Scheduling Injection Molding Operations with Multiple Resource Constraints and Sequence Dependent Setup Times and Costs

Satyaki Ghosh Dastidar • Rakesh Nagi[1]

342 Bell Hall, Department of Industrial Engineering, University at Buffalo (SUNY), Buffalo, NY 14260, USA

Abstract

This paper addresses a production scheduling problem in an injection molding facility. It appears to be the first attempt to schedule parallel machines for multiple items in presence of multiple capacitated resource constraints with sequence-dependent setup costs and times. The objective is to meet customer demands while minimizing the total inventory holding costs, backlogging costs and setup costs. We present a mathematical formulation of the problem. The computational complexity associated with the formulation makes it difficult for standard solvers to address industrial-dimensioned problems in reasonable solution time. To overcome this, a 2-phase workcenter based decomposition scheme has been developed in this paper. The computational results for different problem sizes demonstrate that this scheme is able to solve industrial-dimensioned problems within reasonable time and accuracy.

Keywords: *Scheduling, Sequence Dependent Setup, Changeover Cost, Multiple Capacitated Resources*

[1] To whom correspondence should be directed. nagi@buffalo.edu

1. Introduction

Injection molding is a huge global market. Molders Economic Index (MEI) indicate that the medical molding, in U.S. alone, has shown accelerated growth rates in the past three years. This is due to the massive expansion in health care spending in the U.S. caused by an aging population. North American molding plants have never been as modern and well equipped as they are now. In the 1990s and well into the year 2000 molders spent heavily on capital improvements. New injection machines, advanced automation and inspection devices, automated assembly machinery, sophisticated materials handling equipment, and decorating and printing equipment all allowed molders to significantly lower the cost of direct labor used in the manufacture of even low-cost commodity parts. In addition, injection molding plants generally enjoy the highest possible productivity rates combined with highly advanced product design and materials technology. In this light of the prospects held by the injection molding industry, particularly the healthcare related injection molding industry, we were motivated to work with a healthcare injection molding plant to improve their processes through better production scheduling.

Injection molding operation is typically a single-stage manufacturing process wherein a number of products are manufactured on shared machines with their respective dies. The molding machine uses a vacuum to move the plastic from the "drier" to its initial holding chamber. This chamber is actually a small "hopper" on the back of the "barrel" of the machine. The barrel is where all the real work is done. It is essentially a large screw housed in a heater which moves the plastic closer to the mold. As the screw turns, the plastic traverses the barrel and reaches a molten state. Only when it is molten can it be injected into the mold with a rapid turn of the screw. The tip of the barrel is called the "nozzle" and from this point to the cavity in the mold the material is not heated and is constantly cooled. The "runner" is the cooled/set plastic that extends from the nozzle to the cavity and is process scrap. Actually, the cooled material from the nozzle to the mold is the "sprue" but it is connected to the runner. The runners are ejected into a "conveyor" below the mold and dumped into a "grinder". The grinder chops the runners into bits and prepares them to be moved back into the drier. "Ejector pins" (part of the mold) are then used to eject the molded parts.

A setup needs to be performed for a product to be manufactured. The setup is sequence dependent and parameters like *changeover cost* and *setup time* vary depending on which product is setup after which. Further, a number of resources like grinders, driers and conveyors are required for the production process. These resources are constrained by their limited numbers. Also different products require different machines and different sets of resources and tools. Coming up with a good schedule in presence of all these constraints, is a challenge, even for the single stage case.

We present a mixed integer formulation for scheduling parallel workcenters in presence of multiple tooling and capacitated resource constraints with sequence dependent setup times and costs. In the first stage, we solve the "aggregated" problem for a "part family". In the second stage, we solve the "disaggregated" problem on a rolling-horizon basis. We develop a workcenter based decomposition scheme to decompose the "disaggregated" problem into smaller subproblems and solve them to generate the final feasible schedule. We also developed a software package that would incorporate the solution methodology and provide a GUI based solution for the aggregate and the detailed scheduling problem.

In the following section, a recent literature review of the related topics is presented. The problem definition and formulation is presented is Section 3. Section 4 presents the solution approach including the 2-Phase workcenter based decomposition scheme. Computational results are shown in Section 5. The results indicate that the proposed scheme provides good feasible schedules for industrial sized problems within an acceptable time limit. We conclude with remarks of this work in Section 6. The appendix contains a brief account of the software system.

2. Literature Review

The literature review presented here is divided into two sections. The first section attempts to define the different classes of lot-sizing problems considered in the literature; and the second section summarizes the different solution strategies developed to deal with different classes of lot-sizing problems.

2.1 Lot-Sizing and Scheduling Models

2.1.1 Basic Models

The simplest version of production planning disregards setup cost and for a single stage, single product situation, attempts to minimize inventory holding cost in an uncapacitated environment. The solution to this problem is a trivial lot-for-lot schedule because it clearly minimizes the objective. This model is typically considered with some combination of the following five aspects to result in a version of practical relevance.

1. Presence of Setup Costs. In this case, the simplest lot-sizing problem was formulated by Wagner and Whitin [39].

2. Capacity Constraints.

3. Presence of Backlogs.

4. Multiple Products. The Capacitated Lot-sizing and Scheduling Problem (CLSP), was addressed by Drexl and Kimms [13], for a single-stage multiple product system. The model minimizes total setup and holding costs.

5. Multiple Stages. The multi-stage ("multi-level") capacitated lot-sizing (MSCLS) problem is concerned with the determination of production lot sizes in resource-constrained multi-stage MRP (Materials Requirement Planning) systems so as to minimize the sum of production, set-up and inventory costs. The MSCLS problem, is shown to be NP-Complete by França *et al.* [19]. This is important because it implies that it is unlikely that any algorithm can optimally solve large problems.

2.1.2 Time Period Based Classification of Models

Two terms used in lot-sizing problems, are small and big (or large) bucket, as defined by Belvaux and Wolsey [5]. The CLSP is called a large bucket (or big bucket) problem by Drexl and Kimms [13] and Belvaux and Wolsey [5], because several items may be produced per period. The case where the (macro) periods are subdivided in several micro-periods leads to the Discrete Lot-sizing and Scheduling Problem (DLSP), called a small bucket problem, by Drexl and Kimms [13] and Belvaux and Wolsey [5], because at most one item can be produced per period. The DLSP has the same objective function as the CLSP. DLSP follows the approach. The Continuous Setup Lot-Sizing Problem (CSLSP) constraint allows the system to produce under its full capacity.

The Proportional Lot-sizing and Scheduling Problem (PLSP) occurs when the CSLSP model does not use the full capacity of a period, as described by Drexl and Kimms [13]. Drexl and Haase [15] present the PLSP with setup times (PLSPST) where the capacity constraint now includes the setup. The General Lot-Sizing and Scheduling Problem (GLSP), proposed by Drexl and Kimms [13] and Fleischmann and Meyr [17], features multiple products, single-machine sequence-dependent setup costs, small bucket time, but with neither setup times nor backlogging. Models and applications of the Continuous Time Lot-Sizing and Scheduling Problem (CTLSP), including the Batching and Scheduling (BSP), are discussed by Belvaux and Wolsey [5], Drexl and Kimms [13], Drexl and Haase [14] and Potts and Van Wassenhove [31]. We do not elaborate on these models any further as these models are not within the scope of this paper.

2.1.3 Multiple Machine Models

This section covers the multiple-machine problem which simultaneously preserves the single-stage feature, i.e., machines in parallel. Anderson *et al.* [2] and Barbarosoglu and Ôzdamar [3] distinguish three special cases for single-stage in parallel machines: identical parallel machines, uniform parallel machines and unrelated parallel machines, a feature often omitted by the researchers. Meyr [27] develops the following features: multiple-products, multiple heterogeneous machines, capacitated, sequence-dependent setup times, without backlogging and the objective function has a particularity including the holding, sequence-dependent costs and production costs, in a single-stage scheme. Meyr [27] develops the model of Fleischmann and Meyr [17] to the multiple machine case denoted General Lot-Sizing and Scheduling Problem Parallel Production Line (GLSPPL). A different modeling approach is adopted in Clark and Clark [9], which presents two formulations to lot-sizing when setup times are sequence-dependent in the context of rolling-horizon planning and scheduling for parallel machines. The first formulation is an exact model while the second is an approximation for rolling-horizon use. Instead of using macro and micro periods, the models by Clark and Clark [9] permit multiple setups within each planning period. In the model by Clark and Clark [9], the n^{th} setup can occur at different times on each machine, for which, a binary variable is introduced. The resultant MIP is huge, but solved using a rolling horizon basis, using continuous variable approximations for future setups. Research is continuing on such models.

2.2 Solution Methods

A recent overview of the lot sizing procedures for the multi-stage, multi-item case is given by Katok *et al.* [24]. Bahl *et al.* [4] and Maes *et al.* [26] summarize efforts for different classes of lot-sizing problems.

There are two distinct approaches to solving MLCLSP. The first one uses the optimization approach and the second approach is the heuristic approach. Helber [22] separates the heuristic procedures into three categories. They are: 1) Decomposition approaches, 2) Local search procedures, and 3) Lagrangean based procedures. Katok *et al.* [24] introduces a fourth category to this list: 4) LP based procedures. Often these approaches have been combined to get better results. Helber [22] describes combinations of decomposition approaches with local search procedures, while Salomon [33] and Salomon *et al.* [32] describe combinations of LP based procedures with local search techniques.

2.2.1 Optimization Approaches

Shortest Route

These involve methods for finding strong lower bounds. Eppen and Martin [16] developed the shortest route formulation of the single stage capacitated lot sizing problem (CLSP), where the LP relaxation provided stronger lower bounds than the standard formulation of Billington *et al.* [6]. Tempelmeier and Helber [36] extended this shortest route formulation to MLCLSP. Pochet and Wolsey [30] used the shortest route formulation and added valid inequalities find near-optimal solutions for both uncapacitated, and capacitated (single resource) multi-item problems having either a single-stage or a multi-stage general structure.

Lagrangean Relaxation

Some optimization approaches use Lagrangian relaxation to generate strong lower bounds. Diaby *et al.* [11] used Lagrangean relaxation with subgradient optimization to generate lower bounds, followed by a branch and bound algorithm to locate the optimal solutions for a single level problem with a single capacitated resource, setup costs, and setup times. They showed that capacity constraint relaxations produce better lower bounds than demand constraints relaxations.

2.2.2 Heuristic Approaches

Decomposition Approaches

These ignore the multi-level structure of MLCLSP and solve a sequence of CLSPs. Blackburn and Millen [8] developed a decomposition procedure for the multi level problem by modifying setup and holding costs. They considered uncapacitated assembly systems with constant demand, infinite time horizon, and setup costs. They solve these problems using an existing single level lot-sizing algorithm. Tempelmeier and Helber [36] used the cost adjustment procedures of Blackburn and Millen [8] and Heinrich and Schneeweiss [21] along with the heuristic technique proposed by Dixon and Silver [12] to solve a sequence of single level CLSPs. The final stage of the algorithm restored feasibility using a smoothing heuristic. Stadtler [35] proposed a time-oriented decomposition heuristic to solve the dynamic multi-item multi-level lot-sizing problem in general product structures with single and multiple constrained resources as well as setup times.

Lagrangean Relaxation

Trigeiro *et al.* [38] considered the single level lot-sizing problem with setup costs, setup times, and a single constrained resource. They used Lagrangean Relaxation of the capacity constraints to decompose the problems into uncapacitated single-item problems, which were then solved by dynamic programming. At the last stage, a heuristic smoothing procedure was used to generate feasible production plans. Tempelmeier and Derstroff [37] used Lagrangean relaxation of the multi-level inventory balancing constraints and capacity constraints, and updated Lagrangean multipliers using subgradient optimization to compute lower bounds. Feasible solutions were then generated using a heuristic finite scheduling procedure. Trigeiro *et al.* [38] used Lagrangean Relaxation and Dynamic Programming for MLCLSPs.

Local Search Procedures

These heuristic methods include simulated annealing, tabu search, genetic algorithms, and evolution strategies. Salomon [33] and Helber [22] compared some of these methods. Salomon and Kuik [34] discussed simulated annealing based methods. Helber [22] compared decomposition, local search, decomposition and local search, and the Lagrangean based methods of Tempelmeier and Derstroff [37]. According to Helber [22], simulated annealing located solutions with the best average quality, but required much computational effort which were sometimes prohibitive for small problems. A mix of evolution strategy and decomposition performed the best for medium sized problems.

LP Relaxation

Maes *et al.* [26] presented three LP based heuristics for solving multi-item, multi-stage lot-sizing problems with multiple constrained resources, setup costs, and no setup times. These three heuristics started with a solution to the LP relaxation of the problem, and solved sequences of LP restrictions until an integer solution was found. There was, however, a trade-off between the quality of solutions and the computational effort. The best performing heuristic required branch and bound at the last step of the algorithm.

Coefficient Modification Heuristic

Harrison and Lewis [20] described the Coefficient Modification Heuristic (CMH) which gave good solutions to multi-item, multi-stage lot-sizing problems with multiple constrained resources and serial assembly systems. The CMH was designed to handle setups that consume capacity (setup times) and assumed that the setup costs are small enough to be ignored. Katok *et al.* [24] introduced the Coefficient Modification Heuristic with Cost Balancing and Setup Reduction (CMHBR) heuristic. They extended CMH to handle setup costs and general assembly structures.

Other Heuristics

Akker *et al.* [1] use Column Generation for these class of problems. Degraeve *et al.* [10] used a Branch-and-Price (Column Generation) algorithm for the Capacitated Lot-Sizing Problem with Time Periods (CLSTP). Kuik *et al.* [25] discussed simulated annealing based methods. Helber [22] compared decomposition, local search, decomposition and local search, and the Lagrangean based methods. According to Salomon [33] and Kuik *et al.* [25], simulated annealing and tabu search performed better than pure LP based heuristics. Jordan and Drexl [23] solve DLSP by batch sequencing using a branch-and-bound based heuristic. Miller *et al.* [28] have used branch-and-bound to solve MLCLSP. They used results of Miller *et al.* [29] concerning the polyhedral structure of simplified models obtained from a single time period of MLCLSP, to obtain strong valid inequalities for MLCLSP. But their approach required extensive branch-and-bound techniques to guarantee feasibility (even for small sized problems).

2.3 Summary

The literature review presented here indicates that the production planning and scheduling of parallel workcenters for multiple items have not been properly addressed in conjunction with multiple tooling and capacitated resource constraints, and sequence-dependent setup costs and times. The previous efforts do not consider all of these realistic constraints together as are faced by a modern injection molding facility. Different solution strategies have been developed in the literature. But for large sized problems, there remains a significant tradeoff between the solution quality and the computational times. For practical sized problems, it is desirable to obtain a good solution with a small optimality gap within acceptable processing times. Faster solution generation mechanisms are also helpful for recalculation schedules should there be a change in the demand data. This paper attempts to make the following two-fold contributions:

(1) A new lot-sizing problem formulation with sequence dependent setup times and changeover costs, and multiple capacitated resource constraints, and

(2) A decomposition based heuristic that is capable of handling the problem dimensionality, solution quality and speed required in injection molding operations.

3. Problem Description and Formulation

Our problem is motivated by a production planning and scheduling problem faced by a healthcare products manufacturing plant. The manufacturing facility has 43 injection molding workcenters that produce 175 different parts in a single stage operation. Within the same product "family", products require the same tooling but vary according to color schemes. The operations are conducted over two 12 hours shifts, 7 days a week. Presently high volume batches are usually produced in 7-day runs. Other logical rules like this are used to schedule the facility. There are a number of constraints that affect the production. These include:

1. *Part-Workcenter Matching:* A particular part can only be produced on a particular subset of workcenters that are compatible with tools required to produce the part and are of appropriate tonnage. The workcenters that comprise this compatible "set" for the part are treated as functionally identical parallel workcenters.

2. *Part-Resource Matching:* There are multiple resources that are necessary for a part to be set up for production. These include options from Desiccant Drier, Auger Grinder, Conveyor Grinder, Upright Grinder and Color Mixer. Specific resources that are required by a part must be available for the part to be set up for production.

The setup is a lengthy and expensive process and incurs losses in terms of productivity. It involves line clearance and/or color purging and/or tool changes. A typical tool change requires about 4 hours. If a different part needs to be setup, material change may or may not occur. If the preceding and the following parts belong to the same "family" of parts, then the setup change involves minimal tool change. However there might be a "Color Purging" operation. Color purging operation flushes out the system when the following part is lighter in color than the previous part. Color purging is a "low-cost" operation and the tool-change operation for the same "family" of products is minimal. If the two consecutive parts belong to separate product families, then the *changeover cost* can be high as it involves expensive tool and die change along with material change and/or color purging. Ideally, setups of a "family" of parts should be done together. This minimizes the changeover time and cost and hence the setup change cost. *Sequence*, in such a sequence dependent setup, is therefore an important issue. Also, the company wants to come up with a production schedule that minimizes the total setup, inventory holding and backlogging costs. We model the production situation described above through a mixed integer programming model in the following section.

3.1 Notation

P different part types are processed through M functionally similar workcenters. The workcenter may differ in tonnage or process features. Each part requires a combination of different resource types out of L available resource types. Only one replicate for each of the required resources is to be assigned to the part-workcenter combination. n_l such replicates are available for the l^{th} resource type. Production planning and setup sequencing have to be performed over a planning horizon in response to deterministic orders. We assume at all times that the setup times are smaller than the production time period. Further, setups are not allowed within a time perid, i.e., at most one product type is allowed on a particular workcenter at a particular time period.

Following is the notation for various indices and parameters used in the model.

Indices:

$$\begin{aligned}
i &= \text{Index of part types}, \quad i = 1, 2, \ldots, P, \\
m &= \text{Index of workcenters}, \quad m = 1, 2, \ldots, M, \\
t &= \text{Index of time periods}, \quad t = 1, 2, \ldots, T, \text{ and} \\
l &= \text{Index of resources}, \quad l = 1, 2, \ldots, L
\end{aligned}$$

Data:

$$d_{it} = \text{Demand for part } i \text{ at time period } t,$$

$$C_{im} = \text{Capacity of workcenter } m \text{ to produce part } i \text{ per unit time period},$$

$$\rho_{im} = \text{Production rate for part } i \text{ on workcenter } m,$$

$$\tau_{jim} = \text{Tooling setup time of from part } j \text{ to part } i \text{ on workcenter } m,$$

$$\mu_{ji} = \text{Material changeover time from part } j \text{ to part } i,$$

$$s_{jim} = \text{Cost due to setup change from part } j \text{ to part } i \text{ on workcenter } m,$$

$$h_i = \text{Penalty cost for holding inventory of a unit of part } i, \text{ and}$$

$$p_i = \text{Penalty cost for backlogging of a unit of part } i,$$

$$r_{im}^l = \begin{cases} 1 & \text{if } l^{\text{th}} \text{ resource is required to produce part } i \text{ on workcenter } m \\ 0 & \text{otherwise} \quad (\forall l, \ \ \forall i, \ \ \forall m), \end{cases}$$

$$\pi_{im} = \begin{cases} 1 & \text{if part } i \text{ can be processed on workcenter } m \\ 0 & \text{otherwise} \quad (\forall i, \ \ \forall m). \end{cases}$$

Decision Variables:

Real:

$$x_{imt} = \text{Quantity of part } i \text{ processed on workcenter } m \text{ at time period } t,$$

$$I_{it} = \text{Inventory of part } i \text{ at time period } t, \text{ and}$$

$$b_{it} = \text{Backlog of part } i \text{ at time period } t.$$

Binary:

$$\phi_{imt} = \begin{cases} 1 & \text{if part } i \text{ is assigned to workcenter } m \text{ at time period } t \\ 0 & \text{otherwise} \quad (\forall i, \ \ \forall m, \ \ \forall t,) \end{cases}$$

$$\psi_{jimt} = \begin{cases} 1 & \text{if part } i \text{ is setup from part } j \text{ on workcenter } m \text{ at time period } t \\ 0 & \text{otherwise} \quad (\forall i, j, \ \ \forall m, \ \ \forall t), \quad \textit{and} \end{cases}$$

3.2 Formulation

$$(P) \qquad Min: \sum_{\substack{i,j=1 \\ i \neq j}}^{P} \sum_{m=1}^{M} \sum_{t=1}^{T} s_{jim}\psi_{jimt} + \sum_{i=1}^{P} \sum_{t=1}^{T} h_i I_{it} + \sum_{i=1}^{P} \sum_{t=1}^{T} p_i b_{it}.$$

Subject to:

$$
\begin{aligned}
I_{it} - b_{it} &= I_{i(t-1)} - b_{i(t-1)} + \sum_{m=1}^{M} x_{imt} - d_{it} && \forall i, \ \ \forall t, && (1) \\
x_{imt} &\leq C_{im}\phi_{imt} - \sum_{\substack{j=1 \mid \pi_{jm}=1 \\ i \neq j}}^{P} \rho_{im}(\tau_{jim} + \mu_{ji})\psi_{jimt} \\
& && \forall i, m \mid \pi_{im} = 1, \ \ \forall t, && (2) \\
\sum_{i=1}^{P}\sum_{m=1}^{M} r_{im}^{l}\phi_{imt} &\leq n_l && \forall i, m \mid \pi_{im} = 1, \ \ \forall l, t, && (3) \\
\sum_{i=1}^{P}\phi_{imt} &\leq 1 && \forall i, m \mid \pi_{im} = 1, \ \ \forall t, && (4) \\
\psi_{jimt} &\geq \phi_{imt} + \phi_{jm(t-1)} - 1 && \forall i, j, m \mid \pi_{im} = \pi_{jm} = 1, i \neq j, \ \ \forall t, && (5) \\
\phi_{imt} &\in \{0, 1\} && \forall i, m \mid \pi_{im} = 1, \ \ \forall t, && (6) \\
\psi_{jimt} &\in \{0, 1\} && \forall i, j, m \mid \pi_{im} = \pi_{jm} = 1, i \neq j, \ \ \forall t, && (7) \\
x_{imt}, I_{it}, b_{it} &\geq 0 && \forall i, \ \ \forall m, \ \ \forall t. && (8)
\end{aligned}
$$

The objective function minimizes the total setup cost and the inventory holding and backlogging costs. Constraint (1) maintains the mass balance of parts produced/delivered and the inventory levels across the time periods. Constraint (2) determines the number of parts that can be produced subject to the capacity constraint of the workcenter and the loss of production incurred as a result of setup and/or material change, if any. Constraint (3) ensures that the number of replicates used for a particular resource type is less than or equal to the number of replicates available for that resource type. Constraint (4) ensures that only one part type can be processed by a particular workcenter at a particular time period. Constraint (5) determines when the setup state changes. Constraints (6) and (7) are the binary constraints. Finally, constraint (8) indicates that the number of produced parts, parts in inventory and parts in backlog are non-negative.

Constraint (7) can be relaxed as the variables ψ_{jimt} remain binary even when the binary restriction is relaxed (see Lemma 1).

Lemma 1. *In an optimal solution of P with the binary constraint (7) relaxed, the variables ψ_{jimt} take on binary values.*

Proof: Equation (2) does not guarantee integral values for the variables ψ_{jimt}. Given that the variables ϕ_{imt} are binary, the positive fractional values of ψ_{jimt} are automatically pushed up to 1 by the equation (5) to maintain feasibility. The values of the variables ψ_{jimt} are therefore determined by the equation (5). The variables ψ_{jimt} are in the objective function with positive coefficients. As P is a minimization problem, they take up the minimum of the RHS of equation (5), which provides values of ψ_{jimt} to be -1, 0 and 1. In the $MILP$ the variables ψ_{imt} are positive. Hence the variables ψ_{jimt} take up values 0 and 1. □

4. Solution Approach

The problem described here becomes huge for large number of parts, workcenters, resources and length of the planning time period. Florian *et al.* [18] showed that the lot-sizing problem (LSP) is *NP*-hard. Bitran *et al.* [7] proved that the CLSP is also NP-hard. In our case, the additional resource constraints and the non-zero sequence dependent setup change, make the problem subsume previously proven NP-hard versions. So, without going through a formal proof, we assume that our problem is NP-hard. We attempt to overcome the complexity and dimensionality of the problem by developing a decomposition based solution methodology where we break down the original problem into a set of subproblems. At this point we have the following challenges:

1. The dimensions of the subproblems should be manageable.
2. The set of subproblems must address the entire large-sized industrial problem.
3. The overall scheme should generate good feasible solutions within an "acceptable" time limit.

4.1 Monolithic Approach

The complexity of the problem can be reduced by relaxing the binary constraint (7) due to Lemma (1). The resultant problem is still a mixed integer problem (MIP). We solved the entire problem ("monolithic" approach) for small and medium sized version using CPLEX 7.1 on a 1.7 GHz P4 processor, 256 MB RAM, WinNT 4.0 workstation. However the problem sizes were huge for medium-sized and large-sized problems. The computational time for some medium-sized problems exceeded the specified time limit of 2 hours. For other instances of medium and large sized problems, CPLEX 7.1 ran into "out-of-memory" conditions. The computational test results are presented later in Section 5.

4.2 2-Phase Workcenter Based Decomposition Strategy

Clearly, the problem could not be solved in its original form for the medium and large sized versions. This was the motivation for developing the 2-Phase workcenter based decomposition scheme to tackle the medium and large sized problems. Further, the particular company did not want to invest in expensive commercial solvers like CPLEX. We chose to solve the set of related subproblems using an open-source solver, GLPK 4.0 in our case. The following section describes the workcenter based grouping strategy that serves as the basis for the decomposition.

4.2.1 Grouping Scheme

Let P_i^m denote that part i is processed at workcenter W_m. Let us group the workcenters into n groups such that $G(1) = \{\ W'_1,\ W'_2,\ \ldots,\ W'_k\ \}$, $G(2) = \{\ W'_{k+1},\ W'_{k+2},\ \ldots,\ W'_l\ \}$, $\ldots$, $G(n) = \{\ \ldots,\ W'_{M-1},\ W'_M\ \}$. Here, W'_m denote the workcenter that is mapped to the m^{th} entry in the set $G(\cdot)$, and might be different from the actual workcenter W_m. Figure 1 shows the part-workcenter relation.

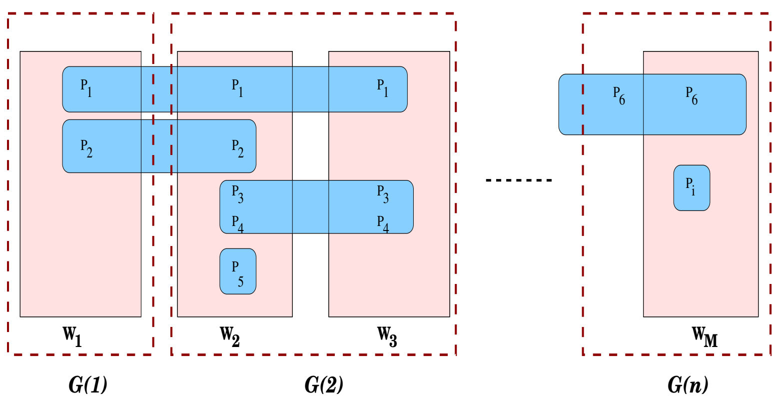

Figure 1: Part-workcenter relation across group of workcenters.

Our choice for selecting the workcenters for a group is based on the following rationale. There are certain parts that can be processed on fewer number of workcenters than some other parts. Such workcenters are "critical" to the processing of those "less-fortunate" parts. For instance, Figure 2 shows the part-workcenter relation before the group formation.

These parts are then sorted to determine the "critical" workcenters and for subsequent group formation. Figure 3 shows the sorted parts in the ascending order of their "fortunes".

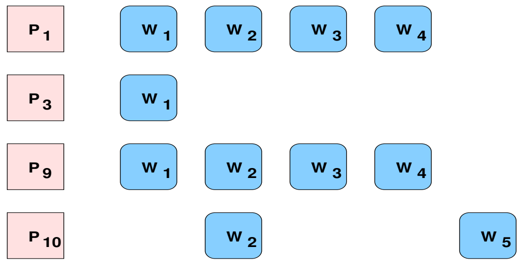

Figure 2: Part and workcenter relation before grouping.

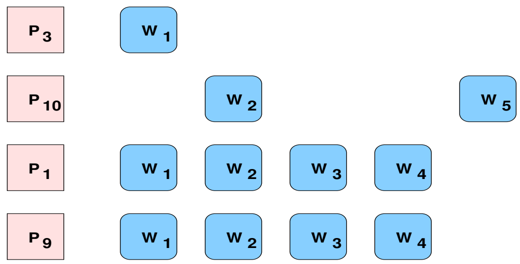

Figure 3: Part and workcenter relation after sorting.

Here, the workcenters W_1, W_2 and W_5 are "critical" to the processing of parts P_3 and P_{10} than to the processing of parts P_1 and P_9. Next, we fix the group size as the nearest integer that is 10% of the total workcenter size. This is an *ad hoc* decision. Next, we group the workcenters in order of their "criticality" until either the group size is reached or all the workcenters are covered. Alternatively, a matrix cross-decomposition can also be performed to group the parts based on their "fortunes". Figure 4 shows groups and the member workcenters for the above example. The grouping algorithm is presented below.

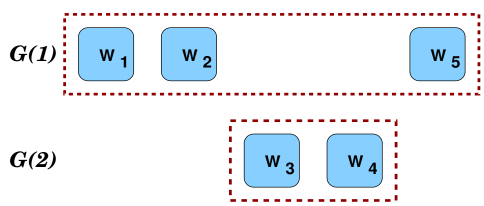

Figure 4: Assignment of workcenter groups.

4.2.2 Grouping Algorithm

Step 1: *Initialize:* $\mathcal{W} = \varnothing$; $G = \varnothing$; $k = 1$.

Step 2: For $i = 1, \ldots, P$

$w_i =$ the set of workcenters $W_m \mid \pi_{im} = 1$

$\mathcal{W} = \mathcal{W} \cup w_i$

End for

Step 3: Sort $\mathcal{W}$ based on the ascending order of the cardinality of its subsets, i.e. $|w_i|$. Let $w_{[k]}$ denote the k^{th} set in the sorted set $\mathcal{W}$.

Step 4: While $\mathcal{W} \neq \varnothing$

$G = G \cup w_{[k]}$

$\mathcal{W} = \mathcal{W} \setminus w_{[k]}$

For all $j = k+1, \ldots \mid \mathcal{W} \mid$

$w_{[j]} = w_{[j]} \setminus w_{[k]}$

End for

$k = k + 1$

End while

Step 5: Partition the set G into n subgroups, each of size s (except the last group which contains the remaining elements of G after $n-1$ partitions), such that $(n-1)\times s + |G(n)| = |G|$. We have $G(1) = \{ W'_1, W'_2, \ldots, W'_s \}$, $G(2) = \{ W'_{s+1}, W'_{s+2}, \ldots, W'_{2s} \}, \ldots, G(n) = \{ \ldots, W'_{M-1}, W'_M \}$ (W'_m maps to the workcenter having m^{th} entry in the set G.)

Step 6: STOP.

4.3 Solution Scheme

We solve the problem in two phases. In the first phase, we use the workcenter based grouping strategy to generate the subproblems. Once the groups are developed, the subproblem for that particular group is generated. In the next phase, we solve the individual subproblem. The solution of a particular subproblem serves as the basis of input for the next subproblem. There are k such subproblems for k such groups. The backlog of part i in the k^{th} subproblem is the “modified” demand of part i in the $k+1^{\text{th}}$ subproblem. The k^{th} workcenter-based subproblem can be defined according to the revised notation that follows.

Revised Notation

Data:

$d_{it}^{\prime\,k}$ = Modified demand for part i at time interval t for subproblem k, and

Decision Variables:

Integer:

$$
\begin{aligned}
I_{it}^k &= \text{Inventory of part } i \text{ at time interval } t \text{ for subproblem } k\text{, and}\\
b_{it}^k &= \text{Backlog of part } i \text{ at time interval } t \text{ for subproblem } k.
\end{aligned}
$$

The k^{th} subproblem is:

$$
(WCSP^k) \qquad Min: \sum_{\substack{i,j=1\\ i\neq j}}^{P} \sum_{m=1}^{M}\sum_{t=1}^{T} s_{jim}\psi_{jimt} + \sum_{i=1}^{P}\sum_{t=1}^{T} h_i I_{it}^k + \sum_{i=1}^{P}\sum_{t=1}^{T} p_i b_{it}^k.
$$

Subject to:

$$
\begin{aligned}
I_{it}^k - b_{it}^k &= I_{i(t-1)}^k - b_{i(t-1)}^k + \sum_{m=1}^{M} x_{imt} - {d'}_{it}^{k} \quad \forall i,m \mid \pi_{im}=1, m\in G(k),\ \forall t, && (9)\\
x_{imt} &\leq C_{im}\phi_{imt} - \sum_{\substack{j=1 \mid \pi_{jm}=1\\ i\neq j}}^{P} \rho_{im}(\tau_{jim}+\mu_{ji})\psi_{jimt} \qquad \forall i,m \mid \pi_{im}=1,\\
&\qquad\qquad i\neq j,\ \ m\in G(k),\ \forall t, && (10)\\
\sum_{i=1}^{P}\sum_{m=1}^{M} r_{im}^l\phi_{imt} &= n_l \qquad \forall i,l,m \mid r_{im}^l=1, \pi_{im}=1, m\in G(k),\ \forall t, && (11)\\
\sum_{i=1}^{P}\phi_{imt} &\leq 1 \qquad \forall i,m \mid \pi_{im}=1, m\in G(k),\ \forall t, && (12)\\
\psi_{jimt} &\geq \phi_{imt} + \phi_{jm(t-1)} - 1 \qquad \forall i,j,m \mid \pi_{im}=\pi_{jm}=1,\\
&\qquad\qquad i\neq j, m\in G(k), \forall t, && (13)
\end{aligned}
$$

$$
\begin{aligned}
0 \leq\ \psi_{jimt} &\leq 1 \qquad \forall i,j,m \mid \pi_{im}=\pi_{jm}=1, m\in G(k),\ \forall t, && (14)\\
\phi_{imt} &\in \{0,1\} \qquad \forall i,m \mid \pi_{im}=1, m\in G(k),\ \forall t, && (15)\\
x_{imt}, I_{it}^k, b_{it}^k &\geq 0 \qquad \forall i,\ \forall m\in G(k),\ \forall t. && (16)
\end{aligned}
$$

We solve the k^{th} subproblem and determine if the demand for the corresponding parts have been satisfied or not. If met, the part is not considered for future consideration and the subproblems are solved for the remaining parts or until all the workcenters have been covered. However, the addition of the resource constraints in the "disaggregated" problem blows up the problem dimension for medium and large sized problems. Figure 5 presents the flow chart of the proposed solution strategy for the 2-phase workcenter based decomposition scheme.

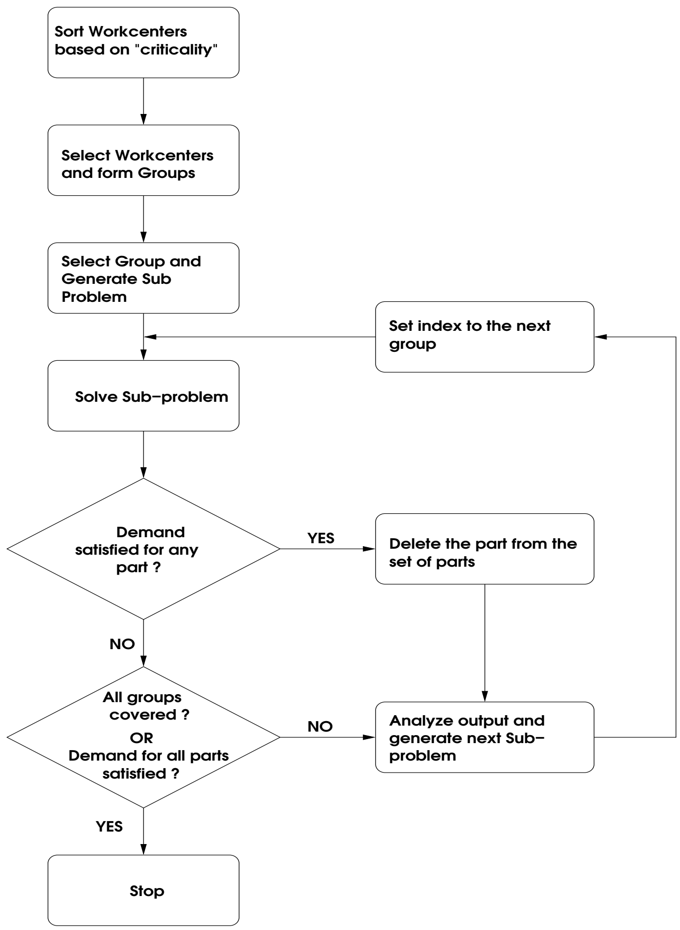

Figure 5: Flow chart representing the 2-phase workcenter based decomposition scheme

5. Numerical Results

Several numerical experiments were conducted to compare the performance of the 2-Phase Decomposition Scheme against the Monolithic approach. The problem instances were generated using data provided by an injection molding company. The testbed considered 51 different product types that were to be scheduled over a maximum of 45 injection molding machines. There were 6 different types of constrained resources. Problem instances were generated for demand over a time period. The parts, workcenters (molding machines) and the resources were selected through a series of demand data, BOM, routing data and inventory data. The resultant problems were then classified into three categories as shown in Table 1.

Problem Size	No. of parts	No. of work-centers	No. of resource types	No. of time periods
small	1-10	10-30	1-6	1-10
medium	11-25	31-42	1-6	11-20
large	26-51	42-45	1-6	21-30

Table 1: Classification of Various Problem Sizes

The actual problem size was determined as the product of the number of parts, work-centers, resources and time periods. The LP relaxed version and the monolithic version of the original MIP problem were solved using CPLEX 7.1. Problems for the 2-Phase Decomposition Scheme were solved using GLPK 4.0. As stated earlier, all the problem instances were run on a 1.7 GHz Pentium 4 processor, 256 MB RAM, WinNT 4.0 workstation. The optimality gap is defined as:

$$\text{Gap}(\%) = \frac{\text{(Integer Optimal Value) - (Objective Value)}}{\text{(Integer Optimal Value)}} \times 100\%$$

For cases when an optimal integer solution is not available, the optimality gap is calculated as follows:

$$\text{Gap}(\%) = \frac{\text{(LP Relaxed Objective) - (Objective Value)}}{\text{(LP Relaxed Objective)}} \times 100\%$$

The average gap is calculated by averaging over the number of problem instances generated for each class of problem.

5.1 Monolithic Scheme

The monolithic scheme attempted to solve the entire MILP problem within a time limit of 7200 secs (2 hrs). The MILP was solved for "small" and "medium" problems. However the MILP could not be solved for some instances of the "medium" and "large" sized problems. Tables 2, 3 and 4 show that the average gaps for the "small" and "medium" sized problems were 0.00% and 0.02% respectively, whereas no integral solution could be found for some "medium" and any "large" sized problems.

5.2 2-Phase Decomposition Scheme

The 2-Phase Decomposition scheme was used to solve the exact same instances of the "small", "medium", and "large" sized problems. Tables 2, 3 and 4 show that the maximum optimality gap for the largest problem instance was 1.75%. There was a significant improvement in the tradeoff between the solution quality and the solution time. The maximum time required to solve the largest instance of the problem using the 2-Phase method was 1266.05 secs (21 mins).

#	i	m	j	t	**LP Lower Bound**	**CPLEX**	**Sol. Time (secs)**	**Gap (%)**	**2-Phase**	**Sol. Time (secs)**	**Gap (%)**
1	6	11	6	2	109246.12	109246.12*	0.12	0.00	109246.12	0.12	0.00
2	9	14	6	11	2156843.82	2156960.63*	15.53	0.00	2156979.68	12.48	0.01
3	22	41	6	4	1794977.25	1795171.90*	27.18	0.00	1795258.47	26.08	0.01
4	24	42	6	6	4708212.90	4708388.55*	34.88	0.00	4708711.43	32.85	0.01
							Average =	0.00		Average =	0.01

Table 2: Results for the Small Size Problems

#	i	m	j	t	**LP Lower Bound**	**CPLEX**	**Sol. Time (secs)**	**Gap (%)**	**2-Phase**	**Sol. Time (secs)**	**Gap (%)**
1	10	18	6	8	5981270.39	5981912.92*	278.87	0.00	5982331.18	93.09	0.01
2	13	39	6	8	898717.91	899645.28*	1504.26	0.00	899935.63	271.42	0.05
3	7	31	6	31	109910.13	109928.36*	2666.09	0.00	109958.70	314.49	0.04
4	32	42	6	20	20310417.51	20315414.27	e.t.l.	0.02	20327494.40	444.62	0.08
							Average =	0.01		Average =	0.05

e.t.l. = Exceeded Time Limit

Table 3: Results for the Medium Size Problems

#	i	m	j	t	**LP Lower Bound**	**CPLEX**	**Sol. Time (secs)**	**Gap (%)**	**2-Phase**	**Sol. Time (secs)**	**Gap (%)**
1	29	42	6	31	16610157.88	16738648.00	e.t.l.	0.77	16748000.00	474.47	0.83
2	39	42	6	30	25595555.50	25942726.75	e.t.l.	1.36	26042672.51	364.79	1.75
3	41	43	6	30	20140000.00	-	o.o.m.	-	20308492.25	965.67	0.84 †
4	51	43	6	30	7205148.77	-	o.o.m.	-	7314174.88	1266.05	1.51 †
							Average =	1.06		Average =	1.23

e.t.l. = Exceeded Time Limit; o.o.m. = Out of Memory; † Gap calculated based on LP Lower Bound.

Table 4: Results for the Large Size Problems

6. Conclusions

In this paper, we cast an injection molding scheduling problem as a mixed integer formulation that schedules parallel workcenters in presence of sequence-dependent setup times, changeover costs, and multiple capacitated resource constraints for a multi-item, multi-class of products in a single-stage case. This paper appears to be the first attempt in this category to combine all these realistic constraints and come up with an efficient method to tackle the problem when the time to generate good schedules is of foremost importance.

The 2-phase workcenter based decomposition scheme has been proposed. The scheme helps to decompose large dimensioned problems into smaller subproblems. It also provides good feasible schedules for cases where the monolithic approach fails to generate any solution, either because of memory management policy of the solver or because of the time limit restrictions (2 hours).

The computational time required by the 2-Phase strategy is significantly less than the solution time of the original MIP problem. Yet the optimality gap is usually less than 2%. In summary, the overall solution approach is applicable for industrial situations that require quick and efficient solutions without any investment in sophisticated solvers and at the same time can provide operational benefits in changeover costs, inventory management costs and in meeting customer due dates.

Acknowledgment

The authors would like to thank the two anonymous referees for their constructive comments. These comments helped strengthen the paper significantly.

References

[1] Akker, J. Van Den, Hoogeveen, J., and Van De Velde, S. Parallel machine scheduling by column generation. Operations Research 1999; 47(6): 862-872.

[2] Anderson, E., Glass C., and Poots, C. Machine scheduling. Local Search in Combinatorial Optimization. Edited by Aarts, E. and Lenstra, J. 1997; 11: 361-414.

[3] Barbarosoglu, G., and Ôzdamar, L. Analysis of solution space-dependent performance of simulated annealing: the case of the multi-level capacited lot sizing problem. Computers and Operations Research. 2000; 27(9): 895-903.

[4] Bahl, H., Ritzman, L., and Gupta, J. Determining lot sizes and resource requirements, a review. Operations Research. 1987; 35(3): 329-345.

[5] Belvaux, G., and Wolsey, L.A. BC-PROD: a specialized branch-and-cut system for lot-sizing problems. Management Science. 2000; 46: 724-738.

[6] Billington, P., McClain, J., and Thomas, L. Mathematical approaches to capacity constrained MRP systems: review, formulation and problem reduction. Management Science. 1983; 29(2): 1126-1141.

[7] Bitran, G.R., and Yanasse, H.H. Computational complexity of the capacitated lot size problem. Management Science. 1982; 28(10): 1174-1186.

[8] Blackburn, J., and Millen, R. Improved heuristics for multi-stage requirements planning systems. Management Science. 1982; 28(1): 44-56.

[9] Clark, A.R., and Clark, S.J. Rolling horizon lot-sizing when setup times are sequence dependent. International Journal of Production Research. 2000; 38(10), 2287-2307.

[10] Degraeve, Z., and Jans, R. A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot sizing problem with setup times. Discussion Paper, Erasmus Research Institute of Management (ERIM). 2003: 285.

[11] Diaby, M., Bahl, H.C., Karwan, M.H., and Zionts, S. A Lagrangean relaxation approach for very large scale capacitated lot-sizing". Management Science. 1992; 38(9). 1329-1339.

[12] Dixon, P., and Silver, E. A heuristic solution procedure for multi-item, single-level, limited capacity lot-sizing problem. Journal of Operational Management. 1981; 2: 23-39.

[13] Drexl, A., and Kimms, A. Lot sizing and scheduling - survey and extensions. European Journal of Operational Research. 1997; 99(2): 221-235.

[14] Drexl A., and Haase K. Sequential analysis based randomized-regret-methods for lot-sizing and scheduling. Journal of the Operational Research Society. 1996; 47(2): 251-265.

[15] Drexl A., and Haase K. Proportional lotsizing and scheduling. International Journal of Production Economics. 1995; 40(1): 73-87.

[16] Eppen, G.D., and Martin, R.K. Solving multi-item capacitated lot-sizing problems using variable redefinition. Operations Research. 1987; 35(6): 832-848.

[17] Fleischmann, B., and Meyr, H. The general lot-sizing and scheduling problem. OR Spektrum. 1997; 19(1): 11-21.

[18] Florian, M., Lenstra, J.K., and Rinnooy Kan H.G. Deterministic production planning: algorithms and complexity. Management Science. 1980; 26(7): 669-679.

[19] França, P., Armentano, V., Berretta, R., Clark A. A heuristic for lot-sizing in multi-stage systems. Computers and Operations Research. 1997; 24: 861-874.

[20] Harrison, T., and Lewis, H. Lot-sizing in serial assembly systems with multiple constrained resources. Management Science. 1996; 42(1): 19-36.

[21] Heinrich, C., and Schneeweiss, C. Multi-stage lot-sizing for general production systems. Lecture Notes in Economics and Mathematical Systems. Editors Axsater S., Schneeweiss C, and Silver, E. Springer Verlag, Heidelberg, Germany, 1986; 266: 150-181.

[22] Helber, S. Lot sizing in capacitated production planning and control systems. OR Spektrum. 1995; 17: 5-18.

[23] Jordan, C. and Drexl, A. Discrete lot-sizing and scheduling by batch sequencing. Management Science. 1998; 44: 698-713.

[24] Katok, E., Lewis, H.S., and Harrison, T.P. Lot-sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. Management Science. 1998; 44(6): 859-877.

[25] Kuik, R., Salomon, M., Van Wassenhove, L.N., and Maes, J. Linear programming, simulated annealing and tabu search heuristics for lot sizing in bottleneck assembly systems. IIE Transactions. 1993; 25(1): 62-72.

[26] Maes, J., McClain, L., and Van Wassenhove, L.N. Multi-level capacitated lot-sizing complexity and LP-based heuristics. European Journal of Operational Research. 1991; 53: 131-148.

[27] Meyr, H. Simultaneous Lot-sizing and scheduling on parallel machines. European Journal of Operational Research. 2001; 139(2): 277-292.

[28] Miller, A., Nemhauser, G., and Savelsbergh, M. A multi-item production planning model with setup times: algorithms, reformulations, and polyhedral characterizations for a special case. Mathematical Programming. 2003; 5(1): 71-90.

[29] Miller, A., Nemhauser, G., and Savelsbergh, M. On the polyhedral structure of a multi-item production planning model with setup times. Mathematical Programming. 2003; 94: 375-405.

[30] Pochet, Y. and Wolsey, L.A. Solving multi-item lot-sizing problems using strong cutting planes. Management Science. 1991; 37(1): 53-67.

[31] Potts, C., and Van Wassenhove, L. Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. Operational Research Society. 1992; 43(5): 395-406.

[32] Salomon, M., Kuik, R., and Van Wassenhove, L. Statistical search methods for lot-sizing problems. Annals of Operations Research. 1993; 41: 453-468.

[33] Salomon, M. Deterministic lot-sizing models for production planning. Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Heidelberg, Germany,1991; 355.

[34] Salomon, M., and Kuik, R. Multi-level lot sizing problem: evaluation of a simulated annealing heuristic. European Journal of Operational Research. 1990; 45(1): 25-37.

[35] Stadtler, H. Multi-level lot-sizing with setup times and multiple resource constrained resources: internally rolling schedules with lot-sizing windows. Operations Research. 2003; 51(3): 487-502.

[36] Tempelmeier, H., and Helber, S. A heuristic for multi-item multi-level capacitated lot-sizing for general product structure. European Journal of Operational Research. 1994; 75(2), 296-311.

[37] Tempelmeier, H., and Derstroff, M. A Lagrangean-based heuristic for dynamic multi-level multi-item constrained lot-sizing with setup times. Management Science. 1996; 42(5): 738-757.

[38] Trigiero, W.W., Thomas, L.J. and McClain, J.O. Capacitated lot-sizing with setup times. Management Science. 1989; 35(3): 353-366.

[39] Wagner, H.M., and Within, T.H. Dynamic version of the economic lot size model. Management Science. 1958; 5(1): 89-96.