CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

The Inventory-Routing Problem with Transshipment

Leandro Callegari Coelho
Jean-François Cordeau
Gilbert Laporte

March 2011

CIRRELT-2011-21

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

The Inventory-Routing Problem with Transshipment

Leandro Callegari Coelho[1,2,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3] Department of Management Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. This paper introduces the Inventory-Routing Problem with Transshipment (IRPT). This problem arises when vehicle routing and inventory decisions must be made simultaneously, which is typically the case in vendor-managed inventory systems. Heuristics and exact algorithms have already been proposed for the Inventory-Routing Problem (IRP), but these algorithms ignore the possibility of performing transshipments between customers so as to further reduce the overall cost. We present a formulation that allows transshipments, either from the supplier to customers or between customers. We also propose an adaptive large neighborhood search heuristic to solve the problem. This heuristic manipulates vehicle routes while the remaining problem of determining delivery quantities and transshipment moves is solved through a network flow algorithm. Our approach can solve four different variants of the problem: the IRP and the IRPT, under maximum level and order-up-to level policies. We perform an extensive assessment of the performance of our heuristic.

Keywords. Inventory-routing problem, transshipment, ALNS, heuristic.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 227837-09 and 39682-10. This support is gratefully acknowledged. The authors thank Glaydston M. Ribeiro for his help with computations at an early stage of this work, as well as Andrew Goldberg and Boris Cherkassky for making their implementation of the scaling push-relabel minimum-cost flow algorithm available.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2011

1 Introduction

Logistics is now widely recognized as a value adding center in organizations through product availability, consistency of deliveries, accuracy in inventory and demand management, and ease of placing orders. Vendor-managed inventory (VMI) is one of the most up-to-date examples of value added through logistics. This practice constitutes a streamlined approach to inventory management in which the supplier makes replenishment decisions based on specific inventory and supply chain policies.

Vendor-managed inventory is a win-win situation: suppliers save on distribution and production costs since they can combine and coordinate demands and shipments for different customers, and buyers gain by not allocating resources to controlling and managing inventories. The supplier then has to make three simultaneous decisions: 1) when to serve a given customer, 2) how much to deliver, and 3) how to combine customers into routes. Regarding the size of deliveries, one of two policies is typically applied: the order-up-to (OU) policy or the maximum level (ML) policy. Under the OU policy the quantity delivered to a customer is that to fill its inventory capacity; under the ML policy the supplier can decide how much to deliver to a customer, as long as its holding capacity is respected [20].

In addition to these features, this paper introduces the concept of *transshipment* within inventory-routing. Under this policy, goods may be shipped to a customer who experiences a shortage, either directly from the supplier or from another customer. This occurs, for example, between stores belonging to a same chain [21, 22] which can ship merchandise to one another when the need occurs. From a practical point of view, the use of a transshipment option contributes to lead-time and cost reductions.

The drawback of VMI is that it requires the solution of a very difficult mathematical problem, called the Inventory-Routing Problem (IRP), itself a combination of two well-studied problems: inventory management and vehicle routing. According to Andersson et al. [1] "no commercially available systems provide decision support for combined inventory management and routing". Scientific research on the IRP is relatively recent compared to that on other optimization problems, such as the Vehicle Routing Problem (VRP). Speranza and Ukovich [30] note the existence of distinct extensive literature reviews on transportation and on inventory management problems, but relatively few studies exploit their integration. This is still true to this day. A quick search on the ABI/INFORM Global database shows over 580 scholarly publications on the VRP, but less than 70 on the IRP. Recent reviews on the IRP found fewer than a hundred papers addressing the combined VRP-inventory management problem [1, 10].

Several variants of the IRP have arisen since this problem was first introduced by Bell et al. [5]. These include the IRP with a single customer [6, 11, 31], the IRP with multiple customers [2, 5, 8, 16], the stochastic IRP [16, 17, 18], the IRP with direct deliveries [12, 13, 15, 16, 19], the multi-item IRP [4, 26, 29, 30], and the IRP with heterogeneous fleet [8, 9, 24]. There are so many ways of modeling and solving IRPs that different authors rarely define the problem in

exactly the same way. In addition, real-life problems combining vehicle routing and inventory management concerns are often dynamic or stochastic. We next describe three important algorithms for the IRP which we will use as benchmarks for our own algorithm.

Bertazzi et al. [7] study the case with multiple retailers, dynamic deterministic demand rates and the OU policy. They compute the quantity to be delivered to each customer for each combination of time instants and insert it in the cheapest position. The solution is then iteratively improved by selecting customer pairs, removing them from the current solution and reinserting them. Any improving move is implemented.

The first exact algorithm for the IRP is that of Archetti et al. [2]. It handles the single-vehicle case without backlogging, under the OU inventory policy. This algorithm is based on a classical branch-and-cut scheme in which the subtour elimination constraints are initially relaxed. Branching is performed in priority on the assignment of customers to delivery periods, and then on routing variables. The instances solved in this paper have become standard benchmarks.

Recently, Archetti et al. [3] have developed a hybrid heuristic combining tabu search and integer programming for the same problem. Starting from a feasible solution, the algorithm explores the neighbourhood of the current solution and performs occasional jumps to new regions of the search space. Infeasible solutions are temporarily accepted, namely due to a stockout at the supplier or exceeded vehicle capacity.

Reverting to the main theme of this paper, the concept of transshipment appears in the work of [21, 22, 23] but has not yet been formally integrated within the context of inventory-routing. Its inclusion adds an extra layer of complexity to an already difficult problem. For this reason, we believe it is unrealistic to contemplate the use of an exact algorithm for the IRPT. We have therefore developed an adaptive large neighbourhood search (ALNS) heuristic for it. This type of algorithm was initially put forward by [27] in the context of the VRP and extends a concept initially proposed by Shaw [28]. The algorithm we propose is designed to handle the specific features of the IRPT. It is flexible and can easily handle the OU and ML replenishment policies.

The main scientific contributions of this paper are the introduction of a transshipment option within the context of inventory-routing and the development of a powerful and flexible ALNS heuristic to solve four variants of the problem: the IRPT with transshipment (IRPT) and the IRP without transshipment (IRP), under an OU or an ML replenishment policy. These four variants will be referred to as IRPT-OU, IRPT-ML, IRP-OU and IRP-ML.

The remainder of the paper is organized as follows. In Section 2 we introduce and describe the IRP. Section 3 presents a mixed-integer linear programming formulation for the four variants of the problem considered in the paper, and for a restriction in which routing is fixed. In Section 4 we present our ALNS algorithm, followed by computational results, in Section 5, and by our conclusions in Section 6.

2 Problem description

We now formally introduce the IRPT. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A}$ is the arc set. Vertex 0 represents the supplier and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. Both the supplier and customers incur unit inventory holding costs h_i per period ($i \in \mathcal{V}$), and each customer has an inventory holding capacity C_i. The length of the planning horizon is p and, at each time period $t \in \mathcal{T} = \{1, ..., p\}$ the quantity of product made available at the supplier is r^t. We assume the supplier has enough inventory to meet all the demand during the planning horizon and that inventories are not allowed to be negative, i.e., the supplier can only ship what he holds in stock with no backlogging option. At the beginning of the planning horizon the decision maker knows the current inventory level of the supplier and of the customers (I_0^0 and I_i^0), and receives the information on the demand d_i^t of each customer i for each time period t.

A single vehicle of capacity Q is available. This vehicle is able to perform one route at the beginning of each time period to deliver products from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$.

Transshipments can be made later in the time period. A transshipment can start from the depot or from any customer in a subset $\mathcal{R} \subseteq \mathcal{V}'$, i.e., these customers can dispatch goods to other customers as needed. Transshipments can occur when it is profitable to ship goods from the depot to a customer on a special request basis, or from customer $i \in \mathcal{R}$ to customer $j \in \mathcal{V}'$. This can be done by subcontracting to a carrier who will pickup goods either at the supplier or from any transshipment point. These outsourced deliveries are only made by direct shipping and the unit cost associated with transshipping products from i to j is b_{ij}.

It is possible that both the supplier's vehicle and the subcontractor visit the same customer within the same time period: the supplier's vehicle first delivers to the customer according to the OU or to the ML policy, and the subcontractor may later deliver to that customer according to the ML policy. We also assume that all orders and deliveries can be performed during the same time period, which means that lead times are negligible.

The objective of the problem is to minimize the total cost while meeting the demand for each customer. The replenishment plan is subject to the following constraints:

- inventories at each customer can never exceed the maximal available inventory capacity;
- inventories are not allowed to be negative, i.e., all demand must be met by previous inventory plus deliveries performed during the time period considered;
- if the supplier's vehicle visits a customer in a time period, an OU or an ML replenishment policy applies;

- the supplier's vehicle can perform at most one route per time period, starting and ending at the supplier;
- the vehicle capacity cannot be exceeded.

The solution to the problem should determine:

- which customers to serve in each time period using the supplier's vehicle;
- which route to use in each time period;
- how much to transship from every $i \in \mathcal{R} \cup \{0\}$ to every $j \in \mathcal{V}'$ in each time period.

3 Mathematical models

In addition to the notation already introduced, we define an artificial period $p+1$ and the set $\mathcal{T}' = \mathcal{T} \cup \{p+1\}$. The model works with the following binary variables: x_{ij}^t is equal to 1 if and only if customer j immediately follows customer i on the route of the supplier's vehicle in period t. Let the quantity of product delivered from the supplier to each customer i at each time period t be y_i^t and let also w_{ij}^t be the amount of product delivered directly from $i \in \mathcal{R} \cup \{0\}$ to customer $j \in \mathcal{V}'$ at period t using the outsourced carrier.

3.1 Model for IRPT-OU

In the IRPT, the total cost to be minimized is the sum of inventory holding costs at the supplier and at the customers, of routing costs for the supplier's vehicle and of transshipment costs:

$$\min \sum_{t\in\mathcal{T}'} h_0 I_0^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}'} h_i I_i^t + \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\sum_{t\in\mathcal{T}} c_{ij} x_{ij}^t + \sum_{i\in\mathcal{R}\cup\{0\}}\sum_{j\in\mathcal{V}'}\sum_{t\in\mathcal{T}} b_{ij} w_{ij}^t. \quad (1)$$

The constraints are as follows:

3.1.1 Inventory definition at the supplier

The inventory level at the supplier in period t is defined at the beginning of the period and is given by its previous inventory level (period $t-1$), plus the quantity r^{t-1} made available in period $t-1$, minus the total quantity shipped to the customers using the supplier's vehicle in period $t-1$, minus the total quantity transshipped to the customers in period $t-1$:

$$I_0^t = I_0^{t-1} + r^{t-1} - \sum_{i\in\mathcal{V}'} y_i^{t-1} - \sum_{i\in\mathcal{V}'} w_{0i}^{t-1} \quad t \in \mathcal{T}', \quad (2)$$

where $r^0 = y_i^0 = w_{0i}^0 = 0, i \in \mathcal{V}'$.

3.1.2 Stockout constraints at the supplier

These constraints impose that the supplier's inventory cannot be less than the total amount of product delivered in period t:

$$I_0^t \geq \sum_{i \in \mathcal{V}'} (y_i^t + w_{0i}^t) \quad t \in \mathcal{T}. \tag{3}$$

3.1.3 Inventory definition at the customers

Likewise, the inventory level at each retailer in period t is given by its previous inventory level in period $t-1$, plus the quantity y_i^{t-1} delivered by the supplier's vehicle in period $t-1$, plus the total quantity transshipped in period $t-1$, minus the total quantity transshipped to other customers in period $t-1$, minus its demand in period $t-1$, that is:

$$I_i^t = I_i^{t-1} + y_i^{t-1} + \sum_{j \in \mathcal{R} \cup \{0\}} w_{ji}^{t-1} - \sum_{j \in \mathcal{V}'} w_{ij}^{t-1} - d_i^{t-1} \quad i \in \mathcal{V}' \quad t \in \mathcal{T}'. \tag{4}$$

3.1.4 Stockout constraints at the customers

These constraints guarantee that for each customer $i \in \mathcal{V}'$ the inventory level I_i^t remains non-negative at all time:

$$I_i^t \geq 0 \quad i \in \mathcal{V} \quad t \in \mathcal{T}'. \tag{5}$$

3.1.5 Maximal inventory level at the customers

These constraints guarantee that for each customer $i \in \mathcal{V}'$ the inventory level I_i^t remains below the maximum level C_i at all time:

$$I_i^t \leq C_i \quad i \in \mathcal{V} \quad t \in \mathcal{T}'. \tag{6}$$

3.1.6 Quantities delivered

These sets of constraints ensure that the quantity delivered by the supplier's vehicle to each customer $i \in \mathcal{V}'$ in each period $t \in \mathcal{T}$ will fill the customer's inventory capacity if the customer is served, and will be zero otherwise:

$$y_i^t \geq C_i \sum_{j \in \mathcal{V}'} x_{ij}^t - I_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{7}$$

$$y_i^t \leq C_i - I_i^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{8}$$

$$y_i^t \leq C_i \sum_{j \in \mathcal{V}} x_{ij}^t \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{9}$$

If customer i is not visited in period t, then constraints (9) mean that the quantity delivered to it will be zero (while constraints (7) and (8) are still respected). If, otherwise, customer i is visited in period t, constraints (9) limit the quantity delivered to the customer's inventory holding capacity, and this bound is tightened by constraints (8), making it impossible to deliver more than what would exceed this capacity. Constraints (7) model the OU replenishment policy, ensuring that the quantity delivered will be exactly the bound provided by constraints (8).

3.1.7 Vehicle capacity

These constraints guarantee that the vehicle's capacity is not exceeded:

$$\sum_{i \in \mathcal{V}'} y_i^t \leq Q \quad t \in \mathcal{T}. \tag{10}$$

3.1.8 Routing constraints

These constraints guarantee that a feasible route is determined to visit all customers served in period t:

a) Flow conservation constraints: these constraints impose that the number of arcs entering and leaving a vertex should be the same:

$$\sum_{i \in \mathcal{V}} x_{ij}^t = \sum_{i \in \mathcal{V}} x_{ji}^t \quad j \in \mathcal{V} \quad t \in \mathcal{T}. \tag{11}$$

b) A single vehicle is available:

$$\sum_{i \in \mathcal{V}} x_{i0}^t \leq 1 \quad t \in \mathcal{T}. \tag{12}$$

c) Subtour elimination constraints:

$$v_i^t - v_j^t + Q x_{ij}^t \leq Q - y_j^t \quad i \in \mathcal{V}' \quad j \in \mathcal{V}' \quad t \in \mathcal{T}; \tag{13}$$

$$y_i^t \leq v_i^t \leq Q \quad i \in \mathcal{V}' \quad t \in \mathcal{T}. \tag{14}$$

3.1.9 Integrality and nonnegativity constraints

$$v_i^t, y_i^t, w_{ji}^t \geq 0 \quad i \in \mathcal{V}' \quad j \in \mathcal{R} \cup \{0\} \quad t \in \mathcal{T}; \tag{15}$$

$$x_{ij}^t \in \{0, 1\} \quad i, j \in \mathcal{V}, i \neq j \quad t \in \mathcal{T}. \tag{16}$$

3.2 Adaptations to IRPT-ML, IRP-OU and IRP-ML

This model can be modified to enforce the ML replenishment policy by dropping constraints (7). Similarly, to forbid transshipments one only has to set all w_{ij}^t variables equal to zero. Thus all four versions of the IRPT can be modeled through the same formulation.

3.3 Model for the IRPT with fixed routes

If one fixes routing variables x_{ij}^t, the remaining problem reduces to a network flow problem defined by the I_i^t, y_i^t and w_{ij}^t variables. The flow conservation equations are given by (2) and (4). The lower and upper bounds on the flows are defined by (3) and (5)−(9). Vehicle capacity constraints (10) still define an upper bound on the quantity delivered by the vehicle, even though the customers to be visited are fixed. Constraints (11)−(13) are not relevant in the flow models because their variables are fixed.

Figure 1 depicts the network flow model for a small network with two customers and two time periods. The supplier and the customers are represented by vertices replicated for each time period, plus one extra set of vertices for initial inventories, and one extra set for the decisions made at the last time period. The supplier and each customer carry their inventories between successive time periods. The corresponding solid arcs in the figure have unit inventory holding costs, and the flows on these arcs are bounded above by the customer's inventory capacity (infinite for the supplier). At each period t the supplier receives r^t units of the product and customer i has a demand equal to d_i^t.

The vehicle is represented by one vertex at each period, receiving an arc from the supplier at the same period with a flow up to Q units and no cost associated to it, and is then connected to each customer receiving a delivery at that period, a decision made by the ALNS heuristic, also with no cost. These arcs are dashed in Figure 1 at period two, supposing that the ALNS has decided to make a visit to customer 2 only. We have added dotted arcs representing transshipment options from the supplier and from every customer to every other customer. For the sake of clarity, Figure 1 only shows these arcs for period 1, but they are actually present in all periods. This allows the network flow solution to serve a given customer through a transshipment if a later routing delivery would violate vehicle capacity, or if any inventory constraint is not satisfied by the routing decisions.

The OU policy is enforced by fixing the flow on the arcs linking customers in different successive time periods: once customer i is visited in period t, the arc connecting it to itself at the next period has a flow equal to $U_i - d_i^t$. The network flow algorithm only computes the quantities delivered from all transshipment arcs, as the quantities delivered by the supplier are fixed by the OU policy. When the ML replenishment policy is in place, no extra action is needed.

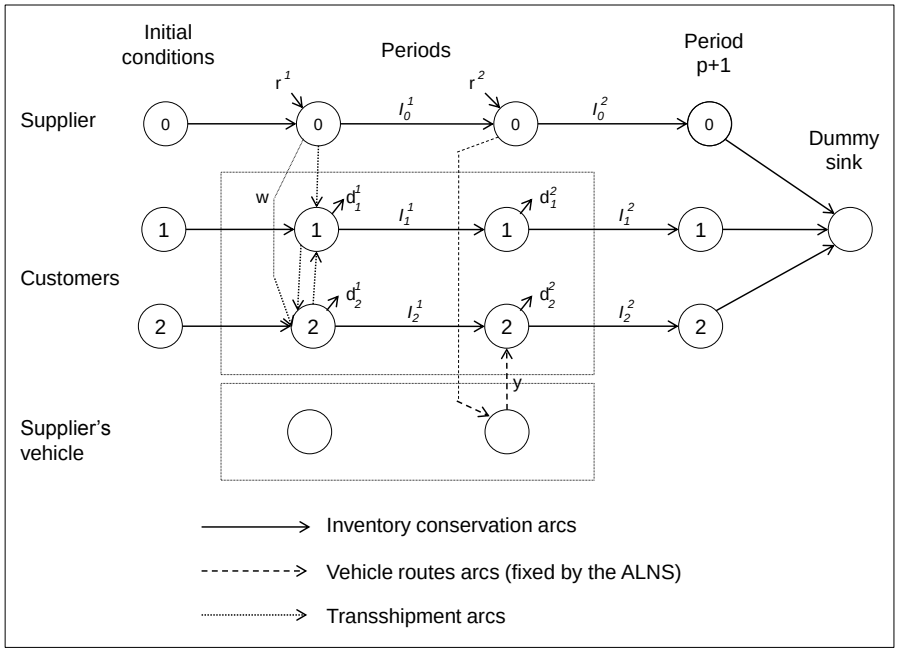

Figure 1: Network flow problem for two customers and two time periods.

4 Adaptive large neighborhood search heuristic

We now describe our ALNS heuristic. The main framework is made up of four components.

1. **Large neighborhood:** At each iteration, a number of customers are removed from their current route and are eventually reinserted. This fixes the decisions regarding routing, and the problem is passed to a network flow solver to optimize all remaining decisions simultaneously (minimize total costs taking into account inventory holding costs, transshipments and delivery quantities), as described in Section 3.3.

2. **Adaptive search engine:** The choice of which movement to perform is governed by a roulette-wheel mechanism in which each movement is assigned a weight depending on its past performance. Let ω_i be a measure of how well movement i has performed in the past; then given h moves with weights ω_i, movement j will be selected with probability $\omega_j / \sum_{i=1}^{h} \omega_i$.

3. **Adaptive weight adjustment:** The search is divided into segments of φ iterations each, and weights are computed by taking into account the performance of the movements during the last segment. In the first

segment all movements have the same weight. At each iteration, scores are updated as follows: if a movement finds a new best solution, its score is increased by σ_1; if it finds a solution better than the incumbent, its score is increased by σ_2; if the solution is not better but is still accepted, the score is increased by σ_3. Obviously $\sigma_1 > \sigma_2 > \sigma_3$. After φ iterations, the weights are updated considering the scores obtained in the last segment and scores are reset to zero. To do so, let π_i and o_{ij} be, respectively, the score of the movement i and the number of times movement i has been used in the last segment j. Then,

$$\omega_{i,j+1} = \begin{cases} \omega_{ij} & \text{if } o_{ij} = 0 \\ (1-\eta)\omega_{ij} + \eta\pi_i/o_{ij} & \text{if } o_{ij} \neq 0, \end{cases} \tag{17}$$

where $\eta \in [0, 1]$ is called the reaction factor, controling how quickly the weight adjustment reacts to changes in the movement performance.

4. **Acceptance and stopping criteria:** We use the same acceptance criteria as in simulated annealing: given a solution s, a neighbor solution s' is accepted if $z(s') < z(s)$, and with probability $e^{-(z(s')-z(s))/\tau}$ otherwise, where $z(s)$ is the solution cost defined by (1) and $\tau > 0$ is the current temperature. The temperature starts at τ_{start} and is decreased by a cooling rate factor ϕ at each iteration, where $0 < \phi < 1$. To avoid long computations for large and difficult instances, we limit the running time to one hour and also to a maximum number of iterations, as described in Section 4.3.

4.1 Initial solution

The initial solution is generated by randomly selecting 75% of the customers and randomly assigning them to some periods of the planning horizon. Their insertion in the routes follows the cheapest insertion rule. Our computational experiments have shown that the initial solution does not have a significant impact on the overall solution cost or running time.

4.2 List of movements

Unlike related problems like the VRP, where every removal is accompanied by an insertion, one may decide to remove a vertex from some periods and not reinsert it back. This partial solution will still be feasible when transshipments are made. Moreover, in the IRPT family, it is feasible and sometimes optimal not to create any route because transshipments can always be made. This observation has motivated us not to use the traditional destroy and repair framework of Shaw [28] and Pisinger and Ropke [25] in which each destroy move is always followed by a repair move, but to keep the option of making only a removal or only an insertion.

As in Pisinger and Ropke [25] we artificially increase the score of complex movements because they are much more time consuming than the others. This reduces the probability of selecting them. In what follows, ρ is an integer randomly drawn from the interval $[1, n]$ using a semi-triangular distribution with a negative slope.

4.2.1 **Randomly remove ρ**

This movement randomly selects one period and removes one customer from it. It is repeated ρ times. This movement is useful for refining the solution, since it does not change it much when ρ is small (which happens frequently), but still yields a major transformation when ρ is large.

4.2.2 **Randomly insert ρ**

This movement randomly inserts ρ customers into the current solution. Specifically, it selects one random customer and one random period, and inserts the customer into the route of that period if the customer is not yet routed. This movement is repeated ρ times.

4.2.3 **Remove worst ρ**

This movement removes the customer that will save the most when removed, considering the total routing, inventory and transshipment cost. It is repeated ρ times.

4.2.4 **Insert best ρ**

This movement is analogous to the previous one. It is repeated ρ times by computing the cheapest insertion with respect to total costs.

4.2.5 **Shaw removal**

Following the ideas developed by Ropke and Pisinger [27] and by Shaw [28] this movement removes customers that are relatively close to each other. Specifically, this heuristic randomly selects one period from the planning horizon and one customer served in this period, computes the distance $dist_{min}$ to the closest customer also being served by the same route, and removes all customers within $2dist_{min}$ units from the selected route.

4.2.6 **Shaw insertions**

This movement is similar to the Shaw removal in the sense that it selects similar customers to be inserted together. It selects one period and one customer not served in that period. The heuristic then computes $dist_{min}$ and all customers within a $2dist_{min}$ distance are inserted in the same period, always following the cheapest insertion rule.

4.2.7 **Remove ρ customers**

This movement is repeated ρ times by selecting one customer and removing it from all routes where it appears. The motivation for this movement is to allow these customers to be assigned to different sets of periods.

4.2.8 Insert ρ customers

This heuristic is repeated ρ times by selecting one customer and assigning it to several randomly selected periods if the customer is not yet present in those periods. The motivation is to diversify the search towards unexplored areas of the search space, allowing customers to be served in different periods from the ones currently selected.

4.2.9 Empty one period

This movement randomly selects one period and removes all customers from it. The motivation is to allow different periods to have opened routes.

4.2.10 Swap routes

This movement randomly selects two periods and swaps their routes.

4.2.11 Randomly move ρ

This movement selects one period and one customer being served in this period, removes it and serves it in a different randomly selected period. It is repeated ρ times.

4.2.12 Multiple Interchanges

This movement is used only in IRP-OU and IRPT-OU since we observed that for these variants the remaining movements were not sufficient: due to the OU policy, the insertion of a customer in the incumbent solution has a great impact e.g. on inventory costs and on the load of the vehicle. This heuristic is an adaptation of the ideas developed by Bertazzi et al. [7] and differs from the original algorithm described in Section 1 by three key elements: 1) it does not iterate while there is an improvement, but the number of iterations is equal to $n/2$; 2) it does not restart afresh at each call, but it is initiated from the incumbent solution; 3) it applies some improvement procedures to take transshipments into account as they were not present in the original algorithm: if a customer has a lower inventory cost than that of the supplier, products are transshipped to that customer if the tradeoff is positive, while respecting its inventory holding capacity; transshipments are combined to an early period if this yields savings, and routes from the last periods are replaced by transshipments, since one could save on inventory holding costs if the OU policy did not have to be enforced.

4.3 Solution procedure

We now describe our ALNS pseudocode and the parameters that govern the algorithm. We have tested different combinations for the parameters during a tuning phase. The starting temperature τ_{start} is set to 30,000 and the cooling rate ϕ is 0.9994, which yields roughly 25,000 iterations, our desired number of repetitions. The stopping criterion is satisfied when the temperature reaches

0.01, when 25,000 iterations have been performed, or when 3,600 seconds have elapsed.

In our implementation, the segment length φ was set to 200 iterations and the reaction factor η was set to 0.7, that is, new weights will be composed by 70% of the performance on the last segment and 30% by the last weight value. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$.

Algorithm 1 shows the pseudocode for our ALNS.

4.4 ALNS applied to the IRPT-OU

In this section we briefly describe how our ALNS is implemented for the IRPT-OU. Once the ALNS has fixed the routing variables, the remaining problem is modeled as a network flow problem and solved by means of a specialized minimum cost network flow algorithm, as described in Section 3.

It is easy to see that if all transshipments are set to zero, the problem reduces to IRP-OU which yields an upper bound on the IRPT-OU optimum.

4.5 ALNS applied to the IRPT-ML

This version of the problem is similar to the IRPT-OU except that the arcs connecting customers between successive time periods do not force the flow to respect the OU policy. The network flow algorithm determines the quantities delivered by the vehicle and from all transshipment arcs. Once again, if all transshipments are zero, the problem reduces to an IRP-ML which yields an upper bound on the IRPT-ML optimum.

4.6 ALNS applied to the IRP-OU

We have applied our algorithm to the IRP-OU without any structural change. To avoid passing an infeasible problem to the network flow algorithm (for instance when vehicle capacity would be exceeded or when a stockout would occur at a customer due to it not being served as often as required), we have kept all transshipment arcs from the supplier and from every customer to every other customer, with large artificial costs; this means that feasible solutions can always be reached, but at very high cost if transshipments are used. These costs act as penalties in the objective function when the vehicle capacity is exceeded or when the master level heuristic does not add all customers to the current solution.

The remaining problem is then similar to the IRPT-OU: decisions regarding routings are fixed by the ALNS algorithm and modeled as a network flow problem with one vertex representing the vehicle for each period, and arcs leaving the vehicle vertex and arriving at each selected customer. The vehicle vertex receives an arc from the supplier with up to Q units of flow. The OU policy is modeled by fixing the flow on the arcs connecting customers in successive time periods: once customer i is visited in period t, the arc linking to it in the next period has a flow equal to $U_i - d_i^t$.

Algorithm 1 ALNS

Initialize: set weights of all removal and insertion heuristics equal to 1 and all scores equal to 0.
$s \leftarrow$ *initial solution.*
$s_{best} \leftarrow s$.
$\tau \leftarrow \tau_{start}$.
while $\tau > 0.01$ and $time < 3{,}600$ and $iterations < 25{,}000$ **do**
    $s' \leftarrow s$.
    Select a movement using the roulette-wheel mechanism based on the weights of the current segment.
    Apply the movement to s' and update the number of times it is used.
    Fix routing decisions, solve the remaining problem taking into account inventory holding costs, inventory policy and transshipment cost.
    if $f(s') < f(s)$ **then**
        $s \leftarrow s'$;
        if $f(s) < f(s_{best})$ **then**
            $s_{best} \leftarrow s$;
            update the score for the heuristic used with σ_1;
        else
            update the score for the heuristic used with σ_2;
        end if
    else
        if s' is accepted by the simulated annealing criterion **then**
            $s \leftarrow s'$;
            update the score for the heuristic used with σ_3.
        end if
    end if
    if the end of the segment is reached **then**
        update the weights of all heuristics and reset their scores.
    end if
    Adjust the cooling factor in order to decrease the temperature to 0.01 by the end of the running time if 25,000 iterations will not be met.
    if $time > 1{,}200$ and $iterations < 25{,}000/3$ **then**
        $c \leftarrow (0.01/\tau)^{1/(2 \cdot iterations)}$;
    end if
    if $time > 2{,}700$ and $iterations < 25{,}000/2$ **then**
        $c \leftarrow (0.01/\tau)^{1/(iterations/2)}$;
    end if
    $\tau \leftarrow \phi\tau$;
end while
Every 200 iterations perform an intra-route 2-opt to improve the sequence of customers.
return s_{best};

4.7 ALNS applied to the IRP-ML

Modeling the IRP-ML as a network flow problem is similar to the IRP-OU, except that arcs connecting the customers in successive time periods have a minimum flow equal to 0. The vehicle vertex is fed from the supplier with up to Q units and the minimum-cost network flow algorithm decides on how much to deliver to each of the customers selected from the master level heuristic. Dummy arcs are again inserted to penalize unvisited customers and solutions that would require exceeded vehicle capacity. It is easy to see that the IRP-OU yields an upper bound on the IRP-ML optimum as we just relaxed one constraint of the former problem.

5 Computational results

Our algorithm was coded in C++ using Microsoft Visual Studio 2008. We used the scaling push-relabel algorithm for the minimum-cost flow problem developed by Goldberg [14] to solve the second level problem. All computations were executed on an Intel Core2Duo 2.4GHz and 4 GB RAM laptop PC unless otherwise noted.

To evaluate the performance of our heuristic, we have used the instances of the IRP generated and solved to optimality by Archetti et al. [2]. These were adapted to account for transshipments as described in Section 2.

For the cases with transshipment, there are no previous reported solutions since we are introducing the problem in this paper. We have compared our algorithm against a truncated execution of CPLEX with a time limit of 3,600s. In Tables 1 to 4 we report both the upper bound (UB) and lower bound (LB) obtained by CPLEX and our solutions for the IRPT-OU. Results for the IRPT-ML are reported in Tables 5 to 8.

Our algorithm was able to find better solutions than CPLEX for most of the instances on all sets. Our solutions were on average 8.2% better than those obtained by CPLEX for the IRPT-OU. They were better on all four sets of instances and up to 15.5% better for the set with six time periods and low inventory holding costs. For the IRPT-ML, our solutions were also better on all sets of instances. They were on average 12.4% better than those obtained by CPLEX, and up to 21.6% better for the same set.

Table 1: Results for the IRPT-OU - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	403.42	403.42	0.00	0.05	**403.42**	**0.00**	6.29
abs2n05	435.55	435.55	0.00	0.02	**435.55**	**0.00**	6.44
abs3n05	1477.9	1477.9	0.00	0.02	**1477.9**	**0.00**	6.86
abs4n05	811.6	811.6	0.00	0.02	**811.6**	**0.00**	5.61
abs5n05	598.5	598.5	0.00	0.2	**598.5**	**0.00**	7.64
abs1n10	1555.97	1555.97	0.00	2.68	**1555.97**	**0.00**	25.98
abs2n10	1791.97	1792.14	0.01	4.72	**1792.14**	**0.01**	25.65
abs3n10	1414.45	1414.45	0.00	1.34	1418.35	0.28	24.46
abs4n10	1589	1589	0.00	1.53	1592.08	0.19	43.81
abs5n10	1729.03	1729.03	0.00	1.18	**1729.03**	**0.00**	27.91
abs1n15	1868.08	1868.2	0.01	8.01	1869.92	0.10	93.41
abs2n15	1799.19	1799.32	0.01	20.51	1821.5	1.24	59.81
abs3n15	2083.5	2083.5	0.00	5.72	**2083.5**	**0.00**	67.32
abs4n15	1768.37	1768.54	0.01	55.85	1777.68	0.53	115.58
abs5n15	1735.96	1736.13	0.01	7.82	1770.93	2.01	75.1
abs1n20	2111.2	2111.4	0.01	303.11	2181.68	3.34	226.12
abs2n20	2107.63	2107.85	0.01	270.84	2156.4	2.31	115.58
abs3n20	2368.33	2368.56	0.01	201.48	2446.22	3.29	135.33
abs4n20	2482.39	2482.64	0.01	951.5	2853.81	14.96	264.2
abs5n20	2301.67	2627.72	14.17	1060.99*	**2574.09**	**11.84**	175.22
abs1n25	2419.52	2491.58	2.98	3600	2723.09	12.55	468.76
abs2n25	2196.78	2416.78	10.01	3600	**2416.78**	**10.01**	288.1
abs3n25	2438.52	3055.47	25.30	1174.95*	**2946.49**	**20.83**	287.87
abs4n25	2644.68	2644.94	0.01	237.34	2809.97	6.25	566.61
abs5n25	2406.76	2941.66	22.22	822.73*	**2727.04**	**13.31**	334.21
abs1n30	2791.91	3594.42	28.74	1219.03*	3934.12	40.91	688.76
abs2n30	2731.55	3352.79	22.74	1599.76*	3686.85	34.97	450.11
abs3n30	3200.53	3268	2.11	3600	3492.05	9.11	473.99
abs4n30	2822.03	2822.31	0.01	167.22	3001.25	6.35	985
abs5n30	2518.92	2578.61	2.37	3600	2593.21	2.95	581.1
abs1n35	2674.14	3410.86	27.55	1636.51*	3413.06	27.63	862.51
abs2n35	2750.55	4050.83	47.27	1161.77*	**3426.5**	**24.58**	820.61
abs3n35	3156.77	4912	55.60	1465.55*	**4071.77**	**28.99**	744.67
abs4n35	2526.47	3412.11	35.05	1563.09*	**3041.2**	**20.37**	1070.35
abs5n35	2755.31	5798.41	110.44	870.64*	**3660.77**	**32.86**	977.45
abs1n40	2847.05	4397.81	54.47	1077.4*	**3590.06**	**26.10**	1749.51
abs2n40	2677.75	4426.09	65.29	1303.44*	**4057.71**	**51.53**	1444.81
abs3n40	2858.37	4218.64	47.59	1051.46*	**3952.64**	**38.28**	1280.53
abs4n40	2814.08	3185.24	13.19	3600	**3486.88**	**23.91**	1561.28
abs5n40	2943.5	6427.93	118.38	1769.30*	**3890.74**	**32.18**	1849.93
abs1n45	3131.23	4397.81	40.45	1077.40*	**4194.48**	**33.96**	2062.53
abs2n45	2891.89	4793.16	65.74	1844.69*	**4127.35**	**42.72**	2166.34
abs3n45	3332.12	4442.81	33.33	1489.38*	**4046.94**	**21.45**	2441.85
abs4n45	3124.94	4689.98	50.08	1220.59*	**4187.81**	**34.01**	2357
abs5n45	2972.11	4379.24	47.34	1923.16*	**3837.87**	**29.13**	2723.47
abs1n50	3102.19	8279.76	166.90	1646.65*	**4627.87**	**49.18**	2659.18
abs2n50	3479.7	5857.72	68.34	1348.19*	**4596.44**	**32.09**	2775.06
abs3n50	3349.33	5862.01	75.02	1391.51*	**4814.94**	**43.76**	2635.58
abs4n50	3751	4780.49	27.45	1555.00*	**4267.96**	**13.78**	3154.88
abs5n50	3200.86	7679.78	139.93	2112.26*	**4598.16**	**43.65**	3265.63
averages			28.40			**16.95**	

*: aborted prematurely with out-of-memory status

Table 2: Results for the IRPT-OU - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	1264.68	1264.68	0.00	0.08	**1264.68**	**0.00**	6.74
abs2n05	1215.1	1215.1	0.00	0.03	**1215.1**	**0.00**	6.54
abs3n05	2700.63	2700.63	0.00	0.03	**2700.63**	**0.00**	13
abs4n05	1374.54	1374.54	0.00	0.02	**1374.54**	**0.00**	6.87
abs5n05	1766.95	1766.95	0.00	0.19	**1766.95**	**0.00**	7.16
abs1n10	4346.14	4346.37	0.01	2.78	**4346.37**	**0.01**	25.4
abs2n10	4064.6	4064.91	0.01	4.82	**4064.91**	**0.01**	25.37
abs3n10	3560.61	3560.61	0.00	1.3	**3560.61**	**0.00**	47.2
abs4n10	3741.2	3741.2	0.00	1.56	**3741.2**	**0.00**	27.27
abs5n10	4506.63	4506.63	0.00	1.55	**4506.63**	**0.00**	25.39
abs1n15	5313.39	5313.91	0.01	9.96	5330.77	0.33	60.26
abs2n15	5111.27	5111.63	0.01	20.94	5185.14	1.45	63.63
abs3n15	5893.92	5894.51	0.01	5.31	**5894.51**	**0.01**	113.57
abs4n15	4574.78	4575.23	0.01	40	4718.36	3.14	71.98
abs5n15	4451.86	4452.29	0.01	27	**4452.29**	**0.01**	63.76
abs1n20	6656.3	6656.97	0.01	149.73	6697.4	0.62	123.86
abs2n20	6687.38	6688.05	0.01	501.72	6757.64	1.05	135.16
abs3n20	7116.31	7117.7	0.02	86.45	7193.38	1.08	219.96
abs4n20	6302.75	6303.39	0.01	1125.22	6382.71	1.27	126.62
abs5n20	7329.12	7619.91	3.97	1559.46*	**7605.69**	**3.77**	150.49
abs1n25	7637.86	7857.37	2.87	3600	7989.53	4.60	256.01
abs2n25	8056.15	8144.19	1.09	3600	8217.01	2.00	298.13
abs3n25	8795.79	9226.77	4.90	1040.78*	**9224.65**	**4.88**	615.94
abs4n25	8037.04	8037.85	0.01	654.55	8122.5	1.06	304.34
abs5n25	9585.88	9790.28	2.13	1608.23*	10216.5	6.58	276.73
abs1n30	11363.33	12253.99	7.84	1173.78*	12803.6	12.67	495.66
abs2n30	10242.24	11015.1	7.55	876.63*	**10727.4**	**4.74**	520.6
abs3n30	11607	11987.65	3.28	2048.46*	12196.7	5.08	652.61
abs4n30	9137.24	9138	0.01	90.56	9151.62	0.16	460.87
abs5n30	9312.08	9430.34	1.27	2623.48*	9456.55	1.55	477.87
abs1n35	10897.68	11762.54	7.94	3600	**11692.6**	**7.29**	740.74
abs2n35	9704.67	10525.81	8.46	1043.51*	**10309.4**	**6.23**	955.9
abs3n35	13117.51	14346.29	9.37	1392.86*	**13801.1**	**5.21**	1048.32
abs4n35	9584.75	10342.5	7.91	1052.61*	**10061.1**	**4.97**	1050.93
abs5n35	10173.45	11067.73	8.79	1022.67*	**10971**	**7.84**	854.51
abs1n40	12495.91	14600.99	16.85	1405.78*	**13394.2**	**7.19**	1210.87
abs2n40	10012.48	11795.18	17.80	1294.77*	**11338.7**	**13.25**	1614.09
abs3n40	12533.76	13579.84	8.35	1699.26*	**13332.6**	**6.37**	1361.27
abs4n40	10553.25	12026.21	13.96	1335.46*	**11389.7**	**7.93**	1395.97
abs5n40	12346.84	13585.76	10.03	1403.73*	**13360.6**	**8.21**	2306.02
abs1n45	13405.93	17873.64	33.33	1309.94*	**14139.3**	**5.47**	1729.56
abs2n45	12291.24	17670.52	43.77	1409.93*	**13475.8**	**9.64**	2167.07
abs3n45	14185.29	17545.85	23.69	1118.45*	**14864.1**	**4.79**	2106.52
abs4n45	12640.85	14802.9	17.10	1678.70*	**13479.3**	**6.63**	3047.63
abs5n45	12824.24	14450.5	12.68	1619.63*	**13648.2**	**6.43**	2169.29
abs1n50	13627.44	17599.87	29.15	2459.93*	**15087.5**	**10.71**	3134.47
abs2n50	13945.74	17171.48	23.13	1353.16*	**14649.8**	**5.05**	360
abs3n50	14159.08	16601.35	17.25	1820.37*	**15180.1**	**7.21**	2920.84
abs4n50	15821.14	16625.97	5.09	2651.43*	**16540.5**	**4.55**	3600
abs5n50	14538.68	18631.79	28.15	1763.14*	**16342.4**	**12.41**	3600
averages			7.56			**4.07**	

*: aborted prematurely with out-of-memory status

Table 3: Results for the IRPT-OU - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	2571.02	2571.02	0.00	0.33	**2571.02**	**0.00**	14.3
abs2n05	1904.36	1904.36	0.00	0.91	**1904.36**	**0.00**	13.85
abs3n05	4002.84	4003.19	0.01	0.04	**4003.19**	**0.01**	16.82
abs4n05	2502.57	2502.57	0.00	1.02	**2502.57**	**0.00**	20.07
abs5n05	1828.09	1828.09	0.00	4.49	**1828.09**	**0.00**	16.14
abs1n10	3305.14	4063.6	22.95	1033.79*	4208.6	27.33	58.56
abs2n10	3135.07	4397.55	40.27	865.38*	**4390.63**	**40.05**	80.45
abs3n10	3471.98	3682.98	6.08	3600	**3682.98**	**6.08**	62.77
abs4n10	3608.24	4024.1	11.53	3600	4074.75	12.93	98.93
abs5n10	4103.32	4103.74	0.01	2008.77	4120.82	0.43	50.78
abs1n15	3996.79	4675.04	16.97	1118.91*	**4633.86**	**15.94**	170.77
abs2n15	3686.22	5275.14	43.10	1009.01*	**4856.49**	**31.75**	184.82
abs3n15	4625.68	5755.8	24.43	1179.77*	**5706.32**	**23.36**	241.33
abs4n15	3264.35	4743.2	45.30	813.84*	**4599.73**	**40.91**	313.17
abs5n15	3830.77	4762.87	24.33	1026.82*	**4612.57**	**20.41**	130.34
abs1n20	3998.34	8266.39	106.75	1321.88*	**5368.02**	**34.26**	358.15
abs2n20	4155.49	5468.97	31.61	1921.86*	5796.29	39.49	472.73
abs3n20	4345.56	6477.09	49.05	1395.75*	**6073.73**	**39.77**	414.18
abs4n20	4816.81	7808.9	62.12	1108.85*	**6726.7**	**39.65**	766.94
abs5n20	4668.53	9072.67	94.34	1165.48*	**7416.88**	**58.87**	444.56
abs1n25	3837.32	9360.67	143.94	1697.63*	**6113.58**	**59.32**	715.73
abs2n25	4413.36	10412.96	135.94	2373.47*	**6289.73**	**42.52**	585.17
abs3n25	5042.47	11793.11	133.88	1520.64*	**7311.17**	**44.99**	999.88
abs4n25	5205.72	9374.41	80.08	1988.31*	**7071.64**	**35.84**	742.2
abs5n25	4707.82	11891.26	152.59	1591.23*	**7246.86**	**53.93**	982.87
abs1n30	5715.18	14707.94	157.35	1610.56*	**9678.23**	**69.34**	2023.78
abs2n30	5520.81	12912.75	133.89	1535.45*	**7455.09**	**35.04**	1812.99
abs3n30	6160.09	13548.36	119.94	1328.50*	**8298.56**	**34.71**	1710.66
abs4n30	5480.05	16317.82	197.77	2111.75*	**7237.08**	**32.06**	1641.95
abs5n30	4938.71	12102.23	145.05	1366.21*	**7236.77**	**46.53**	1062.49
averages			65.98			**29.52**	

*: aborted prematurely with out-of-memory status

Table 4: Results for the IRPT-OU - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	5112.33	5112.33	0.00	0.26	**5112.33**	**0.00**	14.47
abs2n05	4222.7	4222.7	0.00	0.94	**4222.7**	**0.00**	14.91
abs3n05	6167.86	6168.33	0.01	5.67	6175.37	0.12	15.68
abs4n05	4358.37	4358.75	0.01	0.8	**4358.75**	**0.01**	13.65
abs5n05	3935.2	3935.59	0.01	3.39	**3935.59**	**0.01**	16.21
abs1n10	7605.42	8486.31	11.58	879.73*	**8329.06**	**9.51**	60.98
abs2n10	6473.61	7821.87	20.83	739.13*	**7684.48**	**18.70**	62.06
abs3n10	7290.3	7440.54	2.06	3600	**7440.54**	**2.06**	52.53
abs4n10	7159.94	7690.1	7.40	1094.36*	7700.16	7.55	100.36
abs5n10	8946	8946.9	0.01	1074.38	9038.45	1.03	50.52
abs1n15	10583.7	11280.3	6.58	1124.36*	11323	6.99	138.2
abs2n15	10167.89	11619.41	14.28	1141.1*	**11196.1**	**10.11**	156.33
abs3n15	12175.34	13238.85	8.73	1127.72*	**13035.6**	**7.07**	166.47
abs4n15	8454.44	9935.18	17.51	891.32*	**9811.05**	**16.05**	165.66
abs5n15	8805.4	10052.29	14.16	1261.02*	**9772.85**	**10.99**	153.71
abs1n20	12150.24	18092.86	48.91	758.22*	**14070**	**15.80**	418.43
abs2n20	12603.9	14430.81	14.49	1416.83*	**14072.2**	**11.65**	333.85
abs3n20	11782.27	14301.45	21.38	1277.86*	**13168.5**	**11.77**	405.26
abs4n20	11943.74	18294.14	53.17	1369.90*	**14001.2**	**17.23**	447.15
abs5n20	13180.97	18772.02	42.42	1213.92*	**16079.7**	**21.99**	607.47
abs1n25	12334.86	19446.29	57.65	1607.11*	**14871.4**	**20.56**	795.34
abs2n25	13510.01	18382.25	36.06	1007.95*	**16404.5**	**21.42**	853.97
abs3n25	15339.1	24412.8	59.15	2105.59*	**17862.7**	**16.45**	1142.16
abs4n25	13964.71	17617.52	26.16	1532.37*	**15871.9**	**13.66**	669.59
abs5n25	16166.2	24177.3	49.55	1448.93*	**19328**	**19.56**	1071.32
abs1n30	20172.52	31638.5	56.84	2464.61*	**22584.8**	**11.96**	1818.34
abs2n30	17721.38	23734.23	33.93	2853.60*	**20882**	**17.84**	1878.78
abs3n30	21003.7	28328.33	34.87	1339.57*	**23764.9**	**13.15**	1859.32
abs4n30	15433.16	23211.63	50.40	1657.82*	**17087.4**	**10.72**	1459.47
abs5n30	16519.02	25144.04	52.21	1442.89*	**18141.8**	**9.82**	1574.53
averages			24.68			**10.79**	

UB and LB are the upper bound and lower bound obtained after running CPLEX 12.2 with 1 hour-limit
time: run on an Intel Core2Duo 2.4GHz and 4 GB RAM
*: aborted prematurely with out-of-memory status

Table 5: Results for the IRPT-ML - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	403.42	403.42	0.00	0.39	**403.42**	**0.00**	5.18
abs2n05	435.55	435.55	0.00	0.24	**435.55**	**0.00**	5.47
abs3n05	1460.4	1460.4	0.00	0.45	1476.04	1.07	5.75
abs4n05	811.6	811.6	0.00	0.07	**811.6**	**0.00**	4.92
abs5n05	597.83	597.83	0.00	0.3	**597.83**	**0.00**	5.11
abs1n10	1547.22	1547.29	0.00	2.64	**1547.29**	**0.00**	15.37
abs2n10	1702.92	1703.08	0.01	5.8	**1703.08**	**0.01**	15.14
abs3n10	1414.45	1414.45	0.00	1.71	**1414.45**	**0.00**	15.38
abs4n10	1586.38	1586.52	0.01	1.6	**1586.52**	**0.01**	13.41
abs5n10	1635.2	1635.2	0.00	1.47	1683.23	2.94	13.34
abs1n15	1830.46	1830.64	0.01	6.72	1831	0.03	31.82
abs2n15	1783.76	1783.94	0.01	124.32	**1783.94**	**0.01**	28.51
abs3n15	2083.3	2083.5	0.01	5.57	**2083.57**	**0.01**	32.81
abs4n15	1768.36	1768.54	0.01	131.27	1816.96	2.75	31.1
abs5n15	1733.54	1733.71	0.01	19.43	**1733.71**	**0.01**	28.99
abs1n20	1998.89	1999.05	0.01	60.48	**1999.05**	**0.01**	52.68
abs2n20	2102.02	2102.23	0.01	445.75	2158.62	2.69	53.27
abs3n20	2321.78	2322.01	0.01	174.83	2328.19	0.28	53.4
abs4n20	2413.96	2441.35	1.13	3600	**2441.35**	**1.13**	71.88
abs5n20	2325.29	2525.57	8.61	1438.54*	**2525.57**	**8.61**	52.51
abs1n25	2276.53	2482.82	9.06	3600	**2482.82**	**9.06**	77.15
abs2n25	2178.82	2393.38	9.85	3600	2395.29	9.94	94.02
abs3n25	2390.14	3015.87	26.18	978.09*	**2770.31**	**15.91**	115.42
abs4n25	2599.92	2600.17	0.01	634.29	**2600.17**	**0.01**	73.71
abs5n25	2392.37	2794.69	16.82	789.82*	**2647.09**	**10.65**	99.92
abs1n30	2726.77	3692.2	35.41	1783.72*	**3238.33**	**18.76**	137.53
abs2n30	2694.36	3232.74	19.98	3600	**3021.25**	**12.13**	182.23
abs3n30	2961.52	3321.53	12.16	920.35*	3336.58	12.66	165.83
abs4n30	2765.25	2765.53	0.01	127.42	**2765.53**	**0.01**	154.69
abs5n30	2529.73	2568.26	1.52	3600	**2568.26**	**1.52**	185.76
abs1n35	2622.95	3391.19	29.29	1362.56*	**3304.26**	**25.97**	260.65
abs2n35	2779.1	3911.11	40.73	1021.51*	**3251.31**	**16.99**	226.55
abs3n35	3112.63	5531.21	77.70	1016.77*	**3871.63**	**24.38**	255.49
abs4n35	2491.44	3369.47	35.24	1524.17*	**2973.9**	**19.36**	236.09
abs5n35	2705.81	3970.49	46.74	1075.36*	**3839.75**	**41.91**	264.92
abs1n40	2846.23	7375.75	159.14	2003.16*	**3382.91**	**18.86**	327.63
abs2n40	2578.85	3761.29	45.85	1810.21*	**3323.06**	**28.86**	308.65
abs3n40	2905.18	3769.19	29.74	1181.98*	**3447.66**	**18.67**	331.87
abs4n40	2823.84	3358.77	18.94	2761.48*	**3279.74**	**16.14**	291.22
abs5n40	2845.16	4135.58	45.35	2205.33*	**3375.67**	**18.65**	485.32
abs1n45	3087.93	3878.68	25.61	1788.61*	**3484.22**	**12.83**	614.93
abs2n45	2862.79	5267.67	84.00	1994.33*	**4150.2**	**44.97**	470.52
abs3n45	3247.43	5139.83	58.27	1532.85*	**3846.21**	**18.44**	421.25
abs4n45	3123.63	5335.75	70.82	1317.15*	**3541.52**	**13.38**	434.88
abs5n45	2850.69	5933.04	108.13	1436.41*	**3465.94**	**21.58**	366.99
abs1n50	3119.2	11500.45	268.70	1812.45*	**3835.17**	**22.95**	872.09
abs2n50	3481.99	7069.69	103.04	2579.94*	**3982.64**	**14.38**	758.15
abs3n50	3423.73	11159.86	225.96	2728.25*	**4552.21**	**32.96**	903.15
abs4n50	3770.34	4263.94	13.09	3244.02*	**3994.57**	**5.95**	776.54
abs5n50	3187.08	11869.67	272.43	1142.89*	**3990.86**	**25.22**	491.6
averages			37.99			**11.05**	

*: aborted prematurely with out-of-memory status

Table 6: Results for the IRPT-ML - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 3$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	1264.68	1264.68	0.00	0.63	**1264.68**	**0.00**	6.12
abs2n05	1215.1	1215.1	0.00	0.83	**1215.1**	**0.00**	6.4
abs3n05	2683.35	2683.35	0.00	0.81	**2683.35**	**0.00**	5.37
abs4n05	1374.54	1374.54	0.00	0.64	**1374.54**	**0.00**	6.16
abs5n05	1763.67	1763.67	0.00	0.7	**1763.67**	**0.00**	5.9
abs1n10	4316.19	4316.61	0.01	2.24	4318.61	0.06	15.6
abs2n10	3972.6	3973	0.01	8.44	**3973**	**0.01**	15.24
abs3n10	3560.31	3560.61	0.01	2.17	**3560.61**	**0.01**	15.74
abs4n10	3739.86	3739.86	0.00	1.59	3742.02	0.06	16.7
abs5n10	4405.05	4405.05	0.00	1.45	4464.83	1.36	13.54
abs1n15	5278.99	5279.5	0.01	19.4	**5279.5**	**0.01**	33.24
abs2n15	5076.29	5076.77	0.01	42.83	**5076.77**	**0.01**	37.13
abs3n15	5894.51	5894.51	0.00	4.03	**5894.51**	**0.00**	32.74
abs4n15	4574.77	4575.23	0.01	84.52	4612.36	0.82	33.73
abs5n15	4446.06	4446.9	0.02	28.52	**4446.49**	**0.01**	32
abs1n20	6579.92	6580.57	0.01	72.09	**6580.57**	**0.01**	63.63
abs2n20	6659.47	6660.14	0.01	513.9	6805.57	2.19	55.93
abs3n20	7055.51	7056.22	0.01	152.97	**7056.22**	**0.01**	55.09
abs4n20	6185.67	6268.09	1.33	3600	6374.01	3.04	56.39
abs5n20	7283.53	7549.76	3.66	1332.16*	7529.95	3.38	57.66
abs1n25	7704.19	7842.1	1.79	3600	**7842.1**	**1.79**	79.05
abs2n25	7861.84	8335.32	6.02	1293.59*	**8125.86**	**3.36**	92.21
abs3n25	8670.59	9490.04	9.45	806.46*	**9084.56**	**4.77**	112.42
abs4n25	7985	7985.8	0.01	926.11	8001.15	0.20	79.81
abs5n25	9488.54	9917.79	4.52	854.32*	**9759.31**	**2.85**	92.84
abs1n30	11303.29	12277.66	8.62	954.42*	**11780.3**	**4.22**	159.95
abs2n30	10249.68	10963.87	6.97	951.63*	**10591.8**	**3.34**	172
abs3n30	11641.49	11870.68	1.97	2305.24*	11947.4	2.63	147.18
abs4n30	9083.77	9084.67	0.01	121.67	**9084.67**	**0.01**	146.11
abs5n30	9319.6	9383.49	0.69	3600	**9384.01**	**0.69**	170.45
abs1n35	10964.21	11771.05	7.36	1132.09*	**11455.3**	**4.48**	266.65
abs2n35	9669.46	10930.39	13.04	989.09*	**10173.3**	**5.21**	216.16
abs3n35	13078.73	13671.42	4.53	1056.92*	**13543.7**	**3.56**	332.01
abs4n35	9572.4	10099.24	5.50	1622.19*	**9979.83**	**4.26**	261.15
abs5n35	10162.19	11083.17	9.06	1156.73*	11395.2	12.13	336.39
abs1n40	12448.93	16084.59	29.20	1480.34*	**13029.9**	**4.67**	544.99
abs2n40	10034.03	11810.05	17.70	1720.45*	**10680.3**	**6.44**	354.6
abs3n40	12494.33	14353.74	14.88	863.89*	**13093.3**	**4.79**	415.84
abs4n40	10563.5	11368.9	7.62	3092.38*	**10995.4**	**4.09**	235.81
abs5n40	12325.85	18028.67	46.27	1060.97*	**13027.5**	**5.69**	252.96
abs1n45	13323.86	19759.24	48.30	1430.00*	**13604.9**	**2.11**	587.47
abs2n45	12217.37	14759.2	20.81	1543.79*	**13721.6**	**12.31**	563.6
abs3n45	14068.86	16369.23	16.35	1356.51*	**14667.1**	**4.25**	726.24
abs4n45	12608	14481.81	14.86	2785.15*	**13096.4**	**3.87**	823.41
abs5n45	12677.13	13742.65	8.41	3600	**13409.8**	**5.78**	439.09
abs1n50	13609.54	15423.99	13.33	3600	**14159.8**	**4.04**	941.37
abs2n50	13946.48	18389.65	31.86	1862.25*	**14374.2**	**3.07**	1187.11
abs3n50	14163.34	21597.18	52.49	1933.87*	**15003.9**	**5.93**	824.15
abs4n50	15791.82	16867.96	6.81	2279.21*	**16133.3**	**2.16**	862.84
abs5n50	14522.01	18444.44	27.01	2631.84*	**15349.7**	**5.70**	883.91
averages			8.81			**2.79**	

*: aborted prematurely with out-of-memory status

Table 7: Results for the IRPT-ML - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	2571.02	2571.02	0.00	0.75	2571.67	0.03	10.62
abs2n05	1901.87	1901.87	0.00	1.32	**1901.87**	**0.00**	11
abs3n05	3971.23	3971.5	0.01	5.2	**3971.5**	**0.01**	11.68
abs4n05	2502.57	2502.57	0.00	1.09	2504.68	0.08	9.16
abs5n05	1825.16	1825.31	0.01	4.77	1842.15	0.93	12.07
abs1n10	3270.79	4082.43	24.81	940.22*	4244.76	29.78	40.76
abs2n10	3059.62	4409.67	44.12	821.12*	**4245.11**	**38.75**	36.22
abs3n10	3383.4	3680.53	8.78	1861.55*	3682.09	8.83	33.48
abs4n10	3460.3	4038.63	16.71	1079.38*	4186.94	21.00	41.32
abs5n10	4045.11	4070.69	0.63	3600	4116.62	1.77	31.74
abs1n15	3815.5	4614.1	20.93	1025.05*	**4607.52**	**20.76**	83.09
abs2n15	2688.14	5016.88	86.63	1036.87*	**4875.59**	**81.37**	80.7
abs3n15	4586.36	5689.34	24.05	1183.73*	**5611.42**	**22.35**	76.69
abs4n15	3238.72	4967.92	53.39	822.05*	**4519.4**	**39.54**	77.72
abs5n15	3770.23	5285.75	40.20	815.59*	**4559.74**	**20.94**	85.21
abs1n20	3946.59	11228.88	184.52	1483.64*	**5218.72**	**32.23**	146.63
abs2n20	4141.17	6609.2	59.60	1091.49*	**5423.01**	**30.95**	165.34
abs3n20	4264.85	6229.84	46.07	1227.35*	**5773.66**	**35.38**	151.43
abs4n20	4906.78	11286.61	130.02	1895.21*	**6563.02**	**33.75**	216.22
abs5n20	4668.69	14046.67	200.87	1137.69*	**7125.74**	**52.63**	191.59
abs1n25	3844	8154.89	112.15	2162.00*	**6629.91**	**72.47**	267.8
abs2n25	4414.1	10757.72	143.71	1060.15*	**6124.79**	**38.76**	164.73
abs3n25	5149.34	16703.91	224.39	1853.31*	**7425.18**	**44.20**	239.05
abs4n25	5295.19	10739.35	102.81	1518.14*	**6873.24**	**29.80**	295.87
abs5n25	4695.94	22577.03	380.78	2253.39*	**6988.88**	**48.83**	509.09
abs1n30	5629.03	14991.49	166.32	1727.53*	**7579.22**	**34.65**	886.39
abs2n30	5535.06	16188.11	192.46	1541.90*	**7412.54**	**33.92**	700.83
abs3n30	6109.78	14414.26	135.92	2956.53*	**8493.64**	**39.02**	642.68
abs4n30	5448.08	14416.35	164.61	2987.49*	**7047.21**	**29.35**	492.52
abs5n30	5016.7	11473.18	128.70	2468.17*	**6801.87**	**35.58**	633.81
averages			89.77			**29.26**	

*: aborted prematurely with out-of-memory status

Table 8: Results for the IRPT-ML - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 6$ and transshipment cost $b_{ij} = 0.01c_{ij}$

	Truncated CPLEX				ALNS		
Instance	LB	UB	gap (%)	time (s)	z	gap (%)	time (s)
abs1n05	5112.33	5112.33	0.00	0.23	**5112.33**	**0.00**	12.75
abs2n05	4203.67	4203.84	0.00	0.88	**4203.84**	**0.00**	14.29
abs3n05	6104.2	6104.74	0.01	5.22	6117.22	0.21	12.28
abs4n05	4358.75	4358.75	0.00	0.47	4376.88	0.42	12.67
abs5n05	3930.9	3931.29	0.01	4.38	**3931.29**	**0.01**	14.46
abs1n10	7578.09	8380.19	10.58	1015.40*	**8231.27**	**8.62**	36.9
abs2n10	6280.14	7704.91	22.69	738.13*	**7520.2**	**19.75**	43.98
abs3n10	7235.8	7440.22	2.83	3600	**7440.5**	**2.83**	34.8
abs4n10	7209.67	7667.88	6.36	1213.97*	7683.7	6.57	40.93
abs5n10	8846.06	8883.59	0.42	3600	8932.97	0.98	32.97
abs1n15	10513.84	11243.64	6.94	941.82*	11366.9	8.11	90.85
abs2n15	10070.67	11641.73	15.60	620.95*	**11242**	**11.63**	89.78
abs3n15	12052.59	13122.48	8.88	985.66*	**13000.6**	**7.87**	75.87
abs4n15	8297.38	10130.56	22.09	733.68*	**9602.1**	**15.72**	97.41
abs5n15	8733.12	10176.12	16.52	783.17*	**9534.14**	**9.17**	88.36
abs1n20	12120.01	15939.59	31.51	863.40*	**13396.6**	**10.53**	204.31
abs2n20	12619.6	14801.27	17.29	795.02*	**14102.2**	**11.75**	189.54
abs3n20	11766.2	13379.64	13.71	1342.99*	**13310.7**	**13.13**	185.33
abs4n20	11901.02	18849.86	58.39	1285.33*	**13909.3**	**16.87**	171.59
abs5n20	13200.17	24218.41	83.47	1544.11*	**16041.4**	**21.52**	147.76
abs1n25	12341.86	17014.54	37.86	1664.80*	**14821.9**	**20.09**	332.33
abs2n25	13729.97	22863.21	66.52	2300.97*	**15504.3**	**12.92**	291.12
abs3n25	15353.9	27616.76	79.87	1594.66*	**17820.8**	**16.07**	392.52
abs4n25	14024.51	17582.21	25.37	1967.37*	**15448.9**	**10.16**	350.66
abs5n25	16219.27	24736.05	52.51	2157.40*	**18005**	**11.01**	280.92
abs1n30	20173.28	33850.17	67.80	2392.14*	**23164.5**	**14.83**	725.36
abs2n30	17670.31	27567.88	56.01	2762.35*	**20103.2**	**13.77**	913.78
abs3n30	20964.18	31589.5	50.68	1445.17*	**23109.2**	**10.23**	816.25
abs4n30	15566.26	25890.35	66.32	1980.70*	**16933**	**8.78**	596.67
abs5n30	16538.57	27179.34	64.34	1743.81*	**17867.6**	**8.04**	885.33
averages			29.49			**9.72**	

*: aborted prematurely with out-of-memory status

We have also applied our algorithm to the traditional IRP (without transshipment) by setting the transshipment cost sufficiently large ($b_{ij} = 1.00c_{ij}$) so as to avoid their appearance in the final solution. In Tables 9−12, our results are compared to those of Archetti et al. [3] and of Bertazzi et al. [7] on these IRP-OU instances. Our results are significantly better than those of Bertazzi et al. [7] but slightly worse than those of Archetti et al. [3]. We have also tested our algorithm with the IRP-ML and the gaps to the optimal solutions were slightly larger than those observed in the IRP-OU, as observed in tables 13−16.

Table 9: Results for the IRP-OU - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 3$

	ABLS[2]	BPS[7]		ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	1281.68	**1281.68**	**0.00%**	**1281.68**	**0.00%**	3	**1281.68**	**0.00%**	8.88
abs2n05	1176.63	**1176.63**	**0.00%**	**1176.63**	**0.00%**	3	**1176.63**	**0.00%**	10.06
abs3n05	2020.65	2178.94	7.83%	**2020.65**	**0.00%**	4	**2020.65**	**0.00%**	10.26
abs4n05	1449.43	**1449.43**	**0.00%**	**1449.43**	**0.00%**	2	**1449.43**	**0.00%**	9.36
abs5n05	1165.4	1242.07	6.58%	**1165.4**	**0.00%**	3	**1165.4**	**0.00%**	15.02
abs1n10	2167.37	2198.56	1.44%	**2167.37**	**0.00%**	14	**2167.37**	**0.00%**	34.9
abs2n10	2510.13	**2510.13**	**0.00%**	**2510.13**	**0.00%**	12	**2510.13**	**0.00%**	38.49
abs3n10	2099.68	**2099.68**	**0.00%**	**2099.68**	**0.00%**	13	**2099.68**	**0.00%**	32.53
abs4n10	2188.01	**2188.01**	**0.00%**	**2188.01**	**0.00%**	11	**2188.01**	**0.00%**	35.04
abs5n10	2178.15	2231.67	2.46%	**2178.45**	**0.01%**	14	**2178.15**	**0.00%**	35.49
abs1n15	2236.53	2271.68	1.57%	**2236.53**	**0.00%**	40	**2236.53**	**0.00%**	99.39
abs2n15	2506.21	**2506.21**	**0.00%**	**2506.21**	**0.00%**	41	**2506.21**	**0.00%**	85.49
abs3n15	2841.06	**2841.06**	**0.00%**	**2841.06**	**0.00%**	44	**2841.06**	**0.00%**	117.38
abs4n15	2430.07	2632.18	8.32%	**2430.07**	**0.00%**	43	**2430.07**	**0.00%**	100.72
abs5n15	2453.5	2524.95	2.91%	**2453.5**	**0.00%**	39	**2453.5**	**0.00%**	93.63
abs1n20	2793.29	**2793.29**	**0.00%**	**2793.29**	**0.00%**	75	2796.5	0.11%	223
abs2n20	2799.9	2873.51	2.63%	**2799.9**	**0.00%**	105	2809.51	0.34%	240.37
abs3n20	3101.6	3163.94	2.01%	3104.29	0.09%	101	**3101.6**	**0.00%**	259.48
abs4n20	3239.31	**3239.31**	**0.00%**	**3239.31**	**0.00%**	87	**3239.31**	**0.00%**	227.98
abs5n20	3330.99	3814.54	14.52%	**3330.99**	**0.00%**	153	**3330.99**	**0.00%**	247.98
abs1n25	3309.64	3624.49	9.51%	**3309.64**	**0.00%**	341	3311.23	0.05%	563.22
abs2n25	3495.97	3544.55	1.39%	**3495.97**	**0.00%**	214	**3495.97**	**0.00%**	800.72
abs3n25	3481.45	3622.7	4.06%	**3481.45**	**0.00%**	278	3483.45	0.06%	581.56
abs4n25	3272.74	**3272.74**	**0.00%**	**3272.74**	**0.00%**	295	**3272.74**	**0.00%**	450.59
abs5n25	3695.94	**3695.94**	**0.00%**	**3695.94**	**0.00%**	166	**3695.94**	**0.00%**	465.32
abs1n30	3918.76	4022.3	2.64%	**3918.76**	**0.00%**	326	**3918.76**	**0.00%**	884.87
abs2n30	3737.11	3874.22	3.67%	**3737.11**	**0.00%**	497	3743.68	0.18%	1242.37
abs3n30	3761.85	3931.85	4.52%	**3761.85**	**0.00%**	607	**3761.85**	**0.00%**	1159.04
abs4n30	3532.47	3676.9	4.09%	**3532.47**	**0.00%**	452	3534.45	0.06%	807.94
abs5n30	3265.89	3365.76	3.06%	**3269.76**	**0.12%**	693	**3269.76**	**0.12%**	1268.15
abs1n35	3694.48	3737.48	1.16%	**3694.48**	**0.00%**	908	**3694.48**	**0.00%**	1273.92
abs2n35	3796.8	3960.39	4.31%	**3796.8**	**0.00%**	704	3803.58	0.18%	1414.64
abs3n35	4351.09	4665.4	7.22%	4359.08	0.18%	532	**4355.72**	**0.11%**	1736.72
abs4n35	3766.39	3939.11	4.59%	**3766.39**	**0.00%**	1225	3774.84	0.22%	1297.2
abs5n35	3625.57	3807.86	5.03%	**3625.57**	**0.00%**	675	**3625.57**	**0.00%**	1473.93
abs1n40	4263.43	4732.75	11.01%	4274.71	0.26%	903	**4271.11**	**0.18%**	2277.27
abs2n40	4166.95	4415.23	5.96%	**4166.95**	**0.00%**	1539	4186.96	0.48%	2963.37
abs3n40	4337.3	4591.92	5.87%	4340.06	0.06%	1310	4352.5	0.35%	3005.97
abs4n40	3846.84	4000.81	4.00%	**3846.84**	**0.00%**	1441	3856.47	0.25%	2856.11
abs5n40	4013.98	4234.01	5.48%	**4013.98**	**0.00%**	650	4033.79	0.49%	2675.91
abs1n45	4369.38	4568.68	4.56%	**4369.38**	**0.00%**	1493	4372.54	0.07%	3600
abs2n45	4226.82	4655.8	10.15%	**4226.82**	**0.00%**	1224	4239.69	0.30%	3600
abs3n45	4317.08	4572.82	5.92%	**4317.08**	**0.00%**	1904	4346.74	0.69%	3600
abs4n45	4527.95	4962.14	9.59%	4559.36	0.69%	1292	**4537.86**	**0.22%**	3600
abs5n45	3911.82	4215.11	7.75%	**3911.82**	**0.00%**	1387	3919.82	0.20%	2664.01
abs1n50	4629.92	4884.83	5.51%	**4670.41**	**0.87%**	1782	4687.08	1.23%	2540.79
abs2n50	4919.75	5123.98	4.15%	**4919.75**	**0.00%**	2004	4934.71	0.30%	2467.76
abs3n50	4868.36	5269.43	8.24%	**4868.36**	**0.00%**	2648	4888.43	0.41%	2096.98
abs4n50	4972.25	5273.98	6.07%	**5014.17**	**0.84%**	1843	**5013.94**	**0.84%**	2671.83
abs5n50	4664.05	4901.19	5.08%	4687.16	0.50%	3126	**4682.18**	**0.39%**	3600
averages			4.10%		0.07%			0.16%	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 10: Results for the IRP-OU - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 3$

	ABLS[2]	BPS[7]		ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	2149.8	**2149.8**	**0.00%**	**2149.8**	**0.00%**	4	**2149.8**	**0.00%**	8.56
abs2n05	1959.05	**1959.05**	**0.00%**	**1959.05**	**0.00%**	4	**1959.05**	**0.00%**	8.96
abs3n05	3265.44	3408.48	4.38%	**3265.44**	**0.00%**	7	**3265.44**	**0.00%**	10.33
abs4n05	2034.44	**2034.44**	**0.00%**	**2034.44**	**0.00%**	2	**2034.44**	**0.00%**	9.13
abs5n05	2362.16	2413.72	2.18%	**2362.16**	**0.00%**	6	**2362.16**	**0.00%**	10.31
abs1n10	4970.62	5015.82	0.91%	**4970.62**	**0.00%**	15	**4970.62**	**0.00%**	31.34
abs2n10	4803.17	5142.58	7.07%	**4803.17**	**0.00%**	13	**4803.17**	**0.00%**	35.27
abs3n10	4289.84	**4289.84**	**0.00%**	**4289.84**	**0.00%**	13	**4289.84**	**0.00%**	35.16
abs4n10	4347.06	**4347.06**	**0.00%**	**4347.06**	**0.00%**	11	**4347.06**	**0.00%**	35.68
abs5n10	5041.62	5078.05	0.72%	**5044.44**	**0.06%**	14	**5044.44**	**0.06%**	35.66
abs1n15	5713.84	**5713.84**	**0.00%**	**5713.84**	**0.00%**	37	**5713.84**	**0.00%**	84.08
abs2n15	5821.04	5822.88	0.03%	5822.88	0.03%	41	**5821.04**	**0.00%**	99.24
abs3n15	6711.25	6894.43	2.73%	**6711.25**	**0.00%**	45	**6711.25**	**0.00%**	104.64
abs4n15	5227.56	5437.22	4.01%	**5227.56**	**0.00%**	49	5237.82	0.20%	98.78
abs5n15	5210.85	5424.97	4.11%	**5215.02**	**0.08%**	61	5219.39	0.16%	98.72
abs1n20	7353.82	7449.47	1.30%	**7353.82**	**0.00%**	108	**7353.82**	**0.00%**	168.71
abs2n20	7385.03	7450.67	0.89%	**7385.03**	**0.00%**	101	7409.21	0.33%	245.06
abs3n20	7903.97	7974.47	0.89%	7907.4	0.04%	96	**7903.97**	**0.00%**	199.62
abs4n20	7050.91	7601.67	7.81%	**7050.91**	**0.00%**	143	**7050.91**	**0.00%**	244.56
abs5n20	8405.83	8876.11	5.59%	**8405.83**	**0.00%**	73	8416.83	0.13%	263.29
abs1n25	8657.7	8985.82	3.79%	**8657.7**	**0.00%**	205	**8657.7**	**0.00%**	335.05
abs2n25	9266.87	9408.18	1.52%	**9266.87**	**0.00%**	164	**9266.87**	**0.00%**	502.56
abs3n25	9843.6	**9843.6**	**0.00%**	**9843.6**	**0.00%**	208	9845.6	0.02%	415.8
abs4n25	8677.86	**8677.86**	**0.00%**	**8677.86**	**0.00%**	342	8747.67	0.80%	442.01
abs5n25	10857.68	**10857.68**	**0.00%**	**10857.68**	**0.00%**	191	10862.4	0.04%	536.94
abs1n30	12635.55	12847.79	1.68%	**12635.55**	**0.00%**	345	12661.8	0.21%	1048.34
abs2n30	11351.36	11487.63	1.20%	**11351.36**	**0.00%**	273	11363.4	0.11%	953.91
abs3n30	12509.26	12679.26	1.36%	**12613.46**	**0.83%**	621	**12613.5**	**0.83%**	875.09
abs4n30	9928.35	10018.88	0.91%	**9928.35**	**0.00%**	573	9945.73	0.18%	637.57
abs5n30	10178.63	10270.81	0.91%	**10181.69**	**0.03%**	346	10188.9	0.10%	935.91
abs1n35	11984.69	12382.03	3.32%	**11984.69**	**0.00%**	600	**11984.7**	**0.00%**	1376.75
abs2n35	10706.91	10998.26	2.72%	**10706.91**	**0.00%**	1160	10773.1	0.62%	1403.08
abs3n35	14411.58	14732.69	2.23%	14478.78	0.47%	581	**14413.5**	**0.01%**	1567.35
abs4n35	10844.98	11001.99	1.45%	**10844.98**	**0.00%**	1202	10863.7	0.17%	2059.08
abs5n35	11195.87	11365.71	1.52%	**11195.87**	**0.00%**	624	11209.5	0.12%	1596.57
abs1n40	14006.57	14455.9	3.21%	**14006.57**	**0.00%**	931	14006.57	0.00%	2057.93
abs2n40	11722.58	11941.41	1.87%	**11722.58**	**0.00%**	996	11775.6	0.45%	3204.6
abs3n40	14107.14	14655.81	3.89%	**14107.14**	**0.00%**	2035	14109.2	0.01%	2654.68
abs4n40	11684.43	11763.03	0.67%	**11684.43**	**0.00%**	1276	11725.3	0.35%	2918.37
abs5n40	13536.57	13759.25	1.65%	**13536.57**	**0.00%**	1230	13603.1	0.49%	3003.25
abs1n45	14661.2	14867.01	1.40%	**14661.2**	**0.00%**	2099	14727.9	0.45%	3281.36
abs2n45	13675.96	14161.64	3.55%	**13675.96**	**0.00%**	1235	13700.8	0.18%	3600
abs3n45	15316.57	15564.91	1.62%	**15316.57**	**0.00%**	1416	15390	0.48%	2527.2
abs4n45	14096.84	14537.57	3.13%	14121.05	0.17%	1175	**14117.3**	**0.15%**	2474.87
abs5n45	13838.54	14215.84	2.73%	**13840.06**	**0.01%**	1747	14024.2	1.34%	3150.81
abs1n50	15235.83	15543.49	2.02%	**15235.83**	**0.00%**	3989	15310.9	0.49%	3600
abs2n50	15453.78	15630.97	1.15%	**15455.34**	**0.01%**	2412	15484.5	0.20%	3600
abs3n50	15747.73	16055.08	1.95%	15867.45	0.76%	2996	**15844.1**	**0.61%**	2683.44
abs4n50	17163.52	17450.53	1.67%	**17173.36**	**0.06%**	2169	17227.3	0.37%	2339.63
abs5n50	16143.06	16313.73	1.06%	**16143.06**	**0.00%**	2585	16522.5	2.35%	2699.38
averages			1.94%		0.05%			0.24%	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 11: Results for the IRP-OU - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 6$

	ABLS[2]	BPS[7]		ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	3335.24	3348.24	0.39	**3335.24**	**0.00**	16	**3335.24**	**0.00**	21.89
abs2n05	2722.33	**2722.33**	**0.00**	**2722.33**	**0.00**	23	**2722.33**	**0.00**	18.7
abs3n05	4776	4783.42	0.16	**4776**	**0.00**	17	**4776**	**0.00**	20.17
abs4n05	3246.66	3389.66	4.40	**3246.66**	**0.00**	21	**3246.66**	**0.00**	21.77
abs5n05	2419.67	2498.52	3.26	**2419.67**	**0.00**	12	**2419.67**	**0.00**	22.09
abs1n10	4499.24	4534.62	0.79	**4499.24**	**0.00**	78	**4499.24**	**0.00**	103.27
abs2n10	5236.97	**5236.98**	**0.00**	**5236.97**	**0.00**	105	**5236.98**	**0.00**	87.01
abs3n10	4652.52	**4652.53**	**0.00**	**4652.52**	**0.00**	67	**4652.53**	**0.00**	110.35
abs4n10	5104.9	5314.16	4.10	**5104.9**	**0.00**	73	**5104.9**	**0.00**	92.82
abs5n10	4670.75	4760.99	1.93	**4670.75**	**0.00**	61	**4670.75**	**0.00**	85.99
abs1n15	5462.67	5600.34	2.52	**5462.67**	**0.00**	415	**5462.67**	**0.00**	358.27
abs2n15	5494.73	5907.4	7.51	**5494.73**	**0.00**	360	**5494.73**	**0.00**	309.56
abs3n15	6060.37	6168.68	1.79	**6060.37**	**0.00**	276	6101.82	0.68	307.19
abs4n15	5504.64	6027.52	9.50	**5504.64**	**0.00**	380	5545.35	0.74	380.94
abs5n15	5309.47	5311.46	0.04	**5309.47**	**0.00**	256	**5309.47**	**0.00**	332.58
abs1n20	6490.17	6541.11	0.78	6501.26	0.17	559	**6490.17**	**0.00**	1130.63
abs2n20	6082.53	6348.07	4.37	**6093.54**	**0.18**	987	6181.05	1.62	656.74
abs3n20	6950.19	7186.46	3.40	**6953.78**	**0.05**	733	**6953.78**	**0.05**	632.68
abs4n20	7432.77	7729.58	3.99	**7432.77**	**0.00**	698	7453.78	0.28	640.29
abs5n20	7210.72	7369.9	2.21	**7210.72**	**0.00**	1212	**7210.72**	**0.00**	927.83
abs1n25	7095.85	7595.85	7.05	**7095.85**	**0.00**	1495	**7095.85**	**0.00**	1429.03
abs2n25	7484.83	8067.75	7.79	**7504.84**	**0.27**	1188	7570.46	1.14	1775.32
abs3n25	7728.75	7996.62	3.47	**7728.75**	**0.00**	3170	7733.76	0.06	2044.15
abs4n25	7509.01	8035.38	7.01	**7509.01**	**0.00**	1251	7527.16	0.24	1701.65
abs5n25	7452.27	7871.75	5.63	7518.61	0.89	1496	**7511.8**	**0.80**	1102.59
abs1n30	8319.58	8761.13	5.31	8349.59	0.36	2334	**8319.58**	**0.00**	2424.12
abs2n30	7761.52	8049.36	3.71	**7768.53**	**0.09**	3251	7778.27	0.22	2928.55
abs3n30	8214.54	8654.55	5.36	8365.82	1.84	3654	**8276.46**	**0.75**	3280.48
abs4n30	7574.79	8067.12	6.50	**7574.79**	**0.00**	5146	7665.8	1.20	2925.18
abs5n30	7366.46	7538.91	2.34	**7402.71**	**0.49**	2220	**7402.71**	**0.49**	3600
averages			3.51		0.14			0.28	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 12: Results for the IRP-OU - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 6$

	ABLS[2]	BPS[7]		ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	5942.82	**5942.82**	**0.00**	**5942.82**	**0.00**	17	**5942.82**	**0.00**	19.05
abs2n05	5045.91	5047.79	0.04	**5045.91**	**0.00**	20	**5045.91**	**0.00**	23.97
abs3n05	6956.28	6969.76	0.19	**6956.28**	**0.00**	16	**6956.28**	**0.00**	23.45
abs4n05	5163.42	5226.16	1.22	**5163.42**	**0.00**	29	5167.88	0.09	24.65
abs5n05	4581.66	4593.05	0.25	**4581.66**	**0.00**	17	**4581.66**	**0.00**	23
abs1n10	8870.15	8945.11	0.85	**8870.15**	**0.00**	78	**8870.15**	**0.00**	96.67
abs2n10	8569.73	8999.23	5.01	**8569.73**	**0.00**	66	**8569.73**	**0.00**	95.41
abs3n10	8509.81	**8509.81**	**0.00**	**8509.81**	**0.00**	129	**8509.81**	**0.00**	117.49
abs4n10	8792.29	8994.79	2.30	**8792.29**	**0.00**	112	**8792.29**	**0.00**	115.35
abs5n10	9620.07	9735.4	1.20	**9620.07**	**0.00**	67	**9620.07**	**0.00**	109.29
abs1n15	12118.83	**12118.83**	**0.00**	**12118.83**	**0.00**	266	**12118.83**	**0.00**	331.14
abs2n15	11932.1	12158.12	1.89	**11932.1**	**0.00**	305	12012.6	0.67	380.23
abs3n15	13554.15	**13554.15**	**0.00**	**13554.15**	**0.00**	369	13557	0.02	386.16
abs4n15	10618.55	10711.77	0.88	**10618.55**	**0.00**	285	**10618.55**	**0.00**	375.51
abs5n15	10385.54	10721.92	3.24	**10385.54**	**0.00**	220	**10385.54**	**0.00**	380.32
abs1n20	14702.95	15250.81	3.73	14722.95	0.14	507	**14702.95**	**0.00**	1044.56
abs2n20	14646.96	14785.04	0.94	**14670.34**	**0.16**	746	14719.8	0.50	703.16
abs3n20	14532.91	14764.08	1.59	14577.14	0.30	863	**14558.3**	**0.17**	961.49
abs4n20	14539.72	14590.96	0.35	**14539.72**	**0.00**	1084	**14539.72**	**0.00**	1189.16
abs5n20	15896.71	16506.45	3.84	15904	0.05	533	**15896.71**	**0.00**	1207.29
abs1n25	15581.47	15726.5	0.93	**15610.98**	**0.19**	1941	15668.4	0.56	2136.64
abs2n25	16823.16	17017.02	1.15	**16843.16**	**0.12**	1544	16891.4	0.41	1804.32
abs3n25	18098.02	18506.33	2.26	**18121.26**	**0.13**	2198	18168.5	0.39	2598.57
abs4n25	16303.69	17026.13	4.43	**16303.69**	**0.00**	1368	**16303.69**	**0.00**	2253.95
abs5n25	19047.7	19394.1	1.82	**19080.27**	**0.17**	1856	19085.4	0.20	2313.86
abs1n30	23183.99	23754.83	2.46	**23183.99**	**0.00**	2365	23206.5	0.10	3600
abs2n30	20090.29	20712.82	3.10	**20159.96**	**0.35**	2178	20232.7	0.71	2588.49
abs3n30	23382.73	23918.93	2.29	**23439.91**	**0.24**	3978	23478.7	0.41	3600
abs4n30	17649.53	18247.06	3.39	**17746.79**	**0.55**	5063	17772.6	0.70	3600
abs5n30	18979.93	19270.91	1.53	**18997.61**	**0.09**	2239	19071.1	0.48	3600
averages			1.70		0.08			0.18	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 13: Results for the IRP-ML - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 3$

	ABLS[2]	ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	1235.92	**1235.92**	**0.00**	2	**1235.92**	**0.00**	7.97
abs2n05	988.66	**988.66**	**0.00**	2	**988.66**	**0.00**	5.4
abs3n05	1758.02	**1758.02**	**0.00**	3	**1758.02**	**0.00**	5.77
abs4n05	1397.29	**1397.29**	**0.00**	4	**1397.29**	**0.00**	5.93
abs5n05	999.42	**999.42**	**0.00**	2	**999.42**	**0.00**	5.34
abs1n10	1743.07	**1743.07**	**0.00**	10	**1743.07**	**0.00**	28.44
abs2n10	2229.25	**2229.25**	**0.00**	10	**2229.25**	**0.00**	16.96
abs3n10	1871.14	**1871.14**	**0.00**	9	**1871.14**	**0.00**	15.43
abs4n10	1773	**1773**	**0.00**	8	**1773**	**0.00**	17.13
abs5n10	1938.18	**1938.18**	**0.00**	10	**1938.18**	**0.00**	14.98
abs1n15	2131.04	**2131.04**	**0.00**	30	**2131.04**	**0.00**	65.07
abs2n15	2131.58	**2131.58**	**0.00**	24	**2131.58**	**0.00**	45.52
abs3n15	2463.68	**2463.68**	**0.00**	25	**2463.68**	**0.00**	36.75
abs4n15	2151.94	**2151.94**	**0.00**	34	2155.94	0.19	37.66
abs5n15	2160.59	**2160.59**	**0.00**	21	**2160.59**	**0.00**	38.44
abs1n20	2267.32	**2267.32**	**0.00**	43	**2267.32**	**0.00**	92.04
abs2n20	2497.9	**2497.9**	**0.00**	72	2506.9	0.36	97.56
abs3n20	2590.48	**2590.48**	**0.00**	60	**2590.48**	**0.00**	88.99
abs4n20	3122.31	**3122.99**	**0.02**	90	3163.31	1.31	110.64
abs5n20	2849.9	**2849.9**	**0.00**	58	**2849.9**	**0.00**	101.03
abs1n25	2840.92	**2840.92**	**0.00**	107	**2840.92**	**0.00**	174.65
abs2n25	3014.56	**3014.56**	**0.00**	92	3024.56	0.33	176.37
abs3n25	3050.4	**3050.4**	**0.00**	100	**3050.4**	**0.00**	129.88
abs4n25	3078.67	**3078.67**	**0.00**	161	3107.01	0.92	184.02
abs5n25	2954.96	**2954.96**	**0.00**	104	2960.96	0.20	191.96
abs1n30	3427.78	**3427.78**	**0.00**	221	3454.78	0.79	440.05
abs2n30	3328.94	**3328.94**	**0.00**	190	3391.94	1.89	347.55
abs3n30	3471.86	**3471.86**	**0.00**	249	3502.86	0.89	288.86
abs4n30	3321.48	**3321.48**	**0.00**	436	3389.67	2.05	330.64
abs5n30	2914.6	**2914.6**	**0.00**	201	**2914.6**	**0.00**	250.14
abs1n35	3346.12	**3346.12**	**0.00**	446	3406.12	1.79	526.33
abs2n35	3541.71	**3541.71**	**0.00**	601	3559.71	0.51	733.01
abs3n35	3811.78	**3811.78**	**0.00**	340	3906.78	2.49	341.14
abs4n35	3229.34	**3229.34**	**0.00**	402	3233.34	0.12	487.55
abs5n35	3315.26	**3315.26**	**0.00**	284	3369.26	1.63	389.27
abs1n40	3702.14	**3702.14**	**0.00**	589	**3702.14**	**0.00**	671.49
abs2n40	3832.09	**3832.09**	**0.00**	546	3900.56	1.79	1075.83
abs3n40	3874.62	**3874.62**	**0.00**	471	**3874.62**	**0.00**	582.95
abs4n40	3534.8	**3534.8**	**0.00**	550	3579.8	1.27	565.63
abs5n40	3575.46	**3575.46**	**0.00**	450	3623.12	1.33	1074.06
abs1n45	3950.86	**3950.86**	**0.00**	939	**3950.86**	**0.00**	1216.39
abs2n45	3702.72	**3702.72**	**0.00**	700	**3702.72**	**0.00**	871.16
abs3n45	3968.04	**3968.04**	**0.00**	668	3995.04	0.68	1866.71
abs4n45	3998.26	**3998.26**	**0.00**	837	4039.39	1.03	1440.87
abs5n45	3717.54	**3717.54**	**0.00**	744	3753.31	0.96	1633.47
abs1n50	4047.18	**4047.18**	**0.00**	1009	4110.18	1.56	1120.51
abs2n50	4512.96	**4512.96**	**0.00**	1229	4556.98	0.98	1940.11
abs3n50	4451.44	**4451.44**	**0.00**	1112	**4451.44**	**0.00**	2317.85
abs4n50	4405.84	**4405.84**	**0.00**	1105	**4405.84**	**0.00**	1537.38
abs5n50	4218.37	**4218.37**	**0.00**	982	4307.7	2.12	1861.19
averages			0.00			0.54	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 14: Results for the IRP-ML - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 3$

	ABLS[2]	ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	2108.34	**2108.34**	**0.00**	1	**2108.34**	**0.00**	5.68
abs2n05	1767.06	**1767.06**	**0.00**	1	**1767.06**	**0.00**	5.93
abs3n05	2973	**2973**	**0.00**	3	**2973**	**0.00**	6.23
abs4n05	1981.04	**1981.04**	**0.00**	4	**1981.04**	**0.00**	5.35
abs5n05	2170.04	**2170.04**	**0.00**	1	**2170.04**	**0.00**	5.89
abs1n10	4510.61	**4510.61**	**0.00**	10	**4510.61**	**0.00**	15.76
abs2n10	4504.61	**4504.61**	**0.00**	11	4511.51	0.15	17.55
abs3n10	4031.4	**4031.4**	**0.00**	9	**4031.4**	**0.00**	14.83
abs4n10	3933.46	**3933.46**	**0.00**	9	**3933.46**	**0.00**	18.86
abs5n10	4709.79	**4709.79**	**0.00**	10	**4709.79**	**0.00**	18.63
abs1n15	5589.7	**5589.7**	**0.00**	31	**5589.7**	**0.00**	50.2
abs2n15	5443.34	**5443.34**	**0.00**	38	5454.34	0.20	40.76
abs3n15	6300.86	**6300.86**	**0.00**	27	**6300.86**	**0.00**	54.46
abs4n15	4977.58	**4977.58**	**0.00**	44	4981.58	0.08	39.81
abs5n15	4867.53	**4867.53**	**0.00**	21	**4867.53**	**0.00**	46.48
abs1n20	6859.02	**6859.02**	**0.00**	44	**6859.02**	**0.00**	99.61
abs2n20	7087.74	**7087.74**	**0.00**	76	**7087.74**	**0.00**	90.98
abs3n20	7354.68	**7354.68**	**0.00**	68	**7354.68**	**0.00**	98.79
abs4n20	6952.79	**6957.78**	**0.07**	88	7027.51	1.07	100.58
abs5n20	7874.26	**7874.26**	**0.00**	57	7937.26	0.80	75.42
abs1n25	8227.86	**8227.86**	**0.00**	115	**8227.86**	**0.00**	210.02
abs2n25	8765.72	**8765.72**	**0.00**	102	8809.72	0.50	126.47
abs3n25	9382.42	**9382.42**	**0.00**	108	**9382.42**	**0.00**	148.92
abs4n25	8452.93	**8452.93**	**0.00**	165	8528.16	0.89	129.68
abs5n25	10081.42	**10081.42**	**0.00**	116	10087.4	0.06	206.71
abs1n30	12066.86	**12066.86**	**0.00**	234	**12066.9**	**0.00**	278.38
abs2n30	10941.32	**10941.32**	**0.00**	271	10992.5	0.47	295.38
abs3n30	12122.36	**12122.36**	**0.00**	245	12131.4	0.07	276.03
abs4n30	9687.1	**9687.1**	**0.00**	312	9736.04	0.51	349.48
abs5n30	9773.9	**9773.9**	**0.00**	209	9780.9	0.07	379.53
abs1n35	11659.88	**11659.88**	**0.00**	483	**11659.9**	**0.00**	519.62
abs2n35	10466.8	**10466.8**	**0.00**	448	10515.1	0.46	637.33
abs3n35	13776.46	**13776.46**	**0.00**	322	13908.5	0.96	621.28
abs4n35	10307.4	**10307.4**	**0.00**	447	10311.4	0.04	498.91
abs5n35	10847.82	**10847.82**	**0.00**	297	10965.8	1.09	600.54
abs1n40	13364.92	**13364.92**	**0.00**	624	13365.7	0.01	527.47
abs2n40	11317.85	**11317.85**	**0.00**	612	11533.2	1.90	859.44
abs3n40	13598.94	**13598.94**	**0.00**	544	13768.9	1.25	780.53
abs4n40	11353.39	**11353.39**	**0.00**	693	11425.6	0.64	645.41
abs5n40	13070.18	**13070.18**	**0.00**	523	**13070.18**	**0.00**	1195.34
abs1n45	14179.1	**14179.1**	**0.00**	958	14248.1	0.49	1491.94
abs2n45	13142.22	**13142.22**	**0.00**	746	**13142.2**	**0.00**	829.58
abs3n45	14843.6	**14843.6**	**0.00**	717	14970.6	0.86	1036.92
abs4n45	13574.5	**13574.5**	**0.00**	876	**13574.5**	**0.00**	1429.25
abs5n45	13587.26	**13587.26**	**0.00**	732	13705.1	0.87	1197.43
abs1n50	14577.3	**14577.3**	**0.00**	1092	14767.3	1.30	1588.94
abs2n50	15001.64	**15001.64**	**0.00**	1036	**15001.64**	**0.00**	2321.79
abs3n50	15279.49	**15279.49**	**0.00**	1342	15407.6	0.84	2288.75
abs4n50	16517	**16517**	**0.00**	1217	16587	0.42	1959.08
abs5n50	15678.67	**15678.67**	**0.00**	1041	15798.9	0.77	2390.51
averages			0.00			0.34	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 15: Results for the IRP-ML - low inventory cost ($h_i \in [0.01, 0.05]$), $p = 6$

	ABLS[2]	ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	3187.3	**3187.3**	**0.00**	17	**3187.3**	**0.00**	12.01
abs2n05	2565.92	2566.47	0.02	19	**2565.92**	**0.00**	11.35
abs3n05	4489.83	**4489.83**	**0.00**	19	**4489.83**	**0.00**	13.99
abs4n05	3174.35	**3174.35**	**0.00**	14	**3174.35**	**0.00**	13.65
abs5n05	2267.1	**2267.1**	**0.00**	11	**2267.1**	**0.00**	13.51
abs1n10	4141.51	**4141.51**	**0.00**	69	4214.78	1.77	54.27
abs2n10	5044.63	**5046.13**	**0.03**	69	**5046.13**	**0.03**	52.6
abs3n10	4506.83	**4508.8**	**0.04**	71	**4508.8**	**0.04**	61.53
abs4n10	4823.53	**4823.53**	**0.00**	49	**4823.53**	**0.00**	55.04
abs5n10	4545.98	**4545.98**	**0.00**	65	**4545.98**	**0.00**	47.87
abs1n15	5389.08	**5391.17**	**0.04**	166	**5391.17**	**0.04**	121.33
abs2n15	5418.47	**5423.47**	**0.09**	245	5486.72	1.26	186.61
abs3n15	5897.68	**5897.68**	**0.00**	161	5901.36	0.06	160.55
abs4n15	5335.01	**5335.01**	**0.00**	203	5339.35	0.08	217.96
abs5n15	5052.51	**5074.68**	**0.44**	127	**5074.68**	**0.44**	117.62
abs1n20	6114.04	**6114.04**	**0.00**	475	6179.44	1.07	349.32
abs2n20	5957.31	**5958.29**	**0.02**	561	5961.37	0.07	446.55
abs3n20	6784.06	**6786.04**	**0.03**	429	6862.05	1.15	357.04
abs4n20	7309.54	**7394.84**	**1.17**	508	7448.89	1.91	351.66
abs5n20	6961.82	**6967.4**	**0.08**	309	6989.8	0.40	401.17
abs1n25	7052.06	**7064**	**0.17**	1678	7070.75	0.27	672.79
abs2n25	7231.75	**7254.38**	**0.31**	725	7264.84	0.46	808.32
abs3n25	7514.57	**7514.57**	**0.00**	1162	**7514.57**	**0.00**	773.75
abs4n25	7462.08	**7462.08**	**0.00**	744	**7462.08**	**0.00**	198.17
abs5n25	7048.4	**7059.9**	**0.16**	845	**7059.9**	**0.16**	546.33
abs1n30	8052.73	**8092.11**	**0.49**	1930	8102.64	0.62	1438.03
abs2n30	7629.99	**7631.21**	**0.02**	2602	7639.48	0.12	1710.08
abs3n30	8136.21	8137.33	0.01	2422	**8136.21**	**0.00**	1931.62
abs4n30	7502.49	**7502.49**	**0.00**	3712	7509.41	0.09	386.55
abs5n30	7228.63	**7265.35**	**0.51**	2003	7278.49	0.69	1609.11
averages			0.12			0.36	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

Table 16: Results for the IRP-ML - high inventory cost ($h_i \in [0.1, 0.5]$), $p = 6$

	ABLS[2]	ABHS[3]			ALNS		
Instance	z^*	z	gap (%)	time$_1$ (s)	z	gap (%)	time$_2$ (s)
abs1n05	5789.35	**5789.35**	**0.00**	16	**5789.35**	**0.00**	13.19
abs2n05	4883.77	**4883.77**	**0.00**	14	**4883.77**	**0.00**	12.51
abs3n05	6643.29	**6643.29**	**0.00**	24	**6643.29**	**0.00**	11.81
abs4n05	5076.88	**5076.88**	**0.00**	22	**5076.88**	**0.00**	12.29
abs5n05	4377.71	**4377.71**	**0.00**	13	**4377.71**	**0.00**	11.18
abs1n10	8480.17	**8480.17**	**0.00**	86	**8480.17**	**0.00**	69.24
abs2n10	8347.44	8356.35	0.11	79	**8347.44**	**0.00**	49.63
abs3n10	8321.68	8360.05	0.46	107	**8321.68**	**0.00**	54.53
abs4n10	8474.26	**8474.26**	**0.00**	77	8493.26	0.22	57.84
abs5n10	9386.03	**9386.03**	**0.00**	82	**9386.03**	**0.00**	50.1
abs1n15	12052.56	**12052.56**	**0.00**	179	12192.5	1.16	148.48
abs2n15	11823.55	11825.87	0.02	243	**11823.55**	**0.00**	155.75
abs3n15	13305.71	**13305.71**	**0.00**	270	**13305.71**	**0.00**	185.46
abs4n15	10479.29	**10494.11**	**0.14**	239	**10494.11**	**0.14**	173.15
abs5n15	10054.09	**10070.56**	**0.16**	201	10147.4	0.93	110.85
abs1n20	14266.53	**14301**	**0.24**	450	14346.4	0.56	339.59
abs2n20	14477.84	**14551.58**	**0.51**	392	14686.7	1.44	291.46
abs3n20	14319.35	**14328.68**	**0.07**	426	14382.5	0.44	303.5
abs4n20	14390.27	**14417**	**0.19**	398	14586.4	1.36	417.46
abs5n20	15556.7	**15563.44**	**0.04**	341	15919.8	2.33	388.54
abs1n25	15487.66	**15555.79**	**0.44**	805	15802	2.03	747.94
abs2n25	16502.46	**16516.45**	**0.08**	795	**16516.45**	**0.08**	609.68
abs3n25	17833.35	**17833.35**	**0.00**	894	**17833.35**	**0.00**	746.24
abs4n25	16193.96	**16316.37**	**0.76**	1070	16426.8	1.44	922.49
abs5n25	18552.42	**18570.92**	**0.10**	1570	18686.49	0.72	489.17
abs1n30	22837.94	**22874.4**	**0.16**	1922	22981.2	0.63	1609.67
abs2n30	19876.76	**19997.28**	**0.61**	1850	20032.7	0.78	1342.76
abs3n30	23096.93	**23198.29**	**0.44**	2192	23741	2.79	1720.75
abs4n30	17509.82	**17550.33**	**0.23**	1952	17705.3	1.12	1696.01
abs5n30	18731.81	**18770.28**	**0.21**	1694	18856.7	0.67	1620.1
averages			0.17			0.63	

time$_1$: run on an Intel Dual Core 1.86GHz and 3.2 GB RAM
time$_2$: run on an Intel Core2Duo 2.4GHz and 4 GB RAM

We have also evaluated the impact of the transshipment cost on the solution and its cost. To this end, we have gradually increased the transshipment cost, and we have solved a subset of instances under the IRPT-OU and IRPT-ML policies. The results of these experiments are reported in Tables 17 and 18. It can be seen that costs become much closer to those of the traditional IRP when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$, and several instances make no use of transshipment when b_{ij} is set equal to $0.10c_{ij}$. Transshipments start to be economically interesting when the cost of outsourcing the delivery of ten units does not exceed the cost of transporting one unit with the supplier's vehicle, all other costs being identical.

We have also performed tests to assess the impact of individual movements in our ALNS heuristic. We have examined the impact of removing individual movements for a subset of instances, including some small, medium and large instances, with three and six time periods. The results of the experiments for the IRPT-ML are summarized on Table 19.

Table 17: Average increase of the solution cost when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$ for the IRPT-OU

Set of instances	Avg gap (%) to the IRP-OU when $b_{ij} = 0.01c_{ij}$	Avg gap (%) to the IRP-OU when $b_{ij} = 0.05c_{ij}$
$p = 3$, low inventory cost	−20.38	−2.31
$p = 3$, high inventory cost	−11.89	−1.39
$p = 6$, low inventory cost	−10.25	−0.19
$p = 6$, high inventory cost	−6.06	0.00

Table 18: Average increase of the solution cost when the transshipment cost increases from $b_{ij} = 0.01c_{ij}$ to $0.05c_{ij}$ for the IRPT-ML

Set of instances	Avg gap (%) to the IRP-ML when $b_{ij} = 0.01c_{ij}$	Avg gap (%) to the IRP-ML when $b_{ij} = 0.05c_{ij}$
$p = 3$, low cost	−18.21	−0.06
$p = 3$, high cost	−9.63	−0.21
$p = 6$, low cost	−10.75	−0.09
$p = 6$, high cost	−4.96	0.21

Removing some movements can have a major impact both on solution quality and on running time. Specifically, movements that require solving a network flow problem frequently are time consuming. Removing them reduces the CPU time significantly at the expense of a slight decrease in solution quality. Also, movements that increase the diversification of the solution (using randomness) are very fast. They do not impact the CPU time, and may even deteriorate solution quality on some instances. In our preliminary tests we have performed

Table 19: Average increase (%) of the solution cost when individual movements are removed from the ALNS algorithm - IRPT-ML

Movement removed	1	2	3	4	5	6	7	8	9	10	11
Average increase (%)	0.47	1.63	0.52	1.38	0.26	0.39	0.94	0.16	2.46	0.35	0.09

such analyses in order to fine tune some movements and change the ones that were too time consuming or that did not have a positive impact on the solution.

Finally, we also wanted to identify how well some movements would perform if taken in isolation. To this end, we have executed the code with only the few movements we wanted to assess. We offer in Table 20 a summary with three different analyses: randomness only, best (worst) insertion (removal) only, and mixed movements only (those that apply removals and insertions simultaneously). These tests were performed on the IRPT-OU on a subset of instances including small, medium and large ones.

Table 20: Average increase of the solution cost (%) when only a subset of movements are used

	Randomness only	Best/worst only	Mixed only
Average % increase	0.21	24.58	1.42

6 Conclusions

We have introduced a new variant of the Inventory-Routing Problem, in which transshipments are allowed. This problem is very difficult to solve exactly and the lower bounds provided by CPLEX are rather weak. To generate good solutions, we have developed a powerful ALNS heuristic capable of solving four variants of the problem: IRP-OU, IRPT-OU, IRP-ML and IRPT-ML. Comparative tests on a large set of instances have shown that our heuristic can produce high quality solutions within reasonable computing times. We have also shown that the use of transshipment can reduce solution cost significantly, depending on the ratio between unit transshipment cost the cost of using the supplier's vehicle.

References

[1] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management

and routing. *Computers & Operations Research*, 37(9):1515 – 1536, 2010.

[2] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382 – 391, 2007.

[3] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, page ijoc.1100.0439, 2011. doi: 10.1287/ijoc.1100.0439.

[4] D. O. Bausch, G. G. Brown, and D. Ronen. Scheduling short-term marine transport of bulk products. *Maritime Policy & Management*, 25(4):335 – 348, 1998.

[5] W. J. Bell, L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4 – 23, 1983.

[6] L. Bertazzi and M. G. Speranza. Continuous and discrete shipping strategies for the single link problem. *Transportation Science*, 36(3):314 – 325, 2002.

[7] L. Bertazzi, G. Paletta, and M. G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36 (1):119 – 132, 2002.

[8] T. W. Chien, A. Balakrishnan, and R. T. Wong. An integrated inventory allocation and vehicle routing problem. *Transportation Science*, 23(2):67 – 76, 1989.

[9] M. Christiansen. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3 – 16, 1999.

[10] J.-F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo. Vehicle routing. In C. Barnhart and G. Laporte, editors, *Transportation*, pages 367 – 428. North-Holland, Amsterdan, 2007.

[11] M. Dror and M. O. Ball. Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, 34(6):891 – 905, 1987.

[12] G. Gallego and D. Simchi-Levi. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer r-systems. *Management Science*, 36(2):240 – 243, 1990.

[13] G. Gallego and D. Simchi-Levi. Rejoinder to 'a note on bounds for direct shipping costs'. *Management Science*, 40(10):1393, 1994.

[14] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22(1):1 – 29, 1997.

[15] R. W. Hall. A note on bounds for direct shipping cost. *Management Science*, 38(8):1212 – 1214, 1992.

[16] A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94–118, 2002.

[17] A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42 – 70, 2004.

[18] A. S. Minkoff. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research*, 41(1):77 – 90, 1993.

[19] B. K. Mishra and S. Raghunathan. Retailer- vs. vendor-managed inventory and brand competition. *Management Science*, 50(4):445 – 457, 2004.

[20] J. J. Neale, B. T. Tomlin, and S. P. Williems. The role of inventory in superior supply chain performance. In Terry P. Harrison, Hau Leung Lee, and John J. Neale, editors, *The Practice of Supply Chain Management: Where Theory and Application Converge*. Kluwer, Boston, 2003.

[21] L. M. Nonås and K. Jörnsten. Heuristics in the multi-location inventory system with transshipments. In H. Kotzab, S. Seuring, M. Müller, and G. Reiner, editors, *Research Methodologies in Supply Chains Management*, pages 509 – 524. Physica-Verlag, Heidelberg, 2005.

[22] L. M. Nonås and K. Jörnsten. Optimal solution in the multi-location invelltory system with transshipments. *Journal of Mathematical Modelling and Algorithms*, 6(1):47 – 75, 2007.

[23] C. Paterson, G. Kiesmller, R. Teunter, and K. Glazebrook. Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2):125 – 136, 2011.

[24] J. A. Persson and M. Göthe-Lundgren. Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research*, 163(3):631 – 652, 2005.

[25] D. Pisinger and S. Ropke. A general heuristic for the vehicle routing problem. *Computers & Operations Research*, 34(8):2403 – 2435, 2007.

[26] W. W. Qu, J. H. Bookbinder, and P. Iyogun. An integrated inventory-transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115(2):254 – 269, 1999.

[27] S. Ropke and D. Pisinger. An adaptive large neighborghood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455 – 472, 2006.

[28] P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, University of Strathclyde, Glasgow, 1997.

[29] S. Sindhuchao, H. E. Romeijn, E. Akçali, and R. Boondiskulchok. An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. *Journal of Global Optimization*, 32:93 – 118, 2005.

[30] M. G. Speranza and W. Ukovich. Minimizing transportation and inventory costs for several products on a single link. *Operations Research*, 42(5):879 – 894, 1994.

[31] M. G. Speranza and W. Ukovich. An algorithm for optimal shipments with given frequencies. *Naval Research Logistics*, 43(5):655 – 671, 1996.