An Improved Algorithm for the Deterministic Lot-Sizing Problem for General Multi-stage Production Systems

Woonghee Tim Huh
School of Operations Research and Industrial Engineering
Cornell University, Ithaca, NY 14853, USA
huh@orie.cornell.edu

July 18, 2002

Abstract

We consider the production and reorder lot sizing problem in the general multi-stage production-distribution system. If N is the number of stages in the system, the best known bound on the running time is equivalent to N computations of a maximum flow algorithm on N nodes. We present a modified divide-and-conquer algorithm that runs in the same asymptotic bound as a single push-relabel computation for a maximum flow problem on N nodes.

1 Introduction

We study a general multi-stage production-distribution system that is characterized by an arbitrary directed acyclic graph G with node set V and arc set A. Each stage of production or distribution corresponds to a node in V, and an arc $n_1 \to n_2 \in A$ implies that material flows from n_1 to n_2. Demand is deterministic and constant; consequently, there is a one-to-one correspondence between lot-sizes and production/reorder intervals at each stage. Let $K_n > 0$ be the fixed cost of ordering, and $g_n > 0$ be a constant related to the incremental echelon inventory cost at location n. The power-of-two reorder interval problem is

$$
\begin{aligned}
(P)\quad \min_{T_n : n \in V} \quad & \sum_{n \in V} K_n/T_n + g_n T_n \\
\text{s. t.} \quad & T_{n_1} \geq T_{n_2} \geq 0 \quad \forall\ (n_1, n_2) \in A \qquad (1.1)\\
& T_n = 2^{k_n} \cdot T_L \quad \forall\ n \in V. \qquad (1.2)
\end{aligned}
$$

We let (RP) be the problem obtained by discarding the constraint (1.2) from (P). The divide-and-conquer method for solving (RP) is presented in Maxwell and Muckstadt (1985), Roundy (1986), and Muckstadt and Roundy (1993) use it to construct an approximation algorithm that bounds error by six percent. This divide-and-conquer algorithm solves (RP) in $N{\cdot}T_{MF}$ time, where T_{MF} is the time complexity of the max-flow problem on a graph with N nodes and M arcs, and N and M are the cardinalities of V and A respectively. Using $T_{MF} = O(NM\log(N^2/M))$ as in Gallo et al. (1989), the running time is $O(N^2M\log(N^2/M))$.

In this paper, we modify the algorithm so that it runs in $O(NM\log(N^2/M))$, achieving an improvement of the factor N. The decrease in the running time is obtained by an algorithm that solves parametric minimum cut problems as shown by Gallo et al. (1989), Gusfield and Martel (1992) and Gusfield and Tardos (1994). If the directed graph is sparse (i.e. $M = O(N)$), then this running time becomes $O(N^2\log N)$, which compares to $O(N\log N)$ bound for certain structures of G (e.g., Roundy (1990), Jackson and Roundy (1991), and Roundy and Sun (1994)).

The result in this paper can be extended to solve the following problem

$$
\begin{aligned}
\min_{T_n : n \in V} \quad & \sum_{n \in V} f_n(T_n) \\
\text{s. t.} \quad & T_{n_1} \geq T_{n_2} \quad \forall\ (n_1, n_2) \in A,
\end{aligned}
$$

if it satisfies the following regularity assumption given in Jackson and Roundy (1991):

Assumption 1.1. *The functions $f_n(\cdot)$, $n \in V$, are convex on $\Re$. Also, there exists a function $T : 2^V \to \Re \cup \{\infty\}$, such that $T(C)$ uniquely minimize*

$$f(T, C) := \sum_{n \in C} f_n(T)$$

for each $C \subseteq V$. Furthermore, disjoint subsets C_1 and C_2 of V with $T(C_1) < T(C_2)$ satisfy

$$T(C_1) < T(C_1 \cup C_2) < T(C_2).$$

2 Divide-and-Conquer Algorithm

In this section, we present the *divide-and-conquer* method used in Jackson and Roundy (1991) and Picard and Queyranne (1985). For each n, the derivative of $f_n(t)$ is $f_n'(t) = g_n - \frac{K_n}{t^2}$. The convexity of f_n implies that f_n' is increasing.

For any subset C of V, we let $G(C)$ be the subgraph of G spanned by the nodes in C. For any partition V^+ and V^- of V, the value of a directed cut (V^+, V^-) at $t \in \Re$ is defined to be

$$v_t(V^+, V^-) = \begin{cases} \sum_{n \in V^+} f_n'(t) & \text{if no arc in } A \text{ goes from any node in } V^- \text{ to any node in } V^+ \\ \infty & \text{otherwise.} \end{cases}$$

As a part of the divide-and-conquer algorithm, we find a directed cut at a given t minimizing the value $v_t(\cdot, \cdot)$. To do so, we construct an augmented graph $G' = (V', A')$ by adding to G the source node s, the sink node d, and some arcs incident with s or d. In other words, let $V' = V \cup \{s, d\}$, and $A' = A \cup A_s \cup A_d$ where $A_s = \{(s, n) : n \in V\}$ and $A_d = \{(n, d) : n \in V\}$. Associated with each arc in A' is the arc capacity c_t at time t defined by

$$c_t(a) = \begin{cases} f_n'(t) & \text{if } a = (n, d) \in A_d \\ 0 & \text{if } a \in A \cup A_s. \end{cases}$$

This minimum cut problem is well studied in combinatorial optimization. If $(S(t), \overline{S(t)})$, where $s \in S(t)$, is the minimum directed cut of G' with respect to $c_t(\cdot)$, then $(S(t) - \{s\}, \overline{S(t)} - \{d\})$ minimizes $v_t(\cdot, \cdot)$.

For $C \subseteq V$, let

$$T(C) = \sqrt{\frac{\sum_{n \in C} g_n}{\sum_{n \in C} K_n}},$$

which minimizes $\sum_{n \in C} f_n(t)$. The divide-and-conquer algorithm finds the directed cut (V^+, V^-) with the minimum v value at $t = T(V)$. If $v_{T(V)}(V^+, V^-) < 0$, then we apply the divide-and-conquer algorithm recursively to each of two subgraphs spanned by V^+

and V^-. The algorithm terminates with a partition of nodes, and we assign $T_n = T(C)$ if C is in the partition and $n \in C$. The asymptotic running time of this algorithm is equivalent to N applications of the max-flow min-cut algorithm on a graph with N nodes and M arcs.

3 Monotonic Parametric Flow Network

We say a capacitated directed graph is a *monotonic parametric flow network* provided that each arc capacity is defined as a function of one parameter, and the capacities are nondecreasing on all arcs of the form (s, n), $n \neq d$, nonincreasing on all arcs of the form (n, d), $n \neq s$, and nonnegative constant on all other arcs with respect to the parameter. There is an efficient push-relabel algorithm to find minimum cuts of a monotonic parametric flow network corresponding to a set of parameter values.

Theorem 3.1. *In a monotonic parametric flow network with $O(N)$ nodes and $O(M)$ arcs, if parameter values $\lambda_1, \lambda_2, \ldots, \lambda_k$ are specified in any order, an on-line algorithm can compute the minimum cut in each of the networks specified by the parameter values $\lambda_1, \lambda_2, \ldots, \lambda_k$ in $O(N(M + k)\log(N^2/M))$ total time.*

Proof. For the case when all capacities are nonnegative, the proof of this theorem is given in Gusfield and Martel (1992) and Gusfield and Tardos (1994). We can relax the nonnegativity assumption on the arcs incident with the source or sink nodes by adding arcs if necessary, increasing arc capacities on (s, n) and (n, d) by the same amount, and adjusting cut values accordingly. □

The augmented graph G' is a monotone parametric flow network since $c_t(a)$ is constant with respect to t for all $a \in A \cup A_s$, and $c_t(a)$ is a nondecreasing function of t for all $a \in A_d$. Thus, Lemma 2.4 of Gallo et al. (1989) implies the following nesting property.

Lemma 3.2. *If $t_1 < t_2$, and $(V_1, \bar{V_1})$ and $(V_2, \bar{V_2})$ are the cuts minimizing υ_{t_1} and υ_{t_2} respectively, then $V_2 \subseteq V_1$.*

The *modified divide-and-conquer* algorithm takes a partition (L, V_o, R) of V. Initially, set $V_o = V$ and $L = R = \emptyset$. Each recursive step finds a cut of V_o. Using the monotone parametric flow network G' where the parameter t is set to $T(V_o)$, compute the cut (V^+, V^-) minimizing $\upsilon_{T(V_o)}$. We note that V^+ and V^- partition V. Throughout the algorithm, the invariance $L \subseteq V^+$ and $R \subseteq V^-$ is maintained. If $\upsilon_{T(V_o)}(V^+ \cap V_o, V^- \cap V_o) < 0$, we recursively call this algorithm on $(L, V^+ \cap V_o, V^-)$ and $(V^+, V^- \cap V_o, R)$. The invariance is maintained by Lemma 3.2 and Theorem 3.1.

Theorem 3.3. *The modified divided-and-conquer algorithm solves (RP) in* $O(NM \log(N^2/M))$ *time, which is the time complexity of the push-relabel algorithm for the maximum flow problem.*

Proof. The bound on the running time follows from Theorem 3.1 because there are at most $2N$ recursive calls. The correctness of this algorithm follows from the correctness of the divide-and-conquer algorithm. By Lemma 3.2, in each recursive step, the partition of V_o in the algorithm is the same as the partition of nodes in the subgraph spanned by V_o in the divide-and-conquer method. □

Acknowledgement

The author would like to thank Robin O. Roundy for his encouragement and his help in the preparation of this document. He also appreciates comments of Jack Muckstadt.

References

Gallo, G., Grigoriadis, M. D., and Tarjan, R. E. (1989, February), "A Fast Parametric Maximum Flow Algorithm and Applications," *SIAM Journal on Computing*, 18(1), 30–55.

Gusfield, D. and Martel, C. (1992), "A Fast Algorithm for the Generalized Parametric Minimum Cut Problem and Applications," *Algorithmica*, 7, 499–519.

Gusfield, D. and Tardos, E. (1994), "A Faster Parametric Minimum-Cut Algorithm," *Algorithmica*, 11, 278–290.

Jackson, P. L. and Roundy, R. O. (1991, August), "Minimizing Separable Convex Objectives on Arbitrarily Directed Trees of Variable Upper Bound Constraints," *Mathematics of Operations Research*, 16(3), 504–533.

Maxwell, W. L. and Muckstadt, J. A. (1985), "Establishing Consistent and Realistic Reorder Intervals in Production-Distribution Systems," *Operations Research*, 33, 1316–1341.

Muckstadt, J. A. and Roundy, R. O. (1993), "Analysis of Multistage Production Systems," in *Logistics of Production and Invenory*, eds. S. C. Graves, A. H. G. R. Kan, and P. H. Zipkin, volume 4 of *Handbooks in Operations Research and Management Science*, New York: Logisics of Production and Inventory.

Picard, J.-C. and Queyranne, M. (1985), "Integer Minimization of a Separable Convex Function Subject to Variable Upper Bound Constraints," Technical report, Faculty of Commerce and Business Administration, Vancouver, British Columbia.

Roundy, R. O. (1986, November), "98%-Effective Lot-Sizing for Multi-Product, Multi-Stage Production/Inventory Systems," *Mathematics of Operations Research*, 11, 699–727.

Roundy, R. O. (1990, January), "Computing Nested Reorder Intervals for Multi-Item Distribution Systems," *Operations Research*, 38, 37–52.

Roundy, R. O. and Sun, D. (1994, May), "An Improved Algorithm for Finding Optimal Lot Sizing Policies for Finite Production Rate Assembly Systems," *Operations Research*, 42, 562–565.