (10) describes the relationship between the scheduling and assignment decisions in the pickup-side and constraint (11) determines the departure time of the incoming vehicles from the parking yard and schedules the arrival of them at the suppliers.

2.4.4. Constraints on strip doors:

$$\sum_{i} pr_{in} = \sum_{s} g_{sn} \qquad \forall\, n \qquad (12)$$

$$\sum_{s,b,c} x_{isbcn} \leq M \times pr_{in} \qquad \forall\, i,n \qquad (13)$$

$$pr_{in} \leq \sum_{s,b,c} x_{isbcn} \qquad \forall\, i,n \qquad (14)$$

$$\sum_{n} pr_{in} \leq M \times er_i \qquad \forall\, i \qquad (15)$$

$$er_i \leq \sum_{n} pr_{in} \qquad \forall\, i \qquad (16)$$

$$qr_{nn'} - 1 \leq pr_{in} - pr_{in'} \leq 1 - qr_{nn'} \qquad \forall\, i,n,n'\ (n \neq n') \qquad (17)$$

$$\sum_{n'=0,\dots,|N|(n' \neq n)} qr_{n'n} = \sum_{i} pr_{in} \qquad \forall\, n \qquad (18)$$

$$\sum_{n'\epsilon\{1,\dots,|N|+1\}(n'\neq n)} qr_{nn'} = \sum_i pr_{in} \qquad \forall\, n \qquad (19)$$

$$\sum_n qr_{0n} = \sum_i er_i \qquad (20)$$

$$\sum_n qr_{n(|N|+1)} = \sum_i er_i \qquad (21)$$

$$qr_{0n} + qr_{0n'} + pr_{in} + pr_{in'} \leq 3 \qquad \forall\, i,n,n'(n' \epsilon \{1,\dots,|N|+1\}, n \neq n') \qquad (22)$$

$$atr_n \leq M \times \sum_i pr_{in} \qquad \forall\, n \qquad (23)$$

$$atr_n \geq ata_{sn} + \sum_{i,b,c} L_{sc} D_{bc} x_{isbcn} + TRT_{0s} - M\left(1 - \sum_i pr_{in}\right) \qquad \forall\, s,n \qquad (24)$$

$$atr_{n'} \geq atr_n + \sum_{i,s,b,c} U_{ic} D_{bc} x_{isbcn} + TNC - M\left(1 - qr_{nn'}\right) \qquad \forall\, n,n'\ (n \neq n') \qquad (25)$$

Constraint (12) with equations (8), (13) and (14) integrate the assignment of the vehicles to the selected suppliers and also to the strip doors. Constraints (15) and (16) state which strip doors of the cross-dock are used. Equation (17) ensures if an incoming vehicle be immediately subsequence of another vehicle they must be processed at the same strip door. Constraints (18) and (19) ensure each real incoming vehicle precedes exactly one real or dummy incoming vehicle and follows exactly one another real or dummy incoming vehicle. Equations (20), (21) and (22) determine the dummy vehicles '0' and 'N+1' to be respectively the 1^{st} and the last incoming vehicles at each used strip door. Constraints (23) and (24) determine the departure time of the incoming vehicle from the visited supplier and schedule the loading and arrival of them at the strip door. Constraint (25) describes that the arrival time of an incoming vehicle which follows another incoming vehicle at a same strip door must be at least equal to or larger than arrival time of the preceding vehicle in addition its unloading time plus the changeover time of the vehicles.

2.4.5. Constraints on stack doors:

$$z_{bn} \leq \delta_b \qquad \forall\, b,n \qquad (26)$$

$$\sum_j po_{jn} = \sum_b z_{bn} \qquad \forall\, n \qquad (27)$$

$$\sum_b y_{jbn} \leq M \times po_{jn} \qquad \forall\, j,n \qquad (28)$$

$$po_{jn} \leq \sum_b y_{jbn} \qquad \forall\, j\,,n \qquad (29)$$

$$\sum_n po_{jn} \leq M \times eo_j \qquad \forall\, j \qquad (30)$$

$$eo_j \leq \sum_n po_{jn} \qquad \forall\, j \qquad (31)$$

$$qo_{nn'} - 1 \leq po_{jn} - po_{jn'} \leq 1 - qo_{nn'} \qquad \forall\, j,n,n'\ (n \neq n') \qquad (32)$$

$$\sum_{n'\epsilon\{0,\dots,|N|\}(n'\neq n)} qo_{n'n} = \sum_j po_{jn} \qquad \forall\, n \qquad (33)$$

$$\sum_{n'\epsilon\{1,\dots,|N|+1\}(n'\neq n)} qo_{nn'} = \sum_j po_{jn} \qquad \forall\, n \qquad (34)$$

$$\sum_n qo_{0n} = \sum_j eo_j \qquad (35)$$

$$\sum_n qo_{n(|N|+1)} = \sum_j eo_j \qquad (36)$$

$$qo_{0n} + qo_{0n'} + po_{jn} + po_{jn'} \leq 3 \qquad \forall\, j,n,n'(n' \epsilon \{1,\dots,|N|+1\}, n \neq n') \qquad (37)$$

$$ato_n + dto_n \leq M \times \sum_j po_{jn} \qquad \forall\, n \qquad (38)$$

$$ato_n \geq dtp_n + TRT_0 - M\left(1 - \sum_j po_{jn}\right) \qquad \forall\, n \qquad (39)$$

$$ato_{n'} \geq dto_n + TNC - M\left(1 - qo_{nn'}\right) \qquad \forall\, n, n'\ (n \neq n') \qquad (40)$$

$$dto_n \geq ato_n + \sum_b \left(y_{jbn} \sum_c L_{jc} D_{bc}\right) \qquad \forall\, j, n \qquad (41)$$

$$\sum_{i,s,c} x_{isbcn} \leq M \times w_{bn} \qquad \forall\, b, n \qquad (42)$$

$$w_{bn} \leq \sum_{i,s,c} x_{isbcn} \qquad \forall\, b, n \qquad (43)$$

$$ato_n \geq atr_{n'} + \sum_{i,j,s,b,c,n} U_{ic} D_{bc} x_{isbcn'} y_{jbn} + TNT - M\left(2 - w_{bn'} - \sum_j y_{jbn}\right) \qquad \forall\, b, n, n'\ (n \neq n') \qquad (44)$$

Constraint (26) implies if all commodities ordered by a customer spoiled no outgoing vehicle would assign to that. Equation (27) with constraints (8), (28) and (29) integrate the assignment of the vehicles to the customers and also to the stack doors. Equations (30) and (31) state which stack doors of the cross-dock are used. Constraint (32) ensures if an outgoing vehicle be immediately subsequence of another vehicle they must be processed at the same stack door. Equations (33) and (34) ensure each real outgoing vehicle precedes exactly one real or dummy outgoing vehicle and follows exactly one another real or dummy outgoing vehicle. Constraints (35), (36) and (37) determine the dummy vehicles '0' and 'N+1' to be respectively the 1st and the last outgoing vehicles at each used stack door. Equations (38), (39), (40) and (41) determine the departure time of the outgoing vehicles from the cross-dock and schedule the loading and arrival of them at the stack door and also ensure that the arrival time of an outgoing vehicle which follows another outgoing vehicle at a same stack door must be at least equal to or larger than arrival time of the preceding vehicle in addition its loading time plus the changeover time of the vehicles. Equations (42) and (43) ensure that the all commodities ordered by any of the customers are picked up from the suppliers by one of the incoming vehicles. Constraint (44) connects the arrival time of an outgoing vehicle at the cross-dock to the arrival time of an incoming vehicle if any of the commodities is transferred between them.

2.4.6. Constraints on cross-dock:

$$atc_{bc} \leq atr_n + \sum_{i,s} U_{ic} D_{bc} x_{isbcn} + M\left(1 - \sum_{i,s} x_{isbcn}\right) \qquad \forall\, b, c, n \qquad (45)$$

$$atc_{bc} \leq M \times \sum_{i,s,n} x_{isbcn} \qquad \forall\, b, c \qquad (46)$$

$$dtc_b \leq M \times \sum_{j,n} y_{jbn} \qquad \forall\, b \qquad (47)$$

$$dtc_b \geq dto_n - M\left(1 - \sum_j y_{jbn}\right) \qquad \forall\, b, n \qquad (48)$$

$$k_{bc} = 0 \qquad \forall\, b, c \in \{C \backslash CC\} \qquad (49)$$

$$(dtc_b - atc_{bc}) - DD_c \leq M \times k_{bc} \qquad \forall\, b, c \in CC \qquad (50)$$

$$(dtc_b - atc_{bc}) - DD_c \geq M(k_{bc} - 1) \qquad \forall\, b, c \in CC \qquad (51)$$

Equations (45), (46), (47) and (48) describe the arrival and departure times for each commodity type which ordered by each customer to measure the holding time of them in the temporary storage of the cross-dock. Constraints (49), (50) and (51) determine whether each of the commodities is spoiled at cross-dock.

2.4.7. Constraints on delivery-side:

$$\sum_{c}(1-k_{bc}) \geq \delta_b \qquad \forall\, b \qquad (52)$$

$$\sum_{c}(1-k_{bc}) \leq M \times \delta_b \qquad \forall\, b \qquad (53)$$

$$\sum_{j,n} y_{jbn} = \delta_b \qquad \forall\, b \qquad (54)$$

$$\sum_{i,s,c,n} x_{isbcn} \leq M\left(\sum_{j,n} y_{jbn} + 1 - \delta_b\right) \qquad \forall\, b \qquad (55)$$

$$\sum_{j,n} y_{jbn} \leq \sum_{i,s,c,n} x_{isbcn} \qquad \forall\, b \qquad (56)$$

$$\sum_{b}(\sum_{c} V_c D_{bc}(1-k_{bc}) \times \sum_{j} y_{jbn}) \leq CAV_n \sum_{b} z_{bn} \qquad \forall\, n \qquad (57)$$

$$\sum_{j} y_{jbn} \leq M\left(z_{bn} + \sum_{b'(b'\neq b)} zz_{b'bn}\right) \qquad \forall\, b, n \qquad (58)$$

$$z_{bn} + \sum_{b'(b'\neq b)} zz_{b'bn} \leq \sum_{j} y_{jbn} \qquad \forall\, b, n \qquad (59)$$

$$\sum_{b,b'(b'\neq b)} zz_{bb'n} \leq M \times \sum_{b} z_{bn} \qquad \forall\, n \qquad (60)$$

$$z_{bn} + \sum_{b'(b'\neq b)} zz_{b'bn} = \sum_{b'(b'\neq b)} zz_{bb'n} + zl_{bn} \qquad \forall\, b, n \qquad (61)$$

$$atd_{bn} \leq M\left(z_{bn} + \sum_{b'(b'\neq b)} zz_{b'bn}\right) \qquad \forall\, b, n \qquad (62)$$

$$atd_{bn} \geq dto_n + TRT_{0b} - M\left(2 - z_{bn} - \sum_{j} po_{jn}\right) \qquad \forall\, b, n \qquad (63)$$

$$atd_{b'n} \geq atd_{bn} + \sum_{j}(y_{jbn}\sum_{c} U_{bc}D_{bc}) + TRT_{bb'} - M(1 - zz_{bb'n}) \qquad \forall\, b, b'(b' \neq b), n \qquad (64)$$

$$tat_b \geq atd_{bn} - UD_b \qquad \forall\, b, n \qquad (65)$$

$$eat_b \geq LD_b - atd_{bn} - M\left(1 - \sum_{j} y_{jbn}\right) \qquad \forall\, b, n \qquad (66)$$

$$atp_n + dtp_n \leq M\left(\sum_{s} g_{sn} + \sum_{b} z_{bn}\right) \qquad \forall\, n \qquad (67)$$

$$atp_n \geq atr_n + \sum_{i,s,b,c} U_{ic}D_{bc}x_{isbcn} + \sum_{i} TRT_0 pr_{in} \qquad \forall\, n \qquad (68)$$

$$atp_n \geq atd_{bn} + \sum_{c} U_{bc}D_{bc} + TRT_b - M(1 - zl_{bn}) \qquad \forall\, b, n \qquad (69)$$

$$atp_n \leq T \qquad \forall\, n \qquad (70)$$

Equations (52) and (53) ensures if all commodities ordered by customer b spoiled then δ_b would be 0. Constraint (54) determines if at least one of the commodities ordered by a customer not to be spoiled at the cross-dock then only one outgoing vehicle must be assigned to deliver its demand from one stack door. Equations (55) and (56) integrate the assignment decisions for each customer in the pickup and distribution sides. Constraint (57) limits the loading of the outgoing vehicles to their capacity. Equations (58) and (59) state that an outgoing vehicle can deliver commodities to a customer only if the vehicle visits that customer. Constraints (60) and (61) determine the routing of the outgoing vehicles. Constraints (62), (63) and (64) describe the arrival time of the outgoing vehicles at customers and the unloading of them at delivering process. constraint (64) is sub-tour elimination. Equations (65) and (66) measure the earliness and tardiness of the demand delivery for each customer. Constraints (67) limits the arrival and departure times of the vehicles at or from parking yard. Equations (69) and (70) compute the arrival times of the vehicles at the parking yard as an incoming or outgoing vehicle. Constraint (70) ensures that the all of the operations must be done within the planning horizon time.

2.4.8. Group purchasing constraints:

$$P_{sc}^{e} = \begin{Bmatrix} P_{sc}^{1} \rightarrow Q_{c}^{1} \leq oq_{sc} < Q_{c}^{2} \\ P_{sc}^{2} \rightarrow Q_{c}^{2} \leq oq_{sc} < Q_{c}^{3} \\ P_{sc}^{E} \rightarrow Q_{c}^{E} \leq oq_{sc} < Q_{c}^{E+1} \end{Bmatrix} \qquad \forall\, s, c, e \qquad (71)$$

Constraint (71) determines based on the group purchasing strategies that the order quantity assigned to a selected supplier stands at which purchasing level.

It is worth mentioning that the integrating of group purchasing strategies (equation 71) and managing perishability of the commodities (constraints 49-51) into the model is one of the novelties of this paper and these features differentiate the new proposed model from VRPCD model which presented by Nasiri et al. (2018).

As observed above, the new proposed model is non-linear since some variables are multiplied in objective functions and also in equations (44) and (57). Additionally, constraint (71) is a non-linear equation. Thus, at appendix we defined auxiliary variables to linearize the objective functions and mentioned equations.

2.5. Validation and verification of the new proposed model

A small-sized instance which could be solved optimally during a logical time is considered to validate the new proposed model. The detailed figures are showed at table 1 by table 5.

Table 1. supplier information of the test instance

				P_{sc}^{e}				L_{sc}		CAS_{sc}	
				e, c				c		c	
s	FCA_s	TRT_s	TRT_{0s}	1 , 1	1 , 2	2 , 1	2 , 2	1	2	1	2
1	28	2040	1560	33	60	30	59	2	1	90	108
2	5	2520	2640	25	53	22	51	2	2	128	86
3	26	1260	1980	30	57	27	56	1	1	116	96

Table 2. customer information of the test instance

							$TRT_{bb'}$			PC_{bc}		U_{bc}	
							b'			c		c	
b	EPC_b	TPC_b	LD_b	UD_b	TRT_b	TRT_{0b}	1	2	3	1	1	1	2
1	2	1	4500	7500	2400	1320	-	840	480	-	65	1	1
2	0	10	4500	7500	2520	1440	840	-	480	-	55	1	2
3	3	1	4500	7500	2880	1800	480	480	-	-	25	1	1

Table 3. commodity information of the test instance

				U_{ic}	L_{jc}	Q_c^e			D_{bc}		
				i	j	e			b		
c	V_c	H_c	DD_c	1	1	1	2	3	1	2	3
1	1	2	-	2	1	0	45	100000	36	43	43
2	0.75	6	2700	1	2	0	45	100000	22	33	23

Table 4. vehicles information of the test instance

n	1	2	3	4	5
CAV_n	227	180	177	168	173
FCV_n	93	31	28	21	26
VCV_n	2	7	7	5	6

Table 5. cross-dock and planning horizon information of the test instance

TNT	240
TNC	180
TRT_0	1080
T	43200

Agnetis et al. (2014) presented a few solution approaches for multi-agent problems such as linear combination of criteria which called as weighted-sum method. So, the presented MILP model and its linear combination of cost functions is coded in GAMS 25.1 and solved applying CPLEX on PC ASUS Core i5, 3.1 GHs with 8GB RAM. the model is run 10386.50 seconds by GAMS 25.1 until to obtain the optimal solution for weighted-sum objective which includes 145606 as optimal costs of first agent (of_1) and 347746 as optimal costs of second agent (of_2). Fig. 1 is an illustration of an optimal pareto solution for the mentioned test instance.

Also, the optimal solution includes 8164 as supply costs, 106231 as transportation costs by vehicles, no spoilage costs, 373436 as holding costs of commodities at cross-dock and 5521 as earliness and tardiness penalty costs. Two vehicles (n_4 and n_5) are used to pick up the commodities from selected suppliers (s_1 and s_3) and transfer them toward the cross-dock. Also, two vehicles (n_1 and n_3) are used to deliver the demands of customers. We suppose the planning horizon from 7:00 to 19:00 o'clock.

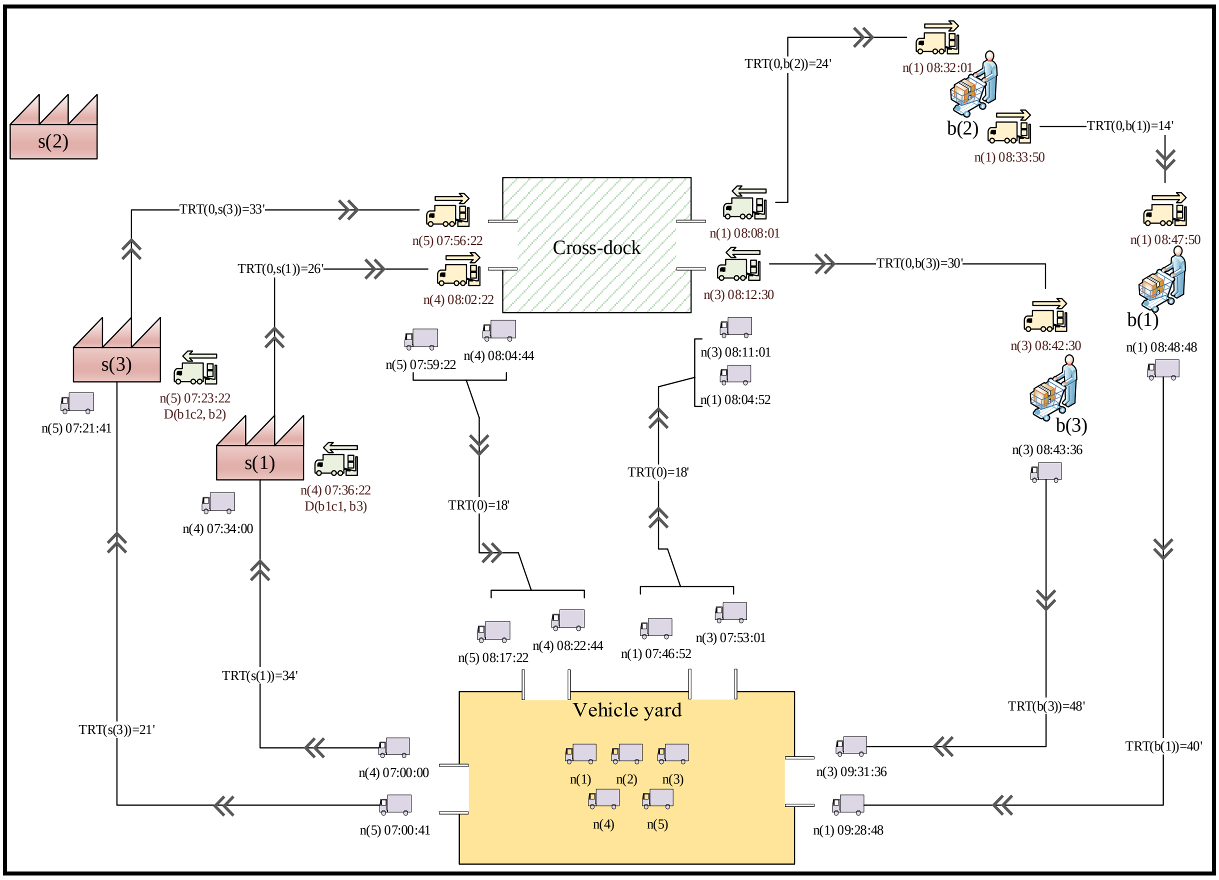

Figure 1. illustration of an optimal solution for the small-sized test instance

On the other hand, the optimal values of two objective functions which are obtained by GAMS 25.1, are normalized such that Fig. 2 shows the behavior of objective functions associated to the agents are not consistent and must be considered separately. On the other hand, Fig. 3 depicts that the CPU-time of instances which are solvable by weighted-sum method increases exponentially when the size of instances enlarges.

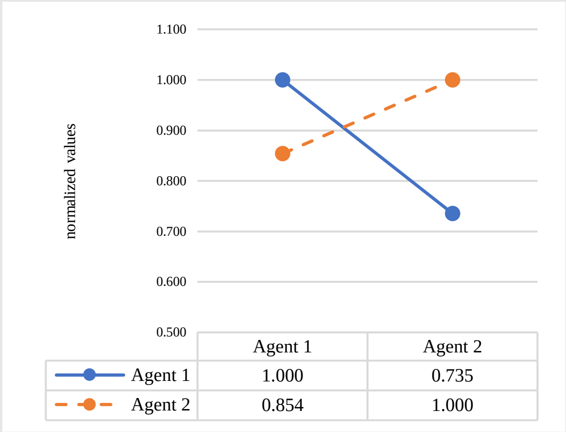

Figure 2. variation of the normalized objective functions related to agents

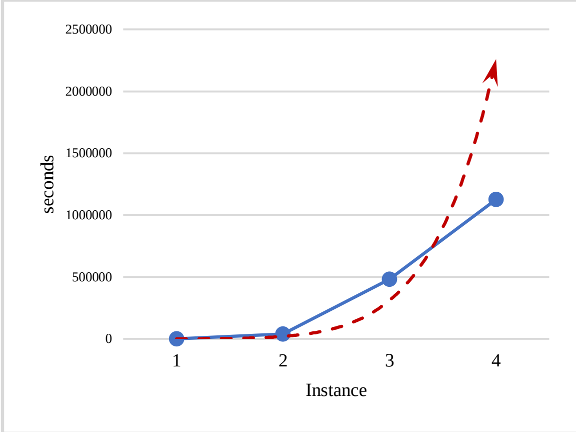

Figure 3. CPU-Time of weighted-sum method

2.6. Sensitivity analysis

Sensitivity analysis is performed for evaluating the effects of variation of the parameters. At first step, we perform the sensitivity analysis on fixed and variable parameters which are fixed and variable cost coefficients of the cost functions and then we examine the effects of variation of above-mentioned parameters on weighted combination of dual objective functions which is named of_T. At this stage, we consider five levels of variability which are {0.1, 0.3, 0.5, 0.7, 1.0}.

The results of analysis on fixed cost coefficients displayed at Fig. 4, show that the behavior of the new proposed model on the variation of the ordering costs of the suppliers is not regular and the variation of of_T is less than five percent and also the sensitivity of the new proposed model on variation of the fixed costs of using the vehicles is less than 1.5 percent and is not meaningful.

On the other hand, the results of analysis on variable cost coefficients displayed at Fig. 5, show when P_{sc}^{e} increases, the total objective function of_T increases at most 3.5 percent. It shows that when the price of commodities rises up the model selects the suppliers which are far from the cross-dock but cheaper than the others. Also, the results show when the holding costs increase, the of_T increases also. The results show when the spoilage costs of perishable commodities increase 10 percent, the of_T decrease about 42% and more increasing of PC_{bc} has no significant effect on of_T. Increasing of the variable costs of vehicles effects the of_T regularly at most about 20 percent. The effects of increasing of earliness and tardiness penalty costs are increasingly on the of_T.

At second stage of sensitivity analysis, we apportion the cost coefficients of the objective functions into two parts. The first part of coefficients is named as supply costs and the second part of coefficients is named as distribution costs. We analyze if the most effective coefficients of part two changed how the supply decisions would make. We consider six levels of variability which are {-1.0, 0.0, 1.0, 2.0, 3.0, 4.0}. The results which are depicted at Fig. 6, show that for the lower levels of variability of distribution costs, the fixed and variable costs of using vehicles effect the supply decisions more than the others but for the higher levels of variability the holding and spoilage costs of commodities effect the supply decisions more. So, obtained results show the significance of managing perishability on cost reduction of supply chain. In addition, more investigation about the results displayed at Fig. 7, shows for the lower levels of variability of distribution costs the cheaper suppliers with regards to the group purchasing strategies are selected and for the higher levels would be vice versa. These results prove the specific benefits of group purchasing such as cost reduction at decisions of supply chain management.

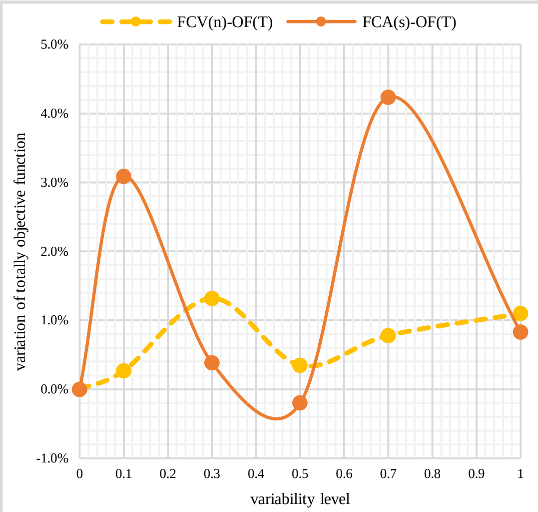

Figure 4. variability effects of fixed costs of using vehicles and suppliers

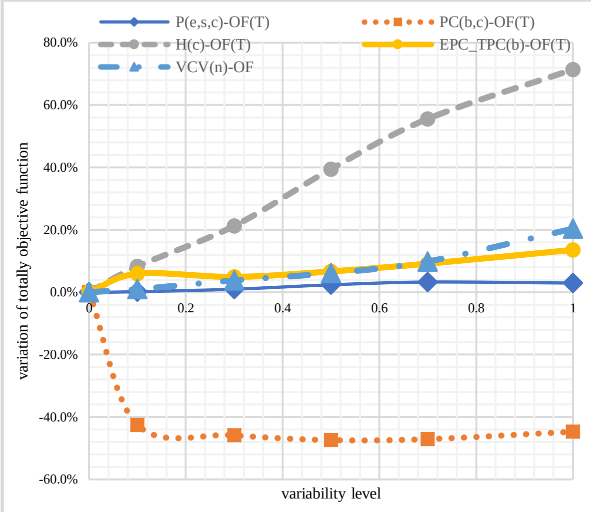

Figure 5. variability effects of variable cost coefficients

Totally, these results prove the profits of considering mentioned strategies into the multi-agent problem of supplier selection and vehicle routing with cross-docking through integrated decision making about the supply-side and distribution-side of supply chain. Finally, considering these strategies leads to cost reduction and enhancement of responsiveness for end consumers and businesses.

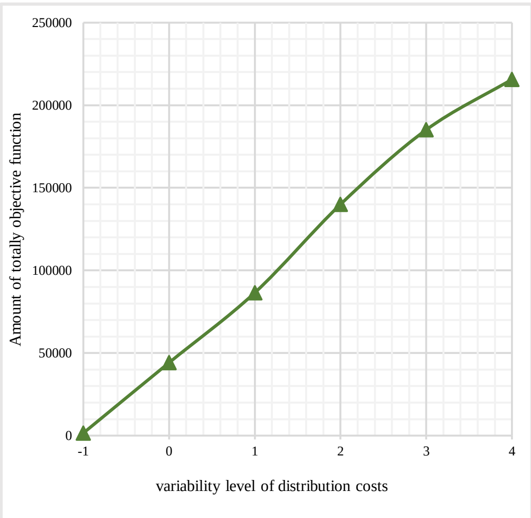

Figure 6. variability effects of distribution costs on totally costs

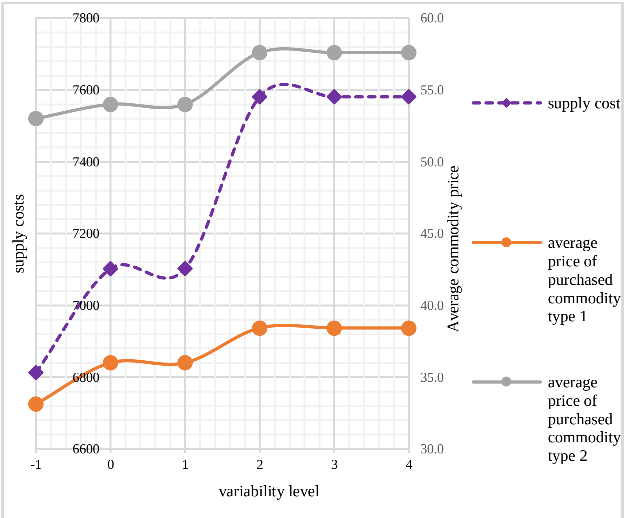

Figure 7. variability effects of distribution costs on supply costs and commodity prices

2.7. Challenges of solving the new proposed model

There are some challenges in solving the new proposed model including NP-hardness of the problem due to two-agent scheduling and their associated different objectives. Additionally, integrating constraints related to managing perishability and group purchasing strategies into the model makes its complexity more. As mentioned before, solving these problems at real scales is a very long time-consuming operation as showed at Fig. 3. So, to address these challenges two meta-heuristic algorithms are proposed to solve the problem at real sizes.

3. Solution approach and methodology

Generally, methodologies for solving multi agent problems are same approaches applying to solve multi-objective problems. In traditional approaches, there are some methodologies to obtain solution for multi-objective instances such as weighted-sum method which compose all the objectives into a single function and the other approach is to determine one of the objectives as a core objective and consider the others as constraints. But both of the above-mentioned approaches may have some difficulties.

So, for the sake of NP-hardness of the real-sized instances, two meta-heuristic approaches including Non-dominated Sorting Genetic Algorithm which is named NSGA-II and Multi Objective Particle Swarm Optimization which is called MOPSO are applied to solve the new proposed model.

These approaches are used to determine the best or optimum solutions for multi agent (as objective) instances where there is not any single solution for all agents. In multi-agent instances, that's possible to generate a set of non-dominated solutions where each solution is not better than the others (Azadeh et al. (2017)). In other words, when a feasible solution $\hat{s}$ is called non-dominated, that means there is not another feasible solution better than $\hat{s}$ in one or more objective functions without worsening the others (Ehrgott (2005)). Ranking of the solutions in meta-heuristic multi-agent (objective) algorithms is based on the non-dominated Pareto fronts. Where the ranking of a set of non-dominated solutions is equal to 1, it's called as F1. While the solutions of F1 are eliminated, the ranking of the remaining non-dominated solutions is equal 2 and are called F2. While the solutions of F2 are eliminated as well as F1, ranks of the remaining non-dominated solutions are equal to 3 and are called F3. Iteration of this process should be done until the ranks of all sets of solutions are determined.

3.1. NSGA-II

Non-dominated Sorting Genetic Algorithm as an evolutionary algorithm firstly presented by Deb (2001) for solving multi-objective optimization problems and then the well-known Non-dominated Sorting Genetic Algorithm II presented by Deb et al. (2000) and for the sake of its simplicity, good convergence and diversity, it has been under attention of much researchers to solve a wide range of multi-objective optimization problems. NSGA-II is a population-based algorithm that implements functions like chromosomes. This approach needs a set of solutions as the initial population to start and based on the inheritance of the algorithm, the past data is extracted to use in the searching procedure. In this paper, a chromosome is a string of mixed integer and binary numbers that are called genes. Firstly, initial population of chromosomes are randomly generated and then in each iteration, a new offsprings (chromosome) are generated by the operators of NSGA-II algorithm which are called crossover and mutation. At crossover process, the parents randomly are selected with a predefined rate (input parameter of NSGA-II) and then the proposed operator (uniform permutation) is applied. After doing the crossover operation, the mutation process is employed with a predetermined parameter. This operation is applied in order to increase the diversity of the solutions and to avoid being trapped into a local optimum. In this paper, we randomly use three different mutation operators (i.e. swap, inversion and insertion).

Then, evaluation of new generated chromosomes is done and by qualification and selection procedures more chromosomes are selected until their population size would be same as the initial population. Then, the next generation of the offsprings by transferring selected chromosomes would be generated and in this process the algorithm converges to the chromosomes that are representations of possible optimum or sub-optimal solutions. Also, in this article, we use crowding-distance (*CD*) as another criterion to sort the non-dominated solutions with the same rank. The Eq. (72) calculates *CD*, where s_{i-1}, s_i and s_{i+1} are consecutive members of non-dominated solutions set. Also, $of_1(s_i)$ and $of_2(s_i)$ are respectively the values of cost functions associated to the first and the second agents of the point *i* of non-dominated solutions set. of_1^{max}, of_2^{max} and of_1^{min}, of_2^{min} are respectively the maximum and minimum values of the first and second objective functions on Pareto front.

$$CD_i = \frac{of_1(s_{i+1}) - of_1(s_{i-1})}{of_1^{max} - of_1^{min}} + \frac{of_2(s_{i+1}) - of_2(s_{i-1})}{of_2^{max} - of_2^{min}} \quad (72)$$

Start
Problem Definition: Instance Parameters and Modeling, Cost Function of each Agent
1. Input:
    a. Population Size (nPop),
    b. it (iteration) = 0,
    c. Maximum Number of Iterations (MaxIt),
    d. Crossover Probability,
    e. Mutation Probability,
    f. Selection Pressure.
    %% Initialization
2. Initialize the Population randomly as Starting Solutions
3. Calculate the Value of each Agent Objective
4. Implement Non-Dominated Sorting and Calculate Crowding Distance
5. Store Non-Dominated Solutions as F1
    %% Main Loop
6. For it < = MaxIt
7. Perform the Crossover and Create New-Generation
8. Perform the Mutation on new-Generation
9. Merge the Parents and new-Generations
10. Implement Non-Dominated Sorting and Calculate Crowding Distance
11. Update the Solutions of F1
12. it = it + 1
13. End
14. Report Cost Functions and Plot F1
End

Figure 8. pseudo-code of NSGA-II

To the best of the authors' knowledge, it is notable that if neighborhood search is only implemented on the members of non-dominated solutions set, *CPU*-time will be decreased without any change in the quality of solutions. Therefore, we done the operations of crossover and mutation only on non-dominated solutions so that a high-quality solution with the greater *CD* has a higher chance to be selected for the next generation or even for the next non-dominated solutions set.

3.2. MOPSO

Multi objective particle swarm optimization is one of the most well-known evolutionary algorithms which is developed based on birds' social behavior (Azadeh et al. (2017)). Usually, in this algorithm the initial population are randomly generated and a velocity is calculated as following Eq. (73) for each of initial solutions. According to a solution's velocity and its distance from the best position in the personal memory and also its distance from the best global founded solutions, a new velocity and position are determined for the solution respectively by the equations (73) and (74). If the calculated position dominates the best position in the personal memory, it will be replaced by the new position, otherwise, the new position will be selected between them randomly (equations (75) and (76)).

Notations of the algorithm are: (1) s_i *shows the current position of the particle*; (2) vel_i *indicates the velocity of the particle*; (3) p_i^{best} *denotes the best personal memory*; (4) g_i^{best} *represents the best position founded by the leader particles*; (5) n_1^r *and* n_2^r *are random numbers which are selected from set* $\{0,1\}$; (6) w *is the weight of inertia*; (7) c_1 *and* c_2 *are acceleration values*.

```
Start
Problem Definition: Instance Modeling and Parameters, Cost Function of each Agent
Input:
        a.   Population Size (nPop),
        b.   it (iteration) = 0,
        c.   Maximum Number of Iterations (MaxIt),
        d.   Repository Size (nRep),
        e.   Personal Learning Coefficient (c1),
        f.   Global Learning Coefficient (c2),
        g.   Mutation Rate (mu),
        h.   Leader Selection Pressure (beta),
        i.   Velocity Limits (VelMax & VelMin).
        %% Initialization
Initialize the Swarm randomly (position & velocity) as Starting Solutions
Calculate the Value of each Agent Objective
Store Personal Best Solutions
Implement Domination Process
Store Non-Dominated Particles as rep
        %% Main Loop
For it < = MaxIt
    Select Leader
    Update position and velocity
    Implement Evaluation of Cost Functions
    Apply the Mutation
    Add Non-Dominated Particles to rep
    Implement Domination Process and Update rep
      if number of rep Members > nRep
        Apply Deletion One Member of rep
      End if
    it = it + 1
End For
Report Cost Functions and Plot rep
End
```

Figure 9. pseudo-code of MOPSO

In the proposed MOPSO algorithm the set of non-dominated solutions considers as the leader particles. As Eq. (73) shows, while the set of non-dominated solutions has several members, for each particle only one leader should be selected. Similar to NSGA-II algorithm we use the *CD* index to select the leader of each particle. On the other hand, if the number of non-dominated solutions set exceeds the size of archive, the non-dominated solutions with the lowest values of *CD* will be eliminated by the algorithm.

$$vel_i(t) = w * vel_i(t) + c_1 n_1^r\{p_i^{best}(t-1) - s_i(t-i)\} + c_2 n_2^r\{g_i^{best}(t-1) - s_i(t-1)\} \quad (73)$$

$$s_i(t) = s_i(t-1) + vel_i(t) \quad (74)$$

$$p_i^{best}(t) = p_i^{best}(t-1) \; if \; of(s_i(t)) \geq of(p_i^{best}(t-1)) \quad (75)$$

$$p_i^{best}(t) = s_i(t) \; if \; of(s_i(t)) \leq of(p_i^{best}(t-1)) \quad (76)$$

4. Computational results

To evaluate the proposed algorithms, ten different numerical test instances based on Nasiri et al. (2018) at every eighteen different sizes are generated and each test instances is solved several times by each algorithm to increase the evaluation accuracy.

The table 6 shows the detailed characteristics of test instances. At every row (group), ten different test instances are created with dissimilar number of strip and stack doors.

Also, to indicate the applicability of the meta-heuristic approaches, a Weighted-Sum Method (WSM) which is coded in GAMS 25.1 is applied to solve some small-sized test instances. Then, they are solved by applying the proposed meta-heuristic algorithms and the performance of solving approaches are compared with each other according to six criteria as following to be explained.

1. The number of Pareto optimal solutions (NPS) which the larger value of this criterion shows the more variability of the Pareto optimal solutions.

Table 6. General Specifications of Test Instances

Instance Group Number	Number of: Suppliers	Buyers	Commodities	Vehicles	Strip & Stack Doors	Number of Solving Replications
1	2	2	2	3	ϵ {1, 2}	12
2	2	2	3	4	ϵ {1, 2}	12
3	3	3	3	5	ϵ {1, 2}	12
4	3	4	4	5	ϵ {1, 2}	12
5	4	4	5	6	ϵ {1, 2}	12
6	4	5	6	6	ϵ {1, 2}	12
7	5	6	7	7	ϵ {2, 3}	8
8	5	6	8	8	ϵ {2, 3}	8
9	6	7	9	10	ϵ {2, 3}	8
10	6	8	10	11	ϵ {2, 3}	8
11	7	9	11	13	ϵ {2, 3}	8
12	7	10	12	15	ϵ {2, 3}	8
13	8	11	13	16	ϵ {3, 4}	5
14	8	11	14	17	ϵ {3, 4}	5
15	9	12	15	19	ϵ {3, 4}	5
16	9	13	16	20	ϵ {3, 4}	5
17	10	14	17	22	ϵ {3, 4}	5
18	10	15	18	24	ϵ {3, 4}	5

2. The CPU-time that indicates the time left to solve an instance and the lower value of it is preferable.

3. The spacing criterion (S) which is used to measure uniformly distribution of the Pareto solutions and is defined as Eq. (77). The Pareto solutions with lower values of this criterion are distributed uniformly in comparison to the larger values.

$$S = \sqrt{\frac{\sum_{i=1}^{NPS}(\overline{dsc} - dsc_i)^2}{NPS - 1}} \tag{77}$$

dsc_i indicates the distance by the *i-th* closest optimal solution which is determined by Eq. (78).

$$dsc_i = \min_{j \text{ } and \text{ } j \neq i}\{|of_1(s_i) - of_1(s_j)| + |of_2(s_i) - of_2(s_j)|\} \tag{78}$$

4. The diversity (*D*) of the Pareto solution that the larger values of this criterion are preferable and is calculated by Eq. (79).

$$D = \sum_{l=1}^{2} \max_{i,j}\{|of_l(s_i) - of_l(s_j)|\} \tag{79}$$

5. The quality of the Pareto solutions which is the most important metric to compare the performance of algorithms that are used to solve multi-agent problems, cannot be calculated individually. Consider s_i and s_j are two sets of solutions that are specified to compare with each other. First of all, $C(s_i, s_j)$ and also $C(s_j, s_i)$ should be indicated by Eq. (80). Then, the $Q(s_i, s_j)$

is defined as quality index of s_i versus s_j and is calculated by Eq. (81). $Q(s_i, s_j)$ would be a ratio between 0 and 1 in which $Q(s_i, s_j) + Q(s_j, s_i) = 1$ and if $Q(s_i, s_j) > Q(s_j, s_i)$, it means that the quality of solution s_i is higher than s_j.

$$C(s_i, s_j) = \frac{the\ quantity\ of\ members\ of\ s_j\ dominated\ by\ members\ of\ s_i}{the\ quantity\ of\ members\ of\ s_j} \tag{80}$$

$$Q(s_i, s_j) = {C(s_i, s_j)}\Big/{C(s_i, s_j) + C(s_j, s_i)} \tag{81}$$

6. Mean ideal distance (MID) of a Pareto solution which is a metric to indicate the distance between the solutions of a generation and the best member of it (Saraafha et al. (2015)). The lower values of MID are preferable and is calculated by Eq. (82).

$$MID = \frac{1}{NPS} \sqrt{\sum_{i=1}^{NPS} (of_{1i}^2 + of_{2i}^2)} \tag{82}$$

All of the test instances are solved by applying the meta-heuristic approaches which are coded in MATLAB R2019b on PC ASUS Core i5, 3.1 GHz with 8GB RAM and the obtained results compared according to the explained six indices.

It is obvious that the proper combination of controllable parameters of meta-heuristic approaches has a significant impact on the performance of the algorithms. So, in this paper by applying Taguchi approach which is a robust method for design of experiments, optimum combination of parameters of meta-heuristic algorithms are determined.

With regards to the number of parameters (at maximum seven factors) which must be determined optimally and considering three levels for each parameter, the appropriate number of experiments based on orthogonal arrays would be 27 (L27). Combination of the parameters is according table 7. Six different test instances randomly are chosen. To choose test instances, six integer numbers are created randomly at intervals (1-3), (4-6), (7-9), (10-12), (13-15), and (16-18), so that at every interval should be selected one number. Then according to the selected number, test instances are chosen from table 6 (at every group, one instance). Each test instance is being solved five times to yield more reliable results. The mean of the results obtained from five times solving of each test instance is considered as the result of solving that instance. Five indices including NPS, CPU-time, spacing (S), diversity (D) and MID are applied to measure the performance of obtained results at each Taguchi experiment. At each experiment, the obtained results of above-mentioned performance indices should be transformed to relative deviation index (RDI) which is calculated by Eq. (83).

$$RDI = \left| \frac{Algorithm_{performance} - Algorithm_{performance}^{best}}{Algorithm_{performance}^{maximum} - Algorithm_{performance}^{minimum}} \right| \tag{83}$$

$\mathrm{Algorithm}_{\mathrm{performance}}$ is the obtained result at each experiment by each performance criterion and $Algorithm_{performance}^{best}$ is the best value between the obtained results. $Algorithm_{performance}^{maximum}$ and $Algorithm_{performance}^{minimum}$ are the maximum and minimum values of each performance criterion, respectively. Then, the mean of the *RDIs* for the six test instances at each Taguchi experiment is calculated as $\overline{RDI}$. The response variable for each Taguchi experiment *i* which is named R_i^{main} is the mean of $\overline{RDIs}$.

Table 7. combination of parameters of MOPSO and NSGA-II

Run	Parameters of MOPSO						
	Number of Generation	Population-size	Repository-size	w	c1	c2	Mutation Rate
	Parameters of NSGA-II						
	Number of Generation	Population-size	Crossover Rate	Mutation Rate	Selection Pressure		
1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2
3	1	1	1	1	3	3	3
4	1	2	2	2	1	1	1
5	1	2	2	2	2	2	2
6	1	2	2	2	3	3	3
7	1	3	3	3	1	1	1
8	1	3	3	3	2	2	2
9	1	3	3	3	3	3	3
10	2	1	2	3	1	2	3
11	2	1	2	3	2	3	1
12	2	1	2	3	3	1	2
13	2	2	3	1	1	2	3
14	2	2	3	1	2	3	1
15	2	2	3	1	3	1	2
16	2	3	1	2	1	2	3
17	2	3	1	2	2	3	1
18	2	3	1	2	3	1	2
19	3	1	3	2	1	3	2
20	3	1	3	2	2	1	3
21	3	1	3	2	3	2	1
22	3	2	1	3	1	3	2
23	3	2	1	3	2	1	3
24	3	2	1	3	3	2	1
25	3	3	2	1	1	3	2
26	3	3	2	1	2	1	3
27	3	3	2	1	3	2	1

Parameters of the algorithms and their levels are presented at tables 8 and 9.

Table 8. parameters and their levels in the NSGA-II

Parameters		Levels		
		1	**2**	**3**
1	Number of Generation	100	150	200
2	Population-size	100	200	300
3	Crossover Rate	0.7	0.8	0.9
4	Mutation Rate	0.1	0.15	0.2
5	Selection Pressure	8	10	12

Table 9. parameters and their levels in the MOPSO

Parameters		Levels		
		1	**2**	**3**
1	Number of Generation	50	100	200
2	Population-size	100	200	300
3	Repository-size	15	30	60
4	w	0.4	0.7	0.9
5	c1	0.5	1.5	2.5
6	c2	0.5	1.5	2.5
7	Mutation Rate	0.1	0.2	0.3

Finally, by applying Taguchi method in MINITAB the best combination of parameters of proposed meta-heuristic algorithms are determined as tables 10 and 11.

Table 10. Input parameters of NSGA-II

	Population-size	Number of Generation	Crossover Rate	Mutation Rate	Selection Pressure
NSGA-II	300	150	0.9	0.15	12

Table 11. Input parameters of MOPSO

	Population-size	Number of Generation	C1	C2	w	Mutation Rate	Repository-size
MOPSO	100	200	0.5	1.5	0.7	0.2	60

To depict the scalability of the new proposed model and solution approaches, we calculate differences between solutions obtained from applying meta-heuristic algorithms and weighted-sum method. The error percentages of the best obtained results of meta-heuristics in comparison to the best amounts of WSM are calculated for some small-sized test instances by Eq. (84) (Khatibi et al. (2017)). Obtained results are written at Table 12. The mean of error percentages for of_1 and of_2 of both meta-heuristic approaches are respectively equal to 2.20% and 2.31%. These results show there is not any limitation for applying the new proposed model at a real environment. Obtained results are strong evidences for the scalability of the new proposed model to apply for a wide range of sizes of problems such as small-sized, medium-sized and large-sized.

$$EP_{of_i} = \frac{Best_{of_i}^{NSGA-II\ or\ MOPSO} - Best_{of_i}^{WSM}}{Best_{of_i}^{WSM}} \qquad \forall\ i \in \{1, 2\} \tag{84}$$

Table 12. Computational Results for Solving Small-sized Test Instances

Test Instance Number	Best Amount						Error Percentage (EP)			
	WSM		NSGA-II		MOPSO		NSGA-II		MOPSO	
	OF1	*OF2*	*OF1*	*OF2*	*OF1*	*OF2*	*OF1*	*OF2*	*OF1*	*OF2*
1	103119	136263	103119	143411	103119	143411	0.00%	5.25%	0.00%	5.25%
2	95726	69329	99923	69330	99923	69330	4.38%	0.00%	4.38%	0.00%
3	95668	80655	95668	80655	95668	80655	0.00%	0.00%	0.00%	0.00%
4	96054	79992	96054	79992	96054	79992	0.00%	0.00%	0.00%	0.00%
5	90235	86214	90235	86214	90235	86214	0.00%	0.00%	0.00%	0.00%
6	188011	125479	200224	125479	200224	125479	6.50%	0.00%	6.50%	0.00%
7	138332	146724	139670	154000	139670	154000	0.97%	4.96%	0.97%	4.96%
8	113319	67896	120029	68489	120029	68489	5.92%	0.87%	5.92%	0.87%
9	99841	128452	99841	136775	99841	136775	0.00%	6.48%	0.00%	6.48%
10	91359	90417	91359	90417	91359	90417	0.00%	0.00%	0.00%	0.00%
Mean	**111166**	**101142**	**113612**	**103476**	**113612**	**103476**	**2.20%**	**2.31%**	**2.20%**	**2.31%**

To prove that the proposed approaches can achieve good Pareto solutions in logical solving time, statistical hypothesis tests are implemented.

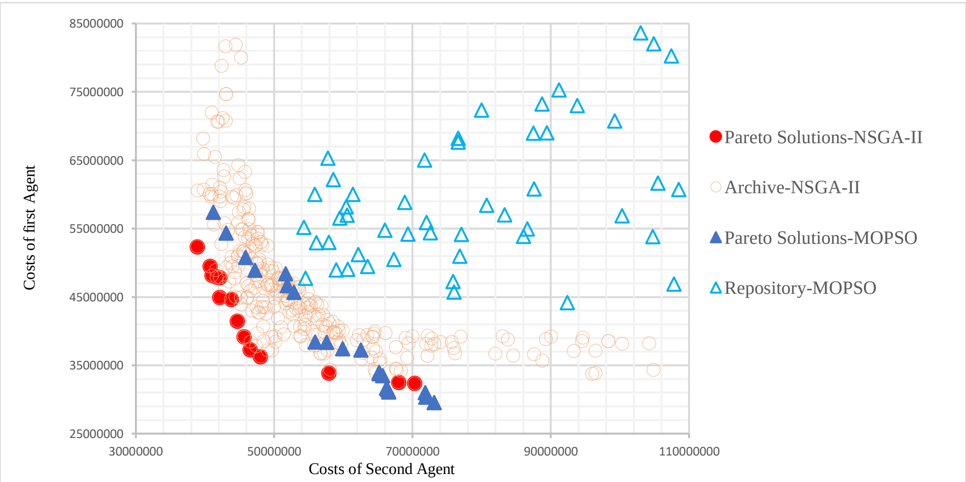

Figure 10. comparison of applying meta-heuristic approaches to solve a large-sized test problem

Table 13.a. Results of Hypothesis Tests (α = 1%)

Test Instance Group Number	Test Results	WSM vs NSGA-II						WSM vs MOPSO					
		NPS	MID	Diversity	Spacing	CPU-Time	Quality	NPS	MID	Diversity	Spacing	CPU-Time	Quality
1	P-value	0.001	0.031	0.003	0.020	0.188	0.000	0.000	0.052	0.003	0.028	0.016	0.000
	best performance	NSGA-II	Non	NSGA-II	Non	Non	WSM	MOPSO	Non	MOPSO	Non	Non	WSM
2	P-value												
	best performance												
3	P-value												
	best performance												
4	P-value												
	best performance												
5	P-value												
	best performance												
6	P-value												
	best performance												
7	P-value												
	best performance												
8	P-value												
	best performance												
9	P-value												
	best performance												
10	P-value	Out of Memory											
	best performance												
11	P-value												
	best performance												
12	P-value												
	best performance												
13	P-value												
	best performance												
14	P-value												
	best performance												
15	P-value												
	best performance												
16	P-value												
	best performance												
17	P-value												
	best performance												
18	P-value												
	best performance												
Mean	Percentage	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	best performance	NSGA-II	Non	NSGA-II	Non	Non	WSM	MOPSO	Non	MOPSO	Non	Non	WSM

Table 14.b. Results of Hypothesis Tests (α = 1%)

Test Instance Group Number	Test Results	MOPSO vs NSGA-II					
		NPS	MID	Diversity	Spacing	CPU-Time	Quality
1	P-value	0.532	0.769	1.000	0.728	0.000	0.168
	best performance	Non	Non	Non	Non	MOPSO	Non
2	P-value	0.061	0.211	0.989	0.225	0.000	0.001
	best performance	Non	Non	Non	Non	MOPSO	NSGA-II
3	P-value	0.001	0.000	0.681	0.012	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
4	P-value	0.000	0.000	0.835	0.000	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	NSGA-II	MOPSO	NSGA-II
5	P-value	0.002	0.001	0.968	0.116	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
6	P-value	0.002	0.000	0.492	0.271	0.000	0.020
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	Non
7	P-value	0.035	0.000	0.721	0.001	0.000	0.000
	best performance	Non	NSGA-II	Non	NSGA-II	MOPSO	NSGA-II
8	P-value	0.001	0.001	0.197	0.150	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
9	P-value	0.005	0.017	0.929	0.536	0.000	0.000

	best performance	NSGA-II	Non	Non	Non	MOPSO	NSGA-II
10	P-value	0.002	0.007	0.572	0.456	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
11	P-value	0.029	0.000	0.483	0.967	0.000	0.000
	best performance	Non	NSGA-II	Non	Non	MOPSO	NSGA-II
12	P-value	0.002	0.000	0.302	0.365	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
13	P-value	0.000	0.000	0.668	0.022	0.000	0.001
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
14	P-value	0.003	0.000	0.164	0.019	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
15	P-value	0.000	0.000	0.596	0.202	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
16	P-value	0.003	0.001	0.271	0.037	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
17	P-value	0.000	0.000	0.239	0.150	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
18	P-value	0.004	0.000	0.166	0.215	0.000	0.000
	best performance	NSGA-II	NSGA-II	Non	Non	MOPSO	NSGA-II
Mean	**Percentage**	**89%**	**83%**	**100%**	**89%**	**100%**	**89%**
	best performance	**NSGA-II**	**NSGA-II**	**Non**	**Non**	**MOPSO**	**NSGA-II**

Hypothesis tests (120 tests) are applied for the six metrics and various sizes of instances at 99 percent confidence level. The results of tests including P-values and determining the approach with best performance are showed at tables 13.a, and 13.b.

As obtained results indicate:

1. The NSGA-II algorithm has better performance than MOPSO in some criteria such as NPS, Spacing and Quality of the solutions. Also, the NSGA-II algorithm performs better than WSM in metrics: NPS and Diversity of solutions.
2. On the other hand, the MOPSO approach has better performance than NSGA-II only in CPU-time and also, the MOPSO performs better in comparison with WSM in metrics: NPS and Diversity of solutions.
3. The WSM has best performance in comparison with meta-heuristics in Quality of solutions.

So, according to the obtained results and above-mentioned facts the NSGA-II is strongly suitable to solve the instances with different sizes in an acceptable CPU-time with high-quality. To prove severely this claim, a technique which is called Technique for Order of Preference by Similarity to Ideal Solution (Hwang and et al. (1993)) is applied to rank the meta-heuristic approaches. This technique has some steps and calculations which are explained as followings:

1. Creation a decision matrix (evl_{ij}) consisting of approaches and evaluating metrics.
2. Calculating the weighted normalized decision matrix (evl_{ij}^{wt}) by Eq. (85). Consider evl_{ij} as performance evaluation of *i-th*, $i \in \{1, 2\}$ meta-heuristic approach according to *j-th*, $j \in \{1, 2, \dots, 6\}$ metric. Assume w_j as weight given to the *j-th* metric.

$$evl_{ij}^{wt} = \frac{w_j . evl_{ij}}{\sqrt{\sum_{k=1}^{2} evl_{kj}^2}} \qquad i \in \{1, 2\} \quad j \in \{1,2,\dots,6\} \tag{85}$$

3. Indicating the best performance (pf_{ij}^{b}) and the worst performance (pf_{ij}^{w}) of the alternatives associated with the impact (positive or negative) of the metrics.
4. Calculating distances between the target alternative and the best performance and the worst performance of the alternatives by equations (86) and (87).

$$s_i^{+} = \sum_{j=1}^{6}(evl_{ij}^{wt}\text{-}pf_{ij}^{b}) \qquad i \in \{1, 2\} \tag{86}$$

$$s_i^{-} = \sum_{j=1}^{6}(evl_{ij}^{wt}\text{-}pf_{ij}^{w}) \qquad i \in \{1, 2\} \tag{87}$$

5. Calculating distance between the target alternative and the worst performance by Eq. (88).

$$CL^{*} = \frac{s_l^{-}}{s_l^{-} + s_l^{+}} \tag{88}$$

The results of calculations of TOPSIS method are represented as tables 14 by 17.

Table 15. Decision (Evaluation) Matrix (Mean of the Results Obtained by Applying Meta-heuristic Approaches to Solve the Test Instances)

Approaches	NPS	CPU-Time	Spacing	Diversity	MID	Quality
NSGA-II	42.83	935.80	689823.5	16292804.9	3069818.7	81.03
MOPSO	16.26	100.46	982354.3	20524121.3	6251686.1	12.31

Table 16. Weighted Normalized Decision Matrix

Approaches	NPS	CPU-Time	Spacing	Diversity	MID	Quality
NSGA-II	0.935	0.994	0.575	0.622	0.441	0.989
MOPSO	0.355	0.107	0.818	0.783	0.898	0.150

Table 17. Distances Between the Approaches and Worst Performance

Approaches	S_l^{+}	S_l^{-}	CL^{*}
NSGA-II	0.902	1.144	0.559
MOPSO	1.144	0.902	0.441

Table 18. Ranking of the Approaches

Approaches	Rank
NSGA-II	1
MOPSO	2

5. Conclusion

In this article, a new model and mathematics has been presented for well-known vehicle routing problem with cross-docking. According to the literature, the new presented model has several novelties such as considering two agents, adding the group purchasing strategies and considering the perishability of the commodities at cross-dock. For the reason why the new model is multi-agent and NP-hard, two meta-heuristic approaches such as NSGA-II and MOPSO have been proposed to solve it. To validate the new proposed model, several test instances with different dimensions have been solved and analyzed. The performance of the solving approaches has been compared according to the six criteria. Also, the proposed algorithms have been ranked by using TOPSIS technique. On the basis of obtained results, it has been strongly recommended to apply NSGA-II for the real (various)-sized instances.

There were some limitations in our proposed model such as fixed travel durations, pre-determined demands of customers and fixed planning horizon. So, at the future studies the corresponding model can be extended with considering different speeds for the vehicles, applying machine learning tools to forecast the demands of customers and also describing variable time horizons. Additionally, full load truck strategies and constraints better describing perishability conditions of commodities (e.g., considering probability distributions) can be integrated to the model. Moreover, other multi-objective approaches can be applied to tackle the provided model and their performance could be evaluated by comparing with the proposed algorithms in this paper.

References

Agnetis, A., J.-Ch. Billaut, S. Gawiejnowic, D. Pacciarelli, and A. Soukhal, *Multiagent scheduling: models and algorithms.* Berlin: Springer, 2014.

Agustina, D., C.K.M. Lee, and R. Piplani, A Review: *Mathematical modles for cross docking planning*. International Journal of Engineering Business Management, 2010. **2**: p. 13.

Ahmadi, A., M.S. Pishvaee, and M. Heydari, *How group purchasing organisations influence healthcare-product supply chains? an analytical approach*. Journal of the Operational Research Society, 2018. **70**(2): p. 280-293.

Ahmadizar, F., M. Zeynivand, and J. Arkat. *Two-level vehicle routing with cross-docking in a three-echelon supply chain: a genetic algorithm approach*. Applied Mathematical Modelling, 2015. **39**: p. 7065-7081.

Alinaghian, M., M. Rezaei Kalantari, A. Bozorgi-Amiri, and N. Golghamat Raad, *A novel mathematical model for cross-dock open-close vehicle routing problem with splitting*. I.J. Mathematical Sciences and Computing, 2016. **3**: p. 21-31.

Amini, A. and R. Tavakkoli-Moghaddam, *A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks*. Computers & Industrial Engineering, 2016. **96**: p. 180-191.

Amorim, P. H. Meyer, C. Almeder, and B. Almada-Lobo, *Managing perishability in production-distribution planning: a discussion and review*. Flexible Services and Manufacturing Journal, 2013. **25**: p. 389-413.

Apte, U.M. and S. Viswanathan, *Effective cross docking for improving distribution efficiencies*. International Journal of Logistics Research and Applications, 2000. **3**(3): p. 291-302.

Azadeh, A., M. Ravanbakhsh, M. Rezaei-Malek, M. Sheikhalishahi, A. Taheri-Moghaddam, *Unique NSGA-II and MOPSO algorithms for improved dynamic CMS by considering human factors*. Applied Mathematical Modelling, 2017. **48**: p. 655-672.

Boysen, N. and M. Fliedner, *Cross dock scheduling: classification, literature review and research agenda*. Omega, 2010. **38**(6): p. 413-422.

Deb, K., *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

Deb, K., S. Agrawal, A. Pratap, T. Meyarivan, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*. In: proceedings of the parallel problem solving from nature VI (PPSN-VI) conference, 2000. p. 849-858.

Ehrgott, M., *Multi-criteria optimization*. Berlin, Heidelberg, New York: Springer, 2005.

Enderer, F., C. Contardo, and I. Contreras, *Integrating dock-door assignment and vehicle routing with cross-docking*. Computers & Operations Research, 2017. **88**: p. 30-43.

Farmand, N., H. Zarei, M. Rasti-Barzoki, *Two meta-heuristic algorithms for optimizing a multi-objective supply chain scheduling problem in an identical parallel machines environment.* International Journal of Industrial Engineering Computations, 2021. **12**: p. 249-272.

Grangier, P., M. Gendreau, F. Lehuede and L.-M Rousseau, *A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking.* Computers & Operations Research, 2017. **84**: p. 116-126.

Grangier, P., M. Gendreau, F. Lehuede and L.-M Rousseau, *The vehicle routing problem with cross-docking and resource constraints*. Journal of Heuristics, 2021. **27**(1): p. 31-61.

Hwang, Ch.-L., Y.-J. Lai, T.-Y. Liu, *A new approach for multiple objective decision making*. Computers & Operations Research, 1993. **20**(8): p. 889-899.

Khatibi, S., M. Khakzar Bafruei, M. Rahmani, *Multi-objective model of airport gate scheduling problem using NSGA-II algorithm and epsilon constraint*. Journal of Modeling in Engineering, 2017. **15**(51): p. 397-410.

Maknoon, Y. and G. Laporte, *Vehicle routing with cross-dock selection.* Computers & Operations Research, 2017. **77**: p. 254-266.

Nasiri, M.M., A. Rahbari, F. Werner and R. Karimi, *Incorporating supplier selection and order allocation into the vehicle routing and multi-cross-dock scheduling problem*. International Journal of Production Research, 2018. **56**(19): p. 6527-6552.

Nikolopoulou, A.I., P.P. Repoussis, Ch.D. Tarantilis and E.E. Zachariadis, *Moving products between location pairs: cross-docking versus direct-shipping.* European Journal of Operational Research, 2017. **256**(3): p. 803-819.

Rahbari, A., and M.M. Nasiri, *Robust vehicle routing and cross-dock scheduling with uncertain loading and unloading time*. In: 1st International Conference on Systems Optimization & Business Management, 2017. Babol, Iran.

Rong, A., R. Akkerman, M. Grunow, *An optimization approach for managing fresh food quality throghout the supply chain*. Int. J. Production Economics, 2011. **131**: p. 421-429.

Sarrafha, K., A. Kazemi, and A. Alinezhad, *Integrated production-distribution planning problem in a multi-echelon supply chain network design and optimization: a multi-objective evolutionary approach*. International Journal of Industrial Engineering & Production Management, 2015. **26**(3): p. 283-298.

Sazvar, Z., S.M.J. Mirzapour Al-e-hashem, A. Baboli, M.R. Akbari Jokar, *A bi-objective stochastic programming model for a centralized green supply chain with deteriorating products*. Int. J. Production Economics, 2014.

150: p. 140-154.
Stephan, K. and N. Boysen, *Cross-docking*. Journal of Management Control, 2011. **22**: p. 129-137.
Van Belle, J., P. Valckenaers, and D. Cattrysse, *Cross-docking: state of the art*. Omega, 2012. **40**(6): p. 827-846.
Yan, Y., R. Zhao, and Y. Lan, *Asymetric retailers with different moving sequences: group buying vs. individual purchasing*. European Journal of Operational Research, 2017. **261**: p. 903-917.
Yin, P.-Y. and Y.-L. Chaung. *Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking*. Applied Mathematical Modeling, 2016. **40**: p. 9302-9315.
Yin, P.-Y. S.-R. Lyu and Y.-L. Chaung, *Cooperative coevolutionary approach for integrated vehicle routing and scheduling using cross-dock buffering*. Engineering Applications of Artificial Intelligence, 2016. **52**: p. 40-53.
Yu, V.F., P. Jewpanya, and V. Kachitvichyanukul, *Particle swarm optimization for the multi-period cross-docking distribution problem with time windows.* International Journal of Production Research, 2016. **54**(2): p. 509-525.

Appendix

At this section the linearization of the non-linear equations is represented.

At cost functions it can be done by replacing the term $x_{isbcn} \times \alpha_{sc}^{e}$ with $x\alpha_{isbcn}^{e}$ and also adding the equations (89-91).

$$x\alpha_{isbcn}^{e} \geq 1 - M(2 - x_{isbcn} - \alpha_{sc}^{e}) \qquad \forall\, i,s,b,c,n,e \qquad (89)$$
$$x\alpha_{isbcn}^{e} \leq x_{isbcn} \qquad \forall\, i,s,b,c,n,e \qquad (90)$$
$$x\alpha_{isbcn}^{e} \leq \alpha_{sc}^{e} \qquad \forall\, i,s,b,c,n,e \qquad (91)$$

At cost functions it can be done by replacing the phrases $\sum_{i,s,b\epsilon BB,c,n} x_{isbcn}(FCA_s + FCV_n + VCV_n(atp_n - dtp_n))/\sum_{i,s,b,c,n} x_{isbcn}$ and $\sum_{i,s,b\epsilon\{B\setminus BB\},c,n} x_{isbcn}(FCA_s + FCV_n + VCV_n(atp_n - dtp_n))/\sum_{i,s,b,c,n} x_{isbcn}$ respectively with terms $\sum_{i,b\epsilon BB,c,n} stcx_{isbcn}$ and $\sum_{i,b\epsilon\{B\setminus BB\},c,n} stcx_{isbcn}$ and adding the equations (92-102).

$$\sum_{i,b,c,n} stcx_{isbcn} = \sum_{i,b,c,n} x_{isbcn}\,(FCA_s + FCV_n + VCV_n(atp_n - dtp_n))) \qquad \forall\, s \qquad (92)$$
$$stcx_{isbcn} = stc_s \times x_{isbcn} \qquad \forall\, i,s,b,c,n \qquad (93)$$
$$stcx_{isbcn} \leq stc_s \qquad \forall\, i,s,b,c,n \qquad (94)$$
$$stcx_{isbcn} \leq M \times x_{isbcn} \qquad \forall\, i,s,b,c,n \qquad (95)$$
$$stcx_{isbcn} \geq stc_s - M \times (1 - x_{isbcn}) \qquad \forall\, i,s,b,c,n \qquad (96)$$
$$atpx_{isbcn} \leq atp_n \qquad \forall\, i,s,b,c,n \qquad (97)$$
$$atpx_{isbcn} \leq M \times x_{isbcn} \qquad \forall\, i,s,b,c,n \qquad (98)$$
$$atpx_{isbcn} \geq atp_n - M \times (1 - x_{isbcn}) \qquad \forall\, i,s,b,c,n \qquad (99)$$
$$dtpx_{isbcn} \leq dtp_n \qquad \forall\, i,s,b,c,n \qquad (100)$$
$$dtpx_{isbcn} \leq M \times x_{isbcn} \qquad \forall\, i,s,b,c,n \qquad (101)$$
$$dtpx_{isbcn} \geq dtp_n - M \times (1 - x_{isbcn}) \qquad \forall\, i,s,b,c,n \qquad (102)$$

At objective functions it can be done by replacing the phrases $\sum_{j,b\epsilon BB,n} y_{jbn}(\,FCV_n + VCV_n(atp_n - dtp_n))/\sum_{j,b,n} y_{jbn}$ and $\sum_{j,b\epsilon\{B\setminus BB\},n} y_{jbn}(\,FCV_n + VCV_n(atp_n - dtp_n))/\sum_{j,b,n} y_{jbn}$ respectively with terms $\sum_{j,b\epsilon BB,n} tdcy_{jbn}$ and $\sum_{j,b\epsilon\{B\setminus BB\},n} tdcy_{jbn}$ and adding the equations (103-113).

$$\sum_{j,b} tdcy_{jbn} = \sum_{j,b} y_{jbn}\,(FCV_n + VCV_n(atp_n - dtp_n)) \qquad \forall\, n \qquad (103)$$
$$tdcy_{jbn} = tdc_n \times y_{jbn} \qquad \forall\, j,b,n \qquad (104)$$
$$tdcy_{jbn} \leq tdc_n \qquad \forall\, j,b,,n \qquad (105)$$
$$tdcy_{jbn} \leq M \times y_{jbn} \qquad \forall\, j,b,,n \qquad (106)$$
$$tdcy_{jbn} \geq tdc_n - M \times (1 - y_{jbn}) \qquad \forall\, j,b,,n \qquad (107)$$
$$atpy_{jbn} \leq atp_n \qquad \forall\, j,b,,n \qquad (108)$$
$$atpy_{jbn} \leq M \times y_{jbn} \qquad \forall\, j,b,,n \qquad (109)$$
$$atpy_{jbn} \geq atp_n - M \times (1 - y_{jbn}) \qquad \forall\, j,b,,n \qquad (110)$$
$$dtpy_{jbn} \leq dtp_n \qquad \forall\, j,b,,n \qquad (111)$$
$$dtpy_{jbn} \leq M \times y_{jbn} \qquad \forall\, j,b,,n \qquad (112)$$
$$dtpy_{jbn} \geq dtp_n - M \times (1 - y_{jbn}) \qquad \forall\, j,b,,n \qquad (113)$$

At cost functions the term $eat_b \times k_{bc}$ can be replacing by $eatk_{bc}$ and also adding the equations (114-116).

$$eatk_{bc} \geq eat_b - M(1 - k_{bc}) \qquad \forall\, j, b, c, n \tag{114}$$
$$eatk_{bc} \leq eat_b \qquad \forall\, j, b, c, n \tag{115}$$
$$eatk_{bc} \leq M \times k_{bc} \qquad \forall\, j, b, c, n \tag{116}$$

At Eq. 44 it can be done by replacing the term $x_{isbcn'} \times y_{jbn}$ with $xy_{ijsbcnn'}$ and also adding the equations (117-119).

$$xy_{ijsbcnn'} \geq 1 - M\big(2 - x_{isbcn'} - y_{jbn}\big) \qquad \forall\, i, j, s, b, c, n, n'\ (n \neq n') \tag{117}$$
$$xy_{ijsbcn} \leq x_{isbcn} \qquad \forall\, i, j, s, b, c, n, n'\ (n \neq n') \tag{118}$$
$$xy_{ijsbcn} \leq y_{jbn} \qquad \forall\, i, j, s, b, c, n, n'\ (n \neq n') \tag{119}$$

At Eq. 57 it can be done by replacing the term $y_{jbn} \times k_{bc}$ with yk_{jbcn} and also adding the equations (120-122).

$$yk_{jbcn} \geq 1 - M\big(2 - y_{jbn} - k_{bc}\big) \qquad \forall\, j, b, c, n \tag{120}$$
$$yk_{jbcn} \leq y_{jbn} \qquad \forall\, j, b, c, n \tag{121}$$
$$yk_{jbcn} \leq k_{bc} \qquad \forall\, j, b, c, n \tag{122}$$

Linearization constraints of Eq. (71) are as followings:

$$Q_c^e \alpha_{sc}^e \leq oq_{sc} \leq Q_c^{e+1} \alpha_{sc}^e \qquad \forall\, s, c, e \tag{123}$$
$$\sum_e \alpha_{sc}^e = 1 \qquad \forall\, s, c \tag{124}$$