Reduction from the Partition Problem: Dynamic Lot Sizing Problem with Polynomial Complexity

Chee-Khian Sim[*]
School of Mathematics and Physics
University of Portsmouth
Lion Gate Building, Lion Terrace
Portsmouth PO1 3HF

Last updated: 06 December 2024

Abstract

In this note, we reduce an instance of the partition problem to a dynamic lot sizing problem in polynomial time, and show that solving the latter problem solves the former problem. We further show that the instance of the partition problem can be solved using polynomial number of addition, multiplication and sort operations in input data using the reduction.

Keywords. Partition problem; dynamic lot sizing model; polynomial complexity.

1 Notations

Let $f(N)$ and $g(N)$ be two nonnegative real-valued functions, where $N \in \mathbb{Z}^{++}$. We write $g(N) = O(f(N))$ to mean that $g(N) \leq Cf(N)$ for some positive constant C and all $N > 0$.

Furthermore, $\mathbb{I}(x)$, where $x \in \mathbb{Z}$, is defined to be 1 for $x > 0$, and 0 otherwise; and $x^{+} = \max\{x, 0\}$, where $x \in \Re$.

2 The Partition Problem and its Reduction to a Dynamic Lot Sizing Problem

Let $S = \{1, \ldots, n\}$ and $a_i \in \mathbb{Z}^{++}$ for $i \in S$, with $\sum_{i \in S} a_i = 2C$. The partition problem is to find a subset A of S such that

$$\sum_{i \in A} a_i = \sum_{i \in S \setminus A} a_i = C.$$

It is known that the partition problem is NP-complete [1, 2, 3]. We call an instance of the partition problem **PPi**.

---

[*]Email address: chee-khian.sim@port.ac.uk

In this section, we reduce **PPi** to a dynamic lot sizing problem in polynomial time and show that solving the dynamic lot sizing problem solves **PPi** in Theorem 2.2. As a consequence, if the dynamic lot sizing problem can be solved with polynomial complexity, **PPi** can also be solved with polynomial complexity.

Dynamic lot sizing problem is introduced in [5], and has since been studied intensively by researchers. We consider a variant of this basic problem which is related to remanufacturing.

In the following, we list down the parameters of the dynamic lot sizing model that we are considering in this note:

Parameters:

- N = number of periods in the time horizon, where $N \geq 1$;

- D_i = demand for serviceable products during the ith period, where $i = 1, \ldots, N$. We assume that $D_i \in \mathbb{Z}^{++}, i = 1, \ldots, N$;

- R_i = returned products as cores at the beginning of the ith period, where $i = 1, \ldots, N$. We assume that $R_i \in \mathbb{Z}^+, i = 1, \ldots, N$, and set $R_{N+1} = 0$;

- $K_{r,i}$ = setup cost when there is remanufacturing at the beginning of the ith period, where $i = 1, \ldots, N$. We let $K_{r,i} \geq 0$ for all $i = 1, \ldots, N$;

- $\Delta K_{m,i}$ = setup cost when there is manufacturing at the beginning of the ith period, where $i = 1, \ldots, N$. We let $\Delta K_{m,i} \geq 0$ for all $i = 1, \ldots, N$;

- $h_{s,i}$ = unit holding cost of serviceable product over the ith period whether from manufacturing or remanufacturing, where $i = 1, \ldots, N$. We let $h_{s,i} \geq 0$ for all $i = 1, \ldots, N$;

- $h_{c,i}$ = unit holding cost of core over the ith period, where $i = 1, \ldots, N$. We let $h_{c,i} \geq 0$ for all $i = 1, \ldots, N$;

- $c_{r,i}$ = unit remanufacturing cost in the ith period, where $i = 1, \ldots, N$. We let $c_{r,i} \geq 0$ for all $i = 2, \ldots, N$;

- $c_{m,i}$ = unit manufacturing cost in the ith period, where $i = 1, \ldots, N$. We let $c_{m,i} \geq 0$ for all $i = 1, \ldots, N$.

We further impose assumptions on the above parameters as follows:

Assumption 2.1   *(a)* $c_{r,i} < c_{m,i}$ *for* $i = 1, \ldots, N$;

  (b) $h_{s,i} > h_{c,i} + K_{r,j}$ *for* $i, j = 1, \ldots, N$, *and* $j \geq i$;

  (c) $\Delta K_{m,i} \geq K_{r,i}$ *for* $i = 1, \ldots, N$;

  (d) $h_{s,i} + c_{m,i} > \Delta K_{m,j} + c_{m,j}$ *for* $i < j$.

These assumptions are crucial to prove Proposition 3.1, which in turn is needed to formulate the dynamic program for the dynamic lot sizing problem we are considering in this note, and to solve it efficiently.

Demand must be satisfied in each period in our model. Our objective for the model is to minimize its total cost, which comprises of setup costs for manufacturing and remanufacturing,

holding costs for serviceable products and cores, manufacturing and remanufacturing costs. We have the following sequence of events in our model - at the beginning of a period, (i) returned products arrive as cores; (ii) number of units of serviceable products to produce through remanufacturing and manufacturing is determined; (iii) demand in the period is satisfied; (iv) any leftover cores and serviceable products are held to the next period.

The dynamic lot sizing problem (**DLSP**) we are considering is given by:

$$\min \sum_{i=1}^{N} (K_{r,i}\mathbb{I}(x_i) + \Delta K_{m,i}\mathbb{I}(y_i) + c_{r,i}x_i + c_{m,i}y_i + h_{c,i}[J_i - x_i] + h_{s,i}I_{i+1})$$

subject to

$$J_{i+1} = J_i + R_{i+1} - x_i, \ i = 1, \ldots, N,$$
$$I_{i+1} = I_i + x_i + y_i - D_i, \ i = 1, \ldots, N,$$
$$x_i \leq J_i, \ i = 1, \ldots, N,$$
$$J_i, I_i \geq 0, \ i = 2, \ldots, N,$$
$$x_i, y_i \in \mathbb{Z}^+, \ i = 1, \ldots, N,$$
$$J_1 = R_1, I_1 = 0.$$

The decision variables in the above minimization problem are:

- x_i = number of units of cores remanufactured in the ith period;

- y_i = number of units of serviceable products obtained by manufacturing in the ith period,

while

- J_i = number of units of cores at the beginning of the ith period;

- I_i = number of units of available serviceable products at the beginning of the ith period.

The objective function in the above minimization problem is the total cost of the model. The first constraint tells us the number of units of cores available at the beginning of the $(i + 1)^{th}$ period, $i = 1, \ldots, N$, after events occurred in the ith period. The second constraint tells us the number of units of serviceable products available at the beginning of the $(i + 1)^{th}$ period, $i = 1, \ldots, N$, after events occurred in the ith period. The third constraint tells us that the number of cores remanufactured in the ith period cannot exceed the cores available in the period. The fourth constraint tells us that the number of units of cores and serviceable products at the beginning of the ith period are never negative. The next constraint is the sign constraint on the decision variables in the problem, while the last constraint sets specific values on J_1, I_1. Note that the parameters in **DLSP** satisfy Assumption 2.1.

Let us call the optimal value of the minimization problem C^*, and its optimal solution x_i^*, y_i^*, $i = 1, \ldots, N$, with $J_i^* = J_{i-1}^* + R_i - x_{i-1}^*, I_i^* = I_{i-1}^* + x_{i-1}^* + y_{i-1}^* - D_{i-1}, i = 2, \ldots, N+1$, $J_1^* = R_1, I_1^* = 0$.

We reduce **PPi** in polynomial time to the above dynamic lot sizing problem by setting appropriate values for the parameters of the model as follows:

- $N = n$;

- $D_i = a_i$, $i = 1, \ldots, N(=n)$;

- $R_1 = C$, $R_i = 0$, $i = 2, \ldots, N$;

- $K_{r,i} = \Delta K_{m,i} = 1$, $i = 1, \ldots, N$;

- $h_{s,i} = 3$, $i = 1, \ldots, N$;

- $h_{c,i} = 0$, $i = 1, \ldots, N$;

- $c_{r,i} = 0$, $i = 1, \ldots, N$;

- $c_{m,i} = 1$, $i = 1, \ldots, N$.

It is easy to check that parameters of the model with the above values satisfy Assumption 2.1. We call the dynamic lot sizing problem with these values for its parameters **DLSPp**, and this problem is a special case of **DLSP**. We have the following theorem:

Theorem 2.2 PPi *can be solved by solving* **DLSPp**.

Proof: *Claim 1:* Suppose there exists a subset A of $S = \{1, \ldots, n\}$ such that

$$\sum_{i \in A} a_i = \sum_{i \in S \setminus A} a_i = C,$$

then the optimal value to **DLSPp** is at most $N + C$.

It is easy to see that by remanufacturing D_i units of cores in the ith period, when $i \in A$, and manufacturing D_i units from raw materials in the ith period, when $i \notin A$, total cost is $N + C$, and it is feasible to **DLSPp**. Hence, the optimal value to **DLSPp** is at most $N + C$.

Claim 2: Suppose the optimal value to **DLSPp** is at most $N + C$, and we let A to contain elements $i \in S = \{1, \ldots, N\}$ such that we remanufacture in the ith period in **DLSPp**. Then we have

$$\sum_{i \in A} a_i = \sum_{i \in S \setminus A} a_i = C.$$

First note that under optimality, whenever we produce, we only produce enough to satisfy demand for the period, and do not hold serviceable products to the next period. To see this, suppose we hold a serviceable product to the next period, then a cost of $h_{s,i} = 3$ is incurred. If we do not produce the serviceable product in the current period, but in the next period, we do not incur the holding cost of $h_{s,i} = 3$ and may even save on its manufacturing cost if this product is obtained by manufacturing, but we incur a possible setup cost of 1 due to remanufacturing or manufacturing, and possible unit manufacturing cost $c_{m,i+1} = 1$. In the new setup, total cost is reduced by at least 1, but this contradicts optimality. Hence, under optimality, whenever we produce, we only produce enough to satisfy demand for the period, and do not hold serviceable products to the next period. It is easy to see that all $R_1 = C$ units of cores are remanufactured to satisfy demand since there is no cost for remanufacturing. Note that these cores need not be all remanufactured in the 1st period and they can be held to later periods for remanufacturing without incurring holding cost since $h_{c,i} = 0$. Now, total demand is $\sum_{i=1}^{N} D_i = \sum_{i=1}^{N} a_i = 2C$, and since half of these demands is satisfied through remanfacturing and that all demand has to be satisfied, the other half of these demands has to

be satisfied through manufacturing, incurring a total manufacturing cost of C, since $c_{m,i} = 1$. In each period, we always have manufacturing and/or remanufacturing to satisfy demand in the period, as we do not have serviceable products held from earlier periods to satisfy demand in the period. Total setup cost is then at least N. Hence, total cost is at least $N+C$. However, the optimal value to **DLSPp** is at most $N+C$. Therefore, under optimality, we must have total cost is exactly $N+C$, leading to total setup cost to be exactly N, and we either remanufacture or manufacture in a period. Claim 2 then follows. □

Note that **DLSPp** and Theorem 2.2 with the claims in its proof follow [4], while the proof of Claim 2 in the theorem is inspired by [4].

3 Dynamic Programming Formulation of Dynamic Lot Sizing Problem and its Efficient Solution

We propose a dynamic programming formulation of **DLSP** in this section. Before we do this, we state and prove the following proposition that is the key which allows us to have the formulation and then solving it efficiently.

Proposition 3.1 *In **DLSP**, suppose we produce in the ith period and we next produce in the jth period, where $j > i$. If $I_i^* + J_i^* \leq D_i + \ldots + D_{j-1}$, then $x_i^* = J_i^*$, $I_j^* = 0$ and $y_i^* = (D_i + \ldots + D_{j-1}) - (I_i^* + J_i^*)$. On the other hand, if $I_i^* + J_i^* \geq D_i + \ldots + D_{j-1}$, then $x_i^* = ((D_i + \ldots + D_{j-1}) - I_i^*)^+$, $I_j^* = (I_i^* - (D_i + \ldots + D_{j-1}))^+$ and $y_i^* = 0$.*

Proof: We first prove that if $I_i^* + J_i^* \leq D_i + \ldots + D_{j-1}$, then $x_i^* = J_i^*$. In order to satisfy demand in the ith period up to the $(j-1)^{th}$ period before we produce again, we need to produce at least $(D_i + \ldots + D_{j-1}) - I_i^*$ units of serviceable products either by remanufacturing from J_i^* units of cores or manufacturing from raw materials in the ith period. Since $c_{r,i} < c_{m,i}$ and $\Delta K_{m,i} \geq K_{r,i}$ by Assumption 2.1(a) and (c), and in order not to incur unnecessary holding costs for cores, under optimality, we remanufacture all available cores, thus $x_i^* = J_i^*$. Now, we prove that $I_i^* + J_i^* \leq D_i + \ldots + D_{j-1}$ implies $I_j^* = 0$ by assuming that $I_j^* > 0$ and show that this leads to a contradiction. Note that since we produce in the ith period and the jth period, we have $x_i^* + y_i^*, x_j^* + y_j^* \geq 1$, and we note that a setup cost of at least $\Delta K_{m,k}$ and at most $\Delta K_{m,k} + K_{r,k}$ is incurred in the kth period, $k = i$ or j. Since we assume that $I_j^* > 0$, we have $I_i^* + x_i^* + y_i^* > D_i + \ldots + D_{j-1}$. Hence, with $x_i^* = J_i^*$ and $I_i^* + J_i^* \leq D_i + \ldots + D_{j-1}$, it leads to $y_i^* \geq 1$. That is, there is manufacturing in the ith period incurring an additional setup cost of $\Delta K_{m,i}$ in the period. Observe that by reducing manufacturing in the ith period by 1 unit, we still satisfy demand in the periods from the ith period to the $(j-1)^{th}$ period, with cost reduced by at least $c_{m,i} + h_{s,i}$, and a further cost reduction of $\Delta K_{m,i}$ if we do not manufacture at all in the ith period, in the new setting. The unit of missing serviceable product produced at the ith period can be "reinstated" in the jth period either by remanufacturing or manufacturing incurring a cost of at most $\Delta K_{m,j} + c_{m,j}$, because $c_{r,j} < c_{m,j}$ and $\Delta K_{m,j} \geq K_{r,j}$ by Assumption 2.1(a), (c). In the new setting, total system cost is then reduced by at least $h_{s,i} + c_{m,i} - \Delta K_{m,j} - c_{m,j}$, which is positive by Assumption 2.1(d). This leads to a contradiction to optimality. We conclude then that $I_i^* + J_i^* \leq D_i + \ldots + D_{j-1}$ implies $I_j^* = 0$. Furthermore, it is easy to see that we have $y_i^* = (D_i + \ldots + D_{j-1}) - (I_i^* + J_i^*)$, since demand from the ith period up to the $(j-1)^{th}$ period have to be satisfied and $x_i^* = J_i^*$.
Next, we show that if $I_i^* + J_i^* \geq D_i + \ldots + D_{j-1}$, then $x_i^* = ((D_i + \ldots + D_{j-1}) - I_i^*)^+$. First, we note that since demand from the ith period up to the $(j-1)^{th}$ period have to be satisfied,

any excess demands not fulfilled by I_i^* units of serviceable products have to be satisfied from remanufacturing from the available J_i^* units of cores at the beginning of the ith period, and hence we have $x_i^* \geq ((D_i + \ldots + D_{j-1}) - I_i^*)^+$. If $x_i^* > ((D_i + \ldots + D_{j-1}) - I_i^*)^+$, then excess serviceable products remanufactured have to be held till the jth period incurring holding cost of $(x_i^* - ((D_i + \ldots + D_{j-1}) - I_i^*)^+) \sum_{k=i}^{j-1} h_{s,k}$, which is higher than the sum of the holding cost of cores of $(x_i^* - ((D_i + \ldots + D_{j-1}) - I_i^*)^+) \sum_{k=i}^{j-1} h_{c,k}$ (if these cores are not remanufactured in the ith period) and possible setup cost of $K_{r,j}$ (to remanufacture these cores to serviceable products in the jth period), since $h_{c,k} + K_{r,j} < h_{s,k}$ by Assumption 2.1(b), and this is a contradiction to optimality. Hence, we have $x_i^* = ((D_i + \ldots + D_{j-1}) - I_i^*)^+$. Furthermore, if $I_i^* + J_i^* \geq D_i + \ldots + D_{j-1}$, then $I_j^* = (I_i^* - (D_i + \ldots + D_{j-1}))^+$, since $x_i^* = ((D_i + \ldots + D_{j-1}) - I_i^*)^+$ and $y_i^* = 0$. Note that $y_i^* = 0$ since demand from the ith period up to the $(j-1)^{th}$ period are satisfied by available serviceable products, I_i^*, and those remanufactured in the ith period, and we do not want to manufacture additional serviceable products that have to be then held till the jth period incurring unnecessary holding cost of serviceable products. $\qquad\square$

The results in the above proposition have the "flavor" of the well-known zero-inventory property of the dynamic lot sizing problem, which first appeared in [5].

Remark 3.2 *Under optimality, it is easy to convince ourselves from the proof of Proposition 3.1 that if the ith period is the last period when we produce before the end of the time horizon, then results in the proposition still hold, where we let $j = N + 1$ in the proposition.*

Let us now consider a dynamic program which we use to solve **DLSP**.

For $i = 1, \ldots, N$, and $J_i, I_i \geq 0$,

$$C_i^{**}(J_i, I_i) \quad := \quad \min \left\{ C_i^{*-}(J_i, I_i), C_i^{*+}(J_i, I_i) \right\}, \tag{1}$$

where

$$
\begin{aligned}
&C_i^{*-}(J_i, I_i) \\
:= \quad &\min \Big\{ K_{r,i}\mathbb{I}((D_i + \ldots + D_{j-1}) - I_i) + c_{r,i}((D_i + \ldots + D_{j-1}) - I_i)^+ + \\
&\sum_{k=i}^{j-2}(h_{s,i} + \ldots + h_{s,k})D_{k+1} + (I_i - (D_i + \ldots + D_{j-1}))^+ \sum_{k=i}^{j-1} h_{s,k} + \\
&\sum_{k=i+1}^{j-1}(h_{c,k} + \ldots + h_{c,j-1})R_k + (J_i - ((D_i + \ldots + D_{j-1}) - I_i)^+) \sum_{k=i}^{j-1} h_{c,k} + \\
&C_j^{**}(J_i - ((D_i + \ldots + D_{j-1}) - I_i)^+ + R_{i+1} + \ldots + R_j, (I_i - (D_i + \ldots + D_{j-1}))^+) \ \Big| \\
&i + 1 \leq j \leq N+1, I_i + J_i \geq D_i + \ldots + D_{j-1} \Big\}, \tag{2}
\end{aligned}
$$

and

$$
\begin{aligned}
&C_i^{*+}(J_i, I_i) \\
:= \quad &\min \Big\{ K_{r,i}\mathbb{I}(J_i) + \Delta K_{m,i}\mathbb{I}((D_i + \ldots + D_{j-1}) - (I_i + J_i)) + c_{r,i}J_i + \\
&\sum_{k=i}^{j-2}(h_{s,i} + \ldots + h_{s,k})D_{k+1} + \sum_{k=i+1}^{j-1}(h_{c,k} + \ldots + h_{c,j-1})R_k + \\
&c_{m,i}[(D_i + \ldots D_{j-1}) - (I_i + J_i)] + C_j^{**}(R_{i+1} + \ldots + R_j, 0) \ \Big|
\end{aligned}
$$

6

$$i + 1 \leq j \leq N + 1, I_i + J_i \leq D_i + \ldots + D_{j-1}\} . \tag{3}$$

We have the convention that $C^*_{N+1}(J_{N+1}, I_{N+1}) = 0$ for all $J_{N+1}, I_{N+1} \geq 0$.

The following lemma relates the above dynamic program to **DLSP**:

Lemma 3.3 *We have $C^* = C^{**}_1(R_1, 0)$.*

Proof: For $i = 1, \ldots, N$, suppose we produce in the ith period, by Proposition 3.1, we can interpret $C^{**}_i(J_i, I_i)$ in (1) as the optimal cost incurred by our model from the beginning of the ith period to the end of the time horizon, where J_i, I_i is the number of units of serviceable products and cores in our model at the beginning of the ith period; $C^{*-}_i(J_i, I_i)$ in (2) as the optimal cost incurred by our model from the beginning of the ith period to the end of the time horizon under the condition that if we produce in the jth period, where $j > i$, then $I_i + J_i \geq D_i + \ldots + D_{j-1}$, with J_i, I_i being the number of units of serviceable products and cores in our model at the beginning of the ith period; $C^{*+}_i(J_i, I_i)$ in (3) as the optimal cost incurred by our model from the beginning of the ith period to the end of the time horizon under the consition that if we produce in the jth period, where $j > i$, then $I_i + J_i \leq D_i + \ldots + D_{j-1}$, with J_i, I_i is the number of units of serviceable products and cores in our model at the beginning of the ith period. Clearly, we then have $C^* = C^{**}_1(R_1, 0)$. $\square$

By the above lemma, we are able to solve **DLSP** by solving the dynamic program (1)-(3) with $J_1 = R_1, I_1 = 0$. We next describe an algorithm to solve **DLSP** by solving this dynamic program. Before we do this, we have a lemma below that is the basis for the algorithm and further allows us to show Theorem 3.6:

Lemma 3.4 *When solving **DLSP** using the dynamic program (1)-(3), where we set $J_1 = R_1, I_1 = 0$, we have, for all $i = 2, \ldots, N$, $I_i = 0$ and J_i takes value $(R_{k_1} + \ldots + R_i) - (D_{k_2} + \ldots + D_{i-1})$ for some k_1, k_2, where $1 \leq k_1 \leq k_2 \leq i$, in (1).*

Proof: Consider $i = 1$ in (1) with $J_1 = R_1, I_1 = 0$. We see from (2) that for $2 \leq j \leq N + 1$, the first argument and second argument of $C^{**}_j(\cdot, \cdot)$ for its minimization problem takes value $(R_1 + \ldots + R_j) - (D_1 + \ldots + D_{j-1})$ and 0 respectively, while from (3), the first and second argument of $C^{**}_j(\cdot, \cdot)$ for its minimization problem takes value $R_2 + \ldots + R_j$ and 0 respectively. Let us now focus on $C^{**}_2(R_2, 0) = \min\{C^{*-}_2(R_2, 0), C^{*+}_2(R_2, 0)\}$. For $C^{*-}_2(R_2, 0)$, we see from (2) that the expression within the minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$ equal to $(R_2 + \ldots + R_j) - (D_2 + \ldots + D_{j-1})$ and 0 respectively for $j \geq 3$. For $C^{*+}_2(R_2, 0)$, we see from (3) that the expression within its minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$ equal to $R_3 + \ldots + R_j$ and 0 respectively for $j \geq 3$. If we consider $C^{**}_i((R_2 + \ldots + R_i) - (D_2 + \ldots + D_{i-1}), 0)$ for some $i \geq 3$, then we see from (2) that the expression within the minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$ equal to $(R_2 + \ldots + R_j) - (D_2 + \ldots + D_{j-1})$ and 0 respectively for $j > i$, and from (3) that the expression within the minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$ equal to $R_{i+1} + \ldots + R_j$ and 0 respectively for $j > i$. Furthermore, if we consider $C^{**}_i(R_3 + \ldots R_i, 0)$ for some $i \geq 3$, then we see from (2) that the expression within the minimization problem has the first and second argument of $C^*_j(\cdot, \cdot)$ equal to $(R_3 + \ldots + R_j) - (D_i + \ldots + D_{j-1})$ and 0 respectively for $j > i$, and from (3) that the expression within the minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$ equal to $R_{i+1} + \ldots + R_j$ and 0 respectively for $j > i$. Now, let us focus on $C^{**}_i(R_2 + \ldots + R_i, 0)$ for $i \geq 3$. We see from (2) that the expression within the minimization problem has the first and second argument of $C^{**}_j(\cdot, \cdot)$

equal to $(R_2 + \ldots + R_j) - (D_i + \ldots + D_{j-1})$ and 0 respectively for $j > i$, and from (3), the expression within the minimization problem has the first and second argument of $C_j^{**}(\cdot, \cdot)$ equal to $R_{i+1} + \ldots + R_j$ and 0 respectively for $j > i$. By arguing in a similar manner for other possibilities when solving the dynamic program (1)-(3) with $J_1 = R_1, I_1 = 0$, we see that the lemma holds. $\square$

Algorithm 3.5

Step 1. *Iterate from $i = N$ to 2, and use previously computed values for $C_j^{**}(J_j, I_j)$, $j = i + 1, \ldots, N$, with $C_{N+1}^{**}(J_{N+1}, I_{N+1}) = 0$, to find $C_i^{**}(J_i, I_i)$ from (1)-(3) with $J_i = (R_{k_1} + \ldots + R_i) - (D_{k_2} + \ldots + D_{i-1})$, for $1 \leq k_1 \leq k_2 \leq i$, and $I_i = 0$.*

Step 2. *Find $C_1^{**}(R_1, 0)$ using (1), where $C_j^{**}((R_1 + \ldots + R_j) - (D_1 + \ldots + D_{j-1}), 0)$ in (2) and $C_j^{**}(R_2 + \ldots + R_j, 0)$ in (3), $j = 2, \ldots N + 1$, have been computed in Step 1.*

While executing the algorithm, we determine $j^*(i)$ which is the $j, i+1 \leq j \leq N+1$ that attained the minimum in (2) or (3), for $i = 1, \ldots, N$. Whether we consider (2) or (3) depends on whether the first term or the second term in the minimization in (1) is lower. These $j^*(i)$ allow us to determine the period when we should produce. In particular, we always remanufacture when $j^*(i)$ is obtained using (2), with possible manufacturing if $j^*(i)$ is obtained using (3).

Theorem 3.6 below states the complexity to solve **DLSP** using its dynamic programming formulation.

Theorem 3.6 **DLSP** *can be solved using $O(N^6)$ multiplication, addition and sort operations.*

Proof: We solve **DLSP** using the dynamic programming formulation (1)-(3) through Algorithm 3.5, and show that solving it takes $O(N^6)$ addition, multiplication and sort operations. For each $i = 1, \ldots, N$, by Lemma 3.4, (1) needs to be solved at most $i(i + 1)/2$ times. Each time (1) is solved, it involves solving (2) and (3). For $i = 1, \ldots, N$ and $i + 1 \leq j \leq N + 1$, the expression in the minimization problem in (2) requires at most $1 + 2(j - i)$ multiplications, $9 + 6(j - i) + \sum_{k=i+1}^{j-1}(k - i) + \sum_{k=i+1}^{j-1}(j - k) = 9 + 6(j - i) + (j - i - 1)(j - i)$ additions to evaluate, hence leading to a total of $O((N - i + 1)^3)$ multiplications and additions. Finding the minimum in the minimization problem in (2) requires $O((N - i + 1)^2)$ operations. Hence, for $i = 1, \ldots, N$, solving (2) requires a total of $O((N - i + 1)^3)$ multiplication, addition and sort operations. On the other hand, for $i = 1, \ldots, N$ and $i + 1 \leq j \leq N + 1$, evaluating the expression within the minimization problem in (3) requires at most $2 + 2(j - i)$ multiplications, $7 + 3(j - i) + \sum_{k=i}^{j-2}(k - i) + \sum_{k=i+1}^{j-1}(j - k) = 7 + 3(j - i) + (j - i - 1)(j - i)$ additions, hence leading to a total of $O((N - i + 1)^3)$ multiplications and additions. Finding the minimum in the minimization problem in (3) requires $O((N - i + 1)^2)$ operations. Therefore, for $i = 1, \ldots, N$, solving (3) requires a total of $O((N - i + 1)^3)$ multiplication, addition and sort operations. Taking into account various values of J_i and I_i involved in (1), we require $O(i^2(N - i + 1)^3)$ operations altogether to solve (1) for each $i = 1, \ldots, N$, which then implies a total multiplication, addition and sort operations of $O(N^6)$ summing i from 1 to N. Hence, **DLSP** can be solved using $O(N^6)$ multiplication, addition and sort operations. $\square$

Corollary 3.7 **PPi** *can be solved using $O(n^6)$ multiplication, addition and sort operations.*

Proof: Follows from Theorems 2.2 and 3.6, noting that **DLSPp** is a special case of **DLSP**. $\square$

References

[1] S. A. Cook. The complexity of theorem-proving procedures. In M. A. Harrison, R. B. Banerji, and J. D. Ullman, editors, *STOC'71: Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158. Association for Computing Machinery, 1971.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., Madison Avenue, New York, 1979.

[3] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[4] W. van den Heuvel. On the complexity of the economic lot-sizing problem with remanufacturing options. Econometric Institute Report EI 2004-46, Erasmus University Rotterdam, 2004.

[5] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.