

Managing Partially Controllable Raw Material Acquisition and Outsourcing in Production Planning

Jian Yang

Department of Mechanical and Industrial Engineering

New Jersey Institute of Technology

Newark, NJ 07102

Email: yang@njit.edu

Xiangtong Qi

Department of Industrial Engineering and Logistics Management

Hong Kong University of Science and Technology

Clear Water Bay, Kowloon, Hong Kong

Email: ieemqi@ust.hk

April 2008; revised, January 2009, June 2009

Abstract

We study a single-item production planning problem for a manufacturing firm. Besides being able to acquire raw material from an external supplier, the firm may also face an incoming stream of internally supplied raw material. In addition, outsourcing may serve as an alternative to in-house production for the firm to satisfy its demands. We concentrate on the case where acquisition, production, and outsourcing costs are setup-linear and inventory holding costs are linear. For this case, we present polynomial algorithms for some situations and show the NP-hardness of the problem for others. Our computational study testifies to the competitiveness of our proposed heuristic.

Keywords: Production Planning; Lot Sizing; Outsourcing; Dynamic Programming

1 Introduction

We consider a single-item production system (henceforward the firm) in which the production process boils down to converting a certain raw material type into a finished product type. First of all, the firm has the obligation to satisfy a demand stream for finished products. In case it being a downstream plant in a vertically integrated conglomerate, the firm may be force-fed an incoming stream of internally supplied raw material; while in all cases, the firm can acquire raw material from an external supplier. Also, when in-house production is no longer the most cost-effective means to satisfy demands, the firm may acquire finished products from an external source.

In each period, the firm has to make decisions on the production level, the quantity of raw material to be acquired from the external supplier, and the level of outsourcing, i.e., the quantity of finished products to be acquired from an external source. While making these decisions, the firm has to make sure that enough raw material exists for production and all demands are satisfied without delay. We study the multi-period deterministic production planning problem for such a firm.

The successful study of the aforementioned problem will have far-reaching practical implications because many production systems, ranging from oil refineries to paper mills, can be thought of as converting a single raw material type into a single finished product type. When a firm is a mid- or down-stream plant in a vertically integrated conglomerate, it is very probable that the amount of raw material shipped to it is at the discretion of both the headquarters and mother nature, and therefore is beyond its own control. To capture the spirit of this phenomenon, we treat a portion of the raw material as being (exogenously) internally supplied. While the headquarters may try to match the supply with the demand, it may be hard to synchronize perfectly at that level, especially when the leadtime is long, or the supply-side capacity is limited.

For instance, a refinery plant within an oil company may use both internal delivery and external acquisition as sources for its crude oil supply. The internal supply may follow a schedule that is dictated by the extraction and transportation conditions at upper-stream facilities. This schedule is not much reactive, at least in the short term, to the demand for gasoline products the plant immediately faces. It is perceivable that, when economically justifiable, the plant will resort to the open spot market to meet its production targets. In fact, many firms in the food, forest and paper products, and metal products industries face similar problems of having to deal with internal supply and acquisition of raw material simultaneously.

The internal supply may also exist when a firm has two supply modes, say a pre-

fixed and cheaper one as a primary source, and a flexible and more expensive one as a supplementary source. This can often be observed in China as it is transitioning from a planned economy to a market economy. A state-owned enterprise can both receive a rigid quantity of cheaper government-subsidized supply and make acquisitions from the external open market.

The modeling of internally supplied raw material is also motivated by the study of disruption management where a pre-determined production plan has to be updated because of disruptions. In such a context, suppose we have drawn a production plan based on future demand information, and thus have ordered raw material for a certain planning horizon. Now, dramatically changed scenarios regarding demand, supply, or costs may force us to revise the existing plan. For the new plan to be efficient, it must account for the raw material units that have been ordered according to the old plan. Our model makes such consideration possible. A company may choose an overseas supplier to reduce supplying cost, and at the same time maintain a local supplier to handle disruptions. Our model fits very well with such situations where the overseas supply has to be determined in advance due to its long leadtime.

There can be substantial benefit of using outsourcing (subcontracting) as a tactical solution to a firm's daily needs, especially when there exist fluctuations in the in-house production costs. Besides the direct production costs, the costliness of in-house production is still influenced by the combined forces of the internal raw material supply levels, the extra acquisition costs, and the finished product demand levels. Hence, the use of an external source can help reduce the overall operational cost of the firm.

Our focus will be on solving a special concave-cost version of the problem, in which the production, acquisition, and outsourcing costs are all setup-linear, and raw material and finished product inventory holding costs are all linear. Due to economy of scale, it is common for a firm to have these concave-cost features. We categorize this problem into a few exhaustive and mutually-exclusive cases. For each of the cases, we either show that it is NP-hard, or provide a polynomial-time algorithm. In particular, cases involving both internal supply of raw material and outsourcing are NP-hard as long as either production or outsourcing cost is setup-linear, while all other cases are polynomially solvable.

Besides the above categorization, we propose a polynomial-time heuristic for the problem which works even when all costs are generally concave. This heuristic is inspired by one in Yang, Golany, and Yu (2005), whose focus was a concave-cost production planning problem involving the remanufacturing and disposal of returned items.

Our computational study reveals the competitiveness of the heuristic with respect to a brute-force DP algorithm and the solving of the problem through a commercial MIP solver.

The remainder of the paper is organized as follows. We survey the relevant literature in Section 2, and introduce the problem’s mathematical formulation and assumptions in Section 3. We cover NP-hard cases in Section 4 and properties useful in the development of algorithms in Section 5. Sections 6 and 7 are devoted to polynomial-time algorithms. We introduce the heuristic in Section 8 and computationally test it in Section 9. The paper is concluded in Section 10.

2 Literature Survey

The (concave-cost) single-item production planning (lot-sizing) problem was first studied in the 1950s. Wagner and Whitin (1959) gave an $O(T^2)$ solution to a special case in which production costs are setup-linear and inventory holding costs are linear. Aggarwal and Park (1993), Federgruen and Tzur (1991), and Wagelmans, van Hoesel, and Kolen (1992) all provided $O(T \log T)$ algorithms for this special case. For the general concave-cost problem, Wagner (1960) recognized it to be a minimum-concave-cost network flow problem and pointed out that the extreme-point optimal solution must form a spanning tree in the network. This observation led to the discovery of an $O(T^2)$ algorithm for the problem by Zangwill (1969).

Sargut and Romeijn (2007b) considered the feature of cumulative capacities which is equivalent to our internal supply of raw material. That is, they effectively studied a version of our problem without the two features of acquisition and outsourcing. They showed the NP-hardness of the problem in general and developed a polynomial-time algorithm for the problem’s concave-cost case. On the other hand, when raw material is neither acquirable nor storable, each period’s internal raw material supply level will serve as that period’s production capacity. In this regard, our problem is somewhat related to the single-item single-stage capacitated production planning problem. With the acquisition option and the non-perishability property for the raw material, our problem is conceivably harder than the aforementioned problem.

There has been extensive research on the capacitated production planning problem. The special case with constant capacities was first found to be polynomially solvable in $O(T^4)$ time by Florian and Klein (1971). Later, van Hoesel and Wagelmans (1996) reduced the time complexity to $O(T^3)$. After the general problem with arbitrary capac-

ities was shown to be NP-hard by Florian, Lenstra, and Rinnooy Kan (1980), Bitran and Yanasse (1982) further investigated the computational complexity of some special cases and identified polynomially solvable cases among them. Approximate solution procedures were proposed for the general capacitated problems by Bitran and Matsuo (1986), Gavish and Johnson (1990), and van Hoesel and Wagelmans (2001). Other extended models on single-stage production planning can be found in Atamtürk and Kucukyavuz (2005), Ganas and Papachristos (2005), Gopalakrishnan et al. (2001), Lee, Cetinkaya, and Wagelmans (2001), and Li, Hsu, and Xiao (2004), and Wolsey (2002).

Having to simultaneously manage both the raw material and finished product inventories, our model is related to the two- or multi-stage production planning problem. Most recent results in this field can be found in Anily and Tzur (2006), Hsu, Li, and Xiao (2005), Jaruphongsa, Cetinkaya, and Lee (2004, 2005), Kaminsky and Simchi-Levi (2003), Lee, Cetinkaya, and Jaruphongsa (2003), Stadtler (2003), Teo and Bertsimas (2001), and van Hoesel et al. (2005). In particular, if internal supply is removed from our model, it will become the special two-product case of the model with one-way product substitution studied in Hsu, Li, and Xiao (2005).

However, the introduction of the internally supplied raw material stream brought in the flavor of a capacitated lot sizing problem and rendered the problem more complex. That being said, there is a certain similarity between our model and the two-stage model with transportation studied in Kaminsky and Simchi-Levi (2003): We can view our self-initiated acquisition as their first-stage production and our production as their second-stage one. So when being stripped of the outsourcing option and the exogenous supply of raw material, and when all costs are linear, ours becomes a special version of theirs, special in our lacking of their transportation stage. Nevertheless, with our analysis concentrating mostly on cases where costs are not all linear, and the exogenous supply of raw material and the outsourcing option are not simultaneously prohibited, we have to resort to solution techniques different than theirs.

Some lot sizing models consider lost sales; see, e.g., Aksen, Altinkemer, and Chand (2003) and Sandbothe and Thompson (1990). Though with the similarity of both being complementary to production, lost sales and outsourcing are fundamentally different. Managerially, outsourcing is something the firm directly controls, while lost sales is a byproduct of its production decisions. Analytically, the cost for outsourcing can have setup components, while the cost for lost sales is always linear. As a consequence, outsourcing in one period can help meet demands in a number of future periods, while

the lost sales level is capped by the current-period demand.

The body of literature on the integrated planning of production and outsourcing has lately become considerable. Atamtürk and Hochbaum (2001) examined a model which contains decisions on capacity acquisition, production planning, and outsourcing. Merzifonluoğlu, Geunes, and Romeijn (2007) and Sargut and Romeijn (2007a) both developed polynomial algorithms for production planning problems involving stationary capacities and the outsourcing option, whereas the former treated demand as controllable through pricing and the latter considered a two-stage system.

Without the firm’s autonomous raw material acquisition, our model is, as has also been realized by van den Heuvel and Wagelmans (2008) in a slightly different context, equivalent to production planning involving the remanufacturing of returned items: Just think of the internally-supplied raw material stream here as the stream of returned items, production here as remanufacturing, and outsourcing here as regular manufacturing. Teunter, Bayindir, and van den Heuvel (2006) considered a lot-sizing problem involving remanufacturing. They developed heuristics for cases where manufacturing and remanufacturing may or may not share setups. The problem was also treated in van den Heuvel (2006).

In Yang, Golany, and Yu (2005), there is in addition the option of discarding returned items, which is like raw material acquisition here, albeit being reversed. This is the main reason why a heuristic developed there could be adapted to suit our current needs. On the other hand, we should caution that this existing remanufacturing problem possesses a certain symmetry between returned items/disposal and demand/manufacturing. Due to the aforementioned reversal in direction in the acquisition activity, however, our current problem does not enjoy the luxury of having this symmetry—internal supply and acquisition both tend to enrich the raw material inventory, while demand and outsourcing work against each other on the finished product inventory.

To the best of our knowledge, this paper is the first to simultaneously deal with both (a) internal supply of raw material and (b) autonomous raw material acquisition. The outsourcing option adds an additional layer of complication to our problem. We contribute mainly in drawing an exact demarcation line between NP-hard and polynomially-solvable cases of the problem, which should serve as a useful guideline to future research and practices revolving around this important problem. To appreciate how intricately woven (a) and (b) can be, we note the following: Without (a), our problem will become a usual two-stage problem which can be polynomially solved

by “routine” two-layered DP algorithms, while without (b), it, as mentioned, will be equivalent to the well-examined production planning problem with remanufacturing.

3 Model Formulation

We suppose that the planning horizon consists of T periods. To describe the problem, we still need the following nonnegative integer parameters:

- U_0 : the initial raw material inventory level;
- V_0 : the initial finished product inventory level;
- B_t for $t = 1, \dots, T$: the internal raw material supply level in period t ;
- D_t for $t = 1, \dots, T$: the demand level in period t ;
- U_T : the terminal raw material inventory level;
- V_T : the terminal finished product inventory level.

For convenience, we let $\bar{B}_{tt'} = \sum_{\tau=t}^{t'} B_\tau$ and $\bar{D}_{tt'} = \sum_{\tau=t}^{t'} D_\tau$, standing, respectively, for cumulative internal supply and external demand in periods $t, t+1, \dots, t'$.

The nonnegative decision variables for this problem are:

- x_t for $t = 1, \dots, T$: the production level in period t ;
- y_t for $t = 1, \dots, T$: the extra raw material acquisition level in period t ;
- z_t for $t = 1, \dots, T$: the outsourcing level in period t ;
- u_t for $t = 0, 1, \dots, T$: the raw material inventory level between periods t and $t+1$;
- v_t for $t = 0, 1, \dots, T$: the finished product inventory level between periods t and $t+1$.

We also need the following nonnegative and nondecreasing cost functions to help define the problem:

- $P_t(x) = \bar{\pi}_t \cdot \mathbf{1}(x > 0) + \bar{p}_t \cdot x$ for $t = 1, \dots, T$: the production cost in period t ,
- $R_t(y) = \bar{\rho}_t \cdot \mathbf{1}(y > 0) + \bar{r}_t \cdot y$ for $t = 1, \dots, T$: the extra acquisition cost in period t ,
- $O_t(z) = \bar{\omega}_t \cdot \mathbf{1}(z > 0) + \bar{o}_t \cdot z$ for $t = 1, \dots, T$: the outsourcing cost in period t ,
- $G_t(u) = \bar{g}_t \cdot u$ for $t = 1, \dots, T-1$: the raw material inventory holding cost between periods t and $t+1$,

$H_t(v) = \bar{h}_t \cdot v$ for $t = 1, \dots, T-1$: the finished product inventory holding cost between periods t and $t+1$,

where $\bar{\pi}_t, \bar{p}_t, \bar{\rho}_t, \bar{r}_t, \bar{\omega}_t, \bar{o}_t, \bar{g}_t$, and \bar{h}_t are nonnegative parameters, and $\mathbf{1}(\cdot)$ stands for the indicator function. In words, we assume that the production, acquisition, and outsourcing costs are all setup-linear, and that the raw material and finished product inventory holding costs are linear.

We can formulate our production planning problem with partially controllable acquisition and outsourcing opportunities (PPPRS) as follows:

$$\text{(PPPRS) } \min \sum_{t=1}^T P_t(x_t) + \sum_{t=1}^T R_t(y_t) + \sum_{t=1}^T O_t(z_t) + \sum_{t=1}^{T-1} G_t(u_t) + \sum_{t=1}^{T-1} H_t(v_t) \quad (1)$$

subject to

$$u_0 = U_0, v_0 = V_0, u_T = U_T, v_T = V_T, \quad (2)$$

$$x_t - y_t + u_t - u_{t-1} = B_t \quad \forall t = 1, \dots, T, \quad (3)$$

$$x_t + z_t + v_{t-1} - v_t = D_t \quad \forall t = 1, \dots, T, \quad (4)$$

$$x_t, y_t, z_t \geq 0 \quad \forall t = 1, \dots, T, \quad (5)$$

$$u_t, v_t \geq 0 \quad \forall t = 1, \dots, T-1. \quad (6)$$

In the formulation, (1) states that the objective is to minimize the total cost resulting from production, extra acquisition, outsourcing, and raw material and finished product inventory holding; (2) relates the given initial and terminal inventory levels with the inventory decision variables; (3) and (4) express flow balances in raw material and finished products, respectively; also, (5) and (6) enforce the non-negativity of decision variables.

Instead of letting terminal inventory levels U_T and V_T be pre-determined, an alternative formulation could use terminal holding costs (negative salvage values) $G_T(\cdot)$ and $H_T(\cdot)$ to induce the most appropriate terminal inventory levels u_T and v_T . However, this alternative formulation is equivalent to a PPPRS involving $T+1$ periods. This PPPRS may inherit all supply, demand, and cost parameters from the alternative formulation. In addition, we may let its supply level $B_{T+1} = 0$, demand level $D_{T+1} = 0$, production cost $P_{T+1}(\cdot) = 0$, acquisition cost $R_{T+1}(\cdot) = 0$, outsourcing cost $O_{T+1}(\cdot) = 0$, and terminal inventory levels $U_{T+1} = M$ and $V_{T+1} = M$ for some integer M that is guaranteed to be greater than any feasible u_T and v_T levels. Every solution of this PPPRS satisfies that $y_{T+1} - x_{T+1} = M - u_T \geq 0$ and $x_{T+1} + z_{T+1} = M - v_T \geq 0$, while activities at these levels x_{T+1} , y_{T+1} , and z_{T+1} are free of charge. Therefore, this PPPRS and the alternative formulation share the same optimal solutions when the aforementioned three levels are ignored; and, they share the same optimal cost.

PPPRS is a min-cost network flow problem in a network (as demonstrated in Figure 1) with $2T+1$ nodes, i.e., I_t and J_t for $t = 1, \dots, T$ and O , and $5T-2$ arcs, i.e., (I_t, J_t) , (O, I_t) , and (O, J_t) for $t = 1, \dots, T$ and (I_t, I_{t+1}) and (J_t, J_{t+1}) for $t = 1, \dots, T-1$. For PPPRS to be feasible, certain conditions on initial and terminal inventory levels, as

well as on supply and demand flows, have to be satisfied. We leave details to Appendix A. From now on, we suppose that PPPRS is feasible.

[Insert Figure 1 here]

We can develop a pseudo-polynomial dynamic programming algorithm to solve PPPRS, whose detailed description we leave to Appendix B.

We now impose the following non-speculative conditions on cost parameters:

$$(C1) \quad \bar{p}_t + \bar{h}_t \geq \bar{g}_t + \bar{p}_{t+1};$$

$$(C2) \quad \bar{r}_t + \bar{g}_t \geq \bar{r}_{t+1};$$

$$(C3) \quad \bar{o}_t + \bar{h}_t \geq \bar{o}_{t+1}; \text{ and}$$

$$(C4) \quad \bar{o}_t + \bar{g}_t + \bar{p}_{t+1} \geq \bar{p}_t + \bar{o}_{t+1}.$$

We have not imposed anything on setup portions of costs. When these cost components are stationary, we will have very natural interpretations for the above four conditions. Conditions (C1), (C2), and (C3) reflect that the performance of any earlier-than-needed production/acquisition/outsourcing activities will not be profitable; and, (C4) states that it is never better to outsource now and keep a unit of raw material for production in the next period, than to produce now and outsource in the next period.

Note that the combination of (C1) and (C3) does not lead to (C4), but that of (C1) and (C2) leads to something comparable to (C4):

$$\bar{p}_t + \bar{r}_t + \bar{h}_t \geq \bar{p}_{t+1} + \bar{r}_{t+1}, \quad (7)$$

which states that production and acquisition costs do not increase as much as to call for any finished-product storage for the sole purpose of beating them. This discrepancy may be caused by the substitutable relation between production and outsourcing but complementary relation between production and acquisition. We also note that, with stationary costs, (C1) would become the natural one of the finished product being more costly to store than the raw material, while the remaining conditions would be automatic.

For mnemonic purposes and to better present our results, we use an $I_P I_O I_R$ notation to help categorize our special concave-cost cases. Here,

- I_P can be G_P , meaning that production costs are setup-linear, or l_P , meaning that production costs are strictly linear;
- I_O can be G_O , meaning that outsourcing costs are setup-linear, l_O , meaning that outsourcing costs are strictly linear, or n_O , meaning that outsourcing is too expensive for it to be a viable option for the firm;

- I_R can be G_R , meaning that acquisition costs are setup-linear, or l_R , meaning that acquisition costs are strictly linear.

We may use the newly-introduced symbols to summarize our results as follows:

(I) cases involving outsourcing are NP-hard as long as either production or outsourcing cost is setup-linear. That is, the $G_{PlO}(l_R/G_R)$, $l_P G_O(l_R/G_R)$, and $G_P G_O(l_R/G_R)$ cases are all NP-hard;

(II) when outsourcing is not allowed, the problem is polynomially solvable—the $G_{PnO}(G_R/l_R)$ cases can be solved in $O(T^7)$ time; and,

(III) when production costs are linear and outsourcing costs are linear or prohibitively high, the problem is polynomially solvable—the $l_{PlO}(G_R/l_R)$ cases can be solved in $O(T^3)$ time, with the $l_{PlO}l_R$ case being solvable in $O(T^2)$ time when no raw material leftover is required at the end of the planning horizon; the $l_{PnO}G_R$ case can be solved in $O(T^3)$ time, and the $l_{PnO}l_R$ case in $O(T)$ time.

For all our cases, we note that the send-and-split method proposed by Erickson, Monma, and Veinott (1987) for minimum-concave-cost network flow problems can offer little help. With internal raw material supply, all $2T+1$ nodes in the underlying planar network, as shown in Figure 1, are nodes with positive supply/demand. Note that, to have all these nodes in the network except node O covered, we need at least $T-1$ faces. Though very efficient in some cases, the aforementioned method is exponential in the number of faces. Also, were the internal supply of raw material absent, we could solve the corresponding cases in no more than $O(T^3)$ time by adapting Hsu, Li, and Xiao (2005)'s methods.

4 The NP-hard Cases

We can show that even the special $G_{PlO}l_R$ and $l_P G_Ol_R$ cases are NP-hard.

Theorem 1 *The special $G_{PlO}l_R$ case, i.e., the case with linear outsourcing and acquisition costs, is NP-hard.*

Proof: We reduce the NP-hard decision version of the SUBSET SUM problem to the $G_{PlO}l_R$ case. A description of SUBSET SUM goes as follows: There are $N+1$ strictly positive integers a_1, a_2, \dots, a_N , and A with $a_1 \geq a_2 \geq \dots \geq a_N$. The problem is to decide whether there is a subset \mathcal{S} of $\{1, 2, \dots, N\}$ such that $\sum_{n \in \mathcal{S}} a_n = A$.

Given an instance of SUBSET SUM, we construct an instance of the $G_{PlO}l_R$ case as follows. In this instance, $T = N$; for $n = 1, 2, \dots, N$, the production cost $P_n(x) =$

$a_n \cdot \mathbf{1}(x > 0)$, the acquisition cost $R_n(y) = 2 \cdot y$, the outsourcing cost $O_n(z) = 1 \cdot z$, the raw material inventory holding cost $G_n(u) = 0$, and the finished product inventory holding cost $H_n(v) = 2 \cdot v$; the raw material supply and finished product demand levels are $B_1 = A$, $B_2 = \dots = B_N = 0$, and for $n = 1, \dots, N$, $D_n = a_n$; and the starting and ending inventory levels U_0, V_0, U_N , and V_N are all 0. The reduction from SUBSET SUM to PPPRS can be done in polynomial time. Note that such a cost structure satisfies all conditions (C1) to (C4). To complete the proof, we shall show that the answer to the SUBSET SUM instance is *yes* if and only if the optimal cost to the PPPRS instance is no more than $\sum_{n=1}^N a_n$.

The *if* part: Suppose the PPPRS instance costs no more than $\sum_{n=1}^N a_n$ optimally. Since the minimum cost of satisfying the demand in each period n is a_n , which is only achievable through sole production or sole outsourcing, the optimal solution cannot involve acquisition or finished product inventory holding. Therefore, total raw material supply A must be used up by a set \mathcal{S} of some periods of which $\sum_{n \in \mathcal{S}} a_n = A$, implying a *yes* answer to SUBSET SUM.

The *only if* part: Suppose there is a subset \mathcal{S} of $\{1, 2, \dots, N\}$ such that $\sum_{n \in \mathcal{S}} a_n = A$. If we produce using given raw material supply in periods in \mathcal{S} and outsource in periods in $\{1, \dots, N\} \setminus \mathcal{S}$, we will incur a total cost of $\sum_{n=1}^N a_n$. ■

Theorem 2 *The special $l_P G_O l_R$ case, i.e., the case with linear production and acquisition costs, is NP-hard.*

Proof: We still reduce SUBSET SUM to the $l_P G_O l_R$ case. Given an instance of SUBSET SUM, we construct an instance of the $l_P G_O l_R$ with everything else being the same as in the proof of Theorem 1 and the exception that for $n = 1, \dots, N$, the production cost $P_n(x) = 1 \cdot x$ and the outsourcing cost $O_n(z) = a_n \cdot \mathbf{1}(z > 0)$. Note that such a cost structure satisfies all conditions (C1) to (C4). It can be shown that the answer to the SUBSET SUM instance is *yes* if and only if the optimal cost to the PPPRS instance is no more than $\sum_{n=1}^N a_n$. This is achievable when the demand of a period n is satisfied by either sole production or sole outsourcing. We omit the details. ■

The above being the case, the more general $G_P l_O G_R$, $l_P G_O G_R$, and $G_P G_O$ (l_R/G_R) cases are certainly NP-hard as well. Actually, the latter two cases are NP-hard even when costs are stationary.

Theorem 3 *The special $G_P G_O l_R$ case, i.e., the case with linear acquisition costs, is NP-hard when costs are stationary.*

Theorem 3 also follows immediately from Theorem 7.4 of van den Heuvel (2006). For the sake of completeness, we present its proof in Appendix C.

5 Useful Properties

In the following three sections, we concentrate on cases summarized in (II) and (III) in Section 3. We present special properties of optimal solutions of these cases in the current section, and elaborate on specific algorithms in the next two sections.

First, note that the V_0 finished products available in the beginning must be consumed by the D_1 demands in period 1, D_2 demands in period 2, \dots , and the $D_T + V_T$ demands in period T . Due to this and also the fact that finished product inventory holding costs are linear, solving the original problem is equivalent to solving a new problem with $U_0 = V_0 = V_T = 0$ and the V_0 being consumed first.

We can obtain the new problem from the given one by going through the following transformations: $V_0 \leftarrow 0$, $D_t \leftarrow (D_t - (V_0 - \bar{D}_{1,t-1})^+)^+$ for $t = 1, \dots, T-1$, $D_T \leftarrow D_T + V_T - (V_0 - \bar{D}_{1,T-1})^+$, $V_T \leftarrow 0$, $B_1 \leftarrow U_0 + B_1$, and $U_0 \leftarrow 0$. Let f_0^* and f^* be the optimal objective value of the original problem and the new problem, respectively. We have

$$f_0^* = \sum_{t=1}^{T-1} H_t((V_0 - \bar{D}_{1t})^+) + f^*. \quad (8)$$

On the other hand, even for the new problem, we shall treat U_T as being arbitrary unless otherwise specified. We deal with the new problem from now on.

There are common features of our 12 cases. When costs are concave, PPPRS is effectively a minimization problem with a jointly concave objective function and a convex feasible region. One of its optimal solutions must be an extreme point of the feasible region whose basis, arcs on which flows are positive, forms a loopless graph in the underlying network. Due to its network flow interpretation and the integrality of the involved parameters, PPPRS's extreme-point solution is guaranteed to be integral.

We describe the optimal properties in this section, and leave the formal statement and proof in Appendix D. We can show that, in one optimal loopless plan,

- GGG-Property 1: between any two periods both with production activities, there must be a period without finished product inventory holding;
- GGG-Property 2: between any two periods both with acquisition activities, there must be a period without raw material inventory holding;

- GGG-Property 3: between any two periods with outsourcing activities, there must be a period without finished product inventory holding; and
- GGG-Property 4: for any three periods, if the first one is with acquisition, the second with production, and the third with outsourcing, then it cannot be the case that between the first two periods, there are always positive levels of raw material inventory holding, while between the last two, there are always positive levels of finished product inventory holding.

We will have slightly more special properties for slightly more special cases. We can show that one optimal loopless plan for the $G_P G_O l_R$ case further satisfies

- GGI-Property 1: there is no raw material inventory holding immediately after acquisition;

one optimal loopless plan for the $G_P l_O G_R$ case further satisfies

- GI-G-Property 1: there is no finished product inventory holding immediately after outsourcing; and
- GI-G-Property 2: any period with a positive production level cannot have finished product inventory carry-over from the previous period that is attributable to production or outsourcing in earlier periods;

and, one optimal loopless plan for the $l_P G_O G_R$ case further satisfies

- IGG-Property 1: there is no finished product inventory holding immediately after production.

To ease the presentation of our algorithms, we expand ranges of the cost functions. Now, for any $x = 1, 2, \dots$, we let $P_t(-x) = R_t(-x) = O_t(-x) = +\infty$ for every $t = 1, 2, \dots, T$ and $G_t(-x) = H_t(-x) = +\infty$ for every $t = 1, 2, \dots, T - 1$.

6 The No-Outsourcing Cases

For the $G_P n_O G_R$ case, we inherit GGG-Properties 1 through 4, and GI-G-Properties 1 and 2.

In the following, a double-horizon (DH), designated by four nonnegative integers s' , s , t' , and t with $s' \leq s - 1$, $t' \leq t - 1$, $s' \leq t'$, $s \leq t$, and $t' \leq s - 1$, refers to a pair of sets of consecutive integers ($\{s' + 1, s' + 2, \dots, s\}, \{t' + 1, t' + 2, \dots, t\}$); the first set is associated with the raw material, while the second set is with the finished product. For convenience, we denote such a DH by $[s', s; t', t]$. This DH's first component is \emptyset

when $s' \geq s$, its second component is \emptyset when $t' \geq t$, and both of its components are \emptyset when $s' \geq t' + 1$, $s \geq t + 1$, or $t' \geq s$.

By GGG-Property 2 and GIG-Property 2, we know that one optimal loopless solution, as demonstrated in Figure 2, must have the following structure:

1) the DH $[0, T; 0, T] \equiv (\{1, \dots, T\}, \{1, \dots, T\})$, representing the overall problem's scope, can be decomposed into K sub-double-horizons (SDHs) for some $K = 1, \dots, T$ and potentially a residue-double-horizon (RDH) with an \emptyset second component; between any two consecutive pairs of the K SDHs, there is no raw material or finished product inventory holding; between the K th SDH and the RDH, there is no raw material inventory holding; for $k = 1, \dots, K$, we may use $[s_{k-1}, s_k; t_{k-1}, t_k]$ to denote the k th SDH, and naturally, the RDH is $[s_K, T; T, T]$;

2) for $k = 1, \dots, K$, acquisition takes place at most once in the k th SDH, and whenever it does, it takes place in some period $r_k = t_{k-1} + 1, t_{k-1} + 2, \dots, s_k$; acquisition takes place at most once in the RDH;

3) for $k = 1, \dots, K$, for some $M_k = 2, \dots, s_k - t_{k-1} + 2$, there exist some $M_k + 1$ integers $t_k^0, t_k^1, \dots, t_k^{M_k}$, not necessarily all different, such that $t_k^0 = s_{k-1} \leq t_k^1 = t_{k-1} < t_k^2 < \dots < t_k^{M_k-2} < t_k^{M_k-1} = s_k - 1 < t_k^{M_k} = t_k$; the SDH $[s_{k-1}, s_k; t_{k-1}, t_k]$ can be further decomposed into $M_k - 1$ sub-sub-double-horizons (SSDHs), namely, the $[t_k^{m-1}, t_k^m + 1; t_k^m, t_k^{m+1}]$'s for $m = 1, \dots, M_k - 1$; also, the only period in which production takes place in the m th SSDH is period $t_k^m + 1$, and there is no finished product inventory holding between any two consecutive SSDHs.

[Insert Figure 2 here]

With the above structure, we can devise a multi-layered dynamic programming process, $DP(G_{PN}OG_R)$, to find f^* . For $t = 0, 1, \dots, T$, let $\tilde{g}(t)$ be the total cost of handling the RDH $[t, T; T, T]$. We know that the total level of raw material internally supplied within $[t, T; T, T]$ is $\bar{B}_{t+1, T}$, and the required terminal raw material level is U_T . Hence, the only acquisition taking place in one of the periods $t + 1, t + 2, \dots, T$ is at the level of $U_T - \bar{B}_{t+1, T}$. So, we have

$$\begin{cases} \tilde{g}(T) = 0, \\ \tilde{g}(t) = \min\{R_r(U_T - \bar{B}_{t+1, T}) + \sum_{\tau=t+1}^{r-1} G_\tau(\bar{B}_{t+1, \tau}) \\ \quad + \sum_{\tau=r}^{T-1} G_\tau(U_T - \bar{B}_{\tau+1, T}) \mid r = t + 1, \dots, T\} \quad \forall t = 0, 1, \dots, T - 1. \end{cases} \quad (9)$$

For $s, t = 0, 1, \dots, T$, and $s \leq t$, let $f(s, t)$ be the minimum total cost of using a solution to handle the problem on DH $[0, s; 0, t]$, while the solution recognizes s and t

as the terminal periods of an SDH. We have the following relationship:

$$f^* = \min\{f(t, T) + \tilde{g}(t) \mid t = 0, 1, \dots, T\}. \quad (10)$$

For $s', t' = 0, 1, \dots, T$, $s = s' + 1, s' + 2, \dots, T$, $t = t' + 1, t' + 2, \dots, T$ satisfying $s' \leq t'$, $s \leq t$, $t' \leq s - 1$, let $g(s', s, t', t)$ be the minimum total cost of using a solution to handle the problem on DH $[s', s; t', t]$, while the solution recognizes the DH as an SDH. Then, we have

$$\begin{cases} f(0, 0) = 0, \\ f(0, t) = +\infty, \quad \forall t = 1, 2, \dots, T, \\ f(s, t) = \min\{f(s', t') + g(s', s, t', t) \mid t' = 0, \dots, s - 1, s' = 0, 1, \dots, t'\}, \\ \quad \forall s = 1, 2, \dots, T, t = s, s + 1, \dots, T. \end{cases} \quad (11)$$

Now we discuss the calculation of $g(s', s, t', t)$. We note that, in the concerned SDH, the total demand level is $\bar{D}_{t'+1, t}$ and the total internally supplied raw material level is $\bar{B}_{s'+1, s}$. When $s < T$, the total acquisition level should be $\bar{D}_{t'+1, t} - \bar{B}_{s'+1, s}$, while when $s = t = T$, the level should be $U_T + \bar{D}_{t'+1, T} - \bar{B}_{s'+1, T}$ due to the terminal raw material requirement U_T .

There is certainly no difference between raw material units from internal supply and those from external acquisition. However, as a technical convenience, we assume that the firm always consumes internally supplied raw material first and externally acquired raw material next. Also, by the non-acquisition cost of a solution, we mean the total cost incurred by the solution while excluding the acquisition cost involved. Now, let $h(r, s', s, t', t)$ be the minimum total non-acquisition cost of using a solution to handle the problem on SDH $[s', s; t', t]$, where in the solution, it is in period r that externally acquired raw material is consumed for the first time.

Suppose further that acquisition occurs in period q in SDH $[s', s; t', t]$. Then, we can decompose the total raw-material-related cost into two portions: the acquisition cost plus the cost of holding the acquired raw material from period q to period r , and the total cost of holding raw material as though acquisition is made in period r . But, we note that the latter portion has been registered in $h(r, s', s, t', t)$. Hence, we have

$$\begin{cases} g(s', T, t', T) = \min\{R_q(U_T + \bar{D}_{t'+1, T} - \bar{B}_{s'+1, T}) \\ \quad + \sum_{\tau=q}^{r-1} G_\tau(U_T + \bar{D}_{t'+1, T} - \bar{B}_{s'+1, T}) + h(r, s', T, t', T) \\ \quad \mid q = s' + 1, \dots, T, r = \max\{q, t' + 1\}, \dots, T\}, \\ g(s', s, t', t) = \min\{R_q(\bar{D}_{t'+1, t} - \bar{B}_{s'+1, s}) + \sum_{\tau=q}^{r-1} G_\tau(\bar{D}_{t'+1, t} - \bar{B}_{s'+1, s}) \\ \quad + h(r, s', s, t', t) \mid q = s' + 1, \dots, s, \\ \quad r = \max\{q, t' + 1\}, \dots, s\} \quad \text{when } s \leq T - 1. \end{cases} \quad (12)$$

When calculating $h(r, s', s, t', t)$, we may rely on the decomposition of the current SDH into SSDHs of the type $[a, b + 1; b, c]$. There are two different types of SSDHs in the form of $[a, b + 1; b, c]$: the early-type where $b \leq r - 1$ and the late-type where $b \geq r$. In Figure 3, we depict the decomposition of an SDH into early- and late-type SSDHs.

[Insert Figure 3 here]

Let $q^E(s', t', a, b, c)$ and $q^L(a, b, c, s, t)$ be the total non-acquisition cost incurred on an SSDH $[a, b + 1; b, c]$ when it is, respectively, of the early- and late-type in SDH $[s', s; t', t]$. After checking flow conservation, we have

$$\left\{ \begin{array}{l} q^E(s', t', a, b, c) = P_{b+1}(\bar{D}_{b+1,c}) + \sum_{\tau=a+1}^b G_{\tau}(\bar{B}_{s'+1,\tau} - \bar{D}_{t'+1,b}) \\ \quad + \sum_{\tau=b+1}^{c-1} H_{\tau}(\bar{D}_{\tau+1,c}), \\ q^L(a, b, c, s, t) = P_{b+1}(\bar{D}_{b+1,c}) + \sum_{\tau=a+1}^b G_{\tau}(\bar{D}_{b+1,t} - \bar{B}_{\tau+1,s}) \\ \quad + \sum_{\tau=b+1}^{c-1} H_{\tau}(\bar{D}_{\tau+1,c}), \quad \text{when } s \leq T - 1; \\ q^L(a, b, c, T, T) = P_{b+1}(\bar{D}_{b+1,c}) + \sum_{\tau=a+1}^b G_{\tau}(U_T + \bar{D}_{b+1,T} - \bar{B}_{\tau+1,T}) \\ \quad + \sum_{\tau=b+1}^{c-1} H_{\tau}(\bar{D}_{\tau+1,c}). \end{array} \right. \quad (13)$$

Let $h^E(s', t', a', a, b', b)$ be the total non-acquisition cost of DH $[a', a; b', b]$ when it can be decomposed solely into early-type SSDHs for SDH $[s', s; t', t]$, and $h^L(a', a, b', b, s, t)$ be the total non-acquisition cost of DH $[a', a; b', b]$ when it can be decomposed solely into late-type SSDHs for SDH $[s', s; t', t]$. To compute the $h^E(s', t', a', a, b', b)$'s and $h^L(a', a, b', b, s, t)$'s, we can use the terminal and recursive relations

$$\left\{ \begin{array}{l} h^E(s', t', a', a, b', b) = \min\{q^E(s', t', a', b', c) + h^E(s', t', b', a, c, b) \mid \\ \quad c = b' + 1, b' + 2, \dots, a\} \quad \text{when } b' \leq a - 2, \\ h^E(s', t', a', b' + 1, b', b) = q^E(s', t', a', b', b), \\ h^L(a', a, b', b, s, t) = \min\{q^L(a', b', c, s, t) + h^L(b', a, c, b, s, t) \mid \\ \quad c = b' + 1, b' + 2, \dots, a\} \quad \text{when } b' \leq a - 2, \\ h^L(a', b' + 1, b', b, s, t) = q^L(a', b', b, s, t). \end{array} \right. \quad (14)$$

Since the latest-appearing early-type SSDH $[a, b + 1; b, c]$ must have $b = r - 1$, we have

$$h(r, s', s, t', t) = \begin{cases} \min\{q^E(s', t', a, r - 1, c) + h^E(s', t', s', a + 1, t', r - 1) \\ + h^L(r - 1, s, c, t, s, t) \mid a = t', t' + 1, \dots, r - 1, \\ c = r, r + 1, \dots, s - 1\} & \text{when } t' + 2 \leq r \leq s - 1 \\ \min\{q^E(s', t', s', t', c) + h^L(t', s, c, t, s, t) \\ \mid c = t' + 1, t' + 2, \dots, s - 1\} & \text{when } r = t' + 1 \leq s - 1, \\ \min\{q^E(s', t', a, s - 1, t) + h^E(s', t', s', a + 1, t', s - 1) \\ \mid a = t', t' + 1, \dots, s - 1\} & \text{when } r = s \geq t' + 2, \\ q^E(s', s - 1, s', s - 1, t), & \text{when } r = s = t' + 1. \end{cases} \quad (15)$$

The algorithm $DP(G_{PnO}G_R)$ contains multiple layers of recursions. The overall problem is solved using (10), where the $\tilde{g}(t)$'s are computed from (9) and the $f(t, T)$'s are obtained from the recursive relations in (11). The $g(s', s, t', t)$'s involved in (11) are obtained from (12), the $h(r, s', s, t', t)$'s involved in (12) are obtained from (15), the $h^E(s', t', a', a, b', b)$'s and $h^L(a', a, b', b, s, t)$'s involved in (15) are obtained from (14), and the $q^E(s', t', a, b, c)$'s and $q^L(a, b, c, s, t)$'s involved in (14) and (15) are obtained from (13). The bulk of the time for solving the problem is spent on tasks (14) and (15), which are both of the order $O(T^7)$.

Besides having all properties of the $G_{PnO}G_R$ case, the $G_{PnO}l_R$ case further enjoys GGI-Property 1, that of no raw material inventory holding immediately after acquisition. Even with this property, one optimal solution for this case will have essentially the same structure as that for the $G_{PnO}l_R$ case. The only difference is that in the current case, in a given SDH $[s', s; t', t]$, acquisition always occurs in period s and there are only early-type SSDHs in it. We can obtain an $O(T^7)$ algorithm $DP(G_{PnO}l_R)$ for this case by keeping (9), (10), and (11) intact, changing (12) into

$$\begin{cases} g(s', T, t', T) = R_T(U_T + \bar{D}_{t'+1, T} - \bar{B}_{s'+1, T}) + h(T, s', T, t', T), \\ g(s', s, t', t) = R_s(\bar{D}_{t'+1, t} - \bar{B}_{s'+1, s}) + h(s, s', s, t', t) & \text{when } s \leq T - 1, \end{cases} \quad (16)$$

using only the first relation in (13), the first two relations in (14), and the last two relations in (15).

7 The Linear-Production-and-Outsourcing Cases

Besides GGG-Properties 1 through 4, GIG-Properties 1 and 2, and IGG-Property 1, we may have an optimal loopless plan for the $l_{PlO}G_R$ case still satisfying,

- llG-Property 1: there is no finished product inventory holding; and
- llG-Property 2: outsourcing will not be invoked if there is raw material inventory being saved for future production.

By GGG-Properties 2 and 4, as well as llG-Properties 1 and 2, we have the following structure for an optimal solution, as demonstrated in Figure 4:

1) the planning horizon $\{1, 2, \dots, T\}$ is decomposed into $K + 1$ sub-horizons for some $K = 0, 1, \dots, T$: K regular sub-horizons and one residue sub-horizon, such that there is no raw material holding between any two adjacent sub-horizons;

2) in any regular sub-horizon, there is at most one period in which acquisition or outsourcing takes place, and whenever outsourcing takes place, it occurs in the last period of a sub-horizon; and

3) in the residue sub-horizon, there can be at most one period in which acquisition takes place and multiple periods in which outsourcing takes place, while except for the last-period acquisition, there can be no production or acquisition after the occurrence of any outsourcing activity.

[Insert Figure 4 here]

Based on the above structure, we can develop a dynamic programming procedure $DP(l_P l_O G_R)$ to solve the problem. For $t = 1, \dots, T + 1$, let $f(t)$ be the minimum total cost of using a solution to handle the problem from period t to period T , while the solution recognizes period t as the starting period of a sub-horizon. The optimal solution for the overall problem is obtained by solving for $f(1)$. For $t = 1, \dots, T$ and $t' = t + 1, \dots, T + 1$, let $g(t, t')$ be the minimum total cost of using a solution to handle the problem on the set of periods $\{t, t + 1, \dots, t' - 1\}$, while the solution recognizes the set as a sub-horizon. We then have the following recursive relations:

$$\begin{cases} f(T + 1) = 0, \\ f(t) = \min\{g(t, t') + f(t') \mid t' = t + 1, \dots, T + 1\} \quad \forall t = 1, 2, \dots, T. \end{cases} \quad (17)$$

From now on, we call a regular sub-horizon with a potential outsourcing activity an O -sub-horizon (OSH) and a regular sub-horizon with a potential acquisition activity an R -sub-horizon (RSH). When $t' \leq T$, the regular sub-horizon $\{t, \dots, t' - 1\}$ is either an OSH or RSH, depending on whether the total cost $g^O(t, t')$ of treating it as an OSH is less than the total cost $g^R(t, t')$ of treating it as an RSH. That is,

$$g(t, t') = \min\{g^O(t, t'), g^R(t, t')\}. \quad (18)$$

If the regular sub-horizon is an OSH, then for the only outsourcing activity in period $t' - 1$, the outsourcing quantity is given by $\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}$. Thus, we have

$$g^O(t, t') = O_{t'-1}(\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}) + \sum_{\tau=t}^{t'-2} P_\tau(D_\tau) + P_{t'-1}(\bar{B}_{t,t'-1} - \bar{D}_{t,t'-2}) + \sum_{\tau=t}^{t'-2} G_\tau(\bar{B}_{t,\tau} - \bar{D}_{t,\tau}). \quad (19)$$

If the regular sub-horizon is an RSH, then for the only acquisition activity in some period $r = t, \dots, t' - 1$, the acquisition quantity is $\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}$. We let $r(t)$ be the smallest $\tau = t, \dots, T - 1$ such that $\bar{B}_{t\tau} \leq \bar{D}_{t\tau} - 1$ when such a τ exists and T otherwise, which is the latest period acquisition should take place. We can compute all $r(t)$'s in $O(T^2)$ time. For our current purpose, we should let $g^R(t, t') = +\infty$ when $r(t) \geq t'$. Otherwise, we shall have

$$g^R(t, t') = \min\{R_r(\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}) + \sum_{\tau=t}^{t'-1} P_\tau(D_\tau) + \sum_{\tau=t}^{r-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}) + \sum_{\tau=r}^{t'-2} G_\tau(\bar{D}_{\tau+1,t'-1} - \bar{B}_{\tau+1,t'-1}) \mid r = t, t+1, \dots, r(t)\}. \quad (20)$$

Now we come to the case where $t' = T + 1$, i.e., the case where the set of periods $\{t, \dots, T\}$ is a residue sub-horizon. Recall that, due mainly to llG-Property 2, outsourcing will only occur after acquisition, unless acquisition takes place in period T . We may use r to denote the period with acquisition and s the first period with outsourcing. From the above, we have that $s = r + 1$ unless $r = T$, at which time s may be any one of $1, \dots, r(t)$. Using $h(t, r, s)$ to denote the total cost on a residue sub-horizon $\{t, \dots, T\}$ as in the above, we have

$$h(t, r, s) = \begin{cases} \sum_{\tau=t}^r P_\tau(D_\tau) + \sum_{\tau=r+1}^T O_\tau(D_\tau) + R_r(\bar{D}_{t,r} + U_T - \bar{B}_{tT}) \\ \quad + \sum_{\tau=t}^{r-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}) + \sum_{\tau=r}^{T-1} G_\tau(U_T - \bar{B}_{\tau+1,T}) \\ \quad \text{when } r = t, \dots, r(t) \text{ and } s = r + 1, \\ \sum_{\tau=t}^{s-1} P_\tau(D_\tau) + \sum_{\tau=s}^T O_\tau(D_\tau) + R_T(\bar{D}_{t,s-1} + U_T - \bar{B}_{tT}) \\ \quad + \sum_{\tau=t}^{s-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}) + \sum_{\tau=s}^{T-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t,s-1}) \\ \quad \text{when } r = T \text{ and } s = t, t+1, \dots, r(t). \end{cases} \quad (21)$$

Finally, we have

$$g(t, T) = \min\{h(t, r, s) \mid r = t, \dots, r(t) \text{ and } s = r + 1, \text{ or } r = T \text{ and } s = t, t+1, \dots, r(t)\}. \quad (22)$$

The bulk of the time taken by the procedure is spent on preparing the $g^R(t, t')$'s, $g^O(t, t')$'s, and the $O(T^2)$ number of $h(t, r, s)$'s, which is of the $O(T^3)$ order. So the overall complexity of the procedure is $O(T^3)$.

The $l_P l_O l_R$ case further has GGI-Property 1. Besides inheriting the structural results above, an optimal plan in this case may also dictate that, a regular RSH starting

from period t always ends in period $r(t)$, and a residue sub-horizon starting from period t does not have $s = r(t) + 1$. From this, we can adapt the existing algorithm into the algorithm $DP(l_P l_O l_R)$ by changing (20) into

$$\begin{cases} g^R(t, r(t) + 1) &= R_{r(t)}(\bar{D}_{tr(t)} - \bar{B}_{tr(t)}) + \sum_{\tau=t}^{r(t)} P_\tau(D_\tau) \\ &+ \sum_{\tau=t}^{r(t)-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}), \\ g^R(t, t') &= +\infty \quad \text{when } t' \neq r(t) + 1, \end{cases} \quad (23)$$

and (22) into

$$g(t, T) = \min\{h(t, T, s) | s = t, t + 1, \dots, r(t)\}. \quad (24)$$

The complexity of the algorithm is still $O(T^3)$.

The $l_P n_O G_R$ case allows no outsourcing. We define $f(t)$ for $t = 1, \dots, T + 1$ and $g(t, t')$ for $t = 1, \dots, T$ and $t' = r(t) + 1, r(t) + 2, \dots, T + 1$ as for the $l_P l_O G_R$ case. We have the slightly different range for t' because in sub-horizon $\{t, t + 1, \dots, t' - 1\}$, acquisition must take place in period $r(t)$. We have

$$\begin{cases} f(T + 1) &= 0, \\ f(t) &= \min\{g(t, t') + f(t') | t' = r(t) + 1, \dots, T + 1\} \quad \forall t = 1, 2, \dots, T. \end{cases} \quad (25)$$

For sub-horizon $\{t, \dots, t' - 1\}$, the acquisition quantity in one of the periods $t, t + 1, \dots, r(t)$ is $\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}$ when $t' \leq T$ and $\bar{D}_{tT} + U_T - \bar{B}_{tT}$ when $t' = T + 1$. We therefore have

$$g(t, t') = \begin{cases} \min\{R_r(\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}) + \sum_{\tau=r}^{r(t)-1} G_\tau(\bar{D}_{t,t'-1} - \bar{B}_{t,t'-1}) \\ | r = t, t + 1, \dots, r(t)\} + \sum_{\tau=t}^{t'-1} P_\tau(D_\tau) + \sum_{\tau=t}^{r(t)-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}) \\ + \sum_{\tau=r(t)}^{t'-2} G_\tau(\bar{D}_{\tau+1,t'-1} - \bar{B}_{\tau+1,t'-1}) \quad \text{when } t' \leq T, \\ \min\{R_r(\bar{D}_{tT} + U_T - \bar{B}_{tT}) + \sum_{\tau=r}^{r(t)-1} G_\tau(\bar{D}_{tT} + U_T - \bar{B}_{tT}) \\ | r = t, t + 1, \dots, r(t)\} + \sum_{\tau=t}^T P_\tau(D_\tau) + \sum_{\tau=t}^{r(t)-1} G_\tau(\bar{B}_{t\tau} - \bar{D}_{t\tau}) \\ + \sum_{\tau=r(t)}^{T-1} G_\tau(\bar{D}_{\tau+1,T} + U_{tT} - \bar{B}_{\tau+1,T}) \quad \text{when } t' = T + 1. \end{cases} \quad (26)$$

The overall complexity of the above algorithm $DP(l_P n_O G_R)$ is dominated by the time spent on the calculation of the $g(t, t')$'s, which is $O(T^3)$.

The $l_P n_O l_R$ case further enjoys GGI-Property 1, that there is no raw material inventory holding immediately after acquisition. So only sub-horizons with $t' = r(t) + 1$ warrants consideration. Furthermore, when $U_T > 0$, we can always obtain an equivalent $(T + 1)$ -period problem with $U_{T+1} = 0$, everything else being the same as the original problem, but with $D_{T+1} = U_T$, $B_{T+1} = U_{T+1} = 0$, $P_{T+1}(x) = 0$, $R_{T+1}(y) = R_T(y)$, and $H_T(u) = G_T(v) = 0$. Acquisition and production in period $T + 1$ for the new

problem are equivalently acquisition and holding of raw material in period T . We need only to face the $l_{PN}o_{lR}$ case with $U_T = 0$ now. For its solution, we can use an $O(T)$ -time greedy algorithm. In this algorithm, we loop from $t = 1$ to $t = T$; and in every period t , we acquire $(D_t - u_{t-1} - B_t)^+$ raw material units, produce D_t finished product units, and carry $u_t = (u_{t-1} + B_t - D_t)^+$ raw material units over to period $t + 1$.

8 A Polynomial-time Heuristic

We present an $O(T^4)$ heuristic edified by one developed for a concave-cost production planning problem involving remanufacturing in Yang, Golany, and Yu (2005). This heuristic will work even when costs are generally concave.

For any given set of flows in the PPPRS network, there always exists a K -partition of the horizon $\{1, \dots, T\}$ into sub-horizons $\{t_0, \dots, t_1-1\}$, $\{t_1, \dots, t_2-1\}$, \dots , $\{t_{K-1}, \dots, t_K-1\}$ with $K = 1, \dots, T$ and $1 \equiv t_0 < t_1 < \dots < t_{K-1} < t_K \equiv T+1$, such that there is no flow on any inventory arc, whether raw material or finished product, crossing between the $(k-1)$ th and k sub-horizon, i.e., between periods $t_k - 1$ and t_k , for any $k = 1, \dots, K-1$, while there is a positive flow on at least one inventory arc between any two adjacent periods both inside the k th sub-horizon for $k = 1, \dots, K$. In this regard, an optimal solution for PPPRS optimally partitions the original time horizon into sub-horizons and in each sub-horizon, it solves the corresponding sub-problem optimally.

In a given sub-horizon $\{t, \dots, t' - 1\}$, our sub-problem is that of finding the optimal sub-solution for periods $t, \dots, t' - 1$, where it is required that there is no flow on any inventory arc through either boundary of the sub-horizon and there is a positive flow on at least one inventory arc in any internal interval in the sub-horizon. Due to the absence of flows on inventory arcs through its boundaries, the cost of each sub-problem that is to be optimized is well defined.

When the sub-problem in every sub-horizon $\{t, \dots, t' - 1\}$ is solved and the optimal cost is found to be $q(t, t')$, we may find the optimal solution using dynamic programming. Define $f(t)$ for t between 1 and $T + 1$ to be the optimal cost for PPPRS being constrained to periods $t, t + 1, \dots, T$. Then, we have

$$\begin{cases} f(T + 1) = 0, \\ f(t) = \min\{q(t, t') + f(t') \mid t' = t + 1, t + 2, \dots, T + 1\} \quad \forall t = T, T - 1, \dots, 1. \end{cases} \quad (27)$$

The computation of $f(1)$ will reveal the optimal cost. This process takes $O(T^2)$ time. However, we will see that it requires an amount of time that is exponential to $t' - t$ to solve the sub-problem on sub-horizon $\{t, \dots, t' - 1\}$ to obtain $q(t, t')$.

The nodes of the PPPRS network and the arcs on which a given loopless solution has positive flows together form a loopless subgraph of the PPPRS network. This loopless subgraph is equal to, or is a proper subgraph of, a spanning tree, which is defined as a subgraph of the PPPRS network that constitutes a tree when the directions of its arcs are ignored. We call a spanning tree containing all positive-flow arcs of a loopless solution the basis of the solution. There is a node-arc removal procedure that will associate every node with a unique arc in a given spanning tree, except for node O . See Yang, Golany, and Yu (2005) for a description of the procedure. Each node I_t may be associated with one of four possible arcs: y_t , x_t , u_{t-1} , and u_t ; and each node J_t may also be associated with one of four possible arcs: x_t , z_t , v_{t-1} , and v_t . Thus, there are $4 \times 4 - 1 = 15$ ways, or 15 flow patterns, in which a (I_t, J_t) pair can be associated with different pairs of arcs.

In our notation for each pattern, the first symbol stands for the type of arc that is associated with node I_t and the second stands for the one that is associated with J_t . When there is a “bar” above a symbol, it means that it is the arc with the subscript of $(t - 1)$, rather than the one with the subscript of t , that is associated with the corresponding node. For instance, we would write (uz) for the pattern in which I_t is associated with u_t and J_t with z_t , while $(\bar{u}x)$ for the pattern in which I_t is associated with \bar{u}_{t-1} and J_t with x_t . Under this notation, the 15 patterns are: (yx) , (yz) , $(y\bar{v})$, (yv) , (xz) , $(x\bar{v})$, (xv) , $(\bar{u}x)$, $(\bar{u}z)$, $(\bar{u}\bar{v})$, $(\bar{u}v)$, (ux) , (uz) , $(u\bar{v})$, and (uv) . Figure 5 illustrates some of the patterns.

[Insert Figure 5 here]

A spanning tree for the whole-horizon PPPRS network is merely a combination of T flow patterns—one for each period. When the spanning tree is the basis of a solution for PPPRS that has $\{t, \dots, t' - 1\}$ as one of its sub-horizons, the subgraph resulting from limiting this spanning tree to the nodes $I_t, J_t, \dots, I_{t'-1}, J_{t'-1}, O$ is a spanning tree for the smaller network on the sub-horizon. So, the basis of a loopless sub-solution on sub-horizon $\{t, \dots, t' - 1\}$ can be described by a sequence of $t' - t$ flow patterns. However, as we shall see, not all $15^{t'-t}$ possible sequences can be counted as spanning trees.

From now on, by a sequence, we mean a series of flow patterns residing at consecutive time periods that, when loopless, make up a spanning tree on a sub-horizon. In appendix E, we introduce maximally polynomial sets (MPSs) of patterns and other related concepts. Simply put, the number of distinct sequences with patterns coming

from a given MPS is polynomial in the length of the sequence, while this number will become exponential if any more pattern is added to the MPS. Our heuristic finds the best solution when each of its sub-horizon must be associated with only one MPS.

On the sub-horizon from $\{t, t+1, \dots, t'-1\}$, note that the two types of sequences produced by following (ux) patterns with $(x\bar{v})$ patterns and following (xv) patterns with $(\bar{u}x)$ patterns, respectively, contain loops. The nine MPSs as identified in Appendix E can generate 30 types of different loopless sequences when $t' \geq t+2$. Also, there are $t' - t$ different sequences of type i for $i = 1, \dots, 27$ and $t' - t - 1$ different sequences of type i for $i = 28, 29, 30$. For instance, the type-1 sequence with some $k = 0, 1, \dots, t' - t - 1$ can be denoted by

$$(uz)[t, t+k-1](xz)[t+k, t+k](\bar{u}x)[t+k+1, t'-1] \text{ type (see Figure 5),}$$

that is, this sequence has (uz) patterns in periods t to $t+k-1$, an (xz) pattern in period $t+k$, and $(\bar{u}x)$ patterns in periods $t+k+1, \dots, t'-1$. To avoid meaningless repetitions, we omit listing all the remaining 29 types here. In total, we have $30 \times (t' - t - 1) + 3$ different loopless sequences. When $t' = t+1$, we have 3 different loopless sequences: $(xz)[t, t]$, $(yz)[t, t]$, and $(yx)[t, t]$. We let $q^H(t, t)$ be the minimum cost among the costs of the three sequences.

For the aforementioned more general case where $t' \geq t+2$, we let $q_i^H(t, t', k)$ be the cost of the type- i sequence at k where either $i = 1, 2, \dots, 27$ and $k = 0, 1, \dots, t' - t - 1$, or $i = 28, 29, 30$ and $k = 0, 1, \dots, t' - t - 2$. We define $q^H(t, t')$ as follows:

$$q^H(t, t') = \min\left\{\min_{i=1}^{27} \min_{k=0}^{t'-t-1} q_i^H(t, t', k), \min_{i=28}^{30} \min_{k=0}^{t'-t-2} q_i^H(t, t', k)\right\}. \quad (28)$$

To compute the above newly defined entities, first, we may redefine $B_1, D_1, B_T,$ and D_T to be $B_1 + U_0, D_1 - V_0, B_T - U_T,$ and $D_T + V_T,$ respectively. Then, we may spend $O(T^2)$ time to calculate $\bar{B}_{tt'}$ and $\bar{D}_{tt'}$ for every (t, t') pair. We now have

$$q^H(t, t) = \min\{P_t(B_t) + S_t(D_t - B_t), R_t(-B_t) + S_t(D_t), R_t(D_t - B_t) + P_t(D_t)\}. \quad (29)$$

For $t' \geq t+2$, we have the expression

$$q_i^H(t, t', k) = \sum_{\tau=t}^{t'-1} R_\tau(y_\tau) + \sum_{\tau=t}^{t'-2} G_\tau(u_\tau) + \sum_{\tau=t}^{t'-1} P_\tau(x_\tau) + \sum_{\tau=t}^{t'-2} H_\tau(v_\tau) + \sum_{\tau=t}^{t'-1} S_\tau(z_\tau), \quad (30)$$

which can be evaluated in $O(t' - t)$ time. Hence, the key in calculating $q_i^H(t, t', k)$ lies in evaluating the $x_\tau, y_\tau, z_\tau, u_\tau, v_\tau$ values, which can be done also in $O(t' - t)$ time. For instance, for $i = 1$, we have $z_\tau = -D_\tau$ for $\tau = t, \dots, t+k-1$; $z_{t+k} = \bar{D}_{t+k, t'-1} - \bar{B}_{t, t'-1}$; $x_{t+k} = \bar{B}_{t, t'-1} - \bar{D}_{t+k+1, t'-1}$; $x_\tau = D_\tau$ for $\tau = t+k+1, \dots, t'-1$; $u_\tau = \bar{B}_{t\tau}$ for $\tau = t, \dots, t+k-1$; and $u_\tau = \bar{D}_{\tau+1, t'-1} - \bar{B}_{\tau+1, t'-1}$ for $\tau = t+k, \dots, t'-2$. The variable

values for the remaining 29 cases can all be easily found. We omit presenting them here.

Now, we define $f^H(t)$ as $f(t)$ is in (27), except that we replace the optimal sub-horizon cost $q(t, t')$ with $q^H(t, t')$. Solving for $f^H(1)$ to obtain a solution to PPPRS constitutes our heuristic. Note that, each $q^H(t, t)$ can be evaluated in $O(1)$ time. For $t' \geq t + 2$, each $q_i^H(t, t', k)$ can be evaluated in $O(t' - t)$ time. So each $q^H(t, t')$ can be calculated in $O((t' - t)^2)$ time, and the $q^H(t, t')$'s can be calculated in $O(T^4)$ time. Since the DP algorithm runs only in $O(T^2)$ time, we see that the entire heuristic runs in $O(T^4)$ time.

9 A Computational Study

We carry out a computational study to exemplify the effectiveness of the heuristic method by comparing it with the brute-force dynamic program introduced in Section 3 and the commercial CPLEX solver. For a common instance, we denote the costs obtained from the three methods as c_{DP} (dynamic programming), c_{CP} (CPLEX solving), and c_{HR} (heuristic), respectively. And, we denote the running times of these methods in seconds as t_{DP} , t_{CP} , and t_{HR} , respectively.

In our study, we let $U_0 = U_T = V_0 = V_T = 0$, and all other problem parameters be statistically independent of each other. We randomly generate cost parameters from uniform distributions, so that $\bar{\pi}_t \sim [\bar{\Delta}_L^P, \bar{\Delta}_U^P]$, $\bar{p}_t \sim [\bar{P}_L, \bar{P}_U]$, $\bar{\rho}_t \sim [\bar{\Delta}_L^R, \bar{\Delta}_U^R]$, $\bar{r}_t \sim [\bar{R}_L, \bar{R}_U]$, $\bar{\omega}_t \sim [\bar{\Delta}_L^O, \bar{\Delta}_U^O]$, $\bar{o}_t \sim [\bar{O}_L, \bar{O}_U]$, $\bar{g}_t \sim [\bar{G}_L, \bar{G}_U]$, and $\bar{h}_t \sim [\bar{H}_L, \bar{H}_U]$. In addition, we generate the demand level D_t from a Poisson random distribution with mean λ_D and the internal raw material supply level B_t from a Poisson random distribution with mean λ_B .

At their default levels, we let $\bar{\Delta}_L^P = 10.0$, $\bar{\Delta}_U^P = 15.0$, $\bar{P}_L = 3.0$, $\bar{P}_U = 5.0$, $\bar{\Delta}_L^R = 3.0$, $\bar{\Delta}_U^R = 5.0$, $\bar{R}_L = 1.0$, $\bar{R}_U = 2.0$, $\bar{\Delta}_L^O = 15.0$, $\bar{\Delta}_U^O = 20.0$, $\bar{O}_L = 4.0$, $\bar{O}_U = 7.0$, $\bar{G}_L = 0.5$, $\bar{G}_U = 1.0$, $\bar{H}_L = 1.0$, $\bar{H}_U = 2.0$, $\lambda_D = 10.0$, and $\lambda_B = 5.0$. Our computation was done on a Dell OPTIPLEX 745 desktop with a Core 2 CPU @ 2.13GHz using CPLEX 11.0.

We first make a rough-cut comparison among all three methods on small-scale problems. For each T , we independently generate 30 instances, to which we apply all three methods. The results are presented in Table 1. Here, every entry is the sample mean from its corresponding 30 independent instances.

Table 1 shows that the DP algorithm and CPLEX both produce optimal solutions as expected, while in so doing the former takes much more time than the latter (the

Table 1: A Comparison of the Three Methods

T	t_{DP}	t_{CP}	t_{HR}	c_{DP}	c_{CP}	c_{HR}
10	9.61	0.01	< 0.01	603.28	603.28	608.65
12	30.86	0.02	< 0.01	728.83	728.83	738.12
14	65.02	0.03	< 0.01	845.12	845.12	856.94
16	140.71	0.04	0.01	971.07	971.07	982.45
18	276.19	0.05	0.02	1091.13	1091.13	1106.24

more so as T becomes larger). Moreover, the heuristic is much faster than CPLEX and yet produces solutions that are close to the optimal ones.

Now, we concentrate on our main study of comparing the heuristic against CPLEX over larger-scale instances. We shall use the gap of optimality $\epsilon = (c_{HR} - c_{CP})/c_{HR}$ to measure the closeness of a solution produced by the heuristic to the optimal one produced by CPLEX. Also in this study, we fix the relationships $\lambda_D = 2\lambda_B$ and $\bar{H}_L = \bar{H}_U = 2\bar{G}_L = 2\bar{G}_U$, and leave all other parameters regarding the costs of production, raw material acquisition, and outsourcing at their default values. In Table 2, we present, for each set of parameters, the sample averages and sample standard deviations for t_{CP} , t_{HR} , and $(100 \times \epsilon)\%$ over 10 independent runs. Each entry for one of the three measures is written in the form of *average \pm standard deviation*.

From Table 2, we see that in most situations, the difference between solutions generated by our heuristic and the true optimal solution is within a 10% factor.

As \bar{G}_U grows along with other inventory holding costs, the optimality gap ϵ decreases. As inventory holding costs increase, an optimal solution tends to have more setups and more sequences with shorter lengths; as a consequence, the difference between the (exponential) number of all allowable sequences and the (polynomial) number of sequences considered by the heuristic on each sub-horizon tends to be smaller, and therefore the polynomial heuristic mimics the optimal solution even better.

As λ_B increases along with λ_D , we see that the gap ϵ drops fairly quickly. This is because higher supply and demand levels make four of the six patterns not considered by the heuristic, (uv) , $(\bar{u}\bar{v})$, $(u\bar{v})$, and $(\bar{u}v)$, which allow only inventory transfers, less likely to appear in the optimal solution. Hence, the heuristic solution can resemble the true optimal solution more closely.

As T grows, t_{CP} grows much faster than t_{HR} . Also, t_{CP} becomes much more unpredictable (as expressed by the increase in its standard deviation). At the same

Table 2: Comparing the Heuristic against CPLEX

T	λ_B	\bar{G}_U	t_{CP}	t_{HR}	$(100 \times \epsilon)\%$
30	5.0	0.25	0.31 ± 0.33	0.12 ± 0.01	$(10.11 \pm 1.22)\%$
30	5.0	0.50	0.50 ± 0.44	0.11 ± 0.01	$(5.28 \pm 0.76)\%$
30	5.0	0.75	0.49 ± 0.57	0.12 ± 0.01	$(2.76 \pm 0.72)\%$
30	5.0	1.00	0.27 ± 0.32	0.12 ± 0.01	$(1.29 \pm 0.58)\%$
30	5.0	1.25	0.12 ± 0.11	0.12 ± 0.01	$(0.62 \pm 0.41)\%$
30	2.0	0.75	2.05 ± 3.66	0.12 ± 0.01	$(9.68 \pm 1.40)\%$
30	3.0	0.75	1.31 ± 1.98	0.12 ± 0.01	$(6.34 \pm 1.03)\%$
30	8.0	0.75	0.11 ± 0.08	0.12 ± 0.01	$(1.14 \pm 0.48)\%$
30	10.0	0.75	0.08 ± 0.08	0.12 ± 0.01	$(0.80 \pm 0.41)\%$
40	5.0	0.25	1.19 ± 1.46	0.35 ± 0.01	$(10.49 \pm 0.99)\%$
40	5.0	0.50	4.12 ± 6.06	0.35 ± 0.01	$(5.37 \pm 0.77)\%$
40	5.0	0.75	8.44 ± 17.40	0.35 ± 0.01	$(2.76 \pm 0.65)\%$
40	5.0	1.00	3.15 ± 9.94	0.35 ± 0.01	$(1.33 \pm 0.49)\%$
40	5.0	1.25	0.50 ± 0.98	0.35 ± 0.01	$(0.61 \pm 0.33)\%$
40	2.0	0.75	94.37 ± 221.85	0.35 ± 0.11	$(10.12 \pm 1.31)\%$
40	3.0	0.75	27.12 ± 61.12	0.35 ± 0.01	$(6.49 \pm 1.01)\%$
40	8.0	0.75	0.55 ± 1.18	0.35 ± 0.01	$(1.10 \pm 0.38)\%$
40	10.0	0.75	0.22 ± 0.30	0.35 ± 0.01	$(0.82 \pm 0.30)\%$
50	5.0	0.75	82.46 ± 216.74	0.82 ± 0.01	$(2.77 \pm 0.68)\%$
60	5.0	0.75	796.11 ± 2433.11	1.66 ± 0.01	$(2.84 \pm 0.54)\%$

time, t_{HR} grows at a much slower rate and remains very predictable.

Noting further that CPLEX works only under the setup-linear setting while the heuristic works under all settings, we may conclude by saying that the heuristic is a stable, efficient, and effective method for solving PPPRS. The advantage of the heuristic is particularly evident when the planning horizon is long, system throughput is high, or inventory handling is costly.

10 Concluding Remarks

We have studied a single-item production planning problem which takes into account the activities of internal raw material supply, acquisition, and outsourcing. We have found solution algorithms for the problem under various assumptions, and given a fairly clear picture of the problem's complexity. Our computational study indicates that a heuristic we developed can achieve reasonably good results in much shorter time than a commercial MIP solver.

We believe that future research is needed for the following extensions of the problem: more than one type of raw material may be needed in the production process; there may be more than one external source of raw material supply; the production process may involve more than one stage, and externally supplied components may be needed for intermediate stages; and, there may be more than one external firm to turn to for outsourcing.

Acknowledgments

Jian Yang was supported by NSF Grant CMMI-0652942, and Xiangtong Qi was supported by the Hong Kong RGC CERG Grant 618606.

References

- Aggarwal, A. and J.K. Park. 1993. Improved Algorithm for Economic Lot Size Problems. *Operations Research*, **41**, pp. 549-571.
- Aksen, D., K. Altinkemer, and S. Chand. 2003. The Single-item lot sizing Problem with Immediate Lost Sales. *European Journal of Operational Research*, **147**, pp. 558-566.

- Anily, S. and M. Tzur. 2006. Algorithms for the Multi-item Multi-vehicles Dynamic Lot Sizing Problem. *Naval Research Logistics*, **53**, pp. 157-169.
- Atamtürk, A. and D.S. Hochbaum. 2001. Capacity Acquisition, Subcontracting, and Lot Sizing. *Management Science*, **47**, pp. 1081-1100.
- Atamtürk, A. and S. Kucukyavuz. 2005. Lot Sizing with Inventory Bounds and Fixed Costs: Polyhedral Study and Computation. *Operations Research*, **53**, pp. 711-730.
- Bitran, G.R. and H. Matsuo. 1986. Approximation Formulations for the Single-Product Capacitated Lot Size Problem. *Operations Research*, **34**, pp. 63-74.
- Bitran, G.R. and H.H. Yanasse. 1982. Computational Complexity of the Capacitated Lot Size Problem. *Management Science*, **28**, pp. 1174-1186.
- Erickson, R.E., C.L. Monma, and A.F. Veinott. 1987. Send-and-split Method for Minimum-concave-cost Network Flows. *Mathematics of Operations Research*, **12**, pp. 634-664.
- Federgruen, A. and M. Tzur. 1991. A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n Periods in $O(n \log n)$ or $O(n)$ Time. *Management Science*, **37**, pp. 909-925.
- Florian, M. and M. Klein. 1971. Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science*, **18**, pp. 12-20.
- Florian, M., J.K. Lenstra, and A.H.G. Rinnooy Kan. 1980. Deterministic Production Planning: Algorithm and Complexity. *Management Science*, **26**, pp. 669-679.
- Ganas, I. and S. Papachristos. 2005. The Single-product lot sizing Problem with Constant Parameters and Backlogging: Exact Results, a New Solution, and All Parameter Stability Regions. *Operations Research*, **53**, pp. 170-176.
- Gavish, B. and R.E. Johnson. 1990. A Fully Polynomial Approximation Scheme for Single Product Scheduling in a Finite Capacity Facility. *Operations Research*, **38**, pp. 70-83.
- Gopalakrishnan, M., K. Ding, J.-M. Bourjolly, and S. Mohan. 2001. A Tabu-Search Heuristic for the Capacitated Lot-Sizing Problem with Set-up Carryover. *Management Science*, **47**, pp. 851-863.
- Hsu, V.N., C.-L. Li, and W. Xiao. 2005. Dynamic Lot Size Problem with One-way Product Substitution. *IIE Transactions*, **37**, pp. 201-215.

- Jaruphongsa, W., S. Cetinkaya, and C.-Y. Lee. 2004. A Two-echelon Inventory Optimization Model with Demand Time Window Considerations. *Journal of Global Optimization*, **30**, pp. 347-366.
- Jaruphongsa, W., S. Cetinkaya, and C.-Y. Lee. 2005. A Dynamic Lot-sizing Model with Multi-mode Replenishments: Polynomial Algorithms for Special Cases with Dual and Multiple Modes, *IIE Transactions*, **37**, pp. 453-467.
- Kaminsky, P. and D. Simchi-Levi. 2003. Production and Distribution Lot Sizing in a Two Stage Supply Chain. *IIE Transactions*, **35**, pp. 1065-1075.
- Lee, C.-Y., S. Cetinkaya, and W. Jaruphongsa. 2003. A Dynamic Model for Inventory Lot Sizing and Outbound Shipment Scheduling at a Third-Party Warehouse. *Operations Research*, **51**, pp. 735-747.
- Lee, C.-Y., S. Cetinkaya, and A.P.M. Wagelmans. 2001. A Dynamic Lot Sizing Model with Demand Time Windows. *Management Science*, **47**, pp. 1384-1395.
- Li, C.-L., V.N. Hsu, and W. Xiao. 2004. Dynamic Lot Sizing with Batch Ordering and Truckload Discounts. *Operations Research*, **52**, pp. 639-654.
- Merzifonluoğlu, Y., J. Geunes, and H.E. Romeijn. 2007. Integrated Capacity, Demand, and Production Planning with Subcontracting and Overtime Options. *Naval Research Logistics*, **54**, pp. 433-447.
- Sandbothe, R.A. and G.L. Thompson. 1990. A Forward Algorithm for the Capacitated Lot Size Model with Stockouts. *Operations Research*, **38**, pp. 474-486.
- Sargut, F.Z. and H.E. Romeijn. 2007a. Capacitated Production and Subcontracting in a Serial Supply Chain. *IIE Transactions*, **39**, pp. 1031-1043.
- Sargut, F.Z. and H.E. Romeijn. 2007b. Lot-sizing with Non-stationary Cumulative Capacities. *Operations Research Letters*, **35**, pp. 549-557.
- Stadtler, H. 2003. Multilevel Lot Sizing with Setup Times and Multiple Constrained Resources: Internally Rolling Schedules with Lot Sizing Windows. *Operations Research*, **51**, pp. 487-502.
- Teo, C.-P. and D. Bertsimas. 2001. Multistage Lot Sizing Problems via Randomized Rounding. *Operations Research*, **49**, pp. 599-608.
- Teunter, R.H., Z.P. Bayindir, and W. van den Heuvel. 2006. Dynamic Lot Sizing with production Returns and Remanufacturing. *International Journal of Production Research*, **44**, pp. 4377-4400.

- van den Heuvel. 2006. *The Economic Lot-sizing Problem: New Results and Extensions*. Ph.D. Thesis. Erasmus Research Institute of Management, Erasmus University, Rotterdam, the Netherlands.
- van den Heuvel, W. and A.P.M. Wagelmans. 2008. Four Equivalent Lot-sizing Models. *Operations Research Letters*, **36**, pp. 465-470.
- van Hoesel, C.P.M. and A.P.M. Wagelmans. 1996. An $O(T^3)$ Algorithm for the Economic Lot Sizing Problem with Constant Capacities. *Management Science*, **42**, pp. 142-150.
- van Hoesel, C.P.M. and A.P.M. Wagelmans. 2001. Fully Polynomial Approximation Schemes for Single-item Capacitated Economic Lot-sizing Problems. *Mathematics of Operations Research*, **26**, pp. 339-357.
- van Hoesel, S., H.E. Romeijn, D.R. Morales, and A.P.M. Wagelmans. 2005. Integrated Lot Sizing in Serial Supply Chains with Production Capacities. *Management Science*, **51**, pp. 1706-1719.
- Wagelmans, A.P.M., S. van Hoesel, and A. Kolen. 1992. An $O(n \log n)$ Algorithm that Runs in Linear Time in the Wagner-Whitin Case. *Operations Research*, **40**, pp. S145-S156.
- Wagner, H.M. 1960. A Postscript to "Dynamic Problems in the Theory of the Firm." *Naval Research Logistics Quarterly*, **7**, pp. 7-12.
- Wagner, H.M. and T.M. Whitin. 1959. Dynamic Version of the Economic Lot Size Model. *Management Science*, **5**, pp. 89-96.
- Wolsey, L.A. 2002. Solving Multi-Item Lot-Sizing Problems with an MIP Solver Using Classification and Reformulation. *Management Science*, **48**, pp. 1587-1602.
- Yang, J., B. Golany, and G. Yu. 2005. A Concave-cost Production Planning Problem with Remanufacturing Options. *Naval Research Logistics*, **52**, pp. 443-458.
- Zangwill, W.I. 1969. A Backlogging Model and a Multiechelon Model of a Dynamic Economic Lot Size Production System-A Network Approach. *Management Science*, **15**, pp. 506-527.

Appendix A: Conditions for Feasibility

Let us examine the sub-network made up of all the I_t and J_t nodes and all existing arcs linking them. If we treat flows in the (O, I_t) and (O, J_t) directions as in-flows and those in opposite directions as out-flows, then this sub-network has un-capacitated arcs, and it allows unlimited in-flows while no out-flows. Hence, PPPRS will be feasible, or equivalently, have a finite cost, if and only if each of the sub-network's cut sets $(\mathcal{A}, \mathcal{B})$ with arcs in only the $\mathcal{A} \rightarrow \mathcal{B}$ direction allows positive flows from \mathcal{A} to \mathcal{B} .

As the only arcs in the sub-network are of forms (I_t, I_{t+1}) , (I_t, J_t) , and (J_t, J_{t+1}) , such a cut set must be of the form that $\mathcal{A} = \{I_1, \dots, I_s, J_1, \dots, J_t\}$ and $\mathcal{B} = \{I_{s+1}, \dots, I_T, J_{t+1}, \dots, J_T\}$ for some $s, t = 0, 1, \dots, T$ and $s \geq t$. We can translate these requirements into the following:

- a) $V_T + \bar{D}_{1T} - V_0 - (U_0 + \bar{B}_{1T} - U_T)^+ \geq 0$;
- b) for $t = 1, 2, \dots, T-1$, $\bar{D}_{t+1,T} + V_T - (U_0 + V_0 + \bar{B}_{1T} - \bar{D}_{1t} - U_T)^+ \geq 0$;
- c) for $s = 1, 2, \dots, T-1$, $U_T + V_T + \bar{D}_{1T} - \bar{B}_{s+1,T} - V_0 - (U_0 + \bar{B}_{1s})^+ \geq 0$;
- d) for $1 \leq t \leq s \leq T-1$, $U_T + V_T + \bar{D}_{t+1,T} - \bar{B}_{s+1,T} - (U_0 + V_0 + \bar{B}_{1s} - \bar{D}_{1t})^+ \geq 0$.

Appendix B: A Pseudo-polynomial DP Algorithm

For $t = 1, \dots, T+1$, let $F_t(u_{t-1}, v_{t-1})$ be the minimum cost of PPPRS on the sub-horizon $\{t, \dots, T\}$ with given initial raw material and finished product inventory levels u_{t-1} and v_{t-1} , respectively. The DP algorithm is based on recursive relationships between different $F_t(u_{t-1}, v_{t-1})$'s.

For the terminal condition, we let

$$F_{T+1}(u_T, v_T) = \begin{cases} 0 & \text{when } u_T = U_T \text{ and } v_T = V_T, \\ +\infty & \text{otherwise.} \end{cases} \quad (31)$$

As for recursive relationships, we have, for $t = T, T-1, \dots, 2, 1$, for $v_{t-1} = 0, 1, \dots, \bar{D}_{tT} + V_T$ and $u_{t-1} = 0, 1, \dots, \bar{D}_{tT} - \bar{B}_{tT} + U_T + V_T - v_{t-1}$ when $t \geq 2$ and $u_0 = U_0$ and $v_0 = V_0$ when $t = 1$,

$$\begin{aligned} F_t(u_{t-1}, v_{t-1}) &= \min\{P_t(x) + R_t(y) + S_t(z) + G_t(u_{t-1} + B_t - x + y) \\ &\quad + H_t(v_{t-1} + x + z - D_t) \\ &\quad + F_{t+1}(u_{t-1} + B_t - x + y, v_{t-1} + x + z - D_t) \\ &\quad \mid x = 0, 1, \dots, \bar{D}_{tT} + V_T - v_{t-1}, \\ &\quad y = 0, 1, \dots, \bar{D}_{tT} - \bar{B}_{tT} + U_T + V_T - u_{t-1} - v_{t-1}, \\ &\quad z = 0, 1, \dots, \min\{\bar{D}_{tT} + V_T - v_{t-1} - x, \\ &\quad \bar{D}_{tT} - \bar{B}_{tT} + U_T + V_T - u_{t-1} - v_{t-1} - y\}\}. \end{aligned} \quad (32)$$

We will obtain the optimal solution for PPPRS by computing $F_1(U_0, V_0)$ using the above dynamic programming procedure. In total, we need to calculate the $F_t(u_{t-1}, v_{t-1})$'s for $O(T)$ periods, and for each period t , we need to calculate $O((\bar{D}_{1T} + V_T) \cdot (\bar{D}_{1T} + U_T + V_T))$ of them. To calculate each $F_t(u_{t-1}, v_{t-1})$, we need to make $O((\bar{D}_{1T} + V_T) \cdot (\bar{D}_{1T} + U_T + V_T)^2)$ comparisons.

Appendix C: Proof of Theorem 3: We again reduce SUBSET SUM to the special case. Given any instance of SUBSET SUM, we define an instance of the $G_P G_O l_R$ case with everything else being the same as in the proof of Theorem 1 and the exception that for $n = 1, \dots, N$, the production cost $P_n(x) = 1 \cdot \mathbf{1}(x > 0)$ and the outsourcing cost $O_n(z) = 1 \cdot \mathbf{1}(z > 0)$. Note that such a cost structure satisfies all conditions (C1) to (C4). We are to show that the answer to the SUBSET SUM instance is *yes* if and only if the optimal cost to the PPPRS instance is no more than N .

The *if* part: Suppose the PPPRS instance costs no more than N optimally. Note that the production or outsourcing cost imputable to one item does not exceed 1, while acquisition and finished product inventory holding both cost 2 per item. So, in an optimal solution, no finished product has ever to come from either of the latter two activities. Without acquisition, the total production level in the solution must be exactly A . However, the cost upper-bound disallows simultaneous production and outsourcing activities in any one period. So, there must be a subset \mathcal{S} of $\{1, \dots, N\}$ with $\sum_{n \in \mathcal{S}} a_n = A$, such that in the optimal solution, production takes place in periods in \mathcal{S} and outsourcing takes place in periods in $\{1, \dots, N\} \setminus \mathcal{S}$.

The *only if* part: Suppose there is a subset \mathcal{S} of $\{1, 2, \dots, N\}$ such that $\sum_{n \in \mathcal{S}} a_n = A$. If we produce in periods in \mathcal{S} and outsource in periods in $\{1, \dots, N\} \setminus \mathcal{S}$, which is feasible, we will incur total cost N . ■

Appendix D: Statements and Proofs of Properties

GGG-Property 1 *Suppose in both periods s and t where $s < t$, production levels $x_s^* > 0$ and $x_t^* > 0$. Then, we may assume that there is a period $\tau = s, \dots, t - 1$ such that the finished product inventory level $v_\tau^* = 0$.*

Proof: If there is not, by decreasing the production level in period s by 1 and increasing the production level in period t by 1, effectively sending a unit of flow through the cycle $I_s \rightarrow I_t \rightarrow J_t \rightarrow J_s \rightarrow I_s$, we would not change the feasibility of the solution and would only potentially save costs due to (C1). Repeating this argument, if necessary, leads

to either $x_s^* = 0$ or $v_\tau^* = 0$ for some $\tau = s, \dots, t-1$. Since the former is not true, the latter must be the case. ■

GGG-Property 2 *Suppose in both periods s and t where $s < t$, acquisition levels $y_s^* > 0$ and $y_t^* > 0$. Then, we may assume that there is a period $\tau = s, \dots, t-1$ such that the raw material inventory level $u_\tau^* = 0$.*

Proof: If there is not, by decreasing the acquisition level in period s by 1 and increasing the acquisition level in period t by 1, effectively sending a unit of flow through the cycle $O \rightarrow I_t \rightarrow I_s \rightarrow O$, we would not change the feasibility of the solution and would only potentially save costs due to (C2). ■

GGG-Property 3 *Suppose in both periods s and t where $s < t$, outsourcing levels $z_s^* > 0$ and $z_t^* > 0$. Then, we may assume that there is a period $\tau = s, \dots, t-1$ such that the finished product inventory level $v_\tau^* = 0$.*

Proof: If there is not, by decreasing the outsourcing level in period s by 1 and increasing the outsourcing level in period t by 1, effectively sending a unit of flow through the cycle $O \rightarrow J_t \rightarrow J_s \rightarrow O$, we would not change the feasibility of the solution and would only potentially save costs due to (C3). ■

GGG-Property 4 *Suppose for some $s, t, t' = 1, \dots, T$ that, $u_s^*, u_{s+1}^*, \dots, u_{t-1}^* > 0$ in case $s \leq t$, while $u_t^*, u_{t+1}^*, \dots, u_{s-1}^* > 0$ in case $s \geq t$; and $v_t^*, v_{t+1}^*, \dots, v_{t'-1}^* > 0$ in case $t \leq t'$, while $v_{t'}^*, v_{t'+1}^*, \dots, v_{t-1}^* > 0$ in case $t \geq t'$. Then $y_s^* \cdot x_t^* \cdot z_{t'}^* = 0$.*

Proof: If otherwise, there would be a loop in the basis of the solution, connecting node O with nodes I_s, I_{s+1}, \dots , and I_t or nodes I_t, I_{t+1}, \dots , and I_s , and with nodes J_t, J_{t+1}, \dots , and $J_{t'}$ or nodes $J_{t'}, J_{t'+1}, \dots$, and J_t . ■

GGI-Property 1 *For any $t = 1, \dots, T-1$, we may assume that $y_t^* \cdot u_t^* = 0$.*

Proof: If both u_t^* and y_t^* are positive, by decreasing the acquisition level in period t by 1 and increasing the acquisition level in period $t+1$ by 1, we would not change the feasibility of the solution, and would only potentially save costs due to the linearity of acquisition costs and (C2). ■

GIG-Property 1 *For any $t = 1, \dots, T-1$, we may assume that $z_t^* \cdot v_t^* = 0$.*

Proof: If both v_t^* and z_t^* are positive, by decreasing the outsourcing level in period t by 1 and increasing the outsourcing level in period $t+1$ by 1, we would not change the feasibility of the solution, and would only potentially save costs due to the linearity of outsourcing costs and (C3). ■

GIG-Property 2 *If t is not the first period when there is production or outsourcing and if the production level $x_t^* > 0$, we may assume that the finished product inventory level $v_{t-1}^* = 0$.*

Proof: If $v_{t-1}^* \geq 1$, let t' be the last period before t when there is a positive level of either outsourcing or production. There must be positive finished product inventory levels between periods t' and t . According to GIG-Property 1, only production can occur in period t' . Now, by decreasing the production level in period t' by 1 and increasing the production level in period t by 1, we would not change the feasibility of the solution and would only potentially save costs due to (C1). ■

IGG-Property 1 *For any $t = 1, \dots, T-1$, we may assume that $x_t^* \cdot v_t^* = 0$.*

Proof: If both v_t^* and x_t^* are positive, by decreasing the production level in period t by 1 and increasing the production level in period $t+1$ by 1, we would not change the feasibility of the solution, and would only potentially save costs due to (C1). ■

IIIG-Property 1 *For any $t = 1, \dots, T-1$, we may assume that the finished product inventory holding level $v_t^* = 0$.*

Proof: If $v_t^* > 0$, let us focus on any one unit of the finished product being carried between periods t and $t+1$. This unit must be either produced or outsourced in a period no later than period t . Note that both production and outsourcing costs are linear. In the first case, by delaying the production of the unit to period $t+1$, we would not change the feasibility of the solution, but would potentially save costs due to (C1); while in the second case, by delaying the outsourcing of the unit to period $t+1$, we would not change the feasibility of the solution, but would potentially save costs due to (C3). ■

IIIG-Property 2 *For any $t = 1, \dots, T-1$ and $t' = t+1, \dots, T$, we may assume that $z_t^* \cdot u_t^* \cdots u_{t'-1}^* \cdot x_{t'}^* = 0$.*

Table 3: Feasible Pattern Transitions for PPPRS

	(uz)	(ux)	$(\bar{u}z)$	$(\bar{u}x)$	(xv)	(yv)	$(x\bar{v})$	$(y\bar{v})$	(uv)	$(u\bar{v})$	$(\bar{u}\bar{v})$	$(\bar{u}v)$	(xz)	(yz)	(yx)
(uz)	Y	Y	r	r	Y	Y	Y	Y	Y	Y	r	r	Y	Y	Y
(ux)	Y	Y	r	r	Y	Y	Y	Y	Y	Y	r	r	Y	Y	Y
$(\bar{u}z)$	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
$(\bar{u}x)$	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
(xv)	Y	Y	Y	Y	Y	Y	r	r	Y	r	r	Y	Y	Y	Y
(yv)	Y	Y	Y	Y	Y	Y	r	r	Y	r	r	Y	Y	Y	Y
$(x\bar{v})$	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
$(y\bar{v})$	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
(uv)	Y	Y	r	r	Y	Y	r	r	Y	r	r	r	Y	Y	Y
$(u\bar{v})$	Y	Y	r	r	Y	Y	Y	Y	Y	Y	r	r	Y	Y	Y
$(\bar{u}\bar{v})$	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
$(\bar{u}v)$	Y	Y	Y	Y	Y	Y	r	r	Y	r	r	Y	Y	Y	Y
(xz)	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
(yz)	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d
(yx)	d	d	Y	Y	d	d	Y	Y	d	Y	Y	Y	d	d	d

Proof: If otherwise, due to the linearity of both outsourcing and production costs, by decreasing the outsourcing level in period t by 1, increasing the production level in period t by 1, increasing the outsourcing level in period t' by 1, and decreasing the production level in period t' by 1, we would not change the feasibility of the solution, and would only potentially save costs due to (C4). ■

Appendix E: The Maximally Polynomial Sets

From previous definitions, we see that two flow patterns can appear in a common sequence consecutively if and only if they are connected by at least one inventory arc and have no redundancy in arcs, while redundancy in arcs occurs when two consecutive patterns happen to use the same arc. Table 3 presents the exhaustive result regarding whether a column pattern can immediately follow a row pattern in a sequence. In the table, a “Y” entry means this is possible; a “d” entry means the two periods are not connected by any inventory arc from the two patterns; and a “r” entry means the two patterns have redundant arcs. For example, Figure 5 illustrates that pattern (uz) can be immediately followed by (xz) , while $(\bar{u}x)$ cannot be immediately followed by (xv) .

Also, not all flow patterns can start or end a sequence by our definition of a sub-horizon. Hence we have the following.

Theorem 4 *A sequence must start with one of the patterns (uz) , (ux) , (xv) , (yv) , (uv) , (xz) , (yz) , or (yx) and end with one of the patterns $(\bar{u}z)$, $(\bar{u}x)$, $(x\bar{v})$, $(y\bar{v})$, $(\bar{u}\bar{v})$, (xz) , (yz) , or (yx) . However, the (yv) or (uv) pattern can only start the first sequence in the case of $V_0 \geq D_1 + 1$, while the $(\bar{u}z)$ or $(\bar{u}\bar{v})$ pattern can only end the last sequence in the case of $U_T \geq B_T + 1$.*

The proof can use the same idea used in the proof for Proposition 2 in Yang, Golany, and Yu (2005). We omit it here.

For any set A , we use $|A|$ to denote its cardinality. For any $i, j \in I^0$, let $a(i, j)$ be 1 when the (i, j) entry in Table 3 is “Y” and 0 otherwise. We let a semi-sequence be an ordered set $P = [i_1^P, i_2^P, \dots, i_{|P|}^P]$ of patterns that are connected and have no redundancy in arcs, i.e., an ordered set P with $\prod_{k=1}^{|P|-1} a(i_k^P, i_{k+1}^P) = 1$. For a semi-sequence P and a set $I \subset I^0$, we write $P \subset I$ when all patterns in P belong to I . Now, any sequence P is merely a semi-sequence whose first member pattern can start a sequence and last member pattern can end a sequence: $i_1^P \in I_S^0$ and $i_{|P|}^P \in I_E^0$.

Let \mathcal{I}_1 be all subsets of I^0 whose member patterns can form at least one sequence. For any set $I \in \mathcal{I}_1$, we say a given member pattern $i \in I$ is contributing if and only if it helps form a sequence, i.e., there exists a sequence $P \subset I$ and an integer $k \in \{1, \dots, |P|\}$ such that $i = i_k^P$. For a given $I \in \mathcal{I}_1$, we say I is compact if and only if every pattern belonging to it is contributing. Also, for a given $I \in \mathcal{I}_1$ and a positive integer τ , we let $N_I(\tau)$ be the total number of distinct sequences of length τ that patterns in I can form. For any compact $I \in \mathcal{I}_1$, we say it is polynomial when for a sufficiently large $\alpha \in [0, +\infty)$,

$$\limsup_{\tau \rightarrow +\infty} \frac{N_I(\tau)}{\tau^\alpha} < +\infty;$$

and we say I is exponential when the opposite is true: no matter how large $\alpha \in [0, +\infty)$ is,

$$\limsup_{\tau \rightarrow +\infty} \frac{N_I(\tau)}{\tau^\alpha} = +\infty.$$

The following result from Yang, Golany, and Yu (2005, Proposition 3) gives an easy-to-check sufficient and necessary condition for a compact set I to be exponential or polynomial.

Theorem 5 *Given a compact $I \in \mathcal{I}_1$, the following three statements are equivalent:*

- 1) I is exponential;

2) there exists $i \in I$ and two different semi-sequences with members in I , both starting from and ending at i ;

3) there exist $i, j_1, j_2 \in I$, such that $j_1 \neq j_2$, $a(i, j_1) = a(i, j_2) = 1$, and there exist semi-sequences $P_1 = [i_1^{P_1} = j_1, i_2^{P_1}, \dots, i_{|P_1|}^{P_1} = i] \subset I$ and $P_2 = [i_1^{P_2} = j_2, i_2^{P_2}, \dots, i_{|P_2|}^{P_2} = i] \subset I$.

A corollary of Theorem 5 is immediately in order.

Corollary 1 *If a compact I is polynomial, then no two-element subset $\{i, j\}$ of I can be such that $a(i, i) = a(i, j) = a(j, i) = 1$.*

For any $I \subset I^0$, we use $A(I)$ to denote the matrix that comprises all the $a(i, j)$'s for $i, j \in I$. Also, define $I^1 = \{(uz), (ux), (xv), (yv), (uv)\}$ and $I^2 = \{(\bar{u}z), (\bar{u}x), (x\bar{v}), (y\bar{v}), (\bar{u}\bar{v})\}$.

Theorem 6 *Suppose I is a polynomial subset of I^0 , then*

- 1) *At most one pattern in I^1 can be in I and at most one pattern in I^2 can be in I ;*
- 2) *Neither $(u\bar{v})$ nor $(\bar{u}v)$ can be in I .*

The proof can use the same idea used in the proof for Proposition 4 in Yang, Golany, and Yu (2005). We omit it here.

We say that I is maximally polynomial if any compact I' containing I is exponential. We are now in a position to identify all maximally polynomial sets (MPSs).

Theorem 7 *PPPRS has exactly 9 MPSs. Each such set contains 5 patterns, one of (uz) , (ux) , and (xv) , all (xz) , (yz) , and (yx) , and one of $(\bar{u}x)$, $(x\bar{v})$, and $(y\bar{v})$.*

The proof can use the same idea as the one used in the proof of Proposition 5 in Yang, Golany, and Yu (2005). We omit it here.

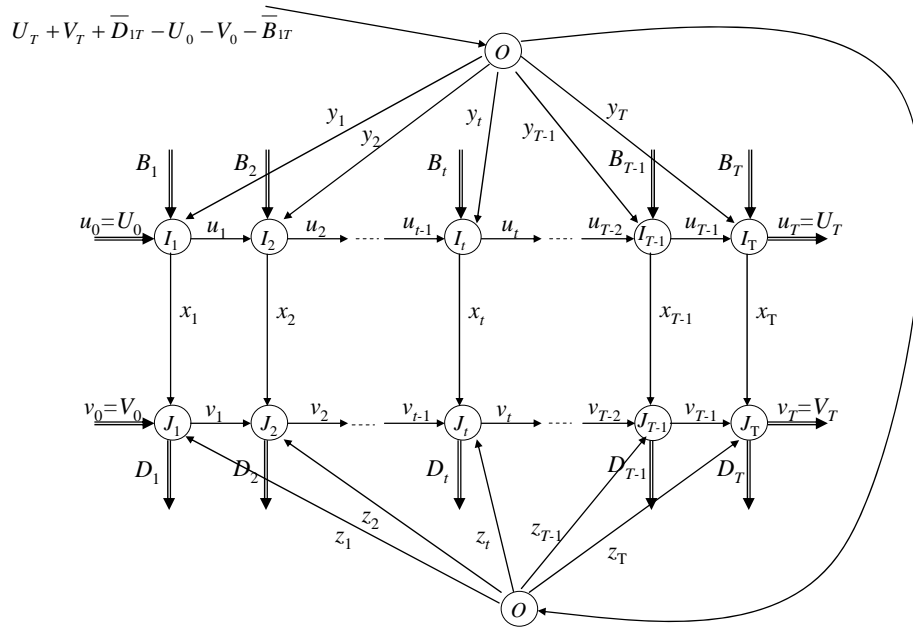
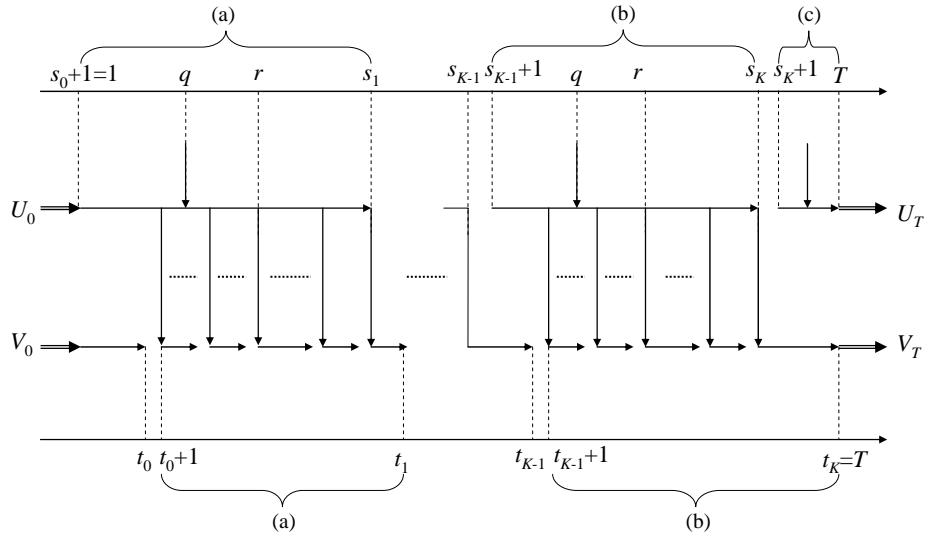
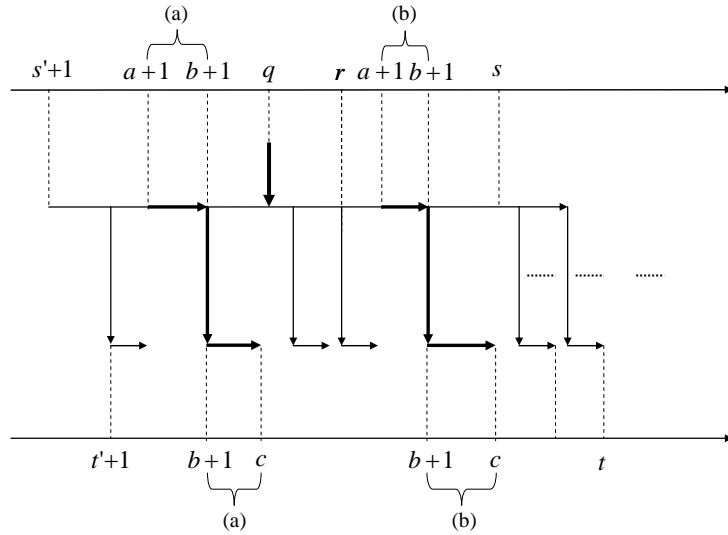


Figure 1. Network flow description of the problem



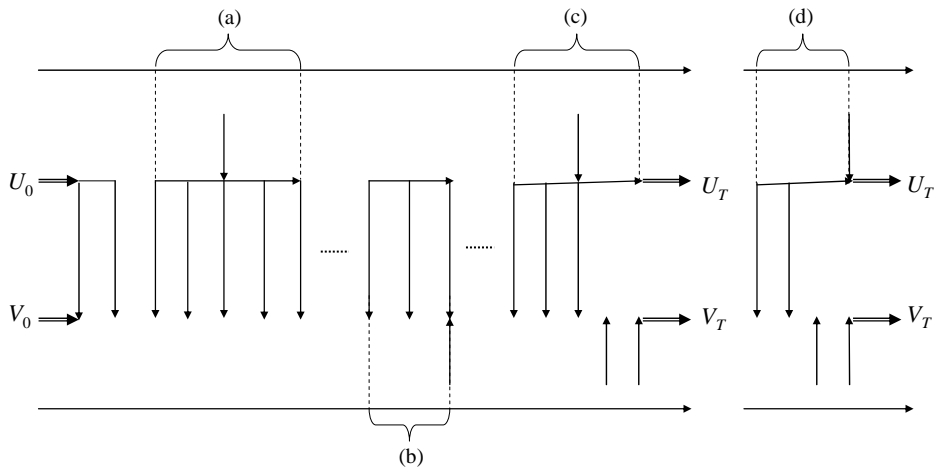
- (a) SDH $[s_0, s_1; t_0, t_1]$
- (b) SDH $[s_{K-1}, s_K; t_{K-1}, t_K]$
- (c) RDH $[s_K, T; T, T]$

Figure 2. Structure of the optimal solution for the $G_{p^n} O G_R$ case



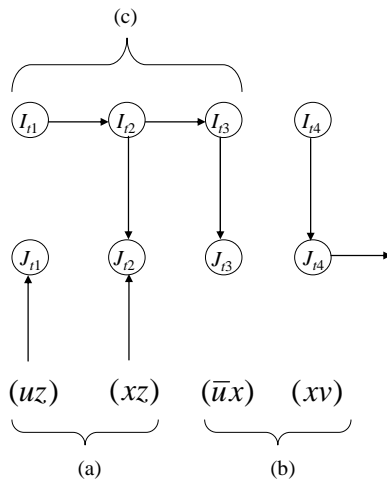
- (a) An early type SSDH $[a, b+1; b, c]$
- (b) A late type SSDH $[a, b+1; b, c]$

Figure 3. SSDHs for SDH $[s', s; t', t]$ in $Y_S G_P n_O G_R$ case



- (a) An R -sub-horizon with one raw material acquisition
- (b) An O -sub-horizon with one outsourcing at the last period
- (c) A residue sub-horizon with raw material acquisition not at the last period
- (d) A residue sub-horizon with raw material acquisition at the last period

Figure 4. Structure of the optimal solution for the $I_P^l O G_R$ case



- (a) Patterns (uz) and (xz) can be possibly connected
- (b) Patterns $(\bar{u}x)$ and (xv) cannot be connected
- (c) Type 1 loopless sequence

Figure 5. Demonstration of patterns and their relationship