5-2008

Heuristic Procedures to Solve Sequencing and Scheduling Problems in Automobile Industry

Jingxu He
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Jingxu He entitled "Heuristic Procedures to Solve Sequencing and Scheduling Problems in Automobile Industry." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Fong-Yuen Ding, Major Professor

We have read this dissertation and recommend its acceptance:

Denise Jackson, Funda Sahin, Dukwon Kim

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Jingxu He entitled "Heuristic Procedures to Solve Sequencing and Scheduling Problems in Automobile Industry." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Fong-Yuen Ding, Major Professor

We have read this dissertation
and recommend its acceptance:

Denise. Jackson

Funda Sahin

Dukwon Kim

Accepted for the Council:

Carolyn R. Hodges, Vice Provost and
Dean of the Graduate School

(Original signatures are on file with official student records.)

HEURISTIC PROCEDURES TO SOLVE SEQUENCING AND SCHEDULING PROBLEMS IN AUTOMOBILE INDUSTRY

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Jingxu He
May 2008

DEDICATION

To God, my Lord, who gives strength and energy.
To my parents, who feed me, educate me, and support me all the time.
To my wife, who raises our angel, Abigail, who brings us joy anywhere and anytime.

ACKNOWLEDGEMENTS

I would like to express my heartfelt thanks and appreciation to Professor Fong-Yuen Ding, for persevering with me as my advisor throughout the time it took me to complete this dissertation. The members of my dissertation committee, Professor Denise F. Jackson, Professor Dukwon Kim, and Professor Funda Sahin, have generously given their time and expertise to better my work. I thank them for their contribution and their support.

ABSTRACT

With the growing trend for greater product variety, mixed-model assembly nowadays is commonly employed in many industries, which can enable just-in-time production for a production system with high variety. Efficient production scheduling and sequencing is important to achieve the overall material supply, production, and distribution efficiency around the mixed-model assembly line. This research addresses production scheduling and sequencing on a mixed-model assembly line for products with multiple product options, considering multiple objectives with regard to material supply, manufacturing, and product distribution. This research also addresses plant assignment for a product with multiple product options as a prior step to scheduling and sequencing for a mixed-model assembly line. This dissertation is organized into three parts based on three papers.

<u>Introduction and literature review</u>

<u>Part 1.</u> In an automobile assembly plant many product options often need to be considered in sequencing an assembly line with which multiple objectives often need to be considered. A general heuristic procedure is developed for sequencing automobile assembly lines considering multiple options. The procedure uses the construction, swapping, and re-sequencing steps, and a limited search for sequencing automobile assembly lines considering multiple options.

<u>Part 2.</u> In a supply chain, production scheduling and finished goods distribution have been increasingly considered in an integrated manner to achieve an overall best efficiency. This research presents a heuristic procedure to achieve an integrated consideration of production scheduling and product distribution with production smoothing for the

automobile just-in-time production assembly line. A meta-heuristic procedure is also developed for improving the heuristic solution.

Part 3. For a product that can be manufactured in multiple facilities, assigning orders to the facility is a common problem faced by industry considering production, material constraints, and other supply-chain related constraints. This paper addresses products with multiple product options for plant assignment with regard to multiple constraints at individual plants in order to minimize transportation costs and costs of assignment infeasibility. A series of binary- and mixed-integer programming models are presented, and a decision support tool based on optimization models is presented with a case study.

Summary and conclusions

TABLE OF CONTENTS

Part **Page**

LIST OF TABLES

Table **Page**

PART 1

PART 2

PART 3

LIST OF FIGURES

Figure **Page**

PART 1

PART 3

INTRODUCTION AND LITERATURE REVIEW

1. Introduction

Material supply, production, and distribution are among the essential operations in most manufacturing companies' supply chains. Many companies have different departments to manage these functions separately without being able to take advantage of a better efficiency in managing these functions in an integrated manner. During the past two decades with the emphasis in integration in the supply chain, material supply, production, and distribution have been increasingly considered jointly.

In many industries such as the automobile industry, production scheduling and sequencing play a key role in achieving efficiency in these above-stated operations. Efficient production scheduling and sequencing can lead to not only better utilization of the manufacturing resources, but also improvement in efficiency in material supply and product distribution. Nowadays, facing the challenges of providing highly diversified and customized products, manufacturers commonly produce various models of similar products on the same assembly line. In this dissertation, production planning, scheduling, and sequencing related to mixed-model assembly will be addressed to enhance material supply, production, and distribution operations.

The car sequencing problem, involving sequencing cars on the assembly line with a spacing requirement, attempts to minimize work overload or material imbalance in an implicit manner. It is combined with a mixed-model sequencing problem in this dissertation to address manufacturing and material supply considerations. Heuristic procedures for production scheduling with an integrated consideration of product distribution and production smoothing in the automobile industry will be developed. A production planning problem that assigns orders to multiple facilities will be also addressed. All these problems will address products with multiple product options.

2. Literature review

2.1 Mixed-model assembly line

Wester and Kilbridge (1964) addressed the model-mix sequencing problem by finding sequences that could avoid the overload or excessive capacity, which leads to high defect rates or low productivity. Wild (1972) referred to this type of production line as mixed-model assembly line. As customers demand more and more customized products which require a highly diversified product portfolio, mixed-model assembly becomes more and more popular as it enables just-in-time production. Nowadays mixed-model assembly has become a common practice not only in automobile industry but also in final assembly processes of many other industries.

In an automobile assembly plant, various models and configurations of vehicles are commonly scheduled and sequenced within a regular scheduling cycle. The commonly accepted objectives of the mixed-model assembly line sequencing problem (MMALSP) are to achieve a smooth pace of material usage (Monden, 1997), even workload, and reduced line stoppage on the assembly line. The mixed-model sequencing problem focuses on considering multiple models instead of considering multiple product options (such as with the car sequencing problem) associated with multiple objectives. The existing literature is reviewed below.

Monden (1997) presented the so-called “goal chasing” method, used in the Toyota production system, to attempt to have the part usage rate as even as possible. This is an iterative construction approach. Miltenburg (1989) developed a heuristic sequencing algorithm that minimized the variability of model quantities, which contributed to the objective of minimizing part usage rates. Cheng and Ding (1996) developed a two-stage mixed-model assembly line sequencing method which considered the weighted variation in model quantities.

Yano and Rachmadugu (1991) formulated a sequencing model to minimize the total work overload on a mixed-model assembly line. A constant-speed assembly line was assumed and an operator could not work across the boundary of the station. Miltenburg and Goldstein (1991) considered both balancing the workload and smoothing the part usage for a just-in-time production system. Most researchers assumed a condition that each part is only used once on the assembly line thus the whole assembly line can be treated as a single station, which simplified the model structure. Zhao and Zhou (1999) consider the situation with a material consumed at multiple stations and discussed the details at each workstation. Other objectives were also considered. Zhao and Ohno (1994, 1997, and 2000) developed approaches to reduce the duration of line stoppages and thus reduced the opportunity cost of lost sale.

As shown by Kubiak (1993), the problem of sequencing mixed-models to smooth part usage or to minimize workload imbalance is NP-hard. To address the part usage smoothing problem for mixed-model assembly line sequencing, the goals commonly considered can be categorized into two. The product-level problem mainly considers the assembled products (Miltenburg, 1989; Cheng and Ding, 1996) while the part-level problem is to keep a constant part usage rate on the assembly line. Kubiak and Sethi (1991) transformed the product-level problem into an assignment problem, the objective is to minimize the total one-level variation. Inman and Bulfin (1991) applied the earliest due date method for production smoothing with an intent to reduce the part level variation. Bautista et al. (1996), and Zhu and Ding (2000) applied two-stage variation methods to reduce the part-level variation. Milternburg and Sinnamon (1989) developed a solution procedure to minimize multi-level usage variation.

Many meta-heuristic approaches have been applied to mixed model sequencing, such as tabu search (McMullen, 1998), simulated annealing (McMullen and Frazier, 2000), genetic algorithm (McMullen et al., 2000), Ant Colony Optimization (McMullen 2001a, 2001b), and beam search heuristic (McMullen and Tarasewich, 2005).

2.2 Car sequencing problem

The car sequencing problem (CSP), studied by many researchers, involves sequencing cars on the assembly line subject to a quantity limit for each of k considered options in each given number of consecutive cars. Instead of a detailed consideration of work or parts content, CSP considers the succession of product options (attributes, that is, such as sunroofs, side airbags) in order to avoid work overload or overuse of material. A quantity limit can be stated as, at most r_i cars can have option i in every s_i consecutive cars in the sequence. Many researchers have attempted to solve this problem by treating the constraints either as hard or soft constraints. CSP was shown to be NP-hard (Gent, 1998).

Solution approaches applied to CSP include the greedy method, local search, and meta-heuristics. Chew et al. (1991) took both upper and lower ratio limits of the options into account. Simulated annealing (SA) was applied to solve this problem. Smith et al. (1996) presented a neural network approach to solve CSP and compared the results with those of traditional heuristics. They set an individual weight for excessive occurrence of each option and started the procedure from a heuristic solution. Davenport and Tsing (1999) modeled the CSP as a constraint satisfaction problem and applied a heuristic improvement procedure to solve the problem and compared the results with those of other procedures. Gottlieb et al. (2003) aimed at minimizing violations of sequencing rules in the objective using the so-called “sliding window” approach. A penalty of one was assigned to a violation of a restriction. The ant-colony optimization (ACO), a meta-heuristic procedure, was then developed to solve the problem. Gravel et al. (2005) also solved CSP using ACO.

2.3 Combined consideration of MMALSP and CSP

In practice, an automobile company often needs to address broader considerations than CSP and MMALSP addressed individually. Drexl and Kimms (2001) considered the joint problem of CSP and MMALSP at the product level by using the column generation approach. Computational experiments were conducted on relatively small-size problems. Bergen et al. (2001) addressed the assembly-line sequencing problem that considered eight types of hard or soft constraints. Vehicles of identical attributes partially related to car options are first split into lots; and similar lots are then grouped into hourly batches. The sequencing problem consists of assignment of batches to hourly slots and sequencing of lots within batches. Three approximation algorithms, a local search, backtracking, and branch-and-bound algorithm were presented to solve the problem.

Estellon et al. (2005) addressed CSP as Renault’s car sequencing problem with the additional constraints on the number of consecutive vehicles having the same color. Reducing the number of color purges thus was also considered. A very-large neighborhood search and a very-fast local search approach were presented to solve the problem. The former approach is an integer-linear-programming based neighborhood search, while the latter is a local search method based on five transformations. Muhl et al. (2003) applied the genetic algorithm, simulated annealing, gradient search, and stochastic search to solve assembly-line sequencing problems that considered requirements of multiple shops.

In today's automobile industry, many options are available on various car models; and in sequencing assembly lines it is common to consider production capacities and manufacturing considerations associated with many options. In this dissertation research, sequencing heuristic procedures addressing broader consideration than the problems of MMALSP and CSP addressed individually will be developed to solve large scale problems. The sequencing problem will deal with multiple objectives and products with multiple options.

2.4 Integration of production scheduling and distribution

Much research considered the production-capacity-limit and material-usage related constraints and objectives when performing sequencing and scheduling. In a manufacturing firm, production and distribution operations constitute a major part of its operational activities. There can be a significant cost benefit to jointly consider these two functions in performing scheduling and sequencing. Scheduling and sequencing ultimately affects the distribution operations. As the integration of the whole supply chain is emphasized by more researchers, there is an increasing emphasis on research on integration of production scheduling and distribution. In this dissertation research, heuristic scheduling procedures considering both manufacturing and distribution requirements will be developed. The literature in the integration of production and distribution is reviewed below.

Cohen and Lee (1988) considered a stochastic four-stage case involving venders, plants, distribution centers (DCs), and customer zones. They developed analytically-based models to coordinate production and distribution control policies to achieve synergies in performance. Their study represented a departure from traditional separate analysis to supply chain systematic approach. Chandra and Fisher (1994) considered a multi-product production-distribution problem with a single production facility and multiple customers. The computational results show a consistent improvement on the total cost by the coordinated approach over the decoupled one. Wilkinson et al. (1996) presented a case study involving an integrated production and distribution scheduling system for several multi-purpose sites over a wide geographical area. To find the most efficient schedule, all the plants were considered in detail and simultaneously as one large production system. Chen and Vairaktarakis (2005) considered a joint scheduling problem of production and distribution with the objective function taking into account both customer service level and product distribution. The possible benefit of using the proposed integrated model relative to a sequential model, where production and distribution operations were scheduled sequentially and separately, was investigated. The computational tests showed that in many cases a significant benefit could be achieved by integration.

The products studied in most of the above cases were interchangeable. In this dissertation an integrated production-scheduling and distribution problem for products with multiple options, such as automobiles, will be considered. The scheduling problem will involve multiple product options in the production smoothing considerations, and multiple destinations and multiple modes of transportation in the transportation considerations.

2.5 Plant assignment for a product produced in multiple facilities

For a product produced in multiple plants, assigning production orders to the multiple plants considering transportation costs and capacity constraint related to multiple product attributes are considered prior to the scheduling and sequencing stage at each plant. Some relevant research is reviewed below.

A related area with the plant assignment problem considered here is the workload allocation problem. The workload allocation problem addresses assigning products to multiple product lines or multiple production systems (Tetzlaff and Pesch, 1996) considering multiple objectives. Tetzlaff and Pesch proposed several nonlinear optimization models to optimize the performance in the throughput, work-in-process inventory, lead time, and utilization rate. Benjaafar and Gupta (1999) considered multi-product, multi-facility workload allocation problem. The objective for the problem studied is to minimize a function of the manufacturing lead time.

In this dissertation, plant assignment for a product with multiple product options is addressed. Multiple constraints are also related to multiple product options.

3. Research Objectives of this dissertation

The research objectives of this dissertation are as follows:

1. To develop sequencing solution procedures to address multiple sequencing objectives associated with multiple product options and to address broader sequencing considerations than MMALSP and CSP addressed individually.

2. To develop effective scheduling procedures to jointly address scheduling and distribution problems for products with multiple options.
3. To develop a solution procedure and computation tool to address plant assignment problem for products with multiple options.

PART 1

A HEURISTIC PROCEDURE FOR THE AUTOMOBILE ASSEMBLY-LINE SEQUENCING PROBLEM CONSIDERING MULTIPLE PRODUCT OPTIONS

This part is a paper to be published in the journal International Journal of Production Research by Fong-Yuen Ding and Jingxu He.

Abstract

Mixed-model assembly nowadays is a common practice in the automobile industry. In an automobile assembly plant, many car options often need to be considered in sequencing an assembly line, for example, the multiple sequencing objectives that consider a pattern, blocking, spacing, and smoothing of options. A general heuristic procedure is developed in this paper for sequencing automobile assembly lines considering multiple options. The procedure obtains an initial sequence by an enhanced constructive procedure, swaps orders for the most deteriorating category of objectives, and performs re-sequencing attempting to improve the swapped sequence. The heuristic procedure was shown to frequently improve the initial sequences by swapping and re-sequencing when swapping opportunities exist. A further improvement step is also proposed to perform a limited search based on the swapped solution. The limited-search improvement step was shown to be effective in further improving solutions from the heuristic procedure in the computational experimentation. Solutions from the heuristic procedure in conjunction with the limited-search improvement step were compared to those from the simulated annealing procedure for large-size problems and showed relatively positive results.

1. Introduction

Just-in-time production calls for manufacturing a variety of finished goods by using a mixed-model assembly line to set a smooth pace of material usage and even workload in the manufacturing system (Monden, 1997). Mixed-model assembly nowadays is a common practice in the automobile industry. Many researchers have attempted to address the mixed-model assembly line sequencing problem (MMALSP) for achieving balanced workload (Yano and Rachmadugu, 1991; Miltenburg and Goldstein, 1991; Sumichrast et al, 1992), smooth production (Miltenburg, 1989; Kubiak and Sethi, 1991; Inman and Bulfin, 1991; Cheng and Ding, 1996), or reduced line stoppage (Zhao and Ohno, 1994). The problem of sequencing mixed-models to smooth part usage or to minimize workload imbalance has been shown to be NP hard (Kubiak, 1993). In a different research focus, assembly-line design with mixed models is a topic that many researchers (for example, Stadzisz and Henrioud, 1998 Fouda et al. 2001) have contributed in order to achieve an efficient assembly process flow.

Methods to solve the mixed-model assembly line problems include: 1) a greedy method to consider either one-stage or two-stage variation (e.g., Monden, 1997; Bautista et al., 1996; Zhu and Ding, 2000); 2) the assignment-problem model (Kubiak and Sethi, 1991); 3) the ideal due-date method (Inman and Bulfin, 1991); 4) a bi-partite graph (Steiner and Yeomans, 1993); 5) a mixed-integer-program based method (Miltenburg, 1989); 6) dynamic programming (Miltenburg et al., 1990); and 7) a meta-heuristic (such as genetic algorithm by Hyun et al., 1998). A mixed-model assembly line sequencing problem for smoothing production can be formulated to minimize the product-level (e.g., Milternburg, 1989; Inman and Bulfin, 1991), part-level (e.g., Monden, 1997; Bautista et al., 1996), or multi-level (Milternburg and Sinnamon, 1989) usage variation. McMullen and Frazier (2000), McMullen et al. (2000), McMullen (2001a), and McMullen (2001b) applied several meta-heuristic approaches, including the tabu search, simulated annealing, genetic algorithm, and ant-colony method in solving mixed-model assembly line sequencing problems with an objective function that combines two objectives by weights.

The car sequencing problem (CSP), studied by many researchers, involves sequencing cars on the assembly line subject to a capacity constraint for each of k considered options, that is, there can be at most r_i cars with option i in every s_i consecutive cars in the sequence. Many researchers have attempted to solve this problem by treating the constraints either as hard or soft constraints. CSP was shown to be NP-hard (Gent, 1998). Solution approaches applied to CSP include the greedy method (Gottlieb et al., 2003), local search (Lee et al., 1998; Davenport and Tsang, 1999; Puchta and Gottlieb, 2002), meta-heuristics including ant-colony optimization (Gottlieb et al., 2003, Gravel et al., 2005), neural networks (Smith et al. 1996), simulated annealing (Chew et al., 1991), and genetic algorithms (Warwick and Tsang, 1995).

In practice, an automobile company often needs to address broader considerations than CSP and MMALSP. In today’s automobile industry, many options are available on various car models; and in sequencing assembly lines it is common to consider production capacities associated with many options and their various requirements. Drexl and Kimms (2001) considered the joint problem of CSP and MMALSP at the product level by using column generation techniques. Computational experiments were conducted on relatively small-size problems. Bergen et al. (2001) addressed the vehicle assembly-line sequencing problem that considered eight types of hard or soft constraints. Vehicles of identical attributes partially related to car options are first split into lots; and similar lots are then grouped into hourly batches. The sequencing problem consists of assignment of batches to hourly slots and sequencing of lots within batches. Three approximation algorithms, local search, backtracking, and branch-and-bound algorithm were presented to solve the problem.

Estellon et al. (2005) addressed CSP as Renault's car sequencing problem with the additional constraints on the number of consecutive vehicles having the same color. Reducing the number of color purges is also included in the objective function. A very-large neighborhood search and a very-fast local search approach were presented to solve the problem. The former approach is an integer-linear-programming based neighborhood search, while the latter is a local search method based on five transformations. Prandtsetter and Raidl (2005) presented a neighborhood-search approach that combines the general variable neighborhood search with integer linear-programming methods. Gagne et al. (2006) presented an ant-colony-optimization approach to consider Renault's car sequencing problem and obtained solutions better than the simulated annealing approach by Chew et al. (1991). Muhl et al. (2003) applied the genetic algorithm, simulated annealing, gradient search, and stochastic search to solve assembly-line sequencing problems that consider requirements of multiple shops.

In this paper, we address the assembly-line sequencing problem considering multiple options. The research is motivated by a U.S. automobile manufacturer, which annually produces millions of vehicles of many vehicle lines from passenger cars to commercial trucks. In the company, mixed-model assembly is applied in practically all assembly plants based on various options on vehicles. The Scheduling department at the headquarters performs scheduling and sequencing for all vehicle lines.

Sequencing considerations

Sequencing considerations for vehicle orders on assembly lines among the various plants of the automobile manufacturer are certain combinations of options-related requirements including the following *categories* of objectives:

a) *Repeating pattern* – Use of certain fixtures among the cars in sequence must follow a certain pattern, e.g., 1-2-4-3-1-2-4-3-1-2-4-3 for 4 fixtures. A fixture choice is usually related to a certain car option in the model.

b) *Paint blocking* – Grouping of same-color cars in sequence gives significant cost savings from frequent paint purges; there is also a limit on the number of cars in a color block. Typically, the number of different colors considered in a day can be from 6-14.

c) *Spacing rules* (SR) – For a certain option i, e.g., moon-roof, a spacing rule is stated in the format of no more than r_i in s_i consecutive orders, or r_i:s_i; e.g., 2:3 means that no more than 2 moon roofs in 3 consecutive orders.

d) *Smoothing* – To evenly spread out cars with a certain option in the sequence.

e) *Jobs per hour* (JPH) – To limit the total number of orders with a certain option per hour ("discrete" hour) to a specified quantity.

Among these considerations, c) is equivalent to the consideration in CSP, and d) is equivalent to the consideration in MMALSP. Except a) that is usually considered as a "hard" requirement, the others are usually considered as "soft" requirements with different priorities. The number and importance levels of various sequencing criteria would vary from plant to plant. There can also be multiple objectives in a category of sequencing consideration; and there can be various combinations of objectives.

The objective function

In this paper the levels of importance will be considered by penalty weights based on *cost impact* and natures of various objectives. The objective function can be evaluated as the sum of [penalty weight] times [amount of violation] in various objectives. It is proposed that a penalty weight for each sequencing objective will represent the approximate "cost impact" of a unit of violation on the production system. Using penalty weights consistent with the cost impact rather than using stepwise weights has the advantage of more directly associating the cost impact and natures of different objectives. A sequencing objective can act as if having absolute or relative dominance over other objectives. An absolute dominance objective can be represented by a very high weight.

2. A General Sequencing Procedure and a Limited-Search Improvement Step

A general sequencing procedure based on an enhanced constructive method is presented in this paper with the capability of swapping and re-sequencing. The procedure can be considered as an alternative to other heuristics as it is relatively fast and frequently provides improved solutions. Moreover, an improvement step based on the swapped solution generally leads to improved solution values. The appropriateness of the proposed procedure generally relies on the nature of the objectives' being local and relational. The procedure considers the sequencing objectives each time when an order is added to the sequence or is swapped with another order in the sequence. The procedure is intended to address large-size sequencing problems with sufficient randomness in vehicle options associated with various vehicle orders.

2.1 Ideal unit

In order to estimate the goodness of the next added vehicle order, the definition of an ideal unit (ideal vehicle order) is used. An ideal unit is represented by a set of option values (e.g., [2, 0, 1], where 2 represents a value of option 1, 0 represents a value in option 2, and 1 represents a value in option 3) that best meet all sequencing objectives without incurring any penalties. For example, the following conditions are desirable for vehicle orders each with 3 options: Objective 1 is to desire color blocking with a block size of 5 for each color (and color is option 1); objective 2 is to desire meeting a spacing rule of 1:2 for option 2; and objective 3 is to desire smoothing vehicles orders with option 3, which are present in the order pool at 50%. Given that the first 3 orders are [3, 1, 1], [3, 0, 0], and [3, 1, 0], the ideal unit for the next selection is thus [3, 0, 1] based on the 3 options.

2.2 The constructive-swapping-resequencing (CSR) heuristic procedure

The heuristic procedure is developed in this paper to solve the assembly-line sequencing problem considering multiple options. The procedure is a combination of an enhanced constructive method by finding the best match in the next unit of selection considering all objectives, and an enhancement based on swapping and re-sequencing. For ease of reference in the paper, the heuristic procedure will be termed as a CSR heuristic. While the procedure is based on a constructive approach that only considers the immediate impact of the next order selection, the constructive approach itself is enhanced by modified definitions in ideal units and use of dynamic weights for some objectives (see Sections 3.1 and 3.2), and is further enhanced by swapping and re-sequencing.

The CSR heuristic procedure steps are stated as follows:

Phase A. Constructing an initial sequence by an enhanced[1] constructive approach

Step 1. Assign the first order. The first order to be placed in the sequence is the order that has the largest number of option selections and the least popular blocking (if considered) option.

Step 2. Add an order to the sequence. The order to be added is the one with the minimum weighted violation-penalty cost considering all sequencing objectives in relation to the *ideal unit*. Repeat Step 2 until all orders are added to the sequence.

Phase B. Swapping for improving strains at the end of the sequence

Step 1. After Phase A, determine a "*targeted*" *category of objectives* that has the greatest sum of [penalty weight] times [increase in amount of deterioration] toward the end of sequence (e.g., the last 10% of the sequence as compared to the first 10% of the sequence).

Step 2. From the end of the sequence, find an order with a high violation in the targeted category of objectives, and *swap* it with a prior order that has the same values for the attributes of intended absolute dominance but will improve the targeted category of objectives. If multiple prior orders exist in this step, choose the one that results in the least total cost increase in the objectives other than the targeted objectives from the swap. Continue Step 2 until no more swapping can be found to improve the targeted category of objectives.

---

[1] The enhancement is achieved through using a modified definition in ideal units and dynamic weights for some objectives, which will be described in Sections 3.1 and 3.2.

(12 orders:)

Fixture	Color	SR	SR	JPH	Order index
1	1	1	1	1	1
2	2	1	0	0	2
3	1	0	0	1	3
1	2	1	0	1	4
2	1	0	0	1	5
3	3	0	1	0	6
1	2	0	1	0	7
2	1	1	1	0	8
3	2	1	1	0	9
1	1	0	1	0	10
2	3	0	0	1	11
3	1	1	0	1	12

Figure 1. 12 orders to be sequenced with 5 options

Phase C. Re-sequencing

After Phase B, re-perform Phase A while keeping the same option values of the objectives of the "targeted" category of objectives considered in Phase B to attempt to improve the swapped sequence.

Phase D. Final Sequence[2]: The best sequence from Phases A), B), and C) is the final sequence.

2.3 An illustrative example of the CSR procedure

To illustrate the CSR procedure, a small example of 12 vehicle orders assembled at 6 units per hour with the following sequencing objectives related to Options 1-5, respectively, are assumed: 1) fixture pattern 1-2-3, 2) desirable color blocking of size 4 ± 1, 3) spacing rule of 1:2, 4) spacing rule of 1:2, and 5) JPH of a maximum of 3 per hour. These objectives have the unit violation penalties of 100, 10, 2, 2, and 18, respectively. These 12 orders in 12 rows are given in Figure 1.

[2] While a better solution from swapping and re-sequencing is attempted, the best solution of the three phases of the CSR procedure will be considered as the final sequence in case that swapping and re-sequencing did not generate a better sequence.

(After Phase A:)

Fixture	Color	SR	SR	JPH	Order index
1	1	1	1	1	1
2	1	0	0	1	5
3	1	1	0	1	12
1	1	0	1	0	10
2	2	1	0	0	2
3	2	1	1	0	9
1	2	0	1	0	7
2	3	0	0	1	11
3	3	0	1	0	6
1	2	1	0	1	4
2	1	1	1	0	8
3	1	0	0	1	3

Figure 2. Sequence after Phase A

(After Phase B:)

Fixture	Color	SR	SR	JPH	Order index
1	1	1	1	1	1
2	1	0	0	1	5
3	1	1	0	1	12
1	2	1	0	1	4
2	2	1	0	0	2
3	2	1	1	0	9
1	2	0	1	0	7
2	3	0	0	1	11
3	3	0	1	0	6
1	1	0	1	0	10
2	1	1	1	0	8
3	1	0	0	1	3

Figure 3. Sequence after Phase B

The sequence after Phase A of the CSR heuristic procedure is obtained and listed below in Figure 2. In this sequence, the weighted deteriorations in the last ½ hour in comparison to the first ½ hour for the 4 categories of objectives (fixture pattern, color, SR, and JPH) are 0, 20, 2, and 0, respectively. Thus, color identified as the swapping target based on its cost improvement potential. The units with order indices 10 and 4 are further identified for swapping to reduce the color-blocking penalties while the fixture pattern is not violated.

The swapped sequence after Phase B of the CSR heuristic is obtained and listed in Figure 3. It can be seen that the color-blocking penalty is reduced by 10 after swapping. The JPH penalties, however, increase after swapping since only color was considered in swapping.

(After Phase C:)

Fixture	Color	SR	SR	JPH	Order index
1	1	1	1	1	1
2	1	0	0	1	5
3	1	1	0	1	12
1	2	0	1	0	7
2	2	1	0	0	2
3	2	1	1	0	9
1	2	1	0	1	4
2	3	0	0	1	11
3	3	0	1	0	6
1	1	0	1	0	10
2	1	1	1	0	8
3	1	0	0	1	3

Figure 4. Sequence after Phase C

(Optimal:)

Fixture	Color	SR	SR	JPH	Order index
1	1	1	1	1	1
2	1	0	0	1	5
3	1	1	0	1	12
1	2	0	1	0	7
2	2	1	0	0	2
3	2	1	1	0	9
1	2	1	0	1	4
2	1	1	1	0	8
3	1	0	0	1	3
1	1	0	1	0	10
2	3	0	0	1	11
3	3	0	1	0	6

Figure 5. Optimal Sequence

In Phase C of the CSR heuristic procedure, re-sequencing is performed by repeating Phase A of the heuristic procedure while keeping color assignment and fixture pattern (as shown in the dotted boxes) unchanged. The final sequence is listed in Figure 4. In this sequence, JPH was improved while fixtures and colors remain the same. The total violation of this sequence is 18 (10 in color and 8 in SR), while the optimal solution value is 16 as found by complete enumeration.

The optimal sequence is given in Figure 5 with penalties of 10 in color violation and 6 in SR violations.

2.4 A limited-search (LS) improvement step for improving the CSR solutions

In the CSR procedure, as swapping generates disturbance to the initial sequence with increased violations among objectives other than the targeted objectives, re-sequencing attempts to recover from such disturbance. However, it is possible that there may be too much swapping disturbance and the increased violations become unrecoverable by re-sequencing. Based on this insight, an improvement step is proposed to reduce the number of swaps in steps (such as in 3 steps), while re-sequencing is performed at the end of each step. With d steps, re-sequencing is repeated for d times; and the best solution among these d steps and the initial CSR procedure becomes the final solution. This gives a "limited search" by moving to d points each with a potential of improvement over the initial solution because each solution has a part of improving swaps and attempted recovering by re-sequencing. The size of "d" can be set in reference to the total number of swaps in the swapping phase. For example, d can be set at 3 which reduces the number of swaps by 25% in three steps to explore the solutions of 75% swaps, 50% swaps, and 25% swaps, respectively.

3. Considerations in the Proposed CSR Procedure

3.1. JPH considerations for enhancing sequence construction in Phase A

For clarity a *content ratio* refers to the ratio of the total number of orders with an option over the total number of orders; a *JPH ratio* refers to the ratio of the hourly limit for the number of orders with an option over the number of orders in an hour; and an *application ratio* is the ratio of the cumulative number of orders with an option over the total number of orders till the considered stage. In order for a JPH requirement to be achievable, [JPH ratio $\geq$ content ratio] is assumed in this paper.

In the proposed procedure, smoothing is applied to achieve the JPH objectives because smoothing generally leads to even JPH. However, to relax the more restrictive nature of smoothing (which follows the content ratio) in comparison to JPH, so that sequencing requirements other than JPH can "compete" better in the constructive procedure (Phase A), a tolerance of an "indifference range" is applied in smoothing for JPH. To this end, a fraction (such as half) of the gap between the JPH and content ratios can be considered as an *indifference range* in smoothing for JPH. For example, with hourly production of 60 units, assume that a maximum JPH of 40 moon-roofs per hour is desirable at a JPH ratio of 40/60 = 0.67, and that the content ratio is 0.5, which is also the target for smoothing. The indifference range can be set to ½(0.67-0.5) = 0.085; that is, if the application ratio is within 0.5±0.085, the choice of a unit becomes indifferent with no penalty regarding the option.

Further consideration in JPH is to apply a dynamic penalty weight that increases as sequence construction progresses to the end of each hour (e.g., the last 10% of each hour). Initially the penalty weight is set to an "algorithmic weight" as [JPH violation weight]/[the number of units per hour] in smoothing for each JPH objective during sequence construction. Since the direct impact of smoothing on JPH violation becomes more and more significant when approaching the end of each hour ("discrete" hour), increasing weights are applied till reaching the full JPH violation weight (in Computer Experimentation, curve fitting is applied to obtain a nonlinear weight function giving weight values that start from the algorithmic weight to increase to the full JPH violation weight in the last 10% units of an hour).

3.2. Spacing-rule consideration for enhancing sequence construction in Phase A

Generally, a spacing rule (SR) specifies when an option value of 1 is *not* desirable (with a violation). In order to have a definition of an ideal unit regarding when to place a “1,” another measure, the *content ratio*, for smoothing may be jointly applied. That is, if the *application ratio* is less than the *content ratio*, then “1” is preferred at the current iteration for smoothing consideration. The following list gives an example of two possible joint definitions for an ideal unit in SR.

Under Definition 1, the multiplication of [spacing-rule preference] (1 is applicable when 0 or 1 is desirable for SR) and [smoothing preference] is considered. Under Definition 2, slightly modified values are considered. Specifically, a “0.75” and “0.25” in rows 2 and 3 are intended to lead to an earlier placement of a “1.” (Even a “0.25” still makes a 0 more desirable, it will increase the chance of choosing a 1 when other objectives are considered simultaneously.) This consideration intends to increase the rate of placing 1’s earlier in the sequence in order to prevent strains later in the sequence. In general, if the content ratio is relatively tight relative to the spacing rule, it is preferred to place 1’s earlier whenever possible to prevent later strains in the sequence.

Table 1. Joint definitions for an ideal unit in SR

Spacing-rule preference	Smoothing preference	Definition 1	Definition 2
0 or 1	1	1	1
0 or 1	0	0	0.75
0	1	0	0.25
0	0	0	0

3.3. Alleviating the weakness of the constructive part by swapping in Phase B

Due to the constructive nature (Phase A) of the proposed procedure, the sequencing results of certain objectives generally deteriorate at the later stages when there are fewer units to choose from (Monden 1997). To alleviate this deterioration effect, swapping (Phase B) is included in the CSR procedure. To determine the most deteriorating category of objectives, the first, say, 10% and last 10% of the sequence regarding each objective are compared. The difference in the numbers of violations in these two sequence sections is multiplied by the corresponding penalty weight to give an estimate of sequence deterioration in each objective. The category of objectives of the most total cost deterioration can be considered as the target for swapping improvement. In each swapping iteration, a unit as the swapping source is first identified. If multiple units exist as candidates for a destination (that can improve the targeted objectives by a swap), the unit that has the least increase in penalties in other objectives will be selected. In case that a sequence does not have obvious deterioration, this indicates that the sequence found by the enhanced constructive phase is a relatively good sequence.

Performing swapping for color, for example, can eliminate small color blocks, such as color blocks of size 1 or 2 at the end of the sequence, by combining a small color block with a large block elsewhere. In doing so, care needs to be taken so that the order in the option values of an apparent "absolute-dominance" objective should not deteriorate. In our computer experimentation, the color objective or spacing-rule objectives will be selected as a swapping target based on an evaluation of the constructed sequence.

In the computational experiments in this paper, however, limited swapping for color blocks and spacing rules is explored. Only small color blocks of size 1 at the end of the sequence are attempted for combination with same-color blocks earlier in the sequence, although it is possible to eliminate color blocks of size larger than 1. Similarly, in swapping for improving the spacing-rule objectives, only backward insertion of units from the later part of the sequence is considered in the computational experiments, although there may exist two-way or three-way swapping opportunities.

3.4. The rationale for re-sequencing in Phase C

After swapping is performed for a targeted category of objectives, the enhanced constructive step is reapplied while keeping the option values of the targeted category of objectives for swapping. This step is referred to as re-sequencing (Phase C). Units will thus be reassigned while maintaining the option values of objectives of the swapping target. Since the sequence after swapping has been "randomized" in objectives other than the targeted objectives, re-sequencing reconsiders the order in the sequence to reduce such randomization and should improve the sequence.

4. Computational Experimentation

4.1. Randomly generated problems

The algorithm was tested on 3 cases of randomly generated problems. Each problem has 4,800 car orders representing a week of orders at a production rate of 60 units per hour. There are 3 fixtures in a specified pattern when fixtures are considered. Ten colors are assumed when color blocking is considered, and the ideal color block size is 17±3 with a penalty cost of P_c. When a color block is <14, a penalty of P_c is applied; however, if a color block is >20, each unit over 20 is penalized for $P_c/17$. There are 7 JPH, 3 SR, and 3 smoothing options, respectively, whenever it is applicable. The penalty weight for JPH is considered at 3 levels: 10/60/240, while in Phase A of the CSR procedure 1/60 of the JPH penalty weight is used for a regular unit of violation. Smoothing penalty calculation is based on the squared variation between the application and content ratios, when smoothing is considered. When a random problem is generated and an attribute (option) is considered, the control parameters for problem generation are as follows: fixtures are assumed to have a perfect mix; the chance of any color is assumed to be equal; and the content ratio of a JPH option is set at 50%. Furthermore, the content ratios of the SR options for Cases 1, 2a) and 2b) and 3 are set at 50%, 49%, 51% and 51%, respectively. In order to generate problems with the most sequence deterioration taking place at different categories of objectives, the content ratios of SR options are set differently in Cases 2a) and 2b). No correlation among various options is assumed.

Table 2. Results of the CSR heuristic and LS improvement step for Case 1

Note: Each value is an average of 10 runs.

		No. of color violations	No. of JPH violations	No. of SR violations	Overall objective value	CPU time (seconds)	No. of swaps
JPH penalty weight = 10/hr	Before swaps	228.9	7.9	373.5	3,572.8	23.33	75.9
	After swaps	141.5	17.2	523.9	2,905.3		
	Re-sequencing	141.5	9.4	427.1	2,645.7		
	Best of 3 phases	141.5	9.4	427.1	2,645.7		
	After LS improvement	141.5	9.4	427.1	2,645.7	9.75	-
JPH penalty weight = 60/hr	Before swaps	234.4	7.4	464.9	4,187.1	22.73	74.7
	After swaps	148.9	18.6	619.6	4,142.5		
	Re-sequence	148.9	8.5	448.3	3,205.9		
	Best of 3 phases	148.9	8.5	448.3	3,205.9		
	After LS improvement	151.9	7.8	447.5	3,186.3	9.72	-
JPH penalty weight = 240/hr	Before swaps	234.4	7.3	459.7	5,484.1	22.44	75.7
	After swaps	148.4	15.5	614.3	6,729.3		
	Re-sequence	148.4	7.2	546.4	4,613.5		
	Best of 3 phases	148.4	7.2	546.4	4,613.5		
	After LS improvement	160.7	6.3	545.4	4,531.1	9.71	-

Case 1. Fixture – Color – JPH – SR (as the considered objectives)

The penalty weights per violation for these objectives are set at 100, 12, 10/60/240, and 2, for fixture pattern, color, JPH, and spacing rule, respectively. Spacing-rule objectives are rather tight (1:2 at a 50% average content ratio). Based on the selection criteria for the swapping target, color is identified as the target. The average test results are summarized in Table 1. It can be seen that the CSR heuristic procedure improved the initial solutions significantly after swapping and re-sequencing in terms of the overall objective values. It can also be seen that re-sequencing effectively improves the sequence that was "randomized" after swapping. Re-sequencing does not affect the number of color blocks but improves other objectives. The best of 3 phases (construction, swapping, and re-sequencing) turned out to have the same solution values as those from re-sequencing in all 3 test groups. Further applying the limited-search (LS) improvement step with d=3 gave further improved solutions in 2 of 3 test groups with a relatively small computing time increase.

Table 3. Results of the CSR procedure and LS improvement step for Case 2a)

		No. of color violations	No. of JPH violations	No. of SR violations	Overall objective value	CPU time (seconds)	No. of swaps
JPH penalty weight = 10/hr	Before swaps	41.3	7.0	88.5	370.9	50.61	9
	After swaps	32.3	8.1	95	367.9		
	Re-sequencing	32.3	8.1	88.1	354.1		
	Best of 3 phases	35.1	6.7	86.8	345.9		
	After LS improvement	34.6	6.4	86.4	340.6	17.84	-
JPH penalty weight = 60/hr	Before swaps	64.5	10.1	118.6	1,036.8	52.67	12.2
	After swaps	51.9	10.7	126.8	1,051.3		
	Re-sequence	51.9	8.6	126.2	924.1		
	Best of 3 phases	51.9	8.0	127.4	890.5		
	After LS improvement	55.8	7.1	121.6	836.6	20.77	-
JPH penalty weight = 240/hr	Before swaps	64.2	9.6	119.5	2,735.5	51.06	17.5
	After swaps	51.8	10.4	127.8	2,907.0		
	Re-sequence	51.8	9.5	125.8	2,687.0		
	Best of 3 phases	55.1	7.3	126.0	2,169.3		
	After LS improvement	57.3	6.3	124.9	1,933.7	19.83	-

Note: Each value is an average of 10 runs.

The computer experimentation is conducted on a Pentium 2.0 GHz notebook computer using codes written in MATLAB. The average CPU times of a test problem are 22.83 seconds and 9.73 seconds for the CSR procedure and the LS improvement step, respectively. These computing times may be further reduced when "model groups" (which contain units of the same option combinations) exist and order selection is simplified to choose a model group.

<u>Case 2. Color –JPH – SR</u> (as the considered objectives)
<u>Case 2 a) (Color as the swapping target)</u>

The penalty weights are set at 3, 10/60/240, and 2 for color, JPH and SR, respectively. Due to the problem parameters (specifically, the average content ratio for each SR option is set at 49%) and penalty weights, color is selected as the swapping target after the enhanced construction steps. The results in Table 2 showed that the CSR procedure noticeably improved the overall objective values, and that the LS improvement

Table 4. Results of the CSR procedure and LS improvement step for Case 2b)

		No. of color violations	No. of JPH violations	No. of SR violations	Overall objective value	CPU time (seconds)	No. of swaps
JPH penalty weight = 10/hr	Before swaps	139.4	8.3	355.1	1,921.7	58.72	61.1
	After swaps	240.6	16.1	293.9	2,274.8		
	Re-sequencing	246.9	9.9	293.9	2,055.8		
	Best of 3 phases	151.5	8.1	345.7	1,920.0		
	After LS improvement	149.2	8.5	346.5	1,918.3	11.86	-
JPH penalty weight = 60/hr	Before swaps	216.1	10.7	394.9	2,869.9	58.47	79.9
	After swaps	329.9	15.8	315	3,474.1		
	Re-sequence	257.7	10.1	315	2,768.2		
	Best of 3 phases	252.4	10.1	329	2,763.2		
	After LS improvement	255.8	8.7	323.7	2,584.2	11.35	-
JPH penalty weight = 240/hr	Before swaps	238.5	8.3	394.3	3,288.7	57.02	79
	After swaps	331.4	15.5	315.3	4,381.4		
	Re-sequence	284.1	9	315.3	3,271.9		
	Best of 3 phases	260.8	8.4	354.8	3,161.8		
	After LS improvement	268	7.1	328.1	3,052.4	9.71	-

Note: Each value is an average of 10 runs.

step with d=3 further improved the solution values. The average CPU time for a test problem of CSR procedure is 51.45 seconds, and the average additional time of the LS improvement step is 19.48 seconds.

Case 2 b) (Spacing rules as the swapping target)

The same penalty weights as in Case 2a) are used here. The average content ratio of each SR option is set at 51%. Due to the problem parameters and penalty weights, the spacing rules are selected as the swapping target after the enhanced construction steps. Limited swapping for SR is performed by moving a unit (1, 1, 1) (in the three SR options) that is not surrounded by two 0's in an SR option from the back of the sequence and inserting it between two units that have consecutive 0's in the SR option at the front of the sequence (it can be shown that the SR penalty score is improved through such an insertion). The average results are given in Table 3. The average computing time is 58.07 seconds for the CSR procedure. A considerable number of swaps were performed

Table 5. Results of the CSR heuristic procedure and LS improvement step for Case 3

		No. of JPH violations	No. of SR violations	No. of Smoothing violations	Overall objective value	CPU time (seconds)	Total No. of Swaps
JPH penalty weight = 10/hr	Before swaps	8.3	298.3	2,784.6	2,668.5	66.56	16.8
	After swaps	9.3	286.2	3,874.2	3,174.9		
	Re-sequencing	7.5	286.2	2,595.6	2,517.6		
	Best of 3 phases	8.4	292.8	2,495.0	2,502.7		
	After LS improvement	7.8	290.9	2,416.0	2,449.6	28.67	-
JPH penalty weight = 60/hr	Before swaps	8.1	303.6	2,662.6	3,031.7	66.20	17.0
	After swaps	8.8	286.6	4,071.5	3,704.2		
	Re-sequencing	8.6	286.6	3,235.5	3,280.1		
	Best of 3 phases	8.2	294.1	2,600.6	2,968.7		
	After LS improvement	6.7	291.4	2,621.8	2,878.5	46.60	-
JPH penalty weight = 120/hr	Before swaps	9.4	303.4	2,757.6	3,720.4	65.86	12.1
	After swaps	10.2	286.6	4,352.2	4,546.5		
	Re-sequencing	8.6	286.6	2,803.9	3,580.4		
	Best of 3 phases	8.5	297.6	2,562.4	3,491.6		
	After LS improvement	7.0	291.9	2,378.4	3,196.8	50.10	-

Note: Each value is an average of 10 runs.

for improving SR violations in these runs; and in 2 of the 3 test groups improved average solution values from the initial solutions were obtained after swapping and re-sequencing (due to the fact that color blocks also increased significantly to offset savings in spacing-rule violations, group 1 did not improve on the average). In all 3 test groups, the best-of-3-phases solutions gave better average solution values than the initial solutions; and in all groups the solutions after the LS improvement with d=3 gave further improved average solution values.

Case 3. SR – JPH – Smoothing (as the considered objectives)

The penalty weights are set at 4, 10/60/120, and 0.5 for SR, JPH, and smoothing, respectively. Based on the criterion in selecting objectives for swapping, SR is chosen as the swapping target for these problems. From the average results in Table 4, it can be seen that after swapping the smoothing objectives deteriorated noticeably; however, re-sequencing further improved the disturbed sequences. The average overall objective values were improved from the combined steps of swapping and re-sequencing in 2 of 3 test groups, while the best of 3 phases gave improved average solution values in all 3 test groups. The solutions from LS improvement step with d=5 also show further improvement in all 3 test groups. The average computing times are 66.21 and 41.79 seconds for the CSR heuristic and the LS improvement step, respectively.

4.2. Comparison with solutions obtained by simulated annealing

In order to provide a base line for the proposed heuristic procedure, simulated annealing (SA) is considered in this paper for a comparison with the CSR heuristic procedure and LS improvement step for randomly generated large-size problems.

(Several other researchers, e.g., Estellon et al., 2005, and Gagne et al., 2006, also compared results with simulated annealing results.) A simulated annealing (Eglese, 1990) approach is based on the analogy to material annealing, which is generally controlled by the starting temperature and cooling rate. In simulated annealing, at each iteration a neighborhood solution of a value z_1 is obtained by a neighborhood-generation method from the current solution of a value z_0. Let $\Delta z = z_1 - z_0$. If $\Delta z < 0$, the new solution replaces the current solution. However, if $\Delta z \geq 0$, a probability P(A) = $e^{-\Delta z / K_b T}$, where K_b is the Boltzman constant and T is the current temperature, will be applied to determine whether the inferior neighborhood solution will replace the current solution. This prevents the search process from being trapped in a locally optimal solution. The temperature is adjusted after every N iterations according to a cooling rate α; and the Boltzman constant is set so that an initial P(A) is equal to a pre-selected value at an inferiority base.

SA is conducted for 5 randomly generated problems in each of Case-1 and 3 problems addressed in Section 4.1. The SA parameters are given in Table 5. The

Table 6. Parameters for the simulated annealing computer experimentation

	No. of problems	Initial solution	Initial temperature	Ending temperature	Cooling rate	N	Total no. of iterations	P(A)	Inferiority base
Case 1	5	Best of 10,000 random solutions	50	0.005	0.9995	50	920,850	0.05	0.05
Case 3	5	Best of 10,000 random solutions	50	0.001	0.999	50	540,750	0.05	0.05

Table 7. Simulated annealing average results

	10,000 random solutions		**Initial solution value**	**Final SA solution**						CPU time (seconds)	**CSR+LS solution value**
				No. of violations					**Total penalty score**		
	Maximum solution value	Minimum solution value		Fixture	Color blocks	JPH	SR	Smoothing			
Case 1	405,545.6	380,719.2	380,719.2	39.6	1,524.6	0	2,077.0	-	**26,409.2**	10,927.6	**3,078.4**
Case 3	16,558,428.0	538,919.8	538,919.8	-	-	1.2	1,749.8	6,700.9	**13,772.1**	11,626.8	**3,095.3**

Note: Each value is an average of 5 trials.

neighborhood-generation method in this SA experimentation is to swap two randomly selected orders in the current sequence; and an initial solution is the best of 10,000 randomly generated sequences (similar to the simulated annealing approach by McMullen and Fraizer (2000) for sequencing mix-model assembly lines). The SA average results of 5 randomly generated problems are given in Table 6 for each tested case.

From data in Table 6, it can be seen that with careful parameter adjustment, SA is generally effective in improving solutions (From 380,719.2 to 26,409.2 and 538,919.8 to 13,772.1, respectively) considered in Cases 1 and 3; and it appears that it will require a relatively long time (>3 hr, on a Pentium M 2.0GHZ PC, the same system used for later on in this research) in order to reach the quality level of the CSR solutions with LS improvement. After the CPU times 3.03 and 3.23 hours, respectively, the SA solutions are still far away from those (3,078.4 and 3,095.3) of the CSR heuristic with d=3 and d=5 in LS improvement, respectively.

A further experiment is conducted by using the solutions from the CSR procedure with LS improvement step as the initial solution in SA. A lower inferiority base (0.005) is used considering the fact that starting from a relatively good solution, as too much mobility to a more inferior solution can cause rapid solution deterioration due to the nature ("deep valleys") of the considered problem. After 920,850 and 770,900 iterations (3.03 hours and 2.22 hours, respectively) of SA runs, these solutions were improved by 2.17% and 3.27% on the average, for the five Case 1 problems and five Case 3 problems, respectively. This showed that the CSR procedure solutions in these tested cases were at a relatively good solution-quality level comparing to the SA approach.

5. Summary and Conclusions

Automobile assembly lines often have various option-related sequencing considerations including color blocking, maximum number of jobs per hour for a certain option, and rules for spacing units of certain options. These commonly encountered sequencing considerations for options attempt to keep assembly-line work loads, processing requirements, or material usage in a good balance. When multiple objectives related to options are considered, the overall results can be evaluated in the objective function by using various penalty weights representing their cost impact levels to the assembly line in violating these objectives.

A constructive-swapping-re-sequencing heuristic procedure was developed to sequence units of various option values. The procedure includes an enhanced sequence-construction approach considering the multiple sequencing objectives. A weakness in sequence deterioration at the end of the constructed sequence may be reduced by using swapping; and the randomized sequence from swapping is re-sequenced for improvement while keeping the option values of the swapped objectives unchanged. A further improvement step is proposed by searching over a limited number of solutions with a reduced number of swaps and re-sequencing.

Computational experimentation showed that the proposed heuristic procedure generally achieves improved overall objective values through the combined steps of swapping and re-sequencing, and that the limited-search improvement step gives further improvement. Comparisons with simulated-annealing solutions showed positive results with the CSR heuristic procedure and LS improvement step. The construction, swapping, and re-sequencing steps work together attempting to achieve good solutions in considering multiple sequencing objectives related to vehicle options. The improvement step gives further improvement by searching through a limited set of potentially better solutions with a relatively small additional computational burden. Further research considerations can apply forward-looking in sequencing by considering the un-sequenced order pool. Tighter lower bounds can also be developed.

6. References

1. Bergen, M, P. Beek, and T. Carchrae, 2001, "Constraint-based Vehicle Assembly Line Sequencing," *Proceedings of the 14th Conference of the Canadian Society for Computational Studies and Intelligence*, Springer, 88-99.

2. Bautista, J., R. Companys, and A. Corominas, 1996, "Heuristics and exact algorithms for solving the Monden problem," *European Journal of Operational Research*, 88, 101-113.

3. Cheng, L. and F. Ding, 1996, "Sequencing mixed-model assembly lines to minimize the weighted variations in just-in-time production systems," *IIE Transactions*, 28 (11), 919-927.

4. Chew, T.-L., J.-M. David, A. Nguyen, and Y. Tourbier, 1991, "Solving constraints satisfaction problem with simulated annealing: The car sequencing problem revisited," in 12th International Conference on Artificial Intelligence, Expert Systems, and Natural Language, 405-426.

5. Davenport A. and E. Tsang, 1999, "Solving constraint satisfaction sequencing problems by iterative repair," *Proceedings of the First International Conference on the Practical Applications of Constraint Technologies and Logic Programming*, 345-357.

6. Drexl, A. and A. Kimms, 2001, "Sequencing JIT Mixed-Model Assembly Lines Under Station-Load and Part-Usage Constraints," *Management Science*, 47(3), 480-491.

7. Eglese, R.W., 1990, "Simulated Annealing: A tool for Operations Research," *European Journal of Operational Research*, 46, 271-281.

8. Estellon, B., F. Gardi, and K. Nouioua, 2005, "Real life car sequencing: very large neighborhood search vs. very fast local search," Working paper, Laboratoire d'Informatique Fondmentale – CNRS UMR 6166, Université de la Méditerranée, France.

9. Fouda, P., J. Danloy, and T. L'Eglise, 2001, "A Heuristic to Generate a Precedence Graph Between Components for a Product Family," *Proceeding of the 4th IEEE International Symposium on Assembly and Task Planning*, Soft Research Park, Fukuoka, Japan.

10. Gagne, C., Gravel, M. and W.L. Price, 2006, "Solving real car sequencing problems with ant colony optimization," *European Journal of Operational Research*, 174(3), 1427-1448.

11. Gent, I.P., 1998 "Two Results on Car-sequencing Problems," Technical report of the APES Group, APES-02-1998.

12. Gottlieb, J., M. Puchta, and C. Solnon, 2003, "A Study of Greedy, Local Search, and Ant Colony Optimization Approaches for Car Sequencing Problems," S. Cagnoni et al. (Eds.): Evo Workshops, LNCS 2611, 246-257.

13. Gravel, M., C. Gagne, W.L. Price, 2005, "Review and comparison of three methods for the solution of car-sequencing problem," *Journal of Operational Research Society*, 56, 1287-1295.

14. Hyun, C., Y. Kim, and Y. K. Kim, 1998, "A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines," *Computers and Operations Research*, 25(7/8), 675-690.

16. Inman, R.R., and R.L. Bulfin, 1991, "Notes: Sequencing JIT Mixed-Model Assembly Lines," *Management Science*, 37(7), 901-904.

17. Kubiak, W., 1993, "Minimizing variation of production rates in just-in-time systems: a survey," *European Journal of Operational Research*, 66 (3), 259-271.

18. Kubiak, W. and S. Sethi, 1991, "A Note on "Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems", *Management Science*, 37(1), 121-122.

19. Lee, J.H.M., H.F. Leung, and H.W. Won, 1998, "Performance of a Comprehensive and Efficient Constraint Library using Local Search," *Proceedings of 11th Australian Joint Conference on Artificial Intelligence*, LNAI 1502, Springer, 191-202.

20. Miltenburg, J., 1989, "Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems," *Management Science*, 35(2), 192-207.

21. Miltenburg, J. and T. Goldstein, 1991, "Developing production schedules which balance part usage and smooth production loads in JIT production systems," *Naval*

Research Logistics, 38 (6), 893-910.

22. Miltenburg, J., G. Steiner, and S. Yeomans, 1990, “A Dynamic Programming Algorithm for Scheduling Mixed-Model, Just-In-Time Production Systems,” *Math. Computation Modeling*, 13(3), 57-66.

23. Miltenburg, J. and G. Sinnamon, 1989, “Scheduling mixed-model multi-level just-in-time production systems,” *International Journal of Production Research*, 27 (9), 1487-1509.

24. McMullen, P. and G. Frazier, 2000, “A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line,” *IIE Transactions*, 32 (8), 679-686.

25. McMullen, P., 2001a, “An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics,” *Computers and Industrial Engineering*, 41, 335-353.

26. McMullen, P., 2001b, “An ant colony optimization approach to address a JIT sequencing problem with multiple objectives,” *Artificial Intelligence in Engineering*, 15, 309-317.

27. McMullen, P.R., P. Tarasewich, and G.V. Frazier, 2000, “Using genetic algorithms to solve the multi-product JIT sequencing problem with setups,” *International Journal of Production Research*, 38 (12), 2653-2670.

28. Monden, Y., 1997, *Toyota Production System*, third edition, Institute of Industrial Engineers Press, Norcross, GA.

29. Muhl, E., P. Charpentier, and F. Chaxel, 2003, “Optimization of physical flows in an automotive manufacturing plant: some experiments and issues,” *Engineering Applications of Artificial Intelligence*, 16 (4), 293-305.

30. Prandtsetter, M. and G. Raidl, 2005, “A Variable neighborhood Search Approach for Solving the Car Sequencing Problem,” MEC-VNS: 18th Mini Euro Conference on VNS.

31. Puchta M. and J. Gottlieb, 2002, “Solving Car Sequencing Problems by Local Optimization,” *Applications of Evolutionary Computing*, LNCS 2279, Springer, 132-142.

32. Smith, K., Palaniswami, M, and M, Krisnamoorthy, 1996, “Traditional heuristics versus Hopfield neural network approaches to a car sequencing problem,” *European Journal of Operational Research*, 93, 300-316.

33. Stadzisz, P.C. and J.-M. Henrioud, 1998, “An integrated approach for the design of multi-product assembly systems,” *Computers in Industry*, 36, 21-29.

34. Steiner, G. and S. Yeomans, 1993, “Level schedules for mixed-model, just-in-time processes,” *Management Science*, 39 (6), 728-735.

35. Sumichrast, R.T., R.S. Russell, and B.W. Taylor, 1992, “A comparison analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system,” *International Journal of Production Research*, 30(1), 199-214.

36. Warwick, T. and E. Tsang, 1995, “Tackling car sequencing problems using a genetic algorithm,” *Evolutionary Computation*, 3(3), 267-298.

37. Yano, C. and R. Rachamadugu, 1991, “Sequencing to Minimize work overload in Assembly lines with product options,” *Management Science*, 37(5), 572-586.

38. Zhao, X. and K. Ohno, 1994, “A Sequencing Problem for a Mixed-model Assembly Line in a JIT Production System,” *Computers and Industrial Engineering*, 27, 71-74.

39. Zhu, J. and F. Ding, 2000, “A Transformed Two-Stage Method for Reducing the Part-Usage Variation and a Comparison of the Product-Level and Part-Level Solutions in Sequencing Mixed-Model Assembly Lines,” *European Journal of Operational Research*, 127(1), 203-216.

PART 2

JOINT CONSIDERATION OF ASSEMBLY-LINE PRODUCTION SMOOTHING AND FINISHED-GOODS DISTRIBUTION IN AN AUTOMOBILE PLANT

This paper is to be submitted.

Abstract

In a supply chain, production scheduling and finished goods distribution have been increasingly considered in an integrated manner. In an automobile assembly plant, typically various models and configurations of finished goods are scheduled on an assembly line on a regular basis for just-in-time production. From a production scheduling point of view, an assembly plant can consider production smoothing in models and product attributes to have even production loading; for distribution, the most important considerations can be even distribution of delivery dates to a destination, and effective grouping of finished goods for a destination in the schedule in order to ship products as they are produced. This paper presents a Heuristic solution procedure to achieve an integrated consideration of production scheduling and distribution with production smoothing for the assembly line and grouping of finished products for transportation. The Heuristic procedure first forms packages of units and then assigns packages to various days. The computational results based on randomly-generated problems are presented to show the advantage in cost savings for such an integrated consideration. Furthermore, a simulated annealing procedure is developed for the package assignment part of the integrated problem. The results for the Heuristic in package formation plus simulated annealing in package assignment achieve the best overall results based on further computational experiments.

1. Introduction

Mixed-model assembly is a common practice nowadays in many industries including the automobile industry. Extensive research has been conducted in sequencing mixed-model assembly lines (for example, Miltenburg, 1989; Yano and

Rachamadugu, 1991; Monden, 1998; and Drexel et al., 2006) in order to achieve a balanced workload, smooth usage of parts, and minimum line stoppage within the sequencing cycle. Even daily loading on an assembly line (Monden, 1998) is important in achieving even material usage and workloads in the production system from day to day. Production smoothing and capacity constraints may not be the only considerations for production scheduling and sequencing. To achieve the minimized overall cost, an integrated model may need to incorporate production and other related operations, such as finished goods distribution. For transportation of finished goods, smoothing models and product attributes is not nearly as important as considering the transportation efficiency while keeping the inventory low.

There is an increasing research emphasis on integration of production scheduling and distribution (Cohen and Lee, 1988; Chandra and Fisher, 1994; Hahm and Yano, 1995; Pundoor and Chen, 2004). Some of these studies address strategic or tactic levels of the decision making. Others focus on operational scheduling for batched production without considering product options. In this paper we will consider an integrated production-scheduling and product-distribution problem in a context of scheduling for a mixed-model assembly line while there are multiple models and product attributes in production smoothing considerations, and multiple destinations and multiple modes of transportation in distribution considerations.

Garcia et al. (2004) dealt with the problem of scheduling of orders and vehicle assignment for production and distribution planning. The products considered are perishable thus need to be delivered immediately with no waiting. An integer programming model was presented and solved for a special case; while in a general case, a heuristic procedure was developed. Pundoor and Chen (2005) considered a make-to-order production-distribution system with one supplier and one or more customers. The objective was to minimize the maximum delivery tardiness and distribution costs. A heuristic procedure was developed and the results showed the integrated approach was better than the sequential approaches in most cases. Stecke and Zhao (2007) considered a case with a make-to-order company which adopted a commit-to-delivery business mode. The integration of production and transportation costs was considered in their study. For the various cases they studied, mixed-integer programming models were developed and solved by heuristic approaches.

A closely related problem to our research topic is the bin-packing problem (BPP). The classic BPP (Johson et al., 1974) is to pack a certain number of different-sized items into a smallest number of bins. The problem was proved to be NP-hard (Garey and Johnson, 1979). This is an intensively studied problem in the field of approximation algorithms (Coffman et al., 1997). The items considered in BPP can be one or multiple dimensional in weight or size. In most research of BPP, the objectives were focused on the number of bins used or a cost function due to different item-grouping procedures. Rao and Iyengar (1994) presented a target workload assignment problem assuming a fixed number of bins each with an unlimited capacity. The objective was to minimize the squared deviation from the workload target, thus to distribute the workload as evenly as possible. They solved this modified bin-packing problem (MBPP) by simulated annealing with a local search. One-dimensional weight was assumed. Brusco et al. (1997) dealt with the similar problem and improved the heuristic by limiting the interchanges to between items of only similarly-sized ones. They compared their results with those of Rao and Iyengar and showed improvement on the solution quality.

The research in this paper was motivated by an automobile manufacturer in the U.S. producing multiple models and configurations of automobiles on a daily basis and delivering finished goods to a large network of locations by rail or trucks. Jointly considering production scheduling for smoothing for the assembly lines while grouping for distribution considerations can result in a significant cost saving by shipping automobiles as they are produced, improved customer service by spreading out the multiple delivery dates for a destination, and a smooth production schedule for even material usage and workload requirements in the production system.

2. Problem Description

For the scope of this paper, an automobile assembly plant is considered. As a general consideration, it is assumed that the automobile assembly plant produces 2 types of automobiles in terms of *high* and *low profiles*. This assumption affects the distribution of finished goods especially by rail because of its impact on the shipment capacity and grouping. It is further assumed that several models of vehicles, for example, full-size sedan or sport-utility-vehicle, are produced on one or more assembly lines. Each model has many possible configurations due to customization in option selections (attribute values), for example, two- or four-wheel drive, with or without a sunroof. Without loss of generality, the daily production volume on an assembly line is assumed to be a constant. Each vehicle is of a certain *model*, has a specific set of *attribute values* representing its configuration, and is to be delivered to a specified *destination* with a predetermined transportation mode, that is, by *rail* or *truck*. In this paper due dates are not differentiated or treated as a vehicle-order characteristic within the same pool of vehicle orders. This is because we consider the production system that at the monthly scheduling level, most orders are dealer orders, and the manufacturer retains flexibility in assigning orders to various days to achieve better production smoothing.

The problem considered here is to schedule a monthly order pool into daily production considering production smoothing and the distribution efficiency. It is assumed that production smoothing is necessary in order to have even usage of critical parts for supplier considerations and even workload for critical production resources for manufacturing considerations. To achieve production smoothing, specifically a roughly equal number of units of each *model* and a roughly equal number of units of certain product *attribute values* are desired everyday. These considered product attribute values are generally associated with critical production resources and critical material supplies.

Transportation considerations for the two modes of transportation, *rail* and *truck*, are primarily to have the least "*dwell*" times (that is, flow time) for finished goods prior to shipment while assuming that transportation operates with consolidation to achieve efficiency. For rail shipment, the total daily shipping capacity is roughly fixed. It is assumed that only vehicles of the same profile and destination can be loaded into a railcar for efficiency consideration, and that the capacity of a railcar for high-profile vehicles is less than that of low-profile vehicles. A rail destination is also called *railhead*. It is assumed that a desirable *frequency* can be specified for each rail destination. The total number of railcars shipped everyday needs to be roughly equal.

A fraction of truck-shipment destinations are assumed to need consolidation into shipments of specified frequencies. Other truck-shipment destinations are assumed to be able to group with regional truck-shipment destinations in a continuous fashion without having consolidation in scheduling. Furthermore, it is assumed that there are multiple *carriers* to provide truck shipments. Thus, in order to minimize the dwell times and satisfy the even spread of delivery dates for each rail destination and some truck destinations, vehicles of the same destination are to be consolidated on various dates in the production schedule. It is also desirable that the daily product quantities for carriers are smoothed along with other smoothing considerations.

2.1. A Nonlinear Mixed-Integer-Programming Formulation

2.1.1. The Objective Function

In this paper, the weighted sum of five sets of the mean absolute deviation (MAD) values is considered in the objective function. These deviations are MADs in cumulative model quantities, cumulative total numbers of units for attribute values, the delivery intervals for rail shipment and selected truck-shipment destinations, the cumulative daily numbers of railcars, and the cumulative daily quantities for various carriers, from the respective ideal quantities. The overall objective function can be expressed briefly as follows: w_1*MAD[models] + w_2*MAD[attributes] + w_3*MAD[delivery intervals] + w_4*MAD[daily rail cars] + w_5*MAD[carriers]

2.1.2. A Mathematical Program

In the formulation, it is assumed that four *types* of vehicles are considered; type 1: high profile rail shipment, type 2: low profile rail shipment, type 3: truck shipment with consolidation, type 4: truck shipment without consolidation. Without loss of generality, it is further assumed that there are an even number of high- or low-profile vehicles for rail shipment at each railhead to fit into full railcars for each railhead.

Decision Variables:

x_{ijk} : 1 if vehicle i of type k is assigned to day j; 0 otherwise

y_{rjk} : Number of orders of type k for destination r on day j

d_{ijk} : Delivery date of shipment j for destination i for rail shipment (k=1) or truck shipment (k=2)

z_{rj1} : 1 if some rail-shipment orders are produced for destination r on day j, 0 otherwise

z_{rj2} : 1 if truck-shipment orders of type 3 (with consolidation) produced for destination r on day j, 0 otherwise

u_{rj1} : number of railcars used on day j, to destination r for high profile vehicles

u_{rj2} : number of railcars used on day j, to destination r for low profile vehicles

v_{rsjk} : 1 if the s-th shipment of destination r on day j for rail (k=1) or truck shipment (k=2), 0 otherwise

Parameters:

w_1, w_2, w_3, w_4, w_5 : Penalty weights for smoothing models, attributes, delivery intervals, number of railcars, and carrier quantities

f_{ri} : No. of shipments for destination r of rail (i=1) or truck (i=2) shipments with consolidation

N_k : Total number of units of type k orders, k = 1, .., 4

e_k : Standard railcar capacity for type k cars, k = 1, 2

n : Total no. of units to be scheduled

H : No. of carriers

D : Total no. of production days

m : No. of models

G : Total no. of attributes among all models

Φ_p : Ideal rate of a model p to be scheduled daily

t_{irk} : 1 if order i of type k is for destination r, 0 otherwise

c_{ikh} : 1 if order i of type k is by carrier h, 0 otherwise

a_{ip} : 1 if order i is a unit of model p, 0 otherwise

b_{ip} : 1 if order i has smoothing requirement p, else 0

$A_j, A_j^{'}$: Minimum and maximum numbers of rail-shipment vehicles on day j

$B_{r1}, B_{r1}^{'}$: The daily minimum and maximum numbers of railcars can be shipped to railhead r

$B_{r2}, B_{r2}^{'}$: The daily minimum and maximum numbers of vehicles can be shipped to truck-shipment destination r

$C_j, C_j^{'}$: Minimum and maximum volumes, respectively, for total production on day j

$E_h, E_h^{'}$: The lower and upper limits of carrier h's daily quantity

R_1, R_2 : Numbers of railheads and truck shipment destinations with consolidation, respectively

Formulation:

$$Minimize \ \frac{w_1}{mD}\sum_{g=1}^{D}\sum_{p=1}^{m}\left|\sum_{i=1}^{N_k}\sum_{j=1}^{g}\sum_{k=1}^{4}x_{ijk}a_{ip} - g\Phi_p\right| + \frac{w_2}{GD}\sum_{g=1}^{D}\sum_{p=1}^{G}\left|\sum_{i=1}^{N_k}\sum_{j=1}^{g}\sum_{k=1}^{4}x_{ijk}b_{ip} - \frac{g}{D}\sum_{k=1}^{4}\sum_{i=1}^{N_k}b_{ip}\right|$$

$$+ \frac{w_3}{(R_1+R_2)}\sum_{i=1}^{2}\sum_{r=1}^{R_i}\frac{1}{(f_{ri}-1)}\sum_{j=2}^{f_{ri}}\left|d_{rji} - d_{r,j-1,i} - \frac{D}{f_{ri}}\right| + \frac{w_4}{\mathrm{D}}\sum_{g=1}^{D}\left|\sum_{j=1}^{g}\sum_{k=1}^{2}u_{rjk} - \frac{g}{D}\sum_{j=1}^{D}\sum_{k=1}^{2}u_{rjk}\right|$$

$$+ \frac{w_5}{HD}\sum_{g=1}^{D}\sum_{h=1}^{H}\left|\sum_{i=1}^{N_k}\sum_{j=1}^{g}\sum_{k=3}^{4}x_{ijk}c_{ik} - \frac{g}{D}\sum_{k=3}^{4}\sum_{i=1}^{N_k}c_{ikh}\right|$$

Subject to:

$C_j \le \sum_{k=1}^{4}\sum_{i=1}^{N_k}x_{ijk} \le C_j^{'}$	j = 1, 2 .., D	(Daily total shipment quantity constraints)
$\sum_{j=1}^{D}x_{ijk} = 1$	i = 1, 2 .., N_k, k = 1, .., 4	(Unique-day assignment)
$\sum_{i=1}^{N_k}x_{ijk}t_{irk} = y_{rjk}$	j = 1, 2 .., D, k = 1, 2, 3, ∀ r	(Total quantity for destination r on a day j)

$$\left.\begin{array}{ll}
\sum_{k=1}^{2} u_{rjk} \geq z_{rj1} B_{r1} & j = 1, 2 .., D,\ r = 1, 2 .., R_1 \\
\sum_{k=1}^{2} u_{rjk} (1 - z_{rj1}) = 0 & j = 1, 2 .., D,\ r = 1, 2 .., R_1 \\
y_{rj3} \geq z_{rj2} B_{r2} & j = 1, 2 .., D,\ r = 1, 2 .., R_2 \\
y_{rj3} (1 - z_{rj2}) = 0 & j = 1, 2 .., D,\ r = 1, 2 .., R_2
\end{array}\right\} \text{(Either the minimum quantity or 0 to a rail or truck destination r)}$$

$$\left.\begin{array}{ll}
y_{rj1} - u_{rj1} e_1 = 0 & j = 1, 2 .., D,\ r = 1, 2 .., R_1 \\
y_{rj2} - u_{rj2} e_2 = 0 & j = 1, 2 .., D,\ r = 1, 2 .., R_1 \\
B_{r1} \leq \sum_{k=1}^{2} u_{rjk} \leq B'_{r1} & j = 1, 2 .., D,\ r = 1, 2 .., R_1
\end{array}\right\} \text{(Total number of railcars)}$$

$$\left.\begin{array}{ll}
\sum_{k=1}^{2} y_{rjk} \leq M z_{rj1} & j = 1, 2 .., D,\ r = 1, 2 .., R_1 \\
y_{rj3} \leq M z_{rj2} & j = 1, 2 .., D,\ r = 1, 2 .., R_2
\end{array}\right\} \text{(z value is set to 1 if there is a shipment)}$$

$$E_h \leq \sum_{k=3}^{4} \sum_{i=1}^{N_k} x_{ijk} c_{ikh} \leq E'_h \qquad j = 1, 2 .., D,\ h = 1, 2 .., H \qquad \text{(Carrier daily capacity)}$$

$$\left.\begin{array}{ll}
\sum_{j=1}^{D} (j z_{rj1}) v_{rsj_1} = d_{rs1} & r = 1, 2 .., R_1,\ s = 1 .., f_{r1} \\
\sum_{j=1}^{D} v_{rsj_1} = 1 & r = 1, 2 .., R_1,\ s = 1 .., f_{r1} \\
1 \leq d_{r,s-1,1} \leq d_{rs1} & r = 1, 2 .., R_1,\ s = 2 .., f_{r1}
\end{array}\right\} \text{(Constraints to set delivery dates of each rail destination)}$$

$$\left.\begin{array}{ll}
\sum_{j=1}^{D} (j z_{rj2}) v_{rsj2} = d_{rs2} & r = 1, 2 \ldots, R_2,\ s = 1, .., f_{r2} \\
\sum_{j=1}^{D} v_{rsj2} = 1 & r = 1, 2 \ldots, R_2,\ s = 2, .., f_{r2} \\
1 \leq d_{r,s-1,2} \leq d_{rs2} & r = 1, 2 \ldots, R_2,\ s = 2, .., f_{r2}
\end{array}\right\} \text{(Constraints to set delivery dates of each truck destination with shipment consolidation)}$$

$x_{ijk} = \{0, 1\}$, $\forall$i,j,k; y_{rjk} integer, $\forall$r,j,k; $z_{rj1}, z_{rj2} = \{0, 1\}$, u_{rj1}, u_{rj2} integer, $\forall$r,j; v_{rsjk} = {0, 1}, $\forall$r,s,j,k;

d_{ijk} integer, $\forall$i,j,k.

A sub-problem (that is, determining the placement to days for the delivery batches of various railheads after the sizes and contents of various batches are determined in order to achieve near-100% of total daily rail assignments) of the above formulation is a multi-dimensional MBPP while considering closeness of due dates. Since the solution space of MBPP is exponential in the size of the item list (Rao and Ivyengar, 1994), a Heuristic procedure will be developed to find a good solution.

3. A Heuristic Procedure for Integrated Scheduling and Distribution Considerations

Scheduling a monthly pool of vehicles to days in the month to jointly address the production smoothing and transportation efficiency is considered here. A proposed Heuristic approach applies mixed-model sequencing in place of scheduling to allow one unit to be scheduled at a time. Also a "packaging" step for the rail shipment is performed first before scheduling packages to days to adequately consider the delivery frequency for each rail destination. Units are first "pre-sequenced" into the packages of each rail destination to achieve a smooth mix. When packages are placed into days, a procedure similar to earliest-due-date (EDD) sequencing based on pre-assigned due dates (Inman and Bulfin, 1991) is applied to attempt to spread the rail shipment packages evenly within the month. Furthermore, to alleviate the complexity of the bin-packing problem nature, all packages are classified by size and placed in the order of large, medium, and small packages by a ratio control, where the daily average numbers of large and medium packages, respectively, are considered as the control targets in daily placement of packages. The procedure with packages for placement is repeated for a fraction of truck-shipment units needing consolidation. Then the remaining truck-shipment units *without* consolidation are placed into each day unit-by-unit to achieve improved production smoothing.

3.1. The Heuristic Procedure

The Heuristic procedure has two parts for rail and truck shipments. For truck shipment, Phase I groups and assigns packages to days while considering production smoothing. Phase II assigns units that do not need consolidation one by one into days. For rail shipment, only Phase I is needed.

Part I. Assigning rail-shipment vehicles

Phase 1

Step 1. (Forming packages)

Given the shipping frequency (f_{r1}) for each railhead r, calculate the numbers of railcars (q_r^H and q_r^L) for high and low profile vehicles, respectively, based on its monthly quantities and standard railcar capacities. Residual quantities are also determined. Let D be the number of days in a month. For each railhead, form the f_{r1} *packages* of railcars by sequentially assign one by one q_r^H and then q_r^L railcars to f_{r1} days. "*Pre-sequence*" vehicles of each railhead into packages to smooth product attributes by minimizing the one-stage deviation for attributes. Initially set the *due date* of each package to ∞.

Step 2. (Designating package sizes and setting initial due dates of the first packages)

Set the average package size of a railhead to large, medium, or small. For large-size packages, this is done by selecting all packages of a railhead at a time in a decreasing order of the average package sizes of railheads until the total number of large-size packages is about to exceed a third (a user specified parameter) of the total number of packages. This process is repeated for the selection of medium-size packages. The remaining roughly one-third packages are then set to small-size. Pre-assign a delivery date $\lceil 0.5*D/f_{r1} \rceil$ for the *first package* of each railhead r.

Repeat Steps 3-6 until assignment is complete for rail shipment for day 1 to day D:

Step 3. (Specifying a package size in daily assignment)

In day j, specify large *size*, or else medium size, or else small size until the cumulative number of packages of the size reaches $\lfloor$j*(average daily ratio of the number of packages of the size)$\rfloor$.

In day j, determine ψ_k, the "current target of number of railcars" at the selection of k-th package in the day j through $\psi_k = \psi_{k-1}$ + [the average number of packages of the desirable size], where the "average number of packages of the desirable size" is set in the order of large, medium, and small until the size reaches $\lfloor$j*(average daily ratio of the number of packages of the size)$\rfloor$. Whenever a ψ_k reaches 100%, the ψ_k value remains unchanged thereafter.

Step 4. (Assigning a package of a selected size in daily assignment)

Choose a package from the packages of the specified size with a due date based on the smallest value of [w_3(absolute deviation in deviating from due date) + $(\frac{\phi}{mD})w_1$(absolute deviation in deviating from the ideal level in models) + $(\frac{\phi}{GD})w_2$(absolute deviation in deviating from the ideal level in attributes) + $(\frac{\phi}{P})w_4$(absolute deviation in deviating from the ideal of number of railcars at the current stage], where ϕ is the fraction of railcars that have been scheduled into the day over the average daily number of railcars, and P is the total number of packages.

Step 5. (Resetting the due dates for packages)

Each time a package of a railhead is assigned to a day, update the due date of the next available package to $\lceil$current date + interval $D'/f_{r1}'$$\rceil$, where D′ is the remaining number of days and f_{r1}' is the remaining frequency.

Step 6. (Checking closeness to completing assignment for a day)

If 100% of the total daily rail shipment target is not reached, go to Step 3. If a package exceeds 100% of the total daily rail shipment target, this package is either kept in the day or delayed depending on which case is closest to 100% of average daily number of railcars. After all packages are assigned to all days, go to Step 7. Otherwise, let j=j+1, go to Step 3. Residuals are assigned to the last (or possibly first) shipment of the railhead.

Step 7. (Performing post-sequencing) – An optional step

Sequencing all vehicles according to the assigned packages of determined dates and package sizes, reassign all rail-shipment vehicles with the objective function: $\frac{\phi}{mD}$ [w_2(absolute deviation in model volumes) + $\frac{\phi}{GD}$ w_3(absolute deviation in attributes)]

Part II. Assigning truck-shipment vehicles

Phase 1. The above Phase 1 procedure for rail shipment is repeated (with a railhead changed to a truck-shipment region) for vehicles that needs consolidation in shipment.

Phase 2. For vehicles without consolidation, sequencing vehicles unit by unit until reaching the daily overall quantity target with objective: $\frac{\phi}{mD}$ [w_2(absolute deviation in model volumes)+ $\frac{\phi}{GD}$ w_3(absolute deviation in attributes)+ $\frac{\phi}{HD}$ w_4(absolute deviation in carrier quantities)]

3.2. Simulated Annealing for enhancing the Heuristic procedure

A further attempt to improve the Heuristic procedure is to apply a simulated annealing (SA) procedure in place of steps 3-6 of the Heuristic procedure. After packages are formed in Step 1 of Phase I of the Heuristic procedure, these packages are first randomly assigned to various days from 1 to 24 by the SA procedure. In each iteration of the SA procedure two solutions are generated and compared in the neighborhood generation step: the *first* solution is obtained by randomly picking any 2 packages and swapping them; the *second* is obtained by assigning the first-selected package to a randomly selected different day (not losing generality, this is equivalent to instead assigning the second package). These two solutions are then compared and the better one in the objective value is selected as the *candidate solution*. This SA neighborhood generation approach bears some similarity to that by Brusco et al. (1997) to keep a balanced assignment among various "bins" while attempting to have an efficient neighborhood generation for efficiently exploring the solution space.

From the SA neighborhood generation step, if the *candidate solution* is better than the *current solution*, it becomes the new current solution. However, if the new solution is worse, then it can become the new current solution based on a probability term P(A) $= e^{-\Delta z / K_b T}$, where K_b is the Boltzman constant and T is the current temperature; that is, a solution with $\Delta z > 0$ will be accepted with a probability P(A).

4. Computational Experiments

The above stated procedures are tested on randomly generated problems. In each randomly generated problem, 36,000 orders of 5 models for production in a month of 24 days are randomly generated each with 10 attributes (each represented by a value of 1 or 0). Models 1 and 2 are low-profile vehicles together constituting 60% of the total volume, and models 3, 4, 5 are high-profile vehicles together constituting 40% of the total volume. Rail-shipment constitutes 65% of total units, and truck-shipment constitutes 35%. For a rail-shipment vehicle order, a railhead number 1-20 is randomly assigned. All rail shipments require consolidation. The total number of railcars of each railhead is calculated based on the standard railcar capacities (15 low-profile vehicles and 10 high-profile vehicles per railcar, respectively), and its numbers of high- and low-profile vehicles. For truck-shipment vehicles, a total of 15 regions are assumed to be shipped by 5 truck carriers (i.e., trucking companies). Each carrier has a number of regions ranging from 1 to 4. In general, 10% of truck-shipment vehicles of each region need consolidation for shipment efficiency, while other 90% truck shipment vehicles are assumed not requiring consolidation and can be routed with vehicles in the same region.

Weights of 3:3:3:3:1 are assigned to the mean absolute deviations from ideal levels for shipment intervals, daily number of railcars, daily smoothing for carriers, daily smoothing for attributes, and daily smoothing for models. The various weights are used in order to place different emphases on different scheduling objectives. For the simulated annealing procedure, the initial temperature is set at 25; and the ending temperature and cooling rate are set at 0.01 and 0.95, respectively. All the tests are conducted on a Pentium M 3.4 GHz PC.

4.1. Experiment 1 [10 randomly generated problems to test the Heuristic procedure with or without SA procedure for rail-package assignment after Phase I, Data Set 1]

The Heuristic procedure was tested using a data set generated with the procedure described above. The computational results of the 10 runs by the Heuristic procedure without the SA procedure are given in Table 1 with the MADs of the various objectives. The SA procedure in conjunction with the Heuristic procedure took a longer CPU time than the Heuristic procedure alone but obtained improved solutions with lower MADs as shown in Table 2. It is noticed that the models and attributes still have high MAD values as there are uneven loads from day to day due to uneven package assignment to various days; and this is improved by the SA procedure.

Table 1. Results from the Heuristic without SA procedure – Rail only

		1	2	3	4	5	6	7	8	9	10	Mean
Rail	Model	4.77	5.16	4.33	4.86	4.88	4.53	4.56	4.75	5.53	4.86	4.82
	Attribute	3.48	3.72	3.25	3.49	3.49	3.66	3.35	3.54	3.73	3.40	3.51
	Interval	1.42	1.66	1.53	1.62	1.40	1.43	1.63	1.43	1.40	1.54	1.51
	Railcar	0.69	0.56	0.72	0.67	0.97	0.74	0.83	0.63	0.93	0.67	0.74
	Overall	21.52	22.98	20.82	22.19	22.44	22.00	22.01	21.52	23.70	21.67	22.09
	Heuristic CPU time (sec)	73.30	75.36	61.48	66.27	88.55	73.13	69.33	82.03	88.83	82.25	76.05

Table 2. Results from the Heuristic with SA procedure – Rail only

		1	2	3	4	5	6	7	8	9	10	Mean
Rail	Model	2.87	2.75	2.97	2.93	2.63	2.41	2.94	2.31	3.12	2.89	2.78
	Attribute	2.92	2.92	2.76	2.80	2.68	2.74	2.76	2.93	2.80	2.80	2.81
	Interval	1.28	1.33	1.30	1.31	1.37	1.19	1.35	1.16	1.35	0.85	1.25
	Railcar	0.50	0.08	0.49	0.47	0.49	0.53	0.54	0.50	0.49	0.22	0.43
	Overall	16.97	15.75	16.60	16.65	16.25	15.81	16.87	16.07	17.02	14.49	16.25
	SA CPU time (sec)	94.03	92.45	92.20	91.92	92.23	92.25	91.48	93.20	91.63	91.83	92.32

Table 3. Total objective values (Rail plus truck shipments) from Heuristic

	1	2	3	4	5	6	7	8	9	10	Mean
Model	0.48	0.39	0.43	0.35	0.34	0.37	0.37	0.37	0.38	0.41	0.39
Attribute	0.43	0.45	0.42	0.44	0.42	0.40	0.44	0.44	0.42	0.45	0.43
Shipment interval	1.42	1.66	1.53	1.62	1.40	1.43	1.63	1.43	1.40	1.54	1.51
No. of railcars	0.69	0.56	0.72	0.67	0.97	0.74	0.83	0.63	0.93	0.67	0.74
Numbers of carriers	1.67	1.87	1.78	1.80	2.23	1.88	2.05	1.58	2.28	1.71	1.88
Overall	11.02	12.36	11.62	11.94	12.50	11.49	12.73	10.71	12.68	11.48	11.85
CPU time (sec)	1,313	1,183	1,671	1,527	1,162	1,219	1,523	1,078	1,098	992	1,277

4.2. Experiment 2 [10 randomly generated problems with RAIL & TRUCK shipments, Data Set 1]

Experiment 2 is further conducted to compare the overall problem including also assigning truck-shipment vehicles. In this experiment, the truck-shipment vehicles are actually assumed to have no need to consolidate for destinations; and they can be assigned directly according to the Phase II of the Heuristic procedure. The total objectives for rail and truck shipments were evaluated. Ten problems were randomly generated using the same problem-generation scheme as stated in Experiment 1. The results from only the Heuristic procedure are shown in Table 3.

From Table 3 it can be seen that truck assignment significantly reduces the MADs for models and attributes as truck-shipment units are assigned unit by unit, that is, it is achieved by effectively using a sequencing step in place of the scheduling requirement. However, even though the weight for smoothing the numbers of carriers is set to 3 in the objective function, the MADs for the numbers of carriers are moderately larger than the other objectives. This is because truck-shipment assignment is performed after the rail-shipment assignment with uneven remaining scheduling capacities.

Table 4. Total objective value (Rail plus truck shipment) from Heuristic procedure with SA

	1	2	3	4	5	6	7	8	9	10	Mean
Model	0.47	0.41	0.48	0.34	0.28	0.37	0.40	0.37	0.39	0.38	0.39
Attribute	0.43	0.43	0.44	0.43	0.41	0.38	0.43	0.41	0.42	0.43	0.42
Shipment interval	1.28	1.33	1.30	1.31	1.43	1.24	1.60	1.25	1.29	1.45	1.35
No. of railcars	0.50	0.08	0.49	0.47	0.49	0.53	0.54	0.50	0.49	0.22	0.43
Numbers of carriers	1.42	0.63	1.43	1.15	1.37	1.19	1.35	1.16	1.35	0.85	1.19
Overall	11.38	7.84	11.46	10.39	11.35	10.40	12.17	10.32	11.04	9.21	10.56
CPU time (sec)	1,389	1,237	1,638	1,522	1,175	1,238	1,562	1,136	1,113	1,042	1,305

For comparison, the results from the combined procedure with the Heuristic procedure and SA (improvement in package assignment) are summarized in Table 4. From Table 4, it can be seen that the overall results for rail and truck shipments as solved by using the Heuristic procedure and SA modestly improved the results by the Heuristic procedure only with a small CPU time increase. In comparing the overall results in Tables 3 and 4, the better approach to solve the overall problem is to use the Heuristic procedure for forming packages, apply the modified SA in the rail package assignment step, and assign truck shipment units unit by unit as stated in the Heuristic procedure.

4.3. Experiment 3 [10 generated problems to test the Heuristic procedure with SA considering consolidation, with RAIL plus TRUCK shipments, Data Set 2 with simulated industrial data]

To simulate an industrial case, the procedure was further tested with Data Set 2. The problem generation method is the same as described earlier, except that the truck shipment has 10% of vehicles needing consolidation, and the number of packages is higher than that in Data Set 1. The average percentage of the number of small packages in Data Set 2 is 48% as compared to 31% in Data Set 1. In Data Set 2, packages of the increased percentage in small packages are shipped with a "priority" which requires the shipping dates to fall into a certain range of the production periods, such as the first 3 weeks.

4.3.1 Results from Heuristic procedure with SA

The results from the Heuristic procedure with SA for Data Set 2 are shown as in Table 5. From Table 5, it can be seen that the solution is generally consistent with the results given in Table 4. The MADs in the number of railcars are noticeably higher than those in Table 2 due to the nature that the "priority" packages could not be swapped freely. This also leads to the larger MADs in shipment intervals and the numbers of carriers.

4.3.2 Inventory-cost comparison for scheduling solution with and without consolidation

Further comparison is conducted for rail shipment with a procedure that does *not* consider shipment consolidation (unconsolidated case) by using a one-stage sequencing method to achieve smoothing for models and attributes. The railheads are divided into two groups based on the total volume; for a high-volume railhead, the shipment is assumed to take place on a daily basis; for a low-volume railhead, only when the quantity reaches a minimum number of railcars (similar to the ones used in Experiment 4.3.1 based on the frequency), it can then be shipped in batch. Residuals

Table 5. Total objective values (Rail plus truck) from Heuristic procedure with SA

Run	1	2	3	4	5	6	7	8	9	10	Mean
Models	0.35	0.35	0.52	0.44	0.34	0.43	0.40	0.34	0.39	0.31	0.39
Attributes	0.44	0.43	0.59	0.46	0.43	0.41	0.44	0.45	0.42	0.52	0.46
Shipment interval	1.58	1.63	1.42	1.30	1.26	1.44	1.90	1.49	1.25	1.56	1.48
No. of railcars	1.51	1.22	1.63	1.81	2.10	2.03	2.18	2.02	2.77	2.39	1.97
Numbers of Carriers	1.57	2.45	1.83	4.43	2.93	2.15	1.26	1.91	2.94	2.50	2.40
Overall objective	15.68	17.54	16.93	24.43	20.51	18.51	17.74	17.95	22.54	21.19	19.30
CPU time (Sec)	448	553	407	531	540	507	541	443	555	392	492

not fitting the whole number of railcars are held until the next shipment and so forth. While both rail and truck shipments are considered, the inventory-cost comparison is based on the rail shipment only. The Heuristic procedure with SA is applied for the case with consolidation.

In the consolidated case it assumes a batch formed and assigned to a certain day is shipped on the same day. The residuals for the consolidated case are scheduled to be produced with the last batch in the month and shipped with the first batch in the following month. The comparison results are given in Table 6. It can be seen that using the scheduling approach without consolidation, which is similar to some existing industrial practices, the inventory and vehicle dwell time are both noticeably higher.

From Table 6 it can also be seen that the overall smoothing (rail plus truck) results are only modestly improved without consolidation. This comparison shows that the overall smoothing results in models and attributes from the Heuristic approach is relatively effective in smoothing for various models and product attributes while taking shipment consolidation into account. Furthermore, a comparison of the average inventory holdings (only rail is considered) between the two cases showed that with the Heuristic procedure that considers consolidation, the average inventory cost saving per month (averaged among 10 runs) due to reduced finished-goods inventory is more than $160,000 based on a carrying cost of $20 per day per unit. This shows that there can be a significant cost advantage in jointly considering the finished-goods distribution and production scheduling.

Table 6. Inventory cost comparison with or without considering shipment consolidations

	Run	1	2	3	4	5	6	7	8	9	10	Mean	Inventory cost (Rail)
No consolidation	Model (Rail + Truck)	0.40	0.34	0.50	0.22	0.41	0.40	0.46	0.40	0.44	0.36	0.39	
	Attribute (Rail + Truck)	0.43	0.42	0.37	0.38	0.38	0.40	0.41	0.40	0.39	0.41	0.40	
	No. of shipping days (Rail)	24	24	24	24	24	24	24	24	24	24	24	
	Daily inventory (Rail)	386	381	357	391	394	381	410	379	397	386	386	
	Total inventory.day (Rail)	9272	9134	8572	9393	9445	9139	9836	9107	9536	9262	9270	$185,392
with consolidation	Model (Rail + Truck)	0.35	0.35	0.52	0.44	0.34	0.43	0.4	0.34	0.39	0.31	0.39	
	Attribute (Rail + Truck)	0.44	0.43	0.59	0.46	0.43	0.41	0.44	0.45	0.42	0.52	0.46	
	Average shipping interval (Rail)	3.85	4.01	3.52	4.29	4.49	3.83	4.83	4.04	5.09	3.96	4.19	
	Daily inventory in a cycle (Rail)	204	189	192	171	219	162	175	185	210	210	192	
	Total inventory.day (Rail)	785	758	676	735	982	620	845	747	1067	832	803	$16,063
Average cost savings	-	-											$169,392

4.3.3 An illustrative example

An illustrative example for rail-shipment scheduling taking rail-shipment consolidation into account is given in Table 7. It can be seen that the Heuristic procedure in conjunction with SA consolidates orders into a given number of shipments and relatively evenly spreads consolidated orders into days of the month. The daily total quantities are also kept relatively consistent. Residuals are combined with the last shipment for each railhead in this table.

5. Problem Variants

5.1 Special transportation requirements

Due to the requirement of the rail transportation system, a shipping practice can limit shipping to certain regions on certain days of the week. For example, orders to certain regions can only be shipped out on Mondays, Wednesdays, and Fridays; orders of certain other regions can only be shipped on Tuesdays, Thursdays, and Saturdays; and orders of some high-volume regions may be shipped everyday.

Table 7. An illustrative example for a production schedule that considers rail-shipment consolidation

		Days																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Railhead	1	330	330	0	330	0	335	0	0	330	0	335	0	0	0	0	320	0	0	0	0	0	333	0	0
	2	0	0	0	0	395	0	0	395	0	0	0	400	0	0	0	0	400	0	0	395	0	0	425	0
	3	0	0	0	370	0	0	370	0	0	0	370	0	0	385	0	0	0	380	0	0	385	0	0	389
	4	0	0	295	0	0	0	295	0	0	295	0	0	295	0	295	0	0	0	280	0	0	280	0	284
	5	0	330	0	0	0	315	0	0	320	0	0	315	0	320	0	320	0	315	0	317	0	0	0	0
	6	0	0	105	0	105	0	105	200	0	100	100	100	100	0	100	0	112	0	0	0	0	0	0	0
	7	80	80	75	0	75	150	0	150	0	150	0	0	75	75	89	0	0	0	0	0	0	0	0	0
	8	75	0	75	0	0	0	65	0	0	65	0	0	65	0	0	65	65	0	130	0	60	60	120	67
	9	0	0	0	0	115	0	0	0	0	115	0	0	115	0	115	0	115	0	0	0	130	0	264	0
	10	0	0	0	0	115	0	0	0	115	0	0	0	115	0	115	115	0	115	110	110	120	0	0	0
	11	75	75	75	75	0	0	0	65	0	65	65	0	65	0	65	0	65	0	130	0	130	130	74	0
	12	50	0	0	55	0	0	0	0	0	0	0	55	0	0	0	0	0	0	68	0	0	0	0	0
	13	0	0	50	0	0	0	0	0	50	0	0	0	0	0	0	0	77	0	0	0	0	0	0	0
	14	0	0	50	0	0	0	0	0	40	0	0	0	0	0	0	40	0	0	0	0	0	46	0	0
	15	50	0	55	0	0	0	0	0	0	50	0	0	0	0	0	0	73	0	0	0	0	0	0	0
	16	0	40	0	0	0	40	0	0	0	0	0	0	0	40	0	0	0	0	0	47	0	0	0	0
	17	50	0	40	0	0	0	0	40	0	0	0	0	0	0	40	0	0	0	35	0	0	35	0	49
	18	55	0	0	0	50	0	0	0	0	0	0	0	0	0	50	0	0	0	72	0	0	0	0	0
	19	40	0	40	0	0	0	0	0	0	0	0	0	40	0	0	0	0	40	0	0	35	0	0	39
	20	50	0	0	40	0	0	0	0	0	0	0	0	0	50	0	0	0	0	61	0	0	0	0	0
	Total	855	855	860	870	855	840	835	850	855	840	870	870	870	870	869	860	907	850	886	869	860	884	883	828

For a region of which orders can only be shipped on certain days, any orders produced not on these days result in additional holding costs for needing to delay to the actual shipping days. To deal with this problem variant, a penalty term can be added to the objective function for this additional holding cost. At each step of assigning a rail package, the objective including model, attribute, carrier, interval, number of railcars, plus a function of duration till the next shipping date is evaluated in the Heuristic procedure and in the modified SA procedure. The best package (with the smallest objective value) should be picked according to the modified objective function in the Heuristic or the modified SA procedure.

5.2 Dealing with weekly orders

It is possible that the problem treats vehicle orders at the weekly level in case that the company only has sufficient order data for a week. In this case, the ideal shipping interval of a railhead is calculated as 6 days (in a week) divided by its shipping frequency. The due date for the first package of any railhead may be determined based on the last package shipped for the same railhead in a prior week. If a calculated due date is not within current week, it can then be set as either Monday or Friday of the current week depending on it is too early or too late in relation to the considered week. With this modification of due dates, the Heuristic procedure and SA procedure can then be applied to orders in a week.

6. Conclusions

This paper presents a Heuristic procedure that addresses the integrated production scheduling problem considering production smoothing and distribution efficiency. The problem is of multi-objectives and multi-dimensions and usually of large size making it difficult to obtain an optimal solution. A Heuristic procedure was developed based on sequencing mixed-model assembly lines to achieve a desirable scheduling result while considering package assignment for shipment consolidation. A simulated annealing procedure was developed and applied to the package assignment part of the Heuristic procedure.

The computational results showed that the Heuristic procedure achieves significant cost savings from addressing scheduling with shipment consolidation for rail shipments. The partial solution for rail package assignment obtained by using the simulated annealing procedure showed noticeable improvement over the Heuristic procedure. The Heuristic procedure in conjunction with the simulated annealing procedure showed promise in effectively obtaining good solutions for the combined problem of rail and truck shipments in jointly addressing production scheduling and product distribution for products with multiple options.

7. References

1. Brusco, MJ, GM Thompson and LW Jacobs, 1997, “A morph-based simulated annealing heuristic for a modified bin-packing problem,” *Journal of the Operational Research Society* 48, 433-439

2. Chandra, P., and M.L. Fisher, 1994, “Coordination of production and distribution planning,” *European Journal of Operational Research*, 72(3), 503-517.

3. Chen, Zhi-Long, George L Vairaktarakis, 2005, “Integrated Scheduling of Production and Distribution Operations,” *Management Science* Vol. 51, No. 4, April 2005, pp. 614-628

4. Coffman, E. G., Jr., Garey, M. R., and Jonson, D. S., 1997, “Approximation algorithms for bin packing: A survey,” *Approximation Algorithms for Np-Hard Problems* (pp. 46-93). Boston, PWS Publishing.

5. Cohen, M. and H. Lee, 1988, “Strategic Analysis of Integrated Production-Distribution Systems,” *Operations Research*, 36(2), 216-228.

6. Drexl, A., A. Kimms, and L. Matthiessen, 2006, “Algorithms for the car sequencing and the level scheduling problem,” *Journal of Scheduling*, 9(2), 153-176.

7. Garcia, J.M., Lozano, S. and Canca, D., “Coordinated scheduling of production and delivery from multiple plants,” *Robotics and Computer-Integrated Manufacturing* Volume 20, Issue 3, June 2004, Pages 191-198.

8. Garey, M. R., Johnson, D. S., 1979, Computers and Intractability: A guide to the theory of NP-completeness, San Francisco, Freeman.

9. Hahm, J. and Candace Yano, 1995, “Economic lot and delivery scheduling problem: models for nested schedules,” *IIE Transactions*, 27(2), 126-139.

10. Inman, R.R., and R.L. Bulfin, 1991, “Notes: Sequencing JIT Mixed-Model Assembly Lines,” *Management Science*, 37(7), 901-904.

11. Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R., 1974, “Worst case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.* 3 299-326

12. Miltenburg, J., 1989, “Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems,” *Management Science*, 35(2), 192-207.

13. Monden, Y., 1997, *Toyota Production System*, third edition, Institute of Industrial Engineers Press, Norcross, GA.

14. Pundoor, G. and Z.-L. Chen., 2005, "Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost," *Naval Research Logistics*, 52(6), 2005, 571-589.

15. Rao, R.L., S. S. Iyengar, 1994, "Bin-Packing by Simulated Annealing," *Computers Math. Applic.* Vol. 27, No. 5, pp. 71-82.

16. Yano, C. and R. Rachamadugu, 1991, "Sequencing to Minimize work overload in Assembly lines with product options," *Management Science*, 37(5), 572-586.

PART 3

A DECISION SUPPORT TOOL FOR PLANT ASSIGNMENT FOR PRODUCTS WITH MULTIPLE PRODUCT OPTIONS WITH AN AUTOMOBILE ASSEMBLY PLANT CASE STUDY

Abstract

Plant assignment is a common operational topic faced by industry when there are multiple facilities to manufacture a product, while each facility has its production-resource, material, and supply-chain related constraints. This paper addresses plant assignment for products with multiple product options, based on which a unit of the product can be uniquely different from other units. The focus of this paper is on assigning such products to multiple plants considering multiple constraints related to various product options in order to minimize transportation costs and costs of assignment infeasibility. These constraints related to multiple product options can be due to consideration in production resources, material supply, and other supply-chain related limitations. A series of binary- and mixed-integer programming models are presented; and a decision-support tool based on the optimization models is presented. A case study is presented to demonstrate the application of the decision support tool in an automobile manufacturing company.

1. Introduction

Mixed-model assembly lines nowadays are a common practice applied in many assembly systems. Production of customizable products with multiple product attributes or options are often limited by capacities on a mixed-model assembly line due to constraint limits on at least some of the product attribute values. For a product that can be assembled in multiple assembly plants, assigning production orders to these plants considering transportation costs while considering the capacity constraints related to multiple product attribute values needs to be performed prior to the scheduling and sequencing stages at each plant. Proper models, solution procedures, and decision-support framework for plant assignment to minimize the transportation and other relevant costs considering constraints related to various product attributes can be developed.

Assigning customer demand to production facilities was termed demand allocation problem; the demand allocating problem has been widely studied in the context of make-to-order systems where the objective is typically to optimize a function of the manufacturing lead time (Benjaafar et al. 2004). Buzacott and Shanthikumar (1993) presented several cases in this area. Green and Guha (1995) considered a multi-facility and multi-server system to allocate servers and demands in a service system with multiple facilities. Poisson processes are assumed for the arrival and service processes. Benjaafar and Gupta (1999) considered multi-product, multi-facility workload allocation problem with setup times. Demands were assumed to arrive according to Poisson processes, but setup and the processing times can be arbitrary distributions. Heuristic workload allocation was developed based on insight from a nonlinear optimization problem.

Benjaafar et al. (2004) jointly considered demand allocation to facility and inventory level for products at each facility for multiple products, thus extended the consideration to make-to-stock systems. The demand is commonly assumed to follow a Poisson distribution. Different cases were considered based on demand splitability and warehousing forms. For each case, a solution procedure is presented to obtain the optimal demand allocations and optimal inventory. The production, transportation, inventory, and backorder costs are considered with a service-level requirement for each product in the model.

The existing research as stated above deals with the problems with probabilistic demand and products each with interchangeable units. Our study is trying to solve the problem of allocating known demands to multiple facilities with products of which each individual unit may be treated as a unique product due to multiple product options. There is also an increasing research attention on scheduling and sequencing for products with multiple options (for example, Ding and He, 2007, Gravel et al., 2005, Puchta and Gottlieb, 2002).

This paper addresses plant assignment for production orders of a product that can be manufactured in multiple plants in an assemble-to-order production system, where the production capacity is related to multiple product options. A product with multiple options results in different capacity requirements for these options at each plan; and each plant has its capacity limit on the amount of each product option in a period. These capacity limits can be due to material-supply limitation, production-resource limitation, or other supply-chain related limitations. In some cases, it is assumed that some of these capacity limits allow minor modifications through negotiation with suppliers or by adding capacity within the production facility, in order that the overall cost can be reduced.

This paper addresses the modeling, solution, and information tool in assigning production orders of a product with multiple options to multiple plants while considering constraint limits related to product options in order to minimize the transportation cost and cost of infeasibility. In Section 2, a problem description is given and the basic mathematical programming models are presented. In Section 3, a decision-support tool is presented. In Section 4, a case study in the automobile industry is presented.

2. Problem Description and Mathematical Programming Models

Assume that a company produces a product with multiple product options of which p options are related to limited production capacities in each of m facilities. That is, each plant j has a limit L_{jk} on the total quantity of product option k. There are a total of n production orders in a period. Limitation on the total quantity with a certain product option is due to limited material supplies, limited capacity in the production resource, and other limitations. The objectives can include the total transportation costs of having orders produced at a plant, costs of adjusting the capacity limits to accommodate more orders, and the costs of unassigned orders due to capacity constraints. It is assumed that the subsequent transportation costs of having an order *i* produced in plant *j* is c_{ij}. The cost of increasing a unit of capacity limit *k* at plant *j* is assumed to be known as b_{jk} in certain cases. Each product with multiple product options can be built at any plant under consideration within the production capacity limits.

In order to provide multiple considerations and sensitivity analysis, multiple mathematical programming models are presented. When plant capacity-limit adjustment is an option, the model may also include the associated adjustment costs in plant assignment. These mathematical-programming models may be solved iteratively to achieve applicable plant assignment results with possible empirical adjustment.

Mathematical Programming Models

Model 1 The first model assumes that all capacity limits are fixed. Any order that can not be assigned due to constraint limitation in the model can be placed into an "infeasible" pool (unassigned-order pool) with a cost of infeasibility due to unmet demand. The objective here is to find an optimal assignment of all orders while minimizing the total transportation plus infeasibility costs.

Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if order i is assigned to plant j} \\ 0, & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{if order i is assigned to the infeasible pool} \\ 0, & \text{otherwise} \end{cases}$$

Parameters:

$$a_{ik} = \begin{cases} 1, & \textit{if order i} \text{ uses attribute k} \\ 0, & \textit{otheriwse} \end{cases}$$

c_{ij} = Transportation cost if order i is assigned to plant j

f_i = Cost of an infeasible order i

L_{jk} = Capacity limit associated with option k at plant j

$$\textit{Minimize} \qquad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i z_i \tag{1}$$

$$\text{Subject to} \qquad \sum_{i=1}^{n} x_{ij}a_{ik} \le L_{jk} \qquad j = 1, 2,.., m,\ \text{k} = 1, 2, ..., \text{p} \tag{2}$$

$$\sum_{j=1}^{m} (x_{ij}) + z_i = 1 \qquad i = 1, 2,.., n \tag{3}$$

$$x_{ij} = 0 \ or\ 1,\ i = 1, 2, ..., n,\ j = 1, 2, ..., m,\ and\ z_i = 0\ or\ 1, i = 1, 2, ..., n$$

This model determines the optimal plant assignment based on the given capacity limits in various product options to minimize the sum of transportation costs and costs of infeasible orders. The optimal solution provides an initial plant assignment and allows further development of a sensitivity analysis for solution improvement purposes.

Model 2 When Model 1 has a high transportation cost or infeasible (unassigned) orders in the optimal solution, Model 2 can be applied to further determine how to reduce the transportation cost or total cost from Model 1. This may be accomplished by including the capacity increases associated with various product options at each plant as decision variables; and the objective can be to minimize the overall increase in the constraint limits and the cost of infeasible orders. In this model a condition is included to limit the total transportation plus infeasible-order costs to be no greater than a user-specified level as a desired limit on these costs. Model 2 can thus be applied to further determine how to achieve the desired total cost by increasing certain amounts of capacity limits, with a same or different level of transportation cost from Model 1. The optimal solution from Model 2 represents a sensitivity analysis of model 1 to assist in identifying how to consider the constraint-limit modification in order to reduce the overall cost. (If a certain capacity limit is not adjustable or can only allows limited adjustment, such a restriction can be easily incorporated in the model.) Model 2 is as follows:

Additional decision variables:

W_{jk} = capacity limit increase for product attribute k at plant j

Additional parameters:

T = The total transportation plus infeasibility costs at a user-specified level

$$\textit{Minimize} \quad \sum_{j=1}^{m}\sum_{k=1}^{p} W_{jk} \tag{4}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} x_{ij} a_{ik} + \leq L_{jk} + W_{jk} \qquad j = 1, 2, .., m,\ k = 1, 2, .., p \tag{5}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} f_i z_i \leq T \qquad i = 1, 2, .., n,\ j = 1, 2, .., m \tag{6}$$

$$\sum_{j=1}^{m} (x_{ij}) + z_i = 1 \qquad i = 1, 2, .., n \tag{7}$$

$$x_{ij} = 0 \textit{ or } 1,\ i = 1, 2, .., n,\ \mathrm{j = 1, 2, .., m; z_i = 0} \textit{ or } 1, \mathrm{i = 1, 2, ...n; W_{jk}} \geq 0 \textit{ and } \text{integer}$$

Model 2 is applied when it is desirable to further reduce the total (T) transportation plus infeasible-order costs. In the case that the infeasibility cost has dominated the transportation costs and that there is no infeasible order, T represents the desirable level of transportation cost. Adding the transportation plus infeasibility cost limit in constraint set (6) in Model 2 provides a sensitivity analysis for Model 1, where integer variables are used and a sensitivity analysis result is generally not obvious. Model 2 can provide guidance regarding which capacity limits to increase while containing the total cost. If decisions are made to allow some positive W_{jk}'s as identified by Model 2, Model 1 can then be further re-solved following these increases.

Model 3 In case that the cost b_{jk} of adjusting a capacity limit k at plant j is known, Model 3 can include these costs in minimizing the total cost.

Additional parameters:

b_{jk} = cost of adjusting a unit of the capacity limit k for at plant j

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i z_i + \sum_{j=1}^{m}\sum_{k=1}^{p} b_{jk}W_{jk} \tag{8}$$

$$\text{Subject to} \quad \sum_{i} x_{ij}a_{ik} \le L_{jk} + W_{jk} \qquad j = 1, 2, .., m,\ \text{k} = 1, 2, .., \text{p} \tag{9}$$

$$\sum_{j=1}^{m} x_{ij} + z_i = 1 \qquad i = 1, 2, .., n \tag{10}$$

$$x_{ij} = 0 \ or\ 1,\ i = 1, 2, ..., n,\ j = 1, 2, ..., m,\ and\ z_i = 0\ or\ 1,\ i = 1, 2, ..., n; W_{jk} \ge 0 \text{ and integer}$$

3. A Decision Support Tool

For practitioners, solving such a plant assignment problem may need to be performed periodically and interactively. A computing tool can be developed with user-friendly interface based on an efficient optimization solver. If needed, heuristic solution procedures can be developed to find a good solution in a shorter time. To be used in a company setting, such a computing tool should be able to retrieve relevant data from its system databases regarding order information, relevant costs, and capacity limits. This program can then solve the binary-integer model (Model 1) or mixed-integer programming model (Model 3). The tool should also allow the user to make further adjustment of the capacity limits (such as Model 2) based on the initial optimal solution or based on user experience and judgment. These models can be run iteratively until the solution is satisfactory and no more capacity limit adjustment is deemed beneficial or feasible. The system can then output results to a spreadsheet or database.

A schematic of the computing tool is given in Figure 1. The decision-support system can be written in a programming language that allows being executed in a user friendly manner; for example, a programming language in the .net framework. An optimization solver may be called from the programming language. If an optimization solver can solve the problem efficiently, then the optimal solution approach can be used instead of a heuristic solution procedure.

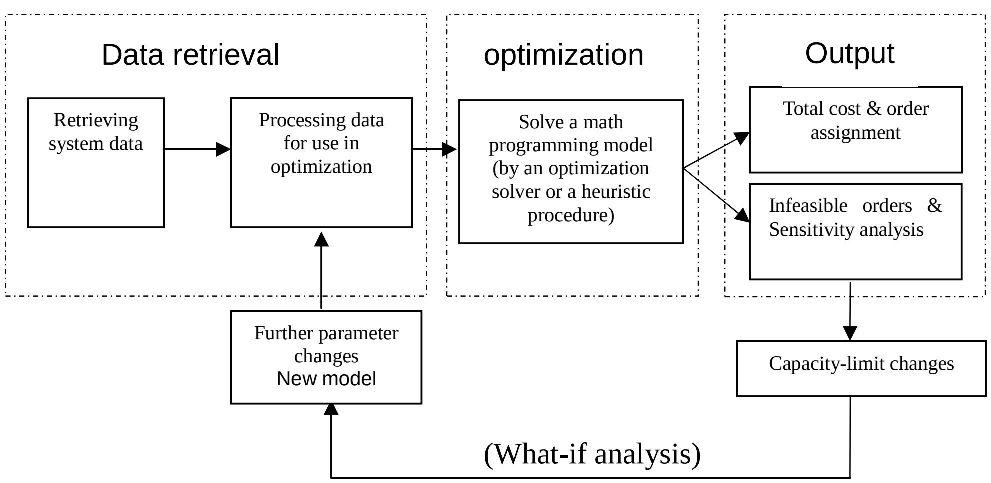

Figure 1. A schematic of the decision support tool for plant assignment

4. A Case Study

This case study considers an automobile manufacturer in the U.S., which manufactures passenger cars, trucks, and sports utility vehicles in multiple assembly plants. The company uses more than 6,000 parts from overseas and domestic suppliers. Some of these parts are limited in total daily or weekly quantities. For each product line, the vehicle has about sixty customer-selectable options, for example, sunroof, leather seat, hybrid, and alloy wheel. Each assembly plant has multiple assembly lines, and each assembly line has capacity limits on the quantities of certain product options of the assembled products. The focus of this case study is on a product line that has a high production volume and needs to be produced in the company's two plants in the U.S.

Once vehicles are produced, at each plant, the company ships vehicles to various sales regions or dealers in North America by truck or rail transportation. The transportation-cost difference by shipping a vehicle from a closer plant can be significant and quickly amplified by the magnitude of its distribution operation. Thus for a vehicle model that can be manufactured in multiple facilities, it is economical to assign to the closest site to minimize the transportation cost while staying within the plant-capacity and material-availability limits related to multiple product options. The total number of options that are considered during plant assignment is around 20 to 60 product options in various periods. The set of active options and their capacity limits vary over time.

The planning horizon considered by the company in plant assignment includes a month or a week depending on the data set considered at the time of application. The monthly production of the considered product is about 20,000 units. For monthly planning, only sales-region information (there are 8 sales regions within the

U.S.) is available for each vehicle order. When it comes to weekly planning, many orders then have specific dealer information. The transportation cost can be associated with a dealer order or with a sales-region order for being produced at each plant. The capacity constraints are given at the time of determining the plant assignment; however, some limits may be modified through negotiation with suppliers or by internal resource or staffing adjustment at a plant. The objective in plant assignment is to minimize the total transportation cost with a minimum achievable capacity-limit increase.

The computer program developed for plant assignment in the company incorporates two mathematical programming models (Models 1 and 2) as stated above. The program is written in C# that invokes OPL-CPLEX to solve the mathematical programming models, and handles input and output data by user-friendly forms. Running Model 1 requires first assigning an "infeasible order" cost based on the best knowledge of the perceived tradeoff or cost estimate. The system will then take the infeasibility and transportation costs into account to obtain an optimal solution. If the infeasible order cost can be close enough to the real opportunity cost of loss of sale, then this model should present the final optimal decision for order assignment. The company in the Case Study, however, requires as many orders as possible to be produced on time. Thus, in Model 1 a very large value was suggested for the infeasible-order cost value.

After solving Model 1, a sensitivity analyses can be beneficial to give the user suggested directions for improving the total cost through modifying the capacity limits of various constraints regarding the product options. Model 2 limits the transportation plus infeasible-order cost to a user specified level while minimizing the total increase in the limits. The output from Model 2 is treated as a guideline for

Table 1. The average test results from 10 randomly generated problems

Scenario	1	2	3	4
(infeasibility cost, % shift in constraint limits)	(\$50, 12%)	(\$150, 12%)	(\$50, 5%)	(\$150, 5%)
No. of infeasible orders	877.8	402	0	0
Transportation cost	\$23,759.2	\$67,171.8	\$7,716.9	\$7,716.9
Total cost	\$67,649.2	127,471.8	\$7,716.9	\$7,716.9
CPU time (Second)	36.773	33.745	47.219	184.49

capacity-limit negotiation or adjustment. (Model 3 is currently not considered in this decision-support tool as the unit cost of increasing a capacity limit is not easy to estimate by the firm.) The user can also run Model 1 in multiple trials with different parameters based on experience and judgment, or based on capacity-limit increases indicated by Model 2 with different cost levels.

5. Computational experience

To further test the experience of the solution tool, 10 problems of a two-plant scenario were randomly generated. Each problem has 18,000 orders, and each order has 10 customer-selectable options. For each option, each order has an equal probability of with or without the option. For each order, the "preferred" plant (with a lower transportation cost) is set randomly. The initial capacity limit for each option at each plant is set as equal to the total number of geographically preferred orders with the option. Each initial capacity limit is then reduced by a certain percentage (5% or 12%) at one plant and moved to the other plant. The transportation cost at an un-preferred plant is randomly set at 2^0, 2^1,.., or 2^7. The cost for an infeasible order is assumed to be \$50 or \$150. Model 1 has 54,000 binary integer variables, and 18,022 constraints. The average test results are shown in the Table 1. The results showed efficient computation with a cost saving potential.

Table 2. Results from real production data sets

Case	1	2	3	4	5	6	7	8
# of orders	4,786	4,786	4,785	4,786	4,789	4,529	5,656	5,659
CPU time (sec)	6.26	6.76	6.15	5.85	7.29	6.01	6.75	7.50
Transportation cost	$54,894	$53,728	$56,007	$57,062	$57,199	$36,120	$45,319	$44,716

Furthermore, 8 weekly production data sets were also tested. In each data set, there are a total of about 5,000 orders (weekly order) to be assigned to two plants in each week. Six product options are considered in the plant capacity limits. Within less than 10 seconds, the optimal solution (Model 1) was obtained with no infeasible orders. The total cost savings for assigning these weekly orders were estimated to be $60,000 as compared with the procedure currently used in the company. The test results are presented in Table 2.

Figure 2 has one of the forms of the developed computer tool. It summarizes the optimal solution, total cost, transportation cost, and cost of infeasibility. The capacity limits and actual capacity usage in these limits in the optimal solution are also displayed.

The user can then take further actions. In this case, the total cost is $54,753.49 with 38 “infeasible” orders. Model 2 can be run to find the minimum capacity increase to reduce the number of “infeasible” orders or to limit the transportation cost by providing a user-specified total cost. After running Model 2, if the results in capacity limit adjustment are acceptable, results can be retrieved, or else with further user adjustment the user should re-run Model 1 to obtain the optimal total cost and final assignment based on the capacity limits obtained in Model 2. At the end of iterative computation, the information of order assignment and capacity limit increase will be passed to the production system for further scheduling, sequencing, and coordination. As shown on the screen, various computing options

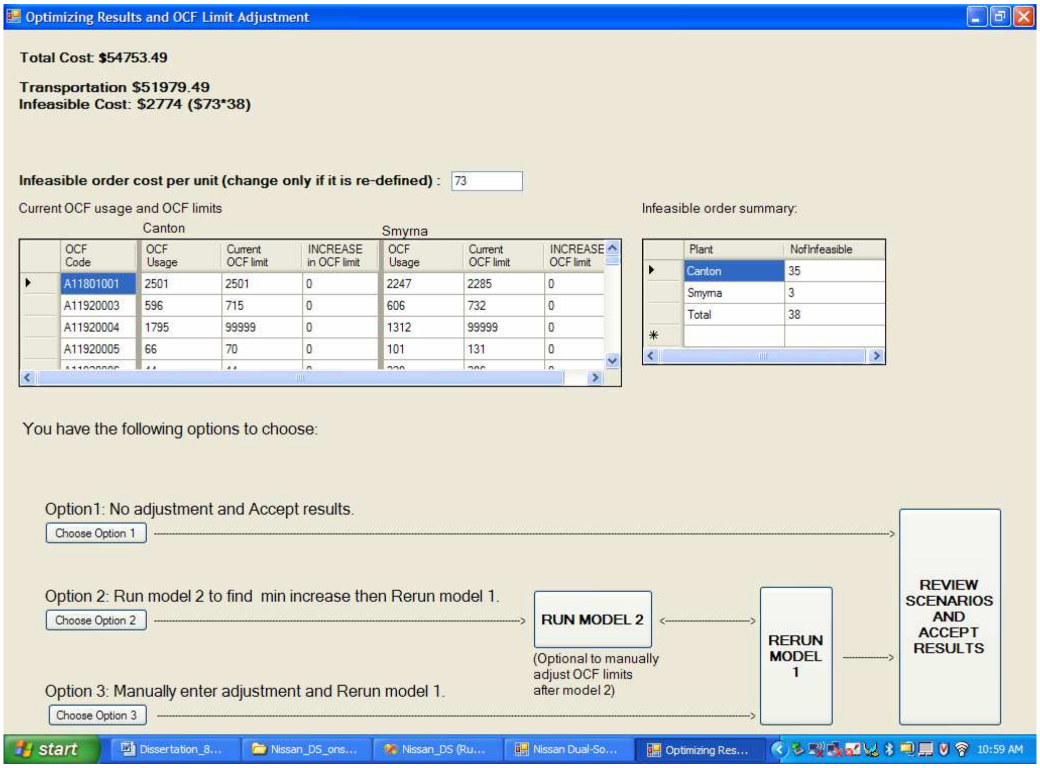

Figure 2. A result screen in the program

are provided for user selection. Since this can become an iterative computing process, the results from different scenarios (based on different parameter settings) can be saved for final selection. The above program has been well received by the company and is in the process of being adopted.

6. Summary and Conclusions

This paper addresses the plant assignment problem considering the transportation cost and cost of assignment infeasibility for a product with multiple product options to be manufacturing at multiple plants. A binary-integer and two mixed-integer programming models were presented for possible iterative solution. A model is to minimize the transportation costs and possibly costs of adjusting the capacity limit. Another model serves as a tool for sensitivity analysis to identify the best choices to modify capacity limits. The framework of a decision support tool was presented.

An industrial case study was also presented, and the binary-integer and mixed-integer programming models were applied in a decision-support tool developed for the considered automobile manufacturing firm. The decision-support tool invoked an optimization tool and solved the problem efficiently. Aided by user-friendly interface, the program allowed the user to retrieve data, solve the problem, and modify the solutions interactively. Noticeable cost savings have been observed through using the optimal solution based on actual data sets in comparison to the currently existing procedure. The application demonstrated that an optimal and efficient solution tool aided by user-friendly interface can provide a beneficial decision-support system in a manufacturing environment.

7. References

1. Benjaafar, S., ElHafsi, M., de Vericourt, F., 2004, Demand Allocation in Multiple-Product, Multiple-Facility, Make-to-Stock Systems, *Management Science*, 50 (10), 1431-1448.

2. Benjaafar, S., D. Gupta. 1999. Workload allocation in multi-product, multi-facility production systems with setup times. *IIE Trans*. 31 339–352.

3. Buzacott, J. A., J. G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems*. Prentice Hall, Engelwood Cliffs, NJ.

4. Gravel, M., C. Gagne, W.L. Price, 2005, "Review and comparison of three methods for the solution of car-sequencing problem," *Journal of Operational Research Society*, 56, 1287-1295.

5. Green, L. V., D. Guha. 1995. On the efficiency of imbalance in multi-facility multi-server service systems. *Management Science*. 41 179–187.

6. Ha, A. 1997. Optimal dynamic scheduling policy for a make-to-stock production system. *Oper. Res*. 45 42–53.

7. Puchta M. and J. Gottlieb, 2002, "Solving Car Sequencing Problems by Local Optimization," *Applications of Evolutionary Computing*, LNCS 2279, Springer, 132-142.

8. de Véricourt, F., F. Karaesmen, Y. Dallery. 2000. Dynamic scheduling in a make-to-stock system: A partial characterization of optimal policies. *Oper. Res*. 48 811–819.

9. de Véricourt, F., F. Karaesmen, Y. Dallery. 2002. Stock allocation for a capacitated supply system. *Management Science*. 48 1486–1501.

10. Wein, L. M. 1992. Dynamic scheduling of a multi-class make-to-stock queue. *Oper. Res*. 40 724–735.

11. Zipkin, P. H. 1995. Performance analysis of a multi-item production-inventory system under alternative policies. *Management Science*. 41 690–703.

SUMMARY AND CONCLUSIONS

Mixed-model assembly line sequencing is a common practice in many industries, especially in the automobile industry. Efficient production scheduling and sequencing can lead to the improvement in the overall production, material-supply, and distribution efficiency. This dissertation research developed a series of methods and heuristic procedures to deal with the assembly-line scheduling and sequencing problems considering automobiles with multiple options as commonly encountered by automobile manufacturers. A common thread of the scheduling and sequencing problems addressed in this dissertation is the definition of products based on multiple options. The scope of the research considerations expands from production to material supply to distribution.

Another research topic considered in this dissertation addressed plant assignment for a product with multiple options on which the plant constraints are based. This is performed before scheduling and sequencing in order to reduce the transportation costs and a possible cost of infeasibility.

Heuristic procedures to solve a multi-objective assembly line sequencing problem for products with multiple options were presented to have broader considerations than mixed-model assembly line sequencing and the car-sequencing problem addressed individually. The computational results showed that the developed procedures achieved a better solution quality than simulated annealing in a relatively short time. These developed procedures may be applied to a sequencing problem for automobiles or other products with multiple options.

A Heuristic procedure addressing the production scheduling problem to jointly consider production smoothing and product distribution was presented. A simulated annealing procedure was developed to further improve part of the heuristic solution. The partial solution in package assignment as obtained by the simulated annealing procedure showed improvement over that by the Heuristic procedure. The Heuristic procedure in conjunction with the simulated annealing procedure can be applied to the scheduling problem of products with multiple options in addressing the production smoothing and distribution efficiency jointly. The test results also showed a significant inventory reduction using the proposed scheduling method than the existing scheduling practice which ignores the distribution efficiency.

To further address the plant assignment problem for product with multiple options, several mixed-integer and binary-integer programs were developed. These mathematical programming models, along with a computer tool, allowed interactively solving the plant assignment problem based on the system constraints and conditions.

In summary, problem models and solution procedures were developed in this dissertation to address the mixed-model assembly line sequencing and scheduling to improve the supply chain operation involving production, material supply, distribution, and production loading of the automobile assembly plants considering multiple product options. Randomly generated problem sets were used to evaluate the developed procedures and compared with meta-heuristic approaches when applicable. A case study and several industrial data sets were included to provide application examples.

Future research may be conducted in the following areas:

1) Developing a more general procedure to deal with the general sequencing problem of products with multiple options considering multiple objectives presented in Part 1 of the dissertation. This may ease the implementation of

the solution procedure for real production systems.

2) In Part 2, the Ant Colony Optimization (ACO) heuristic procedure may be developed to enhance the solution quality.

3) For the decision support tool developed in Part 3, when the order planning horizon changes from monthly to weekly due to changes in customer orders, it needs to be re-run for further update. A more tractable program may be developed to deal with the system change.

4) The sensitivity analysis regarding the binary-integer programming problem presented in Part 3 may be further developed to assist in decision making.

References

1. Bergen, M, P. Beek, and T. Carchrae, 2001, “Constraint-based Vehicle Assembly Line Sequencing,” *Proceedings of the 14th Conference of the Canadian Society for Computational Studies and Intelligence*, Springer, 88-99.

2. Bautista, J., R. Companys, and A. Corominas, 1996, “Heuristics and exact algorithms for solving the Monden problem,” *European Journal of Operational Research*, 88, 101-113.

3. Chen, Zhilong and Vairaktarakis, G.L., 2005 “Integrated scheduling of production and distribution operations,” *Management Science*, v 51, n 4, April 2005, p 614-28

4. Cheng, L. and F. Ding, 1996, “Sequencing mixed-model assembly lines to minimize the weighted variations in just-in-time production systems,” *IIE Transactions*, 28 (11), 919-927.

5. Chew, T.-L., J.-M. David, A. Nguyen, and Y. Tourbier, 1991, “Solving constraints satisfaction problem with simulated annealing: The car sequencing problem revisited,” in 12th International Conference on Artificial Intelligence, Expert Systems, and Natural Language, 405-426.

6. Chandra, P., and M.L. Fisher, 1994, “Coordination of production and distribution planning,” *European Journal of Operational Research,* Vol. 72, No. 3, pp 503-17.

7. Cohen, M.and H. Lee, 1988, “Strategic Analysis of Integrated Production-Distribution Systems,” *Operations Research*, Vol. 36, No. 2, pp 216-228.

8. Davenport A. and E. Tsang, 1999, “Solving constraint satisfaction sequencing problems by iterative repair,” *Proceedings of the First International Conference on the Practical Applications of Constraint Technologies and Logic Programming*, 345-357.

9. Drexl, A. and A. Kimms, 2001, “Sequencing JIT Mixed-Model Assembly Lines Under Station-Load and Part-Usage Constraints,” *Management Science*, 47(3), 480-491.

10. Estellon, B., F. Gardi, and K. Nouioua, 2005, “Real life car sequencing: very large neighborhood search vs. very fast local search,” Working paper, Laboratoire d’Informatique Fondmentale – CNRS UMR 6166, Université de la Méditerranée, France.

11. Gent, I.P., 1998 “Two Results on Car-sequencing Problems,” Technical report of the APES Group, APES-02-1998.

12. Gottlieb, J., M. Puchta, and C. Solnon, 2003, “A Study of Greedy, Local Search, and Ant Colony Optimization Approaches for Car Sequencing Problems,” S. Cagnoni et al. (Eds.): Evo Workshops, LNCS 2611, 246-257.

13. Gravel, M., C. Gagne, W.L. Price, 2005, “Review and comparison of three methods for the solution of car-sequencing problem,” *Journal of Operational Research Society*, 56, 1287-1295.

15. Inman, R.R., and R.L. Bulfin, 1991, “Notes: Sequencing JIT Mixed-Model Assembly Lines,” *Management Science*, 37(7), 901-904.

16. Kubiak, W., 1993, “Minimizing variation of production rates in just-in-time systems: a survey,” *European Journal of Operational Research*, 66 (3), 259-271.

17. Kubiak, W. and S. Sethi, 1991, “A Note on “Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems”, *Management Science*, 37(1), 121-122.

18. McMullen, P.R., 1998, “JIT sequencing for mixed-model assembly lines with setups using Tabu search,” *Production Planning & Control* 9, 504–510.

19. McMullen, P. and G. Frazier, 2000a, “A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line,” *IIE Transactions*, 32 (8), 679-686.

20. McMullen, P., 2001a, “An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics,” *Computers and Industrial Engineering*, 41, 335-353.

21. McMullen, P., 2001b, “An ant colony optimization approach to address a JIT sequencing problem with multiple objectives,” *Artificial Intelligence in Engineering*, 15, 309-317.

22. McMullen, P.R., P. Tarasewich, and G.V. Frazier, 2000, “Using genetic algorithms to solve the multi-product JIT sequencing problem with setups,” *International Journal of Production Research*, 38 (12), 2653-2670.

23. McMullen, P.R., Tarasewich, P., 2005. “A beam search heuristic method for mixed-model scheduling with setups”, *International Journal of Production*

Economics 96, 273–283.

24. Miltenburg, J., 1989, “Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems,” *Management Science*, 35(2), 192-207.

25. Miltenburg, J. and T. Goldstein, 1991, “Developing production schedules which balance part usage and smooth production loads in JIT production systems,” *Naval Research Logistics*, 38 (6), 893-910.

26. Miltenburg, J. and G. Sinnamon, 1989, “Scheduling mixed-model multi-level just-in-time production systems,” *International Journal of Production Research*, 27 (9), 1487-1509.

27. Monden, Y., 1997, *Toyota Production System*, third edition, Institute of Industrial Engineers Press, Norcross, GA.

28. Muhl, E., P. Charpentier, and F. Chaxel, 2003, “Optimization of physical flows in an automotive manufacturing plant: some experiments and issues,” *Engineering Applications of Artificial Intelligence*, 16 (4), 293-305.

29. Smith, K., Palaniswami, M, and M, Krisnamoorthy, 1996, “Traditional heuristics versus Hopfield neural network approaches to a car sequencing problem,” *European Journal of Operational Research*, 93, 300-316.

30. Tetzlaff, U.A.W.; Pesch, E., 1999 “Optimal workload allocation between a job shop and an FMS,” *Robotics and Automation, IEEE Transactions* on Volume 15, Issue 1, Feb. 1999 Page(s):20 – 32

31. Wester, L., Kilbridge, M., 1964, “The assembly line model-mix sequencing problem,” *Third international conference on Operation Research*, Oslo 1963, (1964), 247–260.

32. Wild, R., 1972. *Mass-production management*, London.

33. Wilkinson, S. J., A. Cortier, N. Shah and C.C. Pantelides, 1996, “Integrated production and distribution scheduling on a Europe-wide basis,” *Computers & Chemical Engineering*, v 20, Suppl pt B, p S1275-S1280

34. Yano, C. and R. Rachamadugu, 1991, “Sequencing to Minimize work overload in Assembly lines with product options,” *Management Science*, 37(5), 572-586.

35. Zhao, X. and K. Ohno, 1994, "A Sequencing Problem for a Mixed-model Assembly Line in a JIT Production System," *Computers and Industrial Engineering*, 27, 71-74.

36. Zhao, X. and K. Ohno, 1997, "Algorithms for sequencing mixed models on an assembly line in a JIT production system", *Computers & Industrial Engineering* 32, 47–56.

37. Zhao, X. and K. Ohno, 2000, "Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages", *European Journal of Operational Research* 124, 560–570.

38. Zhao, X. and Zhou, Z., 1999. "Algorithms for Toyota´s goal of sequencing mixed models on an assembly line with multiple workstations," *Journal of Operational Research Society* 50, 704–710.

39. Zhu, J. and F. Ding, 2000, "A Transformed Two-Stage Method for Reducing the Part-Usage Variation and a Comparison of the Product-Level and Part-Level Solutions in Sequencing Mixed-Model Assembly Lines," *European Journal of Operational Research*, 127(1), 203-216.

VITA

Jingxu He received his B.E. in mechanical engineering from the University of Science and Technology, Hefei, China, in 1995. He is currently a Ph.D. candidate at the University of Tennessee, Knoxville, where he started his studies in 2004. Before that he got a M.E. in manufacturing from the University of Michigan, Ann Arbor, and worked for several auto parts companies after graduation.