Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems

Angelo Sifaleras*

Department of Applied Informatics, School of Information Sciences, University of Macedonia, 156 Egnatia Str., Thessaloniki 54636, Greece

Ioannis Konstantaras

Department of Business Administration, School of Business Administration, University of Macedonia, 156 Egnatia Str., Thessaloniki 54636, Greece

Abstract

The multi-product dynamic lot sizing problem with product returns and recovery is an important problem that appears in reverse logistics and is known to be NP-hard. In this paper we propose an efficient variable neighborhood descent heuristic algorithm for solving this problem. Furthermore, we present a new benchmark set with the largest instances in the literature. The computational results, demonstrate that our approach outperforms the state-of-the-art Gurobi optimizer.

Keywords: Inventory, Variable Neighborhood Search, Mathematical Programming, Lot Sizing, Reverse Logistics
2010 MSC: 90B05, 90C59, 65K05, 90-08

1. Introduction

Over the past two decades, reverse logistics and closed-loop supply chain have gained substantial interest in business and academia. Evidences can be found in the very recent review papers given by Srivastava [38], Guide and Van Wassenhove [17], Souza [37], Govindan et al. [16] and Fahimnia et al. [11]. A closed-loop supply chain concentrates on taking back used products from customers and recovering value by remanufacturing the entire product, and/or some of its modules, components, and parts. The remanufactured products have usually the same quality as the new products and are sold for the same price, but they are less costly. So, these products can also be used to fulfil customers' demand.

*Corresponding author. Tel: +302310 891884, Fax: +302310 891881
Email addresses: sifalera@uom.gr (Angelo Sifaleras), ikonst@uom.gr (Ioannis Konstantaras)

The dynamic lot sizing problem, i.e., scheduling manufacturing orders over a number of finite time periods in which demand is known and dynamic, is one of the fundamental and extensively studied OR model in inventory and production control. The work in this field is too extensive and for this reason we refer the basic review papers on dynamic lot sizing problem by Brahimi et al. [5], Robinson et al. [31], Buschkühl et al. [7], Ullah and Parveen [40] and Bushuev et al. [8].

Due to the increasing attention on the sustainability in industry, production planning and inventory control problems have been studied in the field of closed-loop supply chain systems. Manufacturers have started to integrate remanufacturing facilities into the regular production environment. The quantitative models in inventory control and production planning for closed-loop supply chain have increased considerably during the past two decades. This concludes from the detailed review papers given by Akçalı and Çetinkaya [1], Morgan and Gagnon [25] and Brandenburg et al. [6] and from the papers of a special issue on sustainable supply chain management by Govindan and Cheng [15]. One of the first works on dynamic lot-sizing problem in closed-loop supply chain was given by Richter and Sombrutzki [30], who extended the classical Wagner-Whitin model [41] by introducing the alternative of remanufactured products to satisfy customers demand. Golany et al. [14] extended the model of Richter and Sombrutzki [30] assuming disposal of returned products and Pineyro and Viera [28] proposed a Tabu Search procedure for finding near optimal solutions of the dynamic lot sizing with disposal option. Teunter et al. [39] proposed a dynamic programming algorithm for the reverse dynamic lot sizing problem with separate and joint setup for manufacturing and remanufacturing and also they tested some very well known heuristics for the above problem. Schulz [33] proposed some variants of the well known Silver-Meal heuristic to solve the case, proposed by Teunter et al. [39], with separate setup costs for manufacturing and remanufacturing. The last two papers inspired many researchers to develop both single solution and population-based metaheuristics for tackling the dynamic lot sizing problem with separate setup costs for manufacturing and remanufacturing (see Moustaki et al. [26], Li et al. [20], Sifaleras et al. [36], Parsopoulos et al. [27]).

One of the most important issues that should be tackled by a company is the determination of the replenishment policy of the different goods (spare parts, raw material, components or finished goods) involved in the supply chain. This problem becomes more complex whether the demand for each of the products varies with time through a finite planning horizon. Accordingly, problems in this category are usually referred to as multi-product dynamic lot-sizing model. The Multi-product Dynamic Lot Sizing Problem with Remanufacturing activities (MDLSRP) is a very interesting problem that appears in closed-loop supply chain. Till now, there are few papers addressed multi product dynamic lot sizing problems with returns and remanufacturing. Li et al. [21] developed an uncapacitated multi product dynamic lot sizing problem with remanufacturing options and demand substitution and proposed an approximate procedure to compute near-optimal solution. Li et al. [22] studied a variant of the previous

model assuming capacity constraints and proposed a heuristic genetic algorithm to solve the problem. Sahling [32] presented a new model formulation for the multi-product dynamic lot-sizing problem with product returns and remanufacturing. He proposed a column-generation approach with the combination of a truncated branch-and-bound method to solve the proposed model. Figueira et al. [12] presented a closed-loop supply chain case study for the pulp and paper industry and proposed a hybrid solution method to solve the problem.

In this paper we propose a Variable Neighborhood Descent (VND) heuristic algorithm for the solution of the multi-product dynamic lot-sizing problem with remanufacturing options. Variable Neighborhood Search (VNS) is a metaheuristic based on a systematic change of the neighborhood structures within a search introduced by Mladenović and Hansen [18]. Although, VNS has also been applied to other inventory problems [2, 3, 42, 43, 44] or combined inventory / routing problems [23, 24], this is the first time that it is used for this particular problem in the literature. Also, this paper extends the previous work of Sifaleras and Konstantaras [35], presenting more comprehensive experimental results on a larger set of benchmark instances.

The rest of the paper is organized as follows: Section 2 provides the basic formulation of the problem. In Section 3, the VND approach and the neighborhood structures that are used, are briefly described. Section 4 exposes the obtained results, and the paper concludes with Section 5.

2. Model formulation

The problem discussed in this paper is to satisfy the demand of items in each period at the lowest total cost possible. The demand is given for a finite planning horizon and is assumed to be not stationary. It can be satisfied by both manufactured new items and remanufactured returned ones (both known as "serviceables"). Also the number of returns is known for all periods and assumed to be not stationary. The returns can be completely remanufactured and sold as new one. The problem studied in our paper consists of the multi-product dynamic lot sizing model with both remanufacturing and manufacturing setup costs; a modified version of the model introduced by Sahling [32], without capacities. The lot sizing problem under separate manufacturing and remanufacturing set up costs is suitable for situations where there are separate production lines, one for manufacturing and one for remanufacturing. The aim is to determine the number of remanufactured and manufactured items per period, in order to minimize the sum of set up costs of the manufacturing and remanufacturing processes and holding costs for returns and serviceables under various operational constraints. Before presenting the formulation of the MDLSRP, we first introduce some notations:

k: product, $k = 1, 2, \ldots, K$.
t: time period, $t = 1, 2, \ldots, T$.
$D(k, t)$: demand for product k in time period t.

$R(k,t)$: number of returned items of product k in period t that can be completely remanufactured and sold as new.
$h_M(k)$: holding cost for the serviceable items of product k per unit time.
$h_R(k)$: holding cost for the recoverable items of product k per unit time.
$z_M(k,t)$: binary decision variable denoting the initiation of a manufacturing lot of product k in period t.
$z_R(k,t)$: binary decision variable denoting the initiation of a remanufacturing lot of product k in period t.
$x_M(k,t)$: number of manufactured items of product k in period t.
$x_R(k,t)$: number of items that are eventually remanufactured of product k in period t.
$k_M(k)$: manufacturing setup cost of product k.
$k_R(k)$: remanufacturing setup cost of product k.
$p_M(k)$: production cost of product k per unit.
$p_R(k)$: remanufacturing cost of product k per unit.
$y_M(k,t)$: inventory level of serviceable items of product k in period t.
$y_R(k,t)$: inventory level of remanufacturable items of product k in period t.
M: A sufficiently large number.

The formulation of MDLSRP is a generalization of the formulation of the Economic Lot Sizing Problem with remanufacturing options (ELSRP) proposed in [33], by considering i) several products instead of a single product, and ii) production and remanufacturing costs of product k per unit. The following proposed Mixed Integer Linear Programming (MILP) formulation of the MDLSRP differs from the one presented by Sahling [32], only by not taking into account the available capacities of the resources for manufacturing and remanufacturing in a period.

$$\begin{aligned} \min z = & \sum_{k=1}^{K}\sum_{t=1}^{T}\left(h_R(k)y_R(k,t) + h_M(k)y_M(k,t)\right) \\ & + \sum_{k=1}^{K}\sum_{t=1}^{T}\left(p_R(k)x_R(k,t) + p_M(k)x_M(k,t)\right) \\ & + \sum_{k=1}^{K}\sum_{t=1}^{T}\left(k_R(k)z_R(k,t) + k_M(k)z_M(k,t)\right) \end{aligned} \tag{1}$$

where:

$$z_R(k,t) = \begin{cases} 1, & \text{if } x_R(k,t) > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad z_M(k,t) = \begin{cases} 1, & \text{if } x_M(k,t) > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

are binary decision variables denoting the initiation of a remanufacturing or manufacturing lot, respectively.

$$\begin{aligned} & y_R(k,t) = y_R(k,t-1) + R(k,t) - x_R(k,t), \\ & y_M(k,t) = y_M(k,t-1) + x_R(k,t) + x_M(k,t) - D(k,t), \\ & \qquad \forall t = 1,2,\ldots,T, \quad \forall k = 1,2,\ldots,K. \end{aligned} \tag{3}$$

$$
\begin{aligned}
y_M(k,t) &\geq \sum_{s=t+1}^{t+p} (D(k,s) - M(k,s)\,(z_M(k,s) + z_R(k,s))), \\
&\quad \forall k, \quad \forall t = 1,\ldots,T-1, \quad \forall p = 1,\ldots T-t, \\
y_R(k,t) &\geq \sum_{s=t-p}^{t} (R(k,s) - M(k,s) z_R(k,s)), \\
&\quad \forall k, \quad \forall t = 2,\ldots,T, \quad \forall p = 1,\ldots t-1,
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
x_R(k,t) \leq M z_R(k,t), \qquad & x_M(k,t) \leq M z_M(k,t), \\
\forall t = 1,2,\ldots,T, \quad & \forall k = 1,2,\ldots,K.
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
& y_R(k,t), y_M(k,t), x_R(k,t), x_M(k,t) \geq 0, \quad z_R(k,t), z_M(k,t) \in \{0,1\}, \\
& y_R(k,0) = y_M(k,0) = 0, \quad \forall t = 1,2,\ldots,T, \quad \forall k = 1,2,\ldots,K.
\end{aligned} \tag{6}
$$

The constraints defined in Equations (3) are the inventory balance equations which compute the inventory of returns and serviceables, respectively. Relations (4) are the adapted valid inequalities for the MDLSRP proposed in [29]. Relations (5) ensure that a fixed setup cost is incurred when remanufacturing or manufacturing takes place, respectively. Finally, equations (6) ensure that the inventories are initially empty, set the indicator variables and prevent negative (re)manufacturing or inventory.

3. Neighborhood structures for solving MDLSRP

A neighborhood $N(\sigma)$ is a set of neighboring solutions in the solution space that is defined as $N(\sigma) = \{$solution σ' obtained by applying one or more changes to $\sigma\}$. We have used eight different neighborhoods (i.e., $k_{max} = |N| = 8$), based on shifting the values of each binary variable with different ways. These eight different neighborhoods have been successfully employed in the case of the single product Economic Lot Sizing Problem with remanufacturing options (ELSRP) [36], and are now being generalized to tackle the MDLSRP. It should be noted that in this paper, we present more neighborhoods appropriately generalized for the MDLSRP, than in the recent work by Sifaleras and Konstantaras [35]. This way, the local search part becomes more efficient but also requires more time. For this reason, one more improvement compared to the previous work [35], is the fact that the time-consuming computation of the objective value from scratch, has been replaced by efficiently updating the objective value each time a new feasible neighborhood solution has been found. These eight neighborhoods are as follows:

$N_1(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for remanufacturing $z_R(k,t)$ in one period from 1 to 0 (or reduce only that $x_R(k,t)$ variable), by strictly increasing some $y_R(k,t)$ variables and reducing some $y_M(k,t)$ variables, for some $k = 1,2,\ldots,K$, $t = 1,2,\ldots,T\}$.

$N_2(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for remanufacturing $z_R(k,t)$ in one period from 0 to 1, by strictly reducing some $y_R(k,t)$, $y_M(k,t)$, and $x_M(k,t)$ variables, for some $k = 1,2,\ldots,K$, $t = 1,2,\ldots,T\}$.

$N_3(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for manufacturing $z_M(k,t)$ for one product k in one period from 1 to 0, by increasing only some $y_M(k,t)$ variables and a previous $x_M(k,t)$ variable, for some $k = 1,2,\ldots,K$, $t = 1,2,\ldots,T$, and $i < t\}$.

$N_4(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for manufacturing $z_M(k,t)$ for one product k in one period t from 0 to 1, by reducing only some $y_M(k,t)$ variables and a $x_M(k,t)$ variable, for some $k = 1, 2, \ldots, K$, $t = 1, 2, \ldots, T$, and $i < t\}$.

$N_5(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for manufacturing $z_M(k,t)$ in one period from 1 to 0 (or reduce only that $x_M(k,t)$ variable) and shifting the set-up for manufacturing $z_M(k,t)$ in another period from 0 to 1 (or increase only that $x_M(k,t)$ variable), by strictly reducing some $y_M(k,t)$ variables, for some $k = 1, 2, \ldots, K$, $t = 1, 2, \ldots, T\}$.

$N_6(\sigma) = \{\sigma' \mid \sigma'$ is obtained by shifting the set-up for remanufacturing $z_R(k,t)$ in one period from 1 to 0, by strictly increasing some $y_M(k,t)$, $y_R(k,t)$ variables, and one $x_M(k,t)$ variable, for some $k = 1, 2, \ldots, K$, $t = 1, 2, \ldots, T\}$.

$N_7(\sigma) = \{\sigma' \mid \sigma'$ is obtained by reducing some $y_M(k,t)$ variables (and perhaps shifting the set-up for manufacturing $z_M(k,t)$ in one period from 1 to 0) and shifting the set-up for remanufacturing $z_R(k,t)$ in one period from 0 to 1 (or increase that $x_R(k,t)$ variable), by reducing some $y_R(k,t)$ variables, for some $k = 1, 2, \ldots, K$, $t = 1, 2, \ldots, T\}$.

$N_8(\sigma) = \{\sigma' \mid \sigma'$ is obtained in case where between two periods with $z_M(k,t)$ values equal to 1, we reduce as much as possible some $y_M(k,t)$ variables (and perhaps shift the set-up for manufacturing $z_M(k,t)$ in one period from 1 to 0) and increase some $x_M(k,t)$ variables, for some $k = 1, 2, \ldots, K$, $t = 1, 2, \ldots, T\}$.

4. VND heuristic algorithm

The pseudocode of the proposed VND heuristic algorithm (`VND`) is as follows.

Algorithm 1 VND

```
1: procedure VND(K, T, R, D, h_R, h_M, k_R, k_M, p_R, p_M, σ, k_max)
2:     repeat
3:         improvement ← 0
4:         for k ← 1, k_max do
5:             for i ← 1, K do
6:                 for t ← 1, T do
7:                     Find the best neighbor σ' of σ (σ' ∈ N_k(σ))
8:                     if the obtained solution σ' is better than σ then
9:                         Set σ ← σ'
10:                        Set improvement ← 1
11:                    end if
12:                end for
13:            end for
14:        end for
15:    until improvement = 0
16: end procedure
```

4.1. Constructive Heuristic

A simple constructive heuristic has been implemented, where the total demand is fulfilled by a single lot (without remanufacturing units) for each product in the first period, in order to find an initial solution to the MDLSRP.

Algorithm 2 Heuristic initialization method

```
1: procedure HEURISTIC_START(K, T, R, D, x_R, x_M, y_R, y_M, z_R, z_M)
2:     x_R(:,:), z_R(:,:), x_M(:,:), z_M(:,:), y_R(:,:), y_M(:,:) ← 0
3:     y_R(:,1) ← R(:,1)
4:     for i ← 1, K do
5:         for j = 2, T do
6:             y_R(i,j) ← y_R(i,j−1) + R(i,j)
7:             y_M(i,T+1−j) ← y_M(i,T+2−j) + D(i,T+2−j)
8:         end for
9:     end for
10:    x_M(:,1) ← y_M(:,1) + D(:,1)
11:    z_M(:,1) ← 1
12: end procedure
```

5. Numerical testing

5.1. New benchmark set

Our VND heuristic algorithm was applied on a new set of benchmark problems with 300 products, 52 periods, and varying parameters such as $h_R(k) = \{0.2, 0.5, 0.8\}$, $k_R(k) = \{200, 500, 2000\}$, and $k_M(k) = \{200, 500, 2000\}$. We extend the previous work of Sifaleras and Konstantaras [35], and compare our VND algorithm not only with the 27 instances presented in [35], but also with other 135 instances; thus, totally 162 different instances. More specifically, these 108 out of these new 135 instances have been generated using the 27 combinations of the $h_R(k)$, $k_R(k)$, and $k_M(k)$ values for four different data sets of demand and return values (thus $27 \times 4 = 108$) following a normal distribution with small or large variance. In a similar way, the remaining 27 out of these new 135 instances have been generated again using the 27 combinations of the $h_R(k)$, $k_R(k)$, and $k_M(k)$ values but for a different data set of demand and return values following a uniform distribution.

The means and the variances of the normal and uniform distribution regarding the demands and returns, assume three different values per hundred products. The exact configuration of the new 135 test problems is reported in Table 1. For example, the demand values of the first 100 products, for a time period t with small variance, were drawn from a normal distribution $\mathcal{N}\left(\mu_D, \sigma_D^2\right)$ with $\mu_D = 50$, and $\sigma_D^2 = 4$. Finally, the demand and return values of the 27 instances presented in [35] followed uniform distributions $\mathcal{U}$(50,150) and $\mathcal{U}$(15,50), respectively, for all the products.

Moreover, similar to the data set presented in [32], the production and remanufacturing costs $p_M(k)$ and $p_R(k)$, respectively, are assumed to be equal and therefore not crucial (thus, $p_M(k) = p_R(k) = 0$). It is noteworthy, that this new benchmark set is the larger that have ever been used in the literature, for the MDLSRP problem. Each one of the 162 instances consists of 93600 variables (31200 binary and 62400 linear variables), 858000 linear constraints, and 1 linear objective.

This new benchmark set is publicly available from the authors web site: `http://users.uom.gr/~sifalera/benchmarks.html`. Interested readers may

Table 1: Parameter values for the test problems

	Products		
	1 - 100	101 - 200	201 - 300
Normal demands with small variance	$\mathcal{N}$(50,4)	$\mathcal{N}$(150,9)	$\mathcal{N}$(250,16)
Normal demands with large variance	$\mathcal{N}$(50,100)	$\mathcal{N}$(150,400)	$\mathcal{N}$(250,900)
Normal returns with small variance	$\mathcal{N}$(15,1)	$\mathcal{N}$(50,9)	$\mathcal{N}$(80,16)
Normal returns with large variance	$\mathcal{N}$(15,9)	$\mathcal{N}$(50,100)	$\mathcal{N}$(80,400)
Uniform demand	$\mathcal{U}$(20,60)	$\mathcal{U}$(100,200)	$\mathcal{U}$(200,400)
Uniform returns	$\mathcal{U}$(5,20)	$\mathcal{U}$(30,70)	$\mathcal{U}$(80,120)

find all the 162 instances, with all the presented results. Regular updates will be made, once new optimal solutions or new best feasible solutions are found, in the future.

5.2. Computing environment

We ran the experiments on a computer running Windows 8.1 64Bit with an Intel Core i7 4770K CPU at 3.5 GHz with 8 MB L3 cache and 32 GB DDR3 1600MHz main memory. The VND implementation was implemented in Fortran and compiled using the Intel Fortran 64 compiler XE v.15.0.0118.12. Efforts were made to solve each one of them using the latest version of the state-of-the-art Gurobi optimizer v6 within a reasonable amount of time set to 2 hours. The mathematical model of MDLSRP has been implemented using the AMPL mathematical programming language [13]. We should note that, the computational overhead added by AMPL was found to be quite small compared to the solver solution time for our MIP problem, for each instance. The time limit (stopping condition) of our VND approach was 90 seconds.

Moreover, one further technical improvement over the recent work by Sifaleras and Konstantaras [35], was to store all the floating-point parameters ($h_R(k)$) as integer (by simply multiplying with ten). In a similar way, all the other (already integer) parameters were multiplied by ten. Thus, the best objective value at the end was divided by ten, in order to return the correct result. By expressing all numbers in an integer arithmetic expression as integer values we avoid mixing integer and floating-point (real) data in the same computation (update of the objective value at each local search step). This way, we eliminate the need to convert data between fixed and floating-point formats, make easier the auto-vectorization of the innermost loops for the compiler, and thus improve the overall run-time performance [19].

5.3. Experimental results

Due to the increased computational difficulty, Gurobi was not able to solve any instance to optimality. Thus, Gurobi has only computed an upper bound of the optimal objective value. Therefore, the following Tables 2, 3, and 4 report the comparative numerical results regarding the incumbent solutions found by Gurobi (time limit = 7200 secs) and the proposed VND implementation (time

limit = 90 secs), for the instances with demand and return values generated from normal distribution, uniform distribution, and the instances presented in [35], respectively. By `zValue` we denote the best solutions found either by VND or Gurobi. Also, by `Bbound` we denote the lower (linear relaxation) bound computed by Gurobi.

We also report the relative MIP gap (difference between the upper and lower bounds) computed by Gurobi, denoted as `relmipgap`. Furthermore, also due to the increased computational difficulty, Gurobi was not able to compute small gaps. Thus, we also report the percent deviation of the best solution computed by VND from the lower bound computed by Gurobi too, (using again the same notation `relmipgap`).

In cases where the optimal solution is not available, an effective measure of the quality of solution is the percent deviation from a tight lower bound [4]. As it can been seen in the following Tables, although the average gap of VND over all 162 instances is about 20.97% and thus is probably not tight enough to convey useful information, it is quite smaller than the corresponding gap of Gurobi (46.08%). Also, the best objective value obtain is displayed in bold fonts. Moreover, the absolute percentage error of Gurobi against VND is reported, regarding the instances where VND performed better, and is denoted by `error`.

Table 2: Results on instances with demand and return values generated from normal distribution

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
1	**3270282.60**	15.86	3916136.80	2751611.18	29.74	19.75
2	**3495448.50**	13.77	4275150.00	3013973.34	29.50	22.31
3	**3748498.80**	13.09	4604289.60	3257954.05	29.24	22.83
4	**4027173.00**	11.43	5713836.80	3566765.72	37.58	41.88
5	**4479928.00**	8.18	6082050.00	4113522.69	32.37	35.76
6	**4875797.60**	8.04	6413589.60	4483927.76	30.09	31.54
7	**5796958.80**	13.99	7753346.40	4985873.71	35.69	33.75
8	**6984412.00**	7.75	12408648.00	6443427.89	48.07	77.66
9	**8081856.80**	8.08	14167853.40	7428926.36	47.56	75.30
10	**4858495.80**	21.23	7550736.80	3827258.18	49.31	55.41
11	**5089078.50**	20.77	7874550.00	4032199.72	48.79	54.73
12	**5278825.40**	20.79	8950645.40	4181134.06	53.29	69.56
13	**5892350.80**	16.81	9344236.80	4902044.72	47.54	58.58
14	**6321038.50**	15.96	9675950.00	5312252.41	45.10	53.08
15	**6704859.60**	15.94	9994089.60	5636287.28	43.60	49.06
16	**7414346.80**	10.65	16299398.40	6624871.63	59.36	119.84
17	**8742185.00**	8.54	18715450.00	7995556.46	57.28	114.08
18	**9880458.40**	9.52	19040589.60	8940204.86	53.05	92.71
19	**8930651.80**	44.83	23144059.93	4927459.77	78.71	159.15
20	**9380007.00**	47.49	9597074.87	4925833.88	48.67	2.31

Table 2: Results on instances with demand and return values generated from normal distribution

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
21	**9814944.80**	49.14	10036424.07	4992066.19	50.26	2.26
22	**10142751.40**	39.33	10275843.99	6153591.66	40.12	1.31
23	**10621326.00**	35.20	10853866.60	6882310.45	36.59	2.19
24	**11077982.20**	39.05	11470664.68	6752070.81	41.14	3.54
25	**12404914.80**	17.47	37897142.60	10237387.66	72.99	205.50
26	**13681913.00**	20.25	38449994.00	10911996.90	71.62	181.03
27	**14802946.00**	22.31	39002845.40	11499697.48	70.52	163.48
28	**3194939.60**	13.93	4127351.80	2749978.99	33.37	29.18
29	**3470471.50**	13.28	4283751.50	3009449.98	29.75	23.43
30	**3675663.00**	11.81	4614318.20	3241618.79	29.75	25.54
31	**3989579.00**	10.78	5705125.80	3559415.71	37.61	43.00
32	**4492250.00**	8.90	6099551.50	4092451.40	32.91	35.78
33	**4880649.80**	8.53	7171544.20	4464551.75	37.75	46.94
34	**5828545.20**	14.39	7761562.20	4989875.83	35.71	33.16
35	**7115807.50**	9.67	12698158.00	6427657.99	49.38	78.45
36	**8110102.40**	8.88	13481166.80	7389953.82	45.18	66.23
37	**4837334.80**	20.86	7530225.80	3828363.51	49.16	55.67
38	**5086428.00**	20.82	7865651.50	4027308.47	48.80	54.64
39	5295171.40	18.84	**5219774.88**	4297512.64	17.67	–
40	**5824516.80**	15.68	9750851.80	4911423.51	49.63	67.41
41	**6339034.00**	16.07	9687551.50	5320324.10	45.08	52.82
42	**6710371.40**	16.10	10884644.20	5629725.90	48.28	62.21
43	**7442889.60**	10.95	16665116.80	6627615.91	60.23	123.91
44	**8718339.00**	8.54	18750051.50	7973737.81	57.47	115.06
45	**9785732.00**	9.06	19143318.20	8898716.43	53.52	95.62
46	**8952738.60**	44.95	9626643.32	4928576.98	48.80	7.53
47	**9389338.50**	47.81	18493894.44	4900641.02	73.50	96.97
48	**9841728.00**	49.43	10053739.98	4976531.79	50.50	2.15
49	**10120022.20**	39.26	28316351.80	6146913.40	78.29	179.81
50	**10617420.50**	39.32	11050058.18	6442147.55	41.70	4.07
51	**11055554.00**	39.00	11522111.45	6743961.87	41.47	4.22
52	**12410460.80**	17.50	36378025.80	10238789.39	71.85	193.12
53	**13644827.50**	20.03	36658551.50	10912354.50	70.23	168.66
54	**14678743.20**	21.70	39002144.20	11494086.84	70.53	165.70
55	**3276712.20**	16.25	3911774.00	2744207.47	29.85	19.38
56	**3521771.50**	14.55	4276493.50	3009213.81	29.63	21.43
57	**3734584.00**	13.16	4603446.60	3242930.10	29.55	23.27
58	**4030910.60**	12.01	5680872.20	3546811.13	37.57	40.93
59	**4505687.50**	9.06	6100593.50	4097602.71	32.83	35.40
60	**4890417.00**	8.46	6432246.60	4476584.65	30.40	31.53
61	**5719050.40**	13.10	7332649.80	4969952.88	32.22	28.21

Table 2: Results on instances with demand and return values generated from normal distribution

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
62	**7002678.50**	8.27	12120281.50	6423214.83	47.00	73.08
63	**8089596.80**	8.39	15576246.60	7411109.82	52.42	92.55
64	4858676.40	21.22	**4544955.99**	3827578.26	15.78	–
65	**5085849.50**	20.95	7834693.50	4020594.57	48.68	54.05
66	5275865.60	18.58	**5207205.43**	4295775.09	17.50	–
67	**5854389.60**	16.57	9316874.00	4884314.22	47.58	59.14
68	**6327424.50**	16.14	10292310.50	5305968.84	48.45	62.66
69	**6704104.20**	15.97	9971346.60	5633549.98	43.50	48.73
70	**7414142.40**	11.11	12564202.00	6590149.73	47.55	69.46
71	**8724241.50**	8.72	17879547.50	7963815.84	55.46	104.94
72	**9846869.80**	9.54	19115346.60	8907956.91	53.40	94.13
73	**8917010.20**	44.65	19432469.56	4935465.41	74.60	117.93
74	**9398291.50**	47.51	9527063.61	4933201.98	48.22	1.37
75	**9838522.00**	49.19	10001054.01	4999177.97	50.01	1.65
76	**10116626.20**	39.24	28153598.00	6146840.40	78.17	178.29
77	**10619852.00**	39.39	10974273.74	6436601.62	41.35	3.34
78	**11089695.80**	39.13	11535117.10	6750144.41	41.48	4.02
79	**12375946.80**	17.52	36361374.00	10207641.42	71.93	193.81
80	**13634882.50**	20.22	36562493.50	10877998.18	70.25	168.15
81	**14777131.20**	22.40	36810846.60	11467230.73	68.85	149.11
82	**3214135.80**	14.72	4125014.20	2741159.76	33.55	28.34
83	**3477940.00**	13.69	4280574.00	3001835.30	29.87	23.08
84	**3675077.60**	12.16	4615681.80	3228273.61	30.06	25.59
85	**3982416.40**	11.08	5495300.40	3540993.72	35.56	37.99
86	**4505007.50**	9.48	6513580.50	4077989.89	37.39	44.59
87	**4893901.60**	9.00	6462481.80	4453592.70	31.09	32.05
88	**5721915.80**	13.09	7332866.60	4972939.82	32.18	28.15
89	**7067795.00**	9.34	12438695.00	6407469.90	48.49	75.99
90	**8095371.80**	8.93	12308099.40	7372356.21	40.10	52.04
91	4832667.00	20.91	**4563827.73**	3822370.00	16.25	–
92	5084756.50	19.79	**4934202.57**	4078625.67	17.34	–
93	5300186.00	19.19	**5212352.22**	4283024.34	17.83	–
94	**5815482.80**	16.01	9340498.40	4884337.25	47.71	60.61
95	**6319621.50**	16.09	9684674.00	5302811.55	45.25	53.25
96	**6690277.40**	15.93	10879829.80	5624718.03	48.30	62.62
97	**7413442.20**	11.08	13918887.20	6592056.93	52.64	87.75
98	**8697765.00**	8.73	19352661.50	7938441.74	58.98	122.50
99	**9768606.20**	9.25	18891049.20	8864886.14	53.07	93.39
100	**8930673.60**	44.78	23574988.00	4931201.23	79.08	163.98
101	**9383498.00**	47.46	9533880.55	4929907.33	48.29	1.60
102	**9828182.60**	49.21	15787242.05	4991547.15	68.38	60.63

Table 2: Results on instances with demand and return values generated from normal distribution

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
103	**10121253.40**	39.15	28149914.20	6158943.82	78.12	178.13
104	**10602131.00**	39.24	10909558.83	6441933.08	40.95	2.90
105	**11067832.00**	39.06	15104652.45	6745198.84	55.34	36.47
106	**12396198.40**	17.66	37835414.20	10206951.35	73.02	205.22
107	**13606139.00**	20.05	36627174.00	10877478.35	70.30	169.20
108	**14660716.80**	21.82	36927281.80	11462234.46	68.96	151.88

Table 3: Results on instances with demand and return values generated from uniform distribution

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
1	**3288156.20**	18.29	3925439.00	2686625.70	31.56	19.38
2	**3579426.00**	17.23	4336282.00	2962725.66	31.68	21.14
3	**3757871.60**	15.05	5413927.20	3192432.84	41.03	44.07
4	4021857.80	12.48	**3625067.20**	3519825.51	2.90	–
5	**4612075.50**	11.85	5749124.50	4065559.29	29.28	24.65
6	**5030035.00**	11.38	6642878.20	4457766.28	32.89	32.06
7	**5772314.20**	12.12	7276670.60	5072689.71	30.29	26.06
8	**7183598.50**	9.18	13077691.00	6524328.51	50.11	82.05
9	**8229016.00**	8.61	11033798.60	7520347.41	31.84	34.08
10	4762549.00	18.60	**4444640.60**	3876796.92	12.78	–
11	**5007422.50**	19.37	7823182.00	4037259.52	48.39	56.23
12	5259170.00	18.32	**5203010.45**	4295552.36	17.44	–
13	**5743878.00**	14.91	9289939.00	4887606.16	47.39	61.74
14	**6321740.50**	16.00	9671882.00	5310448.59	45.09	52.99
15	**6749591.00**	16.11	10025290.40	5661914.74	43.52	48.53
16	**7511577.20**	10.35	11618945.00	6733785.14	42.04	54.68
17	**8880716.50**	8.99	18112842.00	8081948.52	55.38	103.96
18	**9955046.20**	9.26	19402233.40	9032735.77	53.44	94.90
19	**8917528.00**	42.56	25765441.80	5121874.69	80.12	188.93
20	**9382673.50**	44.67	14033145.99	5191557.15	63.01	49.56
21	**9845523.60**	49.45	9965982.36	4976535.25	50.06	1.22
22	**10168450.40**	37.46	27724741.80	6358855.83	77.06	172.65
23	**10662331.00**	37.34	10969001.37	6680497.17	39.10	2.88
24	**11139763.80**	37.33	11481404.01	6980959.42	39.20	3.07
25	**12569296.40**	17.10	36131439.00	10420263.34	71.16	187.46
26	**13906463.00**	20.27	36324882.00	11088285.32	69.47	161.21
27	**14939059.40**	21.82	36623290.40	11679412.89	68.11	145.15

Table 4: Results on instances presented in [35]

	VND		Gurobi			
ID	zValue	relmipgap (%)	zValue	Bbound	relmipgap (%)	error (%)
1	**2942610.20**	15.70	3977436.40	2480736.74	37.63	35.17
2	**3187454.00**	14.77	4038449.00	2716802.94	32.73	26.70
3	3375163.40	13.01	**3122836.00**	2936149.44	5.98	–
4	**3543668.40**	11.00	3895702.40	3153870.31	19.04	9.93
5	**4145741.00**	12.50	4519693.00	3627720.99	19.74	9.02
6	4543054.20	12.50	**4096972.00**	3975376.38	2.97	–
7	**4804402.40**	13.36	5242669.80	4162370.73	20.61	9.12
8	**5988207.00**	10.21	8792123.00	5377036.51	38.84	46.82
9	**6857867.20**	6.98	11832959.40	6379457.34	46.09	72.55
10	4337867.60	25.10	**4178652.27**	3249069.76	22.25	–
11	4571979.00	24.65	**4545301.49**	3444982.99	24.21	–
12	**4768544.00**	24.40	4869789.93	3605137.53	25.97	2.12
13	**5041837.00**	15.40	9148126.00	4265598.59	53.37	81.44
14	**5623766.00**	18.65	9365849.00	4575053.38	51.15	66.54
15	**6036566.00**	19.85	10366900.60	4838270.26	53.33	71.74
16	**6445542.80**	12.71	9663935.60	5626033.90	41.78	49.93
17	**7508623.00**	9.97	13796597.00	6760349.33	51.00	83.74
18	**8386224.00**	8.49	17538763.60	7674545.49	56.24	109.14
19	**7780606.60**	50.37	21854884.80	3861522.54	82.33	180.89
20	**8138222.50**	50.88	8411016.62	3997882.70	52.47	3.35
21	**8495880.00**	51.11	12036073.61	4153391.20	65.49	41.67
22	**8747360.20**	41.40	9140264.75	5125944.17	43.92	4.49
23	**9256106.00**	42.40	9756145.50	5331166.67	45.36	5.40
24	**9682844.00**	43.05	10208624.62	5513962.59	45.99	5.43
25	**10635183.60**	18.59	35901626.00	8658154.32	75.88	237.57
26	**11702127.50**	21.16	35923849.00	9226160.02	74.32	206.99
27	**12562994.20**	22.53	36041650.20	9732634.49	73.00	186.89

5.4. Discussion

As it was shown in the previous Tables 2, 3, and 4, the proposed VND heuristic approach outperformed Gurobi and found better solutions in the 149 out of the 162 instances (91.98%). Furthermore, the VND method required approximately 80 times less computational time for reaching these solutions. Thus, one can conclude that the VND approach is the best method for computing good quality solutions for large MDLSRP instances in a small amount of time.

The ratio of time to produce a solution within 5 percent of the best-found solution value to the time to produce the best was equal to $r_{0.05} = 0.208$ over all 162 instances. Following, we report the rate at which our VND heuristic converges to a solution close in value to that of the "best found" solution. Figure 1 depicts the tradeoff between the quality of solution versus the time expended, for all data sets.

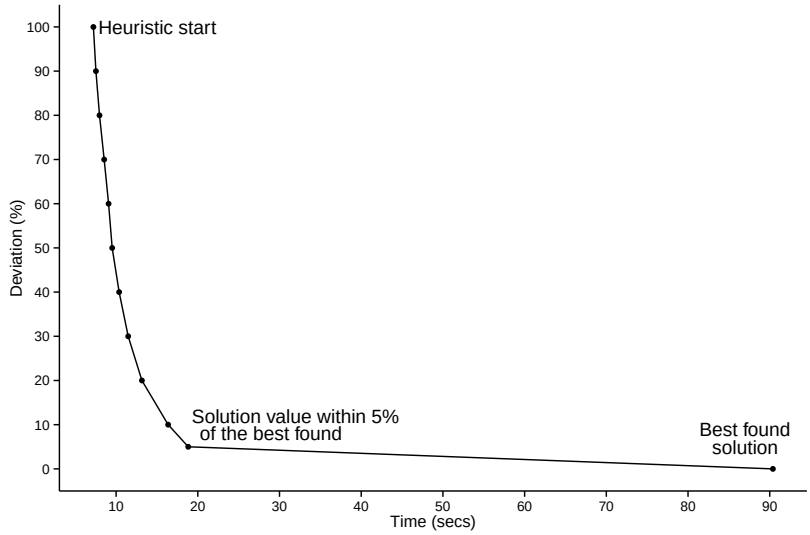

Figure 1: Average quality percent deviation from best found solutions versus computational effort.

The percentage cost error of Gurobi against the proposed VND, over all test instances, for different levels of manufacturing and remanufacturing setup costs as also for different levels of holding cost, is depicted in Table 5. One can note that the test instances with higher manufacturing and remanufacturing setup costs were more difficult for the exact solver to tackle; a finding that was also verified by the recent studies of by Sifaleras et al. [36] and Cunha and Melo [9].

6. Conclusions and future work

In this paper we have addressed the multi-product dynamic lot sizing problem with product returns and recovery in reverse logistics. We have proposed

Table 5: Percentage cost errors for different levels of manufacturing, remanufacturing setup costs and holding cost.

	h_R		
	0.2	0.5	0.8
Average (%)	78.01	59.08	52.01
	k_M		
	200	500	2000
Average (%)	35.59	58.78	94.73
	k_R		
	200	500	2000
Average (%)	34.77	44.05	110.28

a new VND heuristic algorithm for exploring eight neighborhoods for this hard combinatorial optimization problem. Based on extended numerical testing using a new set of benchmark problems with 162 instances, our approach was able to outperform Gurobi on large problems with 300 products and 52 periods. Also, the proposed solution algorithm is a self-contained solver without any link to other callable library API. Thus our approach does not depend on any other, commercial or not, solver for either computing a starting solution or an intermediate computation.

Moreover, since Gurobi using the proposed formulation was not able to solve the proposed instances, it would be interesting to use reformulations for the MDLSRP, similar to the recently proposed reformulation and new valid inequalities suggested by Cunha and Melo [9] for the single-product ELSRP. As it was shown by Cunha and Melo [9], the proposed new valid inequalities achieved very good bounds and allowed the solution of several ELSRP instances (some of them could not be solved otherwise) to optimality.

A subject for a future work is the adaptation of VNS to other combined problems of inventory and network optimization [24, 34]. Finally, the parallel implementation of our proposed VND method using, e.g., MPI [10] or CUDA Fortran consists also a future goal.

References

[1] E. Akçalı, S. Çetinkaya, Quantitative models for inventory and production planning in closed-loop supply chains, International Journal of Production Research 49 (2011) 2373–2407.

[2] B. Almada-Lobo, R.J. James, Neighbourhood search meta-heuristics for capacitated lot-sizing with sequence-dependent setups, International Journal of Production Research 48 (2010) 861–878.

[3] B. Almada-Lobo, J.F. Oliveira, M.A. Carravilla, Production planning and scheduling in the glass container industry: A VNS approach, International Journal of Production Economics 114 (2008) 363–375.

[4] R.S. Barr, B.L. Golden, J.P. Kelly, M.G. Resende, W.R. Stewart Jr, Designing and reporting on computational experiments with heuristic methods, Journal of Heuristics 1 (1995) 9–32.

[5] N. Brahimi, S. Dauzere-Peres, N.M. Najid, A. Nordli, Single item lot sizing problems, European Journal of Operational Research 168 (2006) 1–16.

[6] M. Brandenburg, K. Govindan, J. Sarkis, S. Seuring, Quantitative models for sustainable supply chain management: Developments and directions, European Journal of Operational Research 233 (2014) 299–312.

[7] L. Buschkühl, F. Sahling, S. Helber, H. Tempelmeier, Dynamic capacitated lot-sizing problems: a classification and review of solution approaches, OR Spectrum 32 (2010) 231–261.

[8] M.A. Bushuev, A. Guiffrida, M. Jaber, M. Khan, A review of inventory lot sizing review papers, Management Research Review 38 (2015) 283–298.

[9] J. Cunha, R. Melo, A computational comparison of reformulations for the economic lot-sizing with remanufacturing, 2015. Working paper.

[10] T. Davidović, T.G. Crainic, MPI parallelization of variable neighborhood search, Electronic Notes in Discrete Mathematics 39 (2012) 241–248.

[11] B. Fahimnia, J. Sarkis, H. Davarzani, Green supply chain management: A review and bibliometric analysis, International Journal of Production Economics 162 (2015) 101–114.

[12] G. Figueira, M.O. Santos, B. Almada-Lobo, A hybrid VNS approach for the short-term production planning and scheduling: A case study in the pulp and paper industry, Computers & Operations Research 40 (2013) 1804–1818.

[13] R. Fourer, D. Gay, B. Kernighan, AMPL: A Modeling Language for Mathematical Programming, 2nd ed., Duxbury Press, 2002.

[14] B. Golany, J. Yang, G. Yu, Economic lot-sizing with remanufacturing options, IIE Transactions 33 (2001) 995–1003.

[15] K. Govindan, T. Cheng, Sustainable supply chain management: Advances in operations research perspective, Computers & Operations Research 54 (2015) 177–179.

[16] K. Govindan, H. Soleimani, D. Kannan, Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future, European Journal of Operational Research 240 (2015) 603–626.

[17] V.D.R. Guide Jr, L.N. Van Wassenhove, OR FORUM - the evolution of closed-loop supply chain research, Operations Research 57 (2009) 10–18.

[18] P. Hansen, N. Mladenović, J.A. Moreno Pérez, Variable neighbourhood search: methods and applications, Annals of Operations Research 175 (2010) 367–407.

[19] Intel © Corporation, Tuning performance, User and Reference Guide for the Intel © Fortran Compiler 15.0, 2015. URL: `https://software.intel.com/en-us/node/525271`, (last time accessed on March 26, 2015).

[20] X. Li, F. Baki, P. Tian, B.A. Chaouch, A robust block-chain based tabu search algorithm for the dynamic lot sizing problem with product returns and remanufacturing, Omega 42 (2014) 75–87.

[21] Y. Li, J. Chen, X. Cai, Uncapacitated production planning with multiple product types, returned product remanufacturing, and demand substitution, OR Spectrum 28 (2006) 101–125.

[22] Y. Li, J. Chen, X. Cai, Heuristic genetic algorithm for capacitated production planning problems with batch processing and remanufacturing, International Journal of Production Economics 105 (2007) 301–317.

[23] A. Mjirda, B. Jarboui, R. Macedo, S. Hanafi, A variable neighborhood search for the multi-product inventory routing problem, Electronic Notes in Discrete Mathematics 39 (2012) 91–98.

[24] A. Mjirda, B. Jarboui, R. Macedo, S. Hanafi, N. Mladenović, A two phase variable neighborhood search for the multi-product inventory routing problem, Computers & Operations Research 52, Part B (2014) 291–299.

[25] S.D. Morgan, R.J. Gagnon, A systematic literature review of remanufacturing scheduling, International Journal of Production Research 51 (2013) 4853–4879.

[26] E. Moustaki, K.E. Parsopoulos, I. Konstantaras, K. Skouri, I. Ganas, A first study of particle swarm optimization on the dynamic lot sizing problem with product returns, in: Proceedings of the XI Balkan Conference on Operational Research (BALCOR 2013), Belgrade-Zlatibor, Serbia, 2013, pp. 348–356.

[27] K. Parsopoulos, I. Konstantaras, K. Skouri, Metaheuristic optimization for the single-item dynamic lot sizing problem with returns and remanufacturing, Computers & Industrial Engineering 83 (2015) 307–315.

[28] P. Pineyro, O. Viera, Inventory policies for the economic lot-sizing problem with remanufacturing and final disposal options, Journal of Industrial and Management Optimization 5 (2009) 217–238.

[29] M.J. Retel Helmrich, R. Jans, W. van den Heuvel, A.P. Wagelmans, Economic lot-sizing with remanufacturing: complexity and efficient formulations, IIE Transactions 46 (2014) 67–86.

[30] K. Richter, M. Sombrutzki, Remanufacturing planning for the reverse Wagner/Whitin models, European Journal of Operational Research 121 (2000) 304–315.

[31] P. Robinson, A. Narayanan, F. Sahin, Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms, Omega 37 (2009) 3–15.

[32] F. Sahling, A column-generation approach for a short-term production planning problem in closed-loop supply chains, Business Research 6 (2013) 55–75.

[33] T. Schulz, A new silver–meal based heuristic for the single–item dynamic lot sizing problem with returns and remanufacturing, International Journal of Production Research 49 (2011) 2519–2533.

[34] A. Sifaleras, Minimum cost network flows: Problems, algorithms, and software, Yugoslav Journal of Operations Research 23 (2013) 3–17.

[35] A. Sifaleras, I. Konstantaras, General variable neighborhood search for the multi-product dynamic lot sizing problem in closed-loop supply chain, Electronic Notes in Discrete Mathematics 47 (2015) 69–76.

[36] A. Sifaleras, I. Konstantaras, N. Mladenović, Variable neighborhood search for the economic lot sizing problem with product returns and recovery, International Journal of Production Economics 160 (2015) 133–143.

[37] G.C. Souza, Closed-loop supply chains: A critical review, and future research, Decision Sciences 44 (2013) 7–38.

[38] S.K. Srivastava, Green supply-chain management: a state-of-the-art literature review, International journal of management reviews 9 (2007) 53–80.

[39] R.H. Teunter, Z.P. Bayindir, W. Van den Heuvel, Dynamic lot sizing with product returns and remanufacturing, International Journal of Production Research 44 (2006) 4377–4400.

[40] H. Ullah, S. Parveen, A literature review on inventory lot sizing problems, Global Journal of Researches in Engineering 10 (2010) 21–36.

[41] H.M. Wagner, T.M. Whitin, Dynamic version of the economic lot size model, Management Science 5 (1958) 88–96.

[42] Y. Xiao, I. Kaku, Q. Zhao, R. Zhang, A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems, European Journal of Operational Research 214 (2011) 223–231.

[43] Y. Xiao, I. Kaku, Q. Zhao, R. Zhang, A variable neighborhood search based approach for uncapacitated multilevel lot-sizing problems, Computers and Industrial Engineering 60 (2011) 218–227.

[44] Y. Xiao, R. Zhang, Q. Zhao, I. Kaku, Y. Xu, A variable neighborhood search with an effective local search for uncapacitated multilevel lot-sizing problems, European Journal of Operational Research 235 (2014) 102–114.