IMPROVED FORMULATIONS, HEURISTICS AND METAHEURISTICS FOR THE DYNAMIC DEMAND COORDINATED LOT-SIZING PROBLEM

A Dissertation

by

ARUNACHALAM NARAYANAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2006

Major Subject: Information and Operations Management

IMPROVED FORMULATIONS, HEURISTICS AND METAHEURISTICS FOR THE DYNAMIC DEMAND COORDINATED LOT-SIZING PROBLEM

A Dissertation

by

ARUNACHALAM NARAYANAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	E. Powell Robinson
Committee Members,	Benito Flores
	Guy Curry
	Matt Keblis
Head of Department,	Dean Wichern

August 2006

Major Subject: Information and Operations Management

ABSTRACT

Improved Formulations, Heuristics and Metaheuristics for the Dynamic Demand Coordinated Lot-sizing Problem. (August 2006)

Arunachalam Narayanan, B.E., Anna University, India;

M.S., Texas A&M University

Chair of Advisory Committee: Dr. E. Powell Robinson

Coordinated lot sizing problems, which assume a joint setup is shared by a product family, are commonly encountered in supply chain contexts. Total system costs include a joint set-up charge each time period any item in the product family is replenished, an item set-up cost for each item replenished in each time period, and inventory holding costs. Silver (1979) and subsequent researchers note the occurrence of coordinated replenishment problems within manufacturing, procurement, and transportation contexts. Due to their mathematical complexity and importance in industry, coordinated lot-size problems are frequently studied in the operations management literature.

In this research, we address both uncapacitated and capacitated variants of the problem. For each variant we propose new problem formulations, one or more construction heuristics, and a simulated annealing metaheuristic (SAM).

We first propose new tight mathematical formulations for the uncapacitated problem and document their improved computational efficiency over earlier models. We then develop two forward-pass heuristics, a two-phase heuristic, and SAM to solve the uncapacitated version of the problem. The two-phase and SAM find solutions with an average optimality gap of 0.56% and 0.2% respectively. The corresponding average computational requirements are less than 0.05 and 0.18 CPU seconds.

Next, we propose tight mathematical formulations for the capacitated problem and evaluate their performance against existing approaches. We then extend the two-phase heuristic to solve this more general capacitated version. We further embed the six-phase heuristic in a SAM framework, which improves heuristic performance at minimal additional computational expense. The metaheuristic finds solutions with an average optimality gap of

0.43% and within an average time of 0.25 CPU seconds. This represents an improvement over those reported in the literature.

Overall the heuristics provide a general approach to the dynamic demand lot-size problem that is capable of being applied as a stand-alone solver, an algorithm embedded with supply chain planning software, or as an upper-bounding procedure within an optimization based algorithm.

Finally, this research investigates the performance of alternative coordinated lot-sizing procedures when implemented in a rolling schedule environment. We find the perturbation metaheuristic to be the most suitable heuristic for implementation in rolling schedules.

To my dad

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to Dr. Powell Robinson, without whom I would have neither started nor completed this work. He is the best advisor one could wish for. He has not only been my dissertation chair, but also a mentor and friend. I remain indebted to him for giving me this opportunity.

My appreciation extends to other committee members. I would like to thank Dr. Guy Curry, for his classes which laid the foundations for this research work, Dr. Benito Flores, for his experience, advice and guidance and Dr. Matt Keblis, for his support and camaraderie throughout my PhD career.

My thanks to Dr. Li-Lian Gao of Hofstra University, for providing codes for running some of the initial simulation runs in this research.

I would like to acknowledge the financial support for my doctoral studies from Department of Information and Operations Management, Mays Business School and Office of Graduate Studies.

I have been fortunate to have great colleagues and buddies throughout my student life at Aggieland. Specifically Jeremy Brann, whose has been the greatest office mate one could ask for, my PhD colleagues, Uzma Raja and Haiyan Fan, for their advice and competitive spirit in completing this dissertation. My friends have been the great source of motivation; particularly I would like to mention, Arun Surendran, for his enlivening intellectual discussions, Sharath Bulusu, for his stimulating academic thoughts and Pradip Krishnadevarajan, for his unwavering friendship.

This section would be incomplete without the mention of the two most important women in my life, my mom and sister. Throughout my life, they have supported and encouraged every step I had taken. I dedicate this dissertation to them.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

The coordinated (or) joint lot-size problem determines the time-phased replenishment schedule that minimizes the sum of ordering and inventory costs. A family setup cost is incurred each time one or more items in a product family are replenished, and a minor setup cost is charged for each item replenished. Item demand is assumed to be dynamic but deterministic over the planning horizon and must be met without backorders.

Coordinated lot-size problems are often encountered in supply chain management. Silver (1979) provides several examples of coordinated replenishment including scheduling a packaging line to produce various sizes and types of containers (*production*), purchasing multiple items from a common supplier (*procurement*), and shipping items that share a common mode of transportation (*transportation*). Stowers and Palekar (1997), provide two examples problems, one in a chemical processing plant, where the mix and quantity of products are decided and a second in the manufacturing of plastic dies, where a coordinated lot-sizing problem is solved to determine the economic lot size of the die.

Shapiro, Rosenfield and Stecke (2002) discuss a *flexible manufacturing system* that requires a setup cost when changing from one part family to another and a tool adjustment cost for changing between items within the product family. Robinson and Lawrence (2004) describe the production of industrial lubricants, which requires a setup cost each time a product family is made and a packaging line changeover for each different item produced. They also discuss the coordinated shipment of vaccines from a manufacturing facility to a *distribution* center. Here, each shipment requires a dedicated refrigerated truck (major setup

---

This dissertation follows the style of *Manufacturing & Service Operations Management*.

cost) and due to quality and security concerns, each vaccine type shipped incurs a minor setup cost associated with product labeling, packaging, temperature control, and paper processing for FDA regulatory compliance.

1.1 Background and motivation

Due to their mathematical complexity and importance in industry, coordinated lot-size problems are frequently studied in the operations management literature. However, while effective heuristic and exact algorithms exist for the uncapacitated variant of the problem, the more mathematically challenging problem with capacity constraints remains virtually unsolved. The mathematical structure is NP-complete.

Only Erenguc and Mercan (1990), Robinson and Lawrence (2004), Gao and Robinson (2004) and Federgruen, Meissner and Tzur (2004) propose algorithms and report computational experience for the coordinated capacitated lot sizing problem (CCLSP). In each case, the authors note the computational difficulty of finding optimal solutions and suggest that the literature will develop mainly in the direction of effective and computationally feasible heuristic procedures. Of all the heuristics developed, Federgruen et al.'s (2004) progressive interval/ expanding horizon heuristic provides the best results, but their approach is reasonable for only small problem sizes.

CCLSP is a generalization of the capacitated and uncapacitated, single-item, multi-item and coordinated lot-sizing problem. Hence, an effective heuristic for the coordinated replenishment problem provides a general solution methodology that could be used in requirements planning software as a one-stop solution approach for a variety of commonly encountered lot-sizing problems.

1.2 Scope of the dissertation

In this research, we focus on developing efficient and effective heuristics for both uncapacitated and capacitated variants of the coordinated replenishment problem. We start this study by developing new forward-pass and construction heuristics for the uncapacitated version of the coordinated lot-sizing problem. To evaluate the heuristic we need an efficient method to obtain the performance benchmark, like the optimal solution and its run time to achieve it. For achieving this we propose new mathematical formulations for each type of the coordinated lot-sizing problem. We then test these formulations against a wide range of parameter settings and document their improved computational efficiency over earlier mathematical representations of the problem.

Next, we extend the best performing construction heuristic, namely the two-phase heuristic, to solve the more mathematically challenging problem with capacity constraints. We also embed this new six-phase heuristic in a simulated annealing metaheuristic (SAM) framework to improve its performance. Then the resulting heuristics are evaluated against the existing optimality based procedures under all experimental designs proposed in literature.

Finally, this research investigates the performance of these alternative coordinated lot-sizing procedures in a rolling horizon environment. Prior research for single and multiple item problems (Blackburn and Millen 1980, Zhao et al. 2001) show that the relative ranking among the lot-sizing procedures change when applied to rolling schedules. But there is no published research for such a comparison among coordinated lot-sizing rules under rolling horizon. We intend to fill this void in this research.

1.3 Organization of the dissertation

In Chapter II, we briefly review the literature for dynamic demand lot-sizing procedures in static and rolling horizons. Chapters III and IV focus on the uncapacitated version of the joint replenishment problem. In Chapter III, we propose and compare the alternative MIP representations for the coordinated uncapacitated replenishment problem against the mathematical formulations in literature. In Chapter IV we present and evaluate the performance of two new forward pass heuristics, a two-phase construction heuristic and simulated annealing metaheuristic.

Chapters V and VI focus on the capacitated version of the coordinated replenishment problem. In Chapter V, we evaluate alternative MIP formulations to identify the most efficient mathematical problem representation for use in general purpose optimization software. Chapter VI describes a new six-phase heuristic and a simulated annealing metaheuristic for the capacitated problem. We asses the performance of these heuristics by conducting computational studies involving all the experimental designs presented in the literature for the joint replenishment problem.

In Chapter VII, for the first time, we evaluate the performance of alternative coordinated lot sizing rules in rolling schedule environment. Finally, chapter VIII presents the conclusions and implications of this research.

CHAPTER II

LITERATURE REVIEW

Dynamic demand lot-sizing problems are studied under two distinct planning environments, static and rolling horizon. In static or fixed horizon problem, the demand information is available for the entire horizon. On the other hand, in rolling schedule environment demand data is available only for a limited portion called the planning horizon. Due to the imperfect information about future demand, only a subset of the replenishment decisions is implemented and the problem is re-solved when new demands are appended to the horizon (Baker, 1977 and Blackburn and Millen, 1982a). This study involves the dynamic demand coordinated lot-sizing problem.

2.1 Coordinated lot-sizing problem: Static horizon

As illustrated in Figure 2.1, the dynamic demand coordinated capacitated lot-sizing problem (CCLSP) is a generalization of the single-item uncapacitated lot-sizing problem (ULSP), the single-item capacitated lot-sizing problem (CLSP), multiple-item uncapacitated lot-sizing problem (MULSP), multiple-item capacitated lot-sizing problem (MCLSP) and coordinated uncapacitated lot-sizing problem (CULSP). Karimi et al. (2003) and Robinson and Lawrence (2004) provide recent review of the literature on these problem classes. Earlier literature surveys are by Bahl, Ritzman, and Gupta (1987) for the ULSP, CLSP, and MCLSP and Aksoy and Erenguc (1988) for the CULSP. Drexl and Kimms (1997) provide a broad survey of the lot-sizing and scheduling literature. We summarize the literature most closely related to the research reported here.

2.1.1 Single-item uncapacitated lot-sizing problem (ULSP)

The ULSP, as introduced by Wagner and Whitin (1958), optimizes the timing and quantity of production lot-sizes for a single-item, assuming discrete dynamic demand and an unlimited product supply in each production period. System costs include per-unit inventory holding costs, fixed and variable production costs. Wagner and Whitin (1958) present a dynamic programming algorithm of $O(T^2)$ for its solution, where T is the number of time periods in the planning horizon. Backlogging of demand is not allowed. Zangwill (1966a) extends the model to permit demand backlogging. Evans (1985a) provides an efficient computer implementation of the Wagner and Whitin's solution methodology. Federgruen and Tzur (1991), Wagelmans et al. (1992), and Aggarwal and Park (1993) provide $O(T)$ and $O(T \log T n)$ algorithms for the problem under varying cost assumptions. The algorithmic approaches described in the above literature employ dynamic programming. The ULSP problem continues to receive attention both as an important problem class in its own right, and as a sub problem in more general planning models.

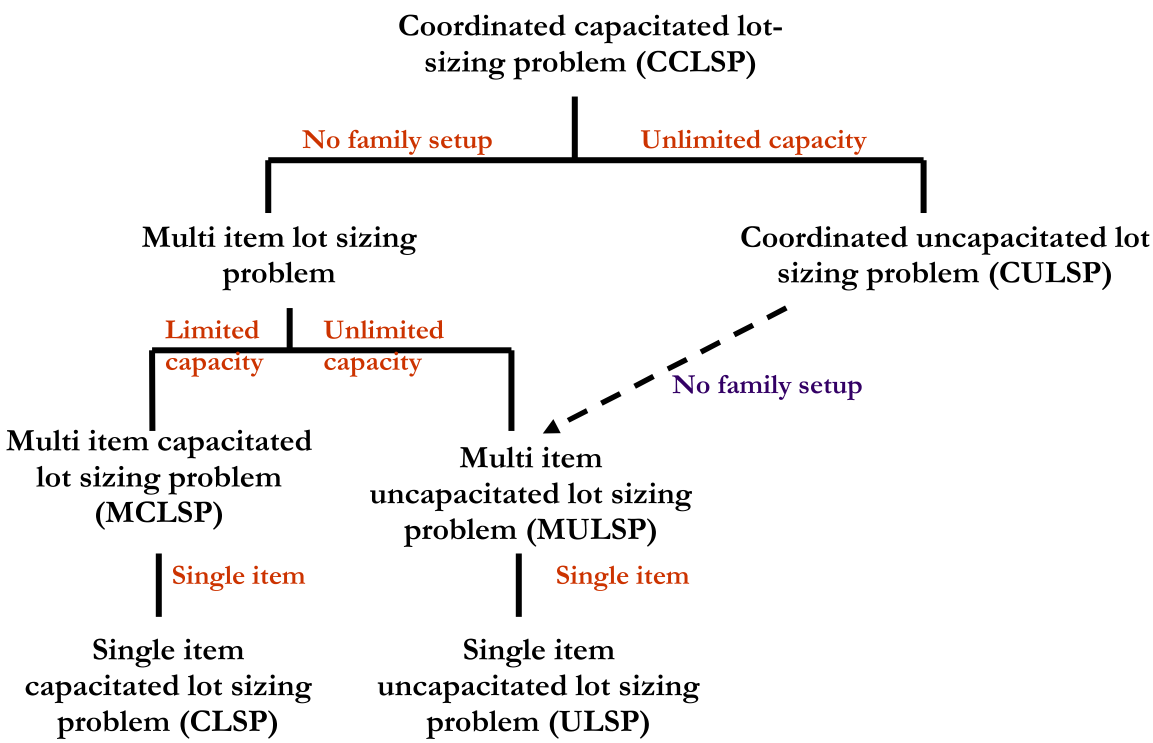


Figure 2.1: Taxonomy of dynamic-demand lot-sizing problem

2.1.2 Multi-item uncapacitated lot-sizing problem (MULSP)

Due to the lack of coupling capacity constraints among items, the MULSP is decomposed by item and solved as a set of independent ULSPs. Hence, all approaches for the ULSP could be used to solve this class of problems.

2.1.3 Single-item capacitated lot-sizing problem (CLSP)

Appending capacity constraints for labor, equipment, storage space, or other factors generalizes the formulation to the CLSP, but results in *NP*-hard or *NP*-complete problems depending upon specific objective function and capacity assumptions (Florian, Lenstra, and Rinnooy Kan, 1980 and Bitran and Yanasse, 1982). Consequently, optimization-based research focuses on special cases of the CLSP for which specialized solution approaches can be developed.

Florian and Klein (1971) describe the structure of optimal solutions of the single-item CLSP both with and without backlogging, and develop shortest route algorithms running in $O(T^4)$ for problems with equal capacities in every time period. For a related problem with bounds specified on inventory rather than production, Love (1973) develops an $O(T^3)$ algorithm.

Baker, Dixon, Magazine and Silver (1978) study the single-item CLSP with time varying capacity and no backlogging. They identify and exploit special mathematical properties of the optimal solution in a tree search algorithm. Specifically, if there is positive inventory carried over from a previous period, then production is either at capacity or zero. On the other hand, if there is positive production at a level less than capacity, then incoming inventory is zero. Experimental results indicate that the computational effort is practical only for a reasonable sized problem and in the worst-case the algorithm is combinatorially not good.

Lambert and Luss (1982) extend Florian and Klein's approach to time-dependent upper bounds by defining capacity in terms of the largest common divisor, c; thus $c_t = n_t c$ where n_t is a nonnegative integer for $t = 1,2,\ldots,T$. Their approach leads to an efficient algorithm when ever $N = \max_t(n_t)$ is relative small. The computational effort is $O(N^2T^4)$.

Erenguc and Aksoy (1990) describe a branch and bound algorithm for a single-item lot sizing problem with limitations on inventory storage and regular and overtime production capacity. The procedure is based on solving a finite number of linear knapsack problems with bounded variables. Problems with up to 12 time periods are solved in less than one second CPU time. Several other exact and polynomial time algorithms were developed for general and special cases of CLSP (Chung and Lin, 1988, Lotfi and Yoon, 1994, Kirca, 1990 and Chung, Flynn, and Lin, 1994)

2.1.4 Multi item capacitated lot-sizing problem (MCLSP)

The MCLSP considers a family of items that share a common constrained resource. Manne (1958) provides a linear programming formulation in which decision variables represent possible production sequences. The optimal linear programming solution, which does not guarantee integer valued production sequences, provides a good approximate solution when the number of items is relatively large when compared to the number of time periods. Dzielinski and Gomory (1965) recognize that Manne's linear programming formulation leads to computational difficulties for practical-sized problems. To overcome this obstacle, they apply Dantzig and Wolfe decomposition principles and a method for creating alternative setup sequences by solving a Wagner and Whitin (1958) type problem. Lasdon and Terjung (1968) improve upon Dzielinski and Gomory's method by using a column generation technique and compact inverse techniques to implicitly represent the generalized upper-bound constraints during simplex implementation. Kleindorfer and Newson (1975) study the Lagrangian dual of the MCLSP that is obtained by relaxing the capacity constraints. Bitran and Matsuo (1986) provide an error bound on Manne's linear programming approximation, and note that its quality is a function of the number of items, the length of the planning horizon and the set-up costs.

Several authors study alternative MIP formulations of the MCLSP in an effort to obtain tighter linear programming relaxations of the problem. Barany, Van Roy, and Wolsey (1984), Leung, Magnanti, and Vachani (1989), and Wolsey (1989) explore the polyhedral structure of the integer programming formulation of MCLSP to identify stronger formulations for the problem. Martin (1987) proposes variable redefinition as an approach for alternating the tightness of the linear programming relaxation of MCLSP sub-problems to obtain improved solution performance. Eppen and Martin (1987) provide several examples of variable redefinition for CLSP and MCLSP problems. Billington, McClain, and Thomas (1983), Evans (1985b), Hindi (1995a, 1995b) and Armentano, Franca, and de Toledo (1999) also provide mathematical programming approaches to the problem.

Newson (1975a, 1975b), Van Nunen and Wessels (1978), and Karni and Roll (1982) approach the problem as a series of uncapacitated single-item problems, and then adjust the production schedules to achieve feasibility. Lagrangian relaxation heuristics are proposed by Thizy and Van Wassenhove (1985), Trigeiro (1987), Trigeiro, Thomas, and McClain (1989), Diaby, Bahl, Karwan, and Zionts (1992a, 1992b), Campbell and Mabert (1991), and Millar and Yang (1994). Millar and Yang (1993) present a Lagrangian decomposition technique. Pratsini (2000) develops a savings heuristic for MCLSP with setup time and learning.

Several researchers propose construction heuristics based on consolidating production lot-sizes across time periods under a maximum savings objective criterion. Eisenhut (1975) develops a forward-pass heuristic based on the part period balance algorithm, but fails to insure capacity feasibility. Lambrecht and Vanderveken (1979) modify the Silver-Meal (1973) heuristic to consider multiple items and apply it with a backtracking subroutine which splits and shifts production into earlier time periods when necessary to obtain capacity feasibility. Dixon and Silver (1981) also utilize a variant of the Silver-Meal (1973) forward-pass heuristic, but guarantee solution feasibility by insuring that sufficient capacity is available in the remaining time periods of the planning horizon prior to lot-size consolidation. They also suggest several performance improvements based on lot-size elimination, merging, interchange and the properties of an optimal solution.

Dogramaci et al. (1981) propose a forward-pass and a four-phase "greedy-period" heuristic for MCLSP. The forward-pass algorithm extends Eisenhut (1975) and Lambrecht and Vanderveken (1979) by providing upper and lower bounds for order lot-sizes. The "greedy-period" heuristic attacks the planning horizon not necessarily from period 1 but from the periods offering the greatest marginal cost reduction. Experimental results confirm the superiority of this greedy-period approach over earlier forward-pass algorithms. Karni and Roll (1982) propose a heuristic that begins with the lower bound solution provided by ignoring the capacity constraint and solving the independent ULSP problems. If the combined ULSPs are capacity feasible, the solution is optimal; otherwise a five component shifting heuristic is employed to obtain feasibility at minimum cost.

Maes and Van Wassenhove (1988) conduct an extensive computational study on the existing MCLSP heuristics and provides general guidelines on the limitations and usefulness of these heuristics. Based on the result they also discuss the effect of parameter settings such as time between orders (TBO) on the performance of the heuristics.

2.1.5 Coordinated uncapacitated lot-sizing problem (CULSP)

Coordinated uncapacitated lot-sizing problems, which assume a joint setup is shared across a product family, are commonly encountered in transportation, procurement, and manufacturing contexts. The objective is to minimize total system costs while serving all customer demand from current production, inventory, and/or backorders. Total system costs include a joint setup charge for each time period any item in the product family is replenished, an item setup cost for each item replenished in each time period, inventory holding costs, and backorder costs. The CULSP is shown to be *NP*-complete by Arkin et al. (1989) and Joneja (1990). However, numerous researchers seek to exploit the specialized mathematical structure of CUSLP with specialized exact algorithms.

The traditional approach for solving CULSP is dynamic programming algorithm (see Zangwill 1966b, Veinott 1969, Kao 1979) in which computation time increase exponentially with problem size. Kao (1979) solve problem sizes containing 12 time periods and 2 products using dynamic programming. Silver (1979) demonstrates that dynamic programming is computationally reasonable when only few items are considered. Haseborg (1982) investigates the optimality of joint ordering policies to mitigate the impact of the number of items.

Erenguc (1988) solves problems with 12 time periods and 20 items by a combined branch-and-bound/dynamic programming procedure based on Veinott's (1969) "major setup pattern" concepts. The algorithm branches on major setup time periods to determine when production may occur and then solves independent ULSP problems to determine each item's replenishment schedule. Joneja (1990) indicates that the CULSP is NP-complete and develops a 'cost covering' heuristic that places a joint order when the total holding cost of all the candidate items exceeds their total ordering cost plus the joint cost. To test the quality of the heuristic solutions, Joneja proposes an integer programming formulation of CULSP for which the linear relaxation provides a very tight lower bound.

Kirca (1995) studies Joneja's integer programming formulation and provides an efficient branch and bound procedure for its solution. The procedure is based on heuristically solving the dual of the linear relaxation to obtain a lower bound on the original problem. An upper bound solution is constructed using complementary slackness conditions. The complementary slackness violations identify the major setup time period on which to branch. Computational experiments illustrate the large-scale capability of the procedures. The procedures effectively solve problems with 24 time periods and 50 items. Federgruen and Tzur (1994) develop a new a class of heuristics called the partitioning heuristic for coordinated lot sizing problem. They describe an efficient branch and bound technique whose upper bound is generated by a new greedy-add heuristic and the tight lower bound is provided by the partitioning heuristic.

Robinson and Gao (1996) propose a tight arborescent fixed charge network programming formulation of the problem with backlogging, and extend Erlenkotter's (1978) dual ascent based B&B procedures for its solution. The RG procedure begins by heuristically solving the dual of the LP realization of DJRP with a dual-ascent procedure. Next, using complementary slackness conditions, a primal feasible solution is constructed for DJRP. As necessary, a dual adjustment procedure attempts to reduce any complementary slackness violations and improve upon the incumbent solution. They report finding optimal solutions to problems with up to 24 (36) time periods and 40 (20) items. Other specialized approaches include branch and cut algorithms and Dantzig Wolfe decomposition (Raghavan 1993). These exact algorithms have not attained widespread application due to their mathematical complexity and inability to efficiently solve the large-scale problems encountered in industry. Hence, several researchers have attempted to develop efficient and effective heuristic solution approaches for CULSP.

Fogarty and Barringer (1987) propose a dynamic programming heuristic for the CULSP which, assumes all items are replenished every time a major setup occurs. Atkins and Iyogun (1988) extend the Silver and Meal (1973) heuristic to the problem, while Iyogun (1991) proposes an extension of the part-period balancing method (De Matteis and Mendoza, 1968). Silver and Kelle (1988) describe an improvement procedure applicable to any feasible heuristic solution which, considers whether a cost saving could be achieved by incorporating the production of each item in its previous production lot-size.

Boctor, Laporte and Renaud (BLR, 2004) evaluate the performance of the best known heuristics and a new perturbation metaheuristic (PM) for solving CULSP. The heuristics include the Fogarty and Barringer (FB, 1987) heuristic with the Silver and Kelle (SK, 1988) improvement procedure (FB-SK), Atkins and Iyogun's (1988) extension of the Silver-Meal (1973) heuristic, Iyogun's (1991) modification of the part-period balancing method, and Federgruen and Tzur's (1994) greedy-add heuristic. Computational results indicate that the PM heuristic, which is based on FB, is the most effective heuristic approach to the problem.

Research Extensions

As noted by Silver and Kelle (1988), the quality of the FB heuristic solutions are expected to be highest when item setup costs are relatively low compared to the joint setup cost and unit carrying costs. Consequently, though not explicitly addressed in BLR one would expect similar results to hold for the perturbation metaheuristic as it is based on FB. Hence, to better understand the effectiveness and limitations of these heuristics, we conduct additional computational studies with an expanded set of test problems and investigate new heuristic approaches for the CULSP. These new approaches and the computational study are presented in Chapter IV.

To evaluate the performance of a heuristic we need an efficient method to obtain the benchmark/optimal solutions. Boctor et al. (BLR, 2004) develop two new mathematical formulations for the uncapacitated problem to obtain optimal solutions in general purpose optimization software. Their experimental studies document the improved computational efficiency of their formulations versus the classical problem formulation seen in Federgruen et al. (1994). There are two other mathematical formulations (Kirca, 1995 and Robinson and Gao, 1996) for the CULSP, but they were not evaluated in BLR's experimental study. This research more fully explores the limitations and potential capabilities of BLR's formulations.

In Chapter III, we first investigate the tightness of the two BLR formulations and propose disaggregating the variable upper bound constraints of each formulation to more tightly constrain the joint setup variables to take on integer values in the linear programming (LP) relaxation. We then evaluate the computational efficiency of the original and revised formulations versus Robinson and Gao's (RG, 1996) plant location type formulation. Kirca's (1995) formulation was not included in the study, since Gao, Altay and Robinson (2004) in their comparative study find the RG formulation to be most efficient formulation for general purpose optimization software.

2.1.6 Coordinated capacitated lot-sizing problem (CCLSP)

The CCLSP contains both the joint capacity constraints that complicate solution of the MCLSP and the family setup decision variables that complicate the mathematical structure of the CULSP. The resulting mathematical structure is *NP*-complete. Erenguc and Mercan (1990) consider multiple product families assuming that labor is a sunk cost and therefore, not relevant in the problem formulation. Capacity is consumed during item run time and product family and item setup. Backorders are not allowed. The problem is solved with a branch-and-bound algorithm that uses linear under-estimators for setup and run times to obtain linear sub problems for lower bounding and a shift heuristic that finds feasible solutions for upper bounding. Computational results for problems with up to eight items, 10 time periods, and four families are reported.

Robinson and Lawrence (2004) propose a Lagrangian heuristic for the single-product family CCLSP with backorders. Computational experiments, over a wide range of environmental parameters, reveal heuristic solutions with average optimality gaps of 0.44%, 3.9%, and 4.72% at the 5%, 45% and 85% capacity utilization levels, respectively. Optimal solutions to 12-period problems with an 85% capacity utilization level and more than two items could not be found within 100 minutes of CPU time by general-purpose optimization software.

Altay (2001) propose and develop a cross decomposition approach to the coordinated replenishment problem. Their implementation shows that the problem is easier to solve when setup costs are negligible and becomes substantially difficult when the ratio of joint setup cost to total cost increase. Their results also indicate the difficulty in attaining computationally efficient optimal solutions. Gao and Robinson (2004) present a tight MIP formulation and a Lagrangian dual-ascent heuristic for the CCLSP. They find solutions within an average 0.67% from optimality for their set of test problems but the performance of the heuristic drops as the capacity utilization and the joint setup cost increase.

Federgruen, Meissner and Tzur (2004) develop a strict partitioning (SP) and a progressive interval/ expanding horizon (EH) heuristic for the capacitated problem. They show that the time-between orders (TBO) for items and family along with capacity utilization are the major factors that effect the performance of the heuristic. The SP heuristics runs faster, but they have a high average optimality gap of 14.7%. The EH heuristics provides solutions with an average optimality gap of 1.2% for a difficult set of test problems, but they have a major drawback. The EH heuristic's computational time increase drastically with problem size. A 25 item-10 period problem requires approximately 5 hours and 30 minutes to solve, in comparison, a problem with 10 item and 10 periods requires just 30 seconds.

Research Extensions

The results in the literature highlight the difficulty of finding both good heuristic and optimal solutions for the CCLSP problem and justify the development of alternative heuristic approaches for the problem. As stated earlier, to evaluate the performance of a heuristic one needs an efficient approach to obtain optimal solutions using general purpose optimization software. It is interesting to note that Robinson and Lawrence (2004), Gao and Robinson (2004) and Federgruen et al. (2004) use different MIP formulations to benchmark their heuristics. In Chapter V, we evaluate these different formulations along with three others to determine the most efficient mathematical model for solving CCLSP in optimization software like CPLEX and Xpress-MP.

In Chapter VI, we develop two new heuristics, a six-phase construction heuristic and simulated annealing metaheuristic, for solving the CCLSP. We then evaluate their performance against current approaches to the problem by conducting four computational studies, three of which are based on previous literature.

2.2 Coordinated lot-sizing problem: Rolling horizon

Due to the lack of information about future demand, the coordinated lot-sizing schedules are implemented on rolling schedule basis. Based on the limited demand information master production schedule (MPS) is constructed. The imminent decisions are implemented and then the schedule is rolled forward as new demand data becomes available (Blackburn and Millen, 1982a). As a result, the production schedule is updated moving through time. In this scenario, an optimal replenishment schedule with respect to a planning horizon may not necessarily be a component of a schedule that would minimize total costs over a period of time (Simpson, 1999). Moreover orders may be rescheduled during subsequent planning cycles, resulting in schedule instability or "nervousness".

Yeung et al. (1998) and Robinson et al. (2005) provide a recent review of literature that examines the multi-item lot-sizing rules and MPS parameters that affect the performance of requirements planning systems. Instead of duplicating these reviews, we present the literature that closely relates to our computational study.

The following Figure 2.2 depicts some of the basic definitions used in rolling horizon or MRP literature,

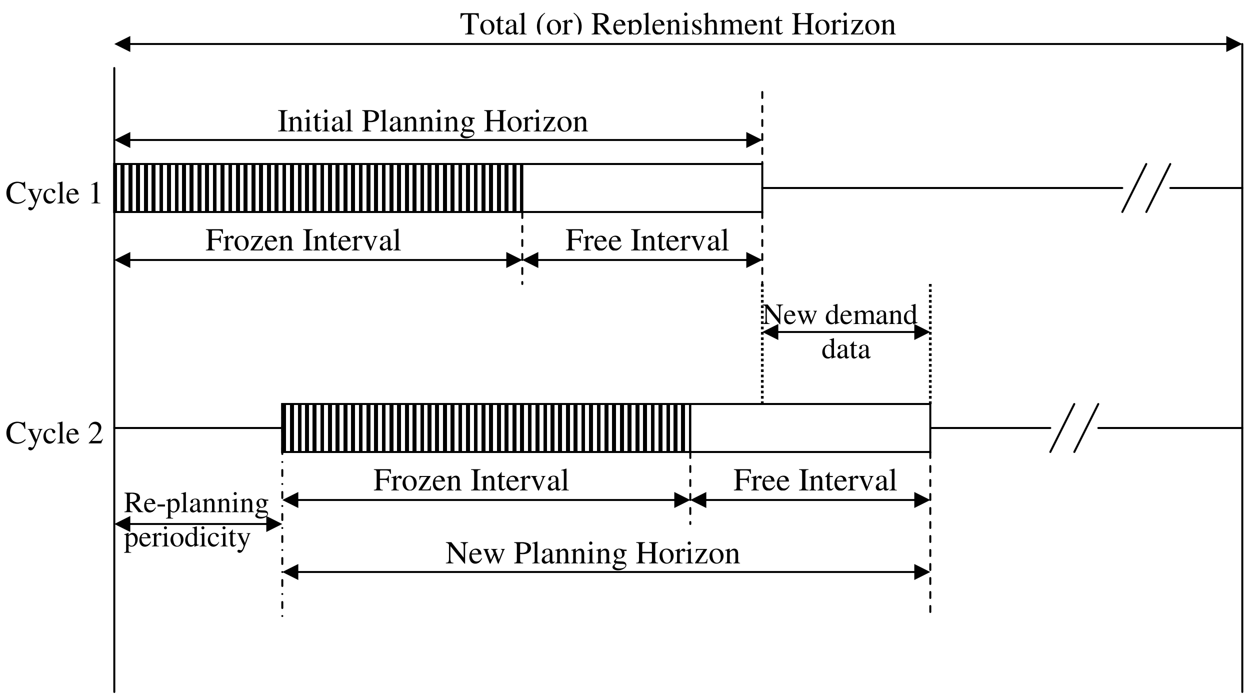


Figure 2.2: Illustration of MPS parameters in rolling horizon environment

The above figure shows the rolling schedule process for two successive planning cycles. The portion of the planning horizon whose order schedules cannot be changed is known as frozen interval. Re-planning periodicity is the interval between two consecutive planning cycles.

Baker (1977) conducts an experimental study to investigate the importance of MPS design parameters in rolling schedules. The findings indicate that the ideal planning horizon length should be equal to integer multiples of natural order cycles (or) time between orders (TBO). Sridharan et al. (1987) explore the effect of freezing method, frozen interval length and planning horizon length on MPS performance. They also show the tradeoff between system cost and instability, longer the freezing horizon ensures system stability whereas shorter freezing horizon ensures cheaper schedules. Sridharan et al. (1988) also evaluate the performance of freezing parameters on MRP system nervousness. Their results indicate the superiority of order based freezing method over period based in a single-level requirement planning system with respect to cost and schedule instability. Furthermore, they also show that shorter planning horizons are favored for the stability of MPS schedule.

Subsequent researchers (Sridharan and Berry 1990, Zhao and Lee 1996, Zhao and Lam 1997 and Zhao et al., 2001) build on the work by Sridharan et al. (1987, 1988). Their results corroborate the initial findings in different MPS settings, such as multi-level and stochastic demand problems. In general, the findings could be summarized as follows, longer planning horizon reduces the schedule cost but at the same time increases schedule instability, on the other hand shorter planning horizon favors stability rather than schedule cost. Schedule costs are lower with shorter frozen intervals. Schedule instability is substantially high when the frozen interval length is less than 50% of the planning horizon and there is marked improvement in stability as the freeze interval approaches the planning horizon. Re-planning at the end of frozen interval reduces both system cost and nervousness.

Blackburn and Millen (1980) evaluate the performance of lot-sizing heuristics such as Wagner-Whitin (1958), Silver-Meal (1973) and part period balancing (De Matteis et al. 1968) heuristics in single level systems. The result indicates that simpler heuristic can perform better than optimization based algorithms such as Wagner-Whitin. Blackburn and Millen (1982a, 1982b) study the performance of lot-sizing rules in multi-level MRP systems. Several other researchers have evaluated the relative performance of different lot-sizing rules and optimal methods for multi-item and multi-level MRP systems, with respect to total cost and system nervousness. (Sridharan et al. 1987, 1988, Zhao and Lam, 1997, Simpson 1999, 2001 and Zhao et al., 2001). Stadtler (2000) provide new modifications of exact algorithms for ULSP such as Wagner-Whitin so that they could perform in par with simple lot-sizing heuristics such as Silver-Meal in a rolling schedule environment.

Research Extensions

There is no literature investigating alternative coordinated lot-sizing heuristic application in a rolling horizon environment. In Chapter VII, we fill this gap by conducting a simulation study to evaluate the performance of existing uncapacitated coordinated lot-sizing heuristics and the new approaches developed in this research.

CHAPTER III

FORMULATIONS FOR COORDINATED UNCAPACITATED LOT-SIZING PROBLEM (CULSP)*

Boctor, Laporte and Renaud's (2004) mathematical programming formulations for uncapacitated problem represent a significant contribution to the coordinated lot-sizing literature. Experimental studies document the improved computational efficiency of their formulations versus the classical problem formulation. This research more fully explores the limitations and potential capabilities of the Boctor, Laporte and Renaud (BLR) formulations. We first investigate the tightness of the two BLR problem formulations and propose disaggregating the variable upper bound constraints of each formulation to more tightly constrain the joint setup variables to take on integer values in the linear programming (LP) relaxation. We then evaluate the computational efficiency of the original and revised formulations versus Robinson and Gao's (RG 1996) plant location type formulation, which is known to promote efficient solution by both general-purpose optimization software and special purpose dual-based algorithms. The performance metrics are the tightness of the LP relaxation, the proportion of the LP solutions which are integer and thus optimal for the mixed-integer programming (MIP) problem, and the computational requirements for finding optimal solutions. Each formulation is tested in a backorder and no backorder environment.

We first present the benchmark RG formulation and then the original BLR formulations (BLR1 and BLR2) and their respective tight formulations (BLR1' and BLR2'). BLR1 and BLR2 refer to DJRP2 and DJRP3, respectively, in the Boctor et al. (2004) paper.

---

* Part of this chapter is reprinted with permission from "More on 'Models and algorithms for the dynamic-demand joint replenishment problem' " by Narayanan A. and E.P. Robinson, *International Journal of Production Research*, 44 (2), 15 January 2006, 383-397.

3.1 RG problem formulation

The RG formulation exploits the hierarchical linkages among the joint setup, item setup and assignment variables to constrain the setup variables to take on a value of a value of 0 or 1 in the optimal solution of its LP relaxation. Let $i = 1, 2, ..., I$, $q = 1, 2, ..., T$, and $t = 1, 2, ..., T$, represent items, replenishment time periods, and demand time periods, respectively. Define: d_{it}, the demand for item type i in period t; S_q, the joint setup cost in time q; s_{iq}, setup cost for item i in period q; h_{it}, the per unit inventory holding cost for item i in period t; and $h_{iqt} = \sum_{r=q}^{t-1} h_{ir}$ is the per unit inventory holding cost for serving demand for item i in period t from a replenishment order in period q. The decision variables include: $Z_q = 1$ if a joint setup occurs in period q, and 0 otherwise; $Y_{iq} = 1$ if item i is replenished period q, and 0 otherwise; and X_{iqt} is the portion of demand for item i in period t that is served from a replenishment order in period q. The RG formulation of CULSP assuming backorders are not allowed is:

$$\text{(RG)} \quad \min \sum_{q=1}^{T} S_q Z_q + \sum_{i=1}^{I} \sum_{q=1}^{T} s_{it} Y_{iq} + \sum_{i=1}^{I} \sum_{q=1}^{T-1} \sum_{t=q}^{T} h_{iqt} X_{iqt} d_{it} \tag{3.1}$$

Subject to

$$\sum_{q=1}^{t} X_{iqt} = 1 \qquad (i = 1, \ldots, I, t = 1, \ldots, T) \tag{3.2}$$

$$Y_{iq} \le Z_q \qquad (i = 1, \ldots, I, q = 1, \ldots, T) \tag{3.3}$$

$$X_{iqt} \le Y_{iq} \qquad (i = 1, \ldots, I, q = 1, \ldots, T, \; t = q, \ldots, T) \tag{3.4}$$

$$Z_q = 0 \text{ or } 1 \qquad (q = 1, \ldots, T) \tag{3.5}$$

$$Y_{iq} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I, q = 1, \ldots, T) \tag{3.6}$$

$$X_{iqt} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I, q = 1, \ldots, T, t = q, \ldots, T) \tag{3.7}$$

Constraints (3.2) insure that each item's demand is satisfied in each period. Constraints (3.3) prevent an item setup from occurring unless there is a joint setup, while constraints (3.4) prohibit replenishment unless the item setup charge is incurred. Constraints (3.5), (3.6), and

(3.7) force decision variables to take on feasible solution values. The above model is extended to consider backorders by changing the summation in constraint (3.2) to consider $q = 1, \ldots, T$; and altering the last summation in the objective to include $t = 1, \ldots, T$ and $q = 1, \ldots, T$; and defining constraint sets (3.4) and (3.7) to consider X_{iqt} from t = 1 to T.

3.2 BLR1 and BLR1' problem formulations

As in Kirca (1995), BLR1 views the problem as multiple Wagner and Whitin (1958) problems that are linked by a complicating joint setup decision variable. The formulation exploits the 'exact requirements' property of Wagner and Whitin (1958) to provide a more compact model than the classical formulation discussed in Boctor et al. (2004). Assign the binary decision variable w_{iqt} the value of 1 if and only if a replenishment order in time q for item i covers all the demand from period q to period t. Define c_{iqt} as the sum of the item setup and inventory holding costs for producing item i in period q and covering its demand from period q through t, i.e., $c_{iqt} = s_{iq} + \sum_{r=q+1}^{t} (\sum_{k=q}^{r-1} h_{ik}) d_{ir}$. BLR1 is formulated as:

$$\text{(BLR1)}\quad \text{Min} \quad \sum_{q=1}^{T} S_q Z_q + \sum_{i=1}^{I} \sum_{q=1}^{T} \sum_{t=q}^{T} c_{iqt} w_{iqt} \qquad (3.8)$$

Subject to

$$\sum_{r=1}^{q} \sum_{t=q}^{T} w_{irt} = 1 \qquad (i = 1, \ldots, I, q = 1, \ldots, T) \qquad (3.9)$$

$$\sum_{i=1}^{I} \sum_{t=q}^{T} w_{iqt} \le I Z_q \qquad (q = 1, \ldots, T) \qquad (3.10)$$

$$w_{iqt} = 0 \text{ or } 1 \quad (i = 1, \ldots, I, q = 1, \ldots, T, t = 1, \ldots, T) \qquad (3.11)$$

$$Z_q = 0 \text{ or } 1 \quad (q = 1, \ldots, T) \qquad (3.12)$$

BLR1's unique and most effective modeling feature is the compact structure of constraint set (3.9), which insures that all demand is met. The weakest feature is constraint set (3.10), which attempts to force joint setups in any time period where one or more individual items are produced. Constraint (3.10) provides a weak bound on Z_q whenever fewer than I items are setup in period q due to the aggregation across items. Specifically, for given values of w_{iqt}, $Z_q = \left(\sum_{i=1}^{I}\sum_{t=q}^{T} w_{iqt}\right)/I$ in the optimal solution of the LP relaxation of BLR1. In addition, from constraint (3.9), we know that when item i is replenished in period q there is a strong tendency for $\sum_{t=q}^{T} w_{iqt} = 1$ and 0 otherwise. Hence, constraint set (3.10) forces Z_q to take on the value of 1 if and only if every item is ordered in time q. We denote BLR1 as a "weak" formulation of CULSP.

Replacing constraint set (3.10) with disaggregated constraint set (3.10') yields the tight formulation BLR1', where

$$\sum_{t=q}^{T} w_{iqt} \le Z_q \qquad (i = 1, \ldots, I, q = 1, \ldots, T) \qquad (3.10')$$

This formulation more tightly constrains the joint setup variable Z_q to take on a value of 1 when any item is replenished in time q. For the cost of I additional constraints, BLR1' provides a potentially tighter bound on Z_q and consequently a tighter lower bound on the optimal value of CULSP.

3.3 BLR2 and BLR2' problem formulations

The BLR2 formulation is similar to the RG formulation, but the item setup decision variable, Y_{iq}, is cleverly eliminated by linking the item setup costs, Y_{iq}, to their respective assignment decision variables, X_{iqq}, in the objective function. The formulation for BLR2 is:

$$\min \sum_{q=1}^{T} S_q Z_q + \sum_{i=1}^{I}\sum_{q=1}^{T} s_{iq} X_{iqq} + \sum_{i=1}^{I}\sum_{q=1}^{T-1}\sum_{t=q+1}^{T} h_{iqt} d_{it} X_{iqt} \qquad (3.13)$$

Subject to

$$\sum_{q=1}^{t} X_{iqt} = 1 \qquad (i = 1, \ldots, I, t = 1, \ldots, T) \qquad (3.14)$$

$$\sum_{i=1}^{I} X_{iqq} \le I Z_q \qquad (q = 1, \ldots, T) \qquad (3.15)$$

$$X_{iqt} \le X_{iqq} \qquad (i = 1, \ldots, I, q = 1, \ldots, T\text{-}1, t = q+1, \ldots, T) \qquad (3.16)$$

$$X_{iqt} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I, q = 1, \ldots, T, t = q, \ldots, T) \qquad (3.17)$$

$$Z_q = 0 \text{ or } 1 \qquad (q = 1, \ldots, T) \qquad (3.18)$$

Disaggregating constraint set (3.15) by item to yield constraint set (3.15') yields BLR2', the tight formulation of BLR2, where

$$X_{iqq} \leq Z_q \qquad (i = 1, \ldots, I, q = 1, \ldots, T) \qquad (3.15')$$

Similar to the RG model, the formulation can handle backorders by altering: the summation in constraint (3.14) to $q = 1, \ldots, T$; the last summation in the objective to include $t = 1, \ldots, T$ and $q = 1, \ldots, T$; the definition of t in constraint set (3.16) to range from $t = 1, \ldots, T$ and $q = 1, \ldots, T$; and the range of t in constraint set (3.17) to $t = 1, \ldots, T$.

3.4 Experimental design

The experimental design is similar to that in Erenguc (1988), Robinson and Gao (1996), and Boctor et al. (2004) but with an expanded set of test problems. The experimental factors include: the number of items, $I \in \{5, 10, 20, 40\}$, planning horizon length, $T \in \{6, 12, 18, 24, 48\}$, joint setup cost, $S_q \in \{60, 120, 480, 960\}$, and demand density, $DD \in \{0.50, 1.0\}$, where demand density is the fraction of time periods experiencing demand for an individual item.

In all problems, demand, d_{it}, is assumed to be normally distributed and it varies by item and time period. Odd numbered items have a mean demand of 50 units and a standard deviation of 20 units: even numbered items have a mean demand of 100 units and a standard deviation of 20 units. When $DD = 0.50$ only 50% of the periods for each item experience demand. Unit production costs are assumed to be equal to zero, inventory holding cost per unit per time period is \$1, and backorder costs are \$1.50 per unit per time period.

Erenguc (1988), Robinson and Gao (1996), and Silver and Kelle (1988) find that the ratio of total item setup costs to joint setup cost impacts problem solution difficulty. To study this factor, we draw the joint and item setup costs from normal distributions. The product family setup cost, S_q, is selected from the set {\$60, \$120, \$480, \$960} with a standard deviation of \$36. Within a test problem, S_q is constant across all time periods. Item setup costs, s_{iq} , are drawn from a distribution with a mean = \$60 and a standard deviation = \$18, where s_{iq} varies by item but is constant in all time periods within a test problem. The mean setup cost ratio ranges from 0.3125 to 40, where the setup cost ratio is defined as $\sum_{i=1}^{I} s_{iq} \Big/ S_q$.

Utilizing a full factorial design results in 160 combinations of factor settings each of which is solved 10 times using randomly generated demands. This yields 1600 test problem instances. The study is conducted using XpressMP Version 2003F (Xpress Optimizer Version 14.24), a state-of-the-art general-purpose optimization software package, on a personal computer running a Pentium® 4 at 1.9 gigahertz.

3.5 Experimental results

Tables 3.1 and 3.2 contain the experimental results by experimental factor for the scenarios without backorders and with backorders, respectively. Each entry in the table represents the average results for ten randomly generated test problems except for the last row which provides overall average results. The first value in each cell is associated with *DD* = 0.5 and the value in parenthesis is for *DD* = 1.0.

The last row of Table 1 indicates that although disaggregating the constraint sets adds *IT* -1 constraints to the formulations, computational times decrease substantially. Specifically, CPU times for BLR1 drop from 4.70(4.06) to 0.52(0.53) and times for BLR2 drop from 1.84 to 0.78 for *DD* =0.5. When *DD* = 1.0, a subset of the test problems of size 40 x 24 and 40 x 48 could not be solved by BLR2 within 600 CPU seconds. However, BLR2′ found solutions in average times of 1.05 and 6.28 CPU seconds, respectively. The enhanced

Table 3.1: Computational times and quality of the LP relaxation with no backorders.

Experimental Factor		RG seconds †	BLR1 seconds†	BLR1′ seconds†	BLR2 seconds†	BLR2′ Seconds†	BLR1/BLR2 LP Gap†	BLR1′/BLR2′ & RG LP Gap†
I x *T*	5 x 6	0.02 (0.10)	0.02 (0.03)	0.02 (0.02)	0.02 (0.03)	0.02 (0.02)	12.46% (4.26%)	0.000% (0.000%)
	5 x 12	0.03 (0.04)	0.04 (0.05)	0.02 (0.03)	0.05 (0.07)	0.04 (0.04)	18.18% (5.33%)	0.000% (0.000%)
	5 x 18	0.05 (0.05)	0.08 (0.10)	0.05 (0.05)	0.09 (0.14)	0.06 (0.06)	19.82% (5.68%)	0.000% (0.001%)
	5 x 24	0.08 (0.06)	0.18 (0.21)	0.08 (0.09)	0.15 (0.23)	0.09 (0.09)	20.40% (5.82%)	0.000% (0.000%)
	5 x 48	0.39 (0.04)	1.58 (1.92)	0.54 (0.60)	0.87 (1.39)	0.50 (0.55)	23.05% (5.96%)	0.000% (0.000%)
	10 x 6	0.02 (0.06)	0.03 (0.03)	0.02 (0.02)	0.03 (0.04)	0.03 (0.03)	15.87% (3.49%)	0.000% (0.000%)
	10 x 12	0.06 (0.13)	0.11 (0.08)	0.05 (0.06)	0.10 (0.11)	0.06 (0.06)	20.46% (4.08%)	0.001% (0.000%)
	10 x 18	0.10 (0.26)	0.27 (0.22)	0.09 (0.09)	0.20 (0.26)	0.12 (0.10)	21.67% (4.28%)	0.000% (0.000%)
	10 x 24	0.20 (0.09)	0.70 (0.54)	0.18 (0.17)	0.42 (0.54)	0.22 (0.19)	22.31% (4.33%)	0.002% (0.000%)
	10 x 48	1.13 (0.12)	6.33 (6.07)	1.25 (1.38)	2.82 (4.63)	1.42 (1.44)	23.14% (4.43%)	0.003% (0.003%)
	20 x 6	0.04 (0.27)	0.06 (0.08)	0.03 (0.03)	0.05 (0.07)	0.04 (0.03)	13.07% (3.72%)	0.000% (0.000%)
	20 x 12	0.09 (0.67)	0.24 (0.22)	0.07 (0.07)	0.18 (0.28)	0.10 (0.09)	16.72% (4.12%)	0.000% (0.000%)
	20 x 18	0.25 (0.10)	0.82 (0.68)	0.19 (0.17)	0.52 (0.92)	0.30 (0.21)	18.39% (4.35%)	0.002% (0.000%)
	20 x 24	0.59 (0.21)	2.15 (1.56)	0.39 (0.33)	1.22 (1.86)	0.67 (0.41)	19.43% (4.34%)	0.002% (0.000%)
	20 x 48	3.48 (0.50)	20.92 (13.75)	2.67 (2.19)	7.90 (13.07)	4.56 (2.32)	20.65% (4.41%)	0.004% (0.000%)
	40 x 6	0.06 (1.60)	0.13 (0.24)	0.05 (0.06)	0.11 (0.55)	0.06 (0.06)	11.09% (3.50%)	0.000% (0.000%)
	40 x 12	0.20 (0.59)	0.66 (1.21)	0.14 (0.14)	0.56 (4.76)	0.20 (0.19)	12.98% (3.61%)	0.000% (0.000%)
	40 x 18	0.52 (1.58)	2.13 (2.96)	0.31 (0.32)	1.30 (48.74)	0.46 (0.46)	13.93% (3.69%)	0.000% (0.000%)
	40 x 24	0.95 (2.72)	5.21 (6.31)	0.60 (0.68)	3.09 (***)	0.88 (1.05)	14.56% (3.75%)	0.000% (0.001%)
	40 x 48	4.90 (7.46)	52.29 (44.95)	3.74 (4.21)	17.25 (***)	5.77 (6.28)	15.28% (3.80%)	0.000% (0.000%)
S_q	\$ 60	0.56 (0.68)	1.02 (1.06)	0.44 (0.47)	0.82 (0.92)	0.62 (0.55)	6.92% (1.97%)	0.000% (0.000%)
	\$120	0.54 (0.69)	1.43 (1.45)	0.45 (0.48)	1.01 (1.21)	0.60 (0.57)	12.69% (3.67%)	0.000% (0.000%)
	\$480	0.58 (0.71)	4.57 (4.06)	0.51 (0.52)	1.75 (4.52)	0.69 (0.61)	24.47% (5.80%)	0.000% (0.000%)
	\$960	0.95 (1.25)	11.77 (9.67)	0.70 (0.68)	3.81 (***)	1.22 (1.01)	26.61% (5.95%)	0.002% (0.001%)
Average		0.66 (0.83)	4.70 (4.06)	0.52 (0.53)	1.84 (***)	0.78(0.68)	17.68% (4.35%)	0.001% (0.000%)

† - The first value represents problems with *DD* = 0.50 and the value within parenthesis is for *DD* = 1.0

*** - Some or all instances couldn't find and verify optimal solution in 600 CPU seconds.

performances of the revised formulations are due to the improvement in the lower bound provided by the LP relaxation of BLR1′ and BLR2′. The last two columns in Table 1 indicate average LP optimality gaps of 17.68% (4.35%) for the weak formulations compared to 0.001% (0.000%) for the tight formulations. Furthermore, the solution values of the LP relaxations of the weak formulations are integer and thus optimal in only 2.5% (2.5%) of the test problems, while 97.75% (99.15%) of the tight LP relaxation solutions are optimal. It is worth noting that the LP solutions and objective function values of BLR1=BLR2 and BLR1' =BLR2'.

The findings for the three tight formulations reveal that RG, BLR1' and BLR2' have equally tight LP relaxations and LP solutions. However, solution of RG and BLR2' requires approximately 42 and 40% more computational resources respectively than required by BLR1'. In addition, the computational times for solving RG increase with demand density, while BLR1' is invariant and BLR2' decreases.

The impact of problem size on CPU time is as expected. Increasing either I or T increases the computational requirements for all problem formulations. Similarly, increasing T increases all LP gaps, but increasing I provides mixed results. Finally, confirming the results in Erenguc (1988) and Robinson and Gao (1996), longer solution times and larger LP gaps are associated with higher levels of the joint set-up cost, S_q.

Table 3.2 contains the experimental results for the tight formulations BLR2' and RG when backorders are allowed. BLR1' is not included in the analysis since the formulation is not capable of handling backorders. Assuming backorder are allowed, RG is the most efficient formulation with average solution times of 1.48(1.44) versus 1.73(1.92) for BLR2'. When compared with the no backorder scenario, solution times approximately double for problems with backorders for both RG and BLR2' due to the increased number of assignment variables, X_{iqt}. Solutions times for the backorder case increase with higher values of I, T and S_q.

General conclusions from the experiments reveal that the three tight formulations can effectively solve large-scale CULSP problems. When backorders are not allowed, the ranking of the formulations in terms of computational efficiency is BLR1', BLR2', RG,

Table 3.2: Computational times and quality of the LP relaxation: with backorders.

Experimental Factor		RG seconds †	BLR2′ seconds†	BLR2′ & RG LP Gap†
I x T	5 x 6	0.03 (0.03)	0.02 (0.03)	0.000% (0.000%)
	5 x 12	0.05 (0.06)	0.05 (0.07)	0.000% (0.005%)
	5 x 18	0.10 (0.12)	0.13 (0.17)	0.000% (0.003%)
	5 x 24	0.17 (0.17)	0.18 (0.21)	0.000% (0.002%)
	5 x 48	0.79 (0.76)	1.10 (1.22)	0.000% (0.001%)
	10 x 6	0.03 (0.04)	0.03 (0.03)	0.000% (0.000%)
	10 x 12	0.11 (0.10)	0.11 (0.11)	0.003% (0.000%)
	10 x 18	0.29 (0.24)	0.30 (0.25)	0.002% (0.000%)
	10 x 24	0.37 (0.42)	0.41 (0.47)	0.000% (0.002%)
	10 x 48	1.71 (1.56)	2.28 (2.31)	0.000% (0.000%)
	20 x 6	0.06 (0.06)	0.06 (0.06)	0.000% (0.000%)
	20 x 12	0.24 (0.21)	0.21 (0.21)	0.000% (0.000%)
	20 x 18	0.92 (0.55)	0.81 (0.59)	0.002% (0.001%)
	20 x 24	0.95 (1.22)	0.96 (1.32)	0.001% (0.004%)
	20 x 48	8.11 (9.10)	8.13 (11.29)	0.002% (0.005%)
	40 x 6	0.11 (0.11)	0.10 (0.10)	0.000% (0.000%)
	40 x 12	0.45 (0.43)	0.41 (0.46)	0.000% (0.000%)
	40 x 18	0.98 (0.84)	0.96 (1.00)	0.000% (0.000%)
	40 x 24	1.91 (2.47)	1.97 (2.41)	0.000% (0.001%)
	40 x 48	12.25 (10.28)	16.32 (16.24)	0.000% (0.000%)
S_q	$ 60	1.08 (1.01)	1.43 (1.42)	0.000% (0.000%)
	$120	1.22 (1.00)	1.60 (1.44)	0.000% (0.000%)
	$480	1.63 (1.32)	1.79 (1.78)	0.002% (0.003%)
	$960	1.99 (2.42)	2.09 (3.07)	0.000% (0.002%)
Average		1.48 (1.44)	1.73(1.92)	0.000%(0.001%)

† - The first value represents problems with DD = 0.50 and the value within parenthesis is for DD = 1.0

BLR1 and BLR2. However, in a backorder environment the rankings are RG and BLR2'. The broader findings of the study are threefold. First, the research clearly illustrates the importance of tightly constraining set-up decision variables in order to improve the quality of the LP relaxation and reduce computational time when applying general-purpose software. While this modeling feature is generally well known in the literature (Denizel et al. 1996), the specific impact on the CULSP problem formulation is documented in this research. Second, the BLR1', BLR2', and RG problem formulations have identical LP solutions and LP objective function values, varying capabilities in representing problem features (e.g. backorders), and different computational efficiencies. These results encourage the development and evaluation of other formulations for the CULSP and other dynamic-demand lot-sizing problem classes in an attempt to discover a broader set of ‘tight’ formulations and the unique characteristics of each. Finally, the new BLR1' and BLR2' formulations provide additional research opportunities. Where Robinson and Gao (1996) exploit the special mathematical structure of their formulation to obtain an efficient dual-based optimization procedure, BLR1' and BLR2' provide new mathematical structures upon which other computationally efficient algorithms can potentially be built.

CHAPTER IV

HEURISTICS FOR COORDINATED UNCAPACITATED LOT-SIZING PROBLEM (CULSP) *

Boctor et al. (2004) evaluate the performance of several well-known and a new perturbation metaheuristic (PM) for solving the coordinated uncapacitated lot-sizing problem (CULSP). The heuristics include the Fogarty and Barringer (1987) heuristic with the Silver and Kelle (1988) improvement procedure (FB-SK), Atkins and Iyogun's (1988) extension of the Silver-Meal (1973) heuristic, Iyogun's (1991) modification of the part-period balancing method, and Federgruen and Tzur's (1994) greedy-add heuristic. Experimental results indicate the superiority of the FB-SK heuristic over those tested. However, the perturbation metaheuristic, which utilizes the FB-SK heuristic, finds the highest quality solutions.

This research develops and evaluates two forward-pass heuristics, a two-phase heuristic and a simulated annealing metaheuristic (SAM) for the CULSP. The heuristics extend the concepts proposed by Eisenhut (1975), Lambrecht and Vanderveken (1979), Dixon and Silver (1981), and Dogramaci et al. (1981) for the multi-item capacitated lot-size problem (MCLSP) to solve the CULSP. While the item replenishment decisions are economically coupled by shared capacitated resource in the MCLSP, the CULSP optimal item replenishment schedules are coupled by the product family's joint setup costs. The interplay between joint and individual item fixed costs in the CULSP results in a more generalized cost function than that for the MCLSP, which increases algorithm complexity and necessitates more generalized algorithms. SAM utilizes the two-phase heuristic to generate an initial solution and as the neighborhood search procedure.

---

* Part of this chapter is reprinted with permission from "Effective heuristics for the dynamic demand joint replenishment problem " by Robinson E.P., A. Narayanan and L-L. Gao, *Journal of Operational Research Society* (forthcoming; published online : April 19th, 2006). © 2006 by Operational Research Society Ltd. This article could be accessed online at http://www.palgrave-journals.com/jors

4.1 Forward-pass heuristics

A forward-pass heuristic attempts to reduce total schedule costs by shifting replenishment orders into earlier time periods when the setup cost saving exceeds the increase in inventory holding costs. The heuristic builds a production schedule beginning in period 1 and then rolls forward through time. Two forward-pass heuristics are proposed – one applies a modified Eisenhut (1975) decision criterion and the other uses a modified Lambrecht and Vanderveken (LV) (1979) decision criterion. We first develop the decision criterion for selecting which order to reschedule into period 1 and then outline the steps of the forward-pass heuristics. For presentation clarity, we assume the setup and holding cost parameters are constant over time. Extensions to model time varying costs are straight-forward.

For the single-item dynamic demand lot-sizing problem, Silver and Meal (1973) define $C_i(t)$ as the average cost per time period for satisfying the demand for item i from time 1 through time t, where,

$$C_i(t) = (s_i + I_i(t))/t \qquad t \leq T, \tag{4.1}$$

s_i is the item setup cost, and $I_i(t) = h_i \sum_{j=1}^{t} (j-1) d_{ij}$ is the inventory holding cost.

We extend the concept to the CULSP by defining $C(t)$ as the cost per time period for serving all item demand through time t from a replenishment in time period 1, where

$$C(t) = S/t + \sum_{i}^{I} C_i(t). \tag{4.2}$$

Modified Eisenhut Decision Criterion

Eisenhut (1975) proposes using the coefficient, $C_i^{'}(t)$, provided by an approximation of the first derivative of equation (4.1) as a decision criterion for lot-size aggregation, where

$$C_i^{'}(t) \approx \left(I_i(t) - s_i\right)/t^2 \tag{4.3}$$

If $C_i^{'}(t) < 0$, total cost can be reduced by rescheduling item i's replenishment quantities through period t into period 1. Hence, at the optimal t, inventory costs approach but are less than the setup cost, which is analogous to the decision criterion applied in the part period balancing algorithm. We restate equation (4.3) on a per unit basis as

$$U_i(t) = \left(s_i - I_i(t)\right) / t^2 d_{it} \tag{4.4}$$

where $U_i(t) > 0$ indicates that costs can be reduced by shifting the order quantities for item i through period t into period 1.

Extending Eisenhut's decision criterion to the CULSP yields the following per unit savings coefficient for rescheduling all of the product family's orders through period t into period 1

$$U(t) = \left(S/t^2 + \sum_i U_i(t) d_{it} \right) \Big/ \sum_i d_{it} \tag{4.5}$$

$U(t) > 0$ implies that costs can be reduced by rescheduling all of the product family's orders through period t into period 1. During the calculation of (4.4) if there is no setup for item i in period 1, then $s_i = 0$ since we are not saving a setup. Similarly, $S = 0$ in (4.5) if a joint setup is not scheduled in period 1.

Modified LV Decision Criterion

LV recognizes that the minimum cost lot-size in period 1 for item i will satisfy demand over t periods if $C_i(t) < C_i(t-1)$ and $C_i(t) < C_i(t+1)$. LV, as do Dogramaci et al. (1981), uses the following coefficient to denote the cost savings derived from rescheduling item i's orders through time t into the period 1 lot-size

$$V_i(t) = \left(s_i + I_i(t-1) - h_i(t-1)^2 d_{it}\right) / t(t-1) d_{it} \tag{4.6}$$

The analogous savings coefficient for the CULSP is

$$V(t) = \left(S/t(t-1) + \sum_i V_i(t) d_{it} \right) \Big/ \sum_i d_{it} \tag{4.7}$$

Forward-Pass Heuristic

The difference in the two forward-pass heuristics rests solely on the decision criterion employed. We briefly outline the heuristic using the Eisenhut decision criterion.

Step 1. *Initialization.* Initialize the order schedule matrix, ***A***, where $a_{it} = d_{it}$ is the order quantity for item *i* in time period *t*. Set the current period as $r = 1$.

Step 2. *Calculation of savings matrix.* For the current order schedule matrix, construct the item cost savings matrix, $\boldsymbol{U} = \{U_i(t) \mid \forall i, t\}$, and the product family cost savings coefficients, $U(t)$ for all *t.*

Step 3. *Forward-pass.* Include all requirements for time period 1 in the replenishment schedule for period 1. Next, schedule the order(s) associated with Max$\{U_i(t) > 0, U(t) > 0\}$ into period 1 and update the order schedule matrix accordingly. Continue rescheduling orders into period 1 until no further savings are possible.

Step 4. *Roll.* If $r = T$, stop. Otherwise set $r = r + 1$. Re-index period $r + 1$ as new period 1 for establishing the updated order schedule matrix. Go to *Step 2*.

4.2 Two-phase heuristic

In the forward-pass heuristic, an earlier order decision may adversely impact the quality of later decisions without any recourse for future schedule adjustments. To overcome this limitation, Dogramaci et al. (1981) propose solving the MCLSP by building orders in the time periods offering the greatest cost reduction, not necessarily from period 1. We extend this concept and develop a two-phase heuristic for solving the CULSP. The heuristic is based on two savings coefficients. The savings coefficient, $C_i(t', t)$, for shifting item *i'*s order from period *t* into setup period t', ($t' < t$), is

$$C_i(t', t) = (Y_{it'} - 1)s_{it'} + s_{it} - I_i(t', t) \tag{4.8}$$

where $Y_{it'} = 1$ if an order is scheduled for item *i* in period t', $s_{it'}$ is the item's setup cost in time t', and the inventory carrying cost from time t' to time *t* is $I_i(t', t) = (t - t')h_i a_{it}$. The savings coefficient, $C(t', t)$, for re-scheduling the product family from *t* into t' is

$$C(t', t) = (Z_{t'} - 1)S_{t'} + S_t + \sum_{i \in I} C_i(t', t) \tag{4.9}$$

where, $Z_{t'} = 1$ if an order is scheduled for the product family in period t' and $S_{t'}$ is the joint family setup cost in period t'. The heuristic begins by initializing the order schedule matrix by setting $a_{it} = d_{it}$ for every item and time period. Next, the two-phase procedure is implemented as follows:

Phase I: Left-Shift Orders Earlier in Time

Phase I reduces schedule cost by rescheduling orders into earlier time periods, i.e., left-shifting orders, when economically attractive.

Step 1. Compute savings : For the current order schedule matrix, calculate

$C_i(t^{'},t)$ and $C(t^{'},t)$ for all *i, t*, and $t^{'} < t$. If all $C_i(t^{'},t) \leq 0$ and $C(t^{'},t) \leq 0$, stop.

Step 2. Left-Shift. Select the Maximum { $C_i(t^{'},t)$, $C(t^{'},t)$ } for all *i*, *t*, and $t^{'} < t$ for left-shifting into time period $t^{'}$ and update the order schedule matrix. Go to *Step 1.*

Phase II: Right-Shift Orders Later in Time

Phase II reduces costs by rescheduling orders into later time periods, thereby lowering inventory costs. Dogramaci et al.'s (1981) right-shift procedure examines adjacent setup periods and attempts to shift order quantities from the earlier period to the later period if inventory costs can be reduced. However, during computational testing we observed several instances were costs could be further reduced by right-shifting orders into periods without a positive order quantity. This is due to unique joint fixed cost structure of CULSP. Consequently, we propose a more comprehensive procedure, which evaluates the economic impact of right-shifting orders into time *t* irrespective of whether there is currently an item order scheduled in time *t* or not.

The maximum quantity of item *i* that can be shifted from period *t'* to *t*, $v_{it'}$, is:

$$v_{it'} = a_{it'} - d_{it'} - (\textit{net requirement for item i from period } t'+1 \textit{ to } t-1) \qquad (4.10)$$

The procedures for calculating the net requirement for item *i* from period *t'* +1 to *t* -1 is detailed in the Appendix A.

For $v_{it'} > 0$, the resulting cost savings is $H_i(t',t) = h_i v_{it'}(t-t') - s_{it}(1-Y_{it}) + s_{it'}X_{it'}$, where $X_{it'}$ = 1 if the full lot-size is shifted and 0, otherwise. For $v_{it'} = 0$, $H_i(t',t) = 0$. The cost savings for right-shifting the entire product family from t' to t is

$$H(t',t) = S_{t'} - S_t(1 - Z_t) + \sum_{i=1}^{I} H_i(t',t) .$$

Phase II consists of four steps.

Step 1. *Initialize.* Set $t = T$.

Step 2. *Calculate savings.* For $i = 1, 2, \ldots, I$ and $t' = 1, 2, \ldots, t-1$ compute $v_{it'}$, $H_i(t',t)$ and $H(t',t)$. If all $H_i(t',t) \le 0$ and $H(t',t) \le 0$ go to *Step 4.*

Step 3. *Right-shift.* Select the Maximum { $H_i(t',t)$, $H(t',t)$ } for all i and t' for right-shifting into period t and update order schedule matrix. Go to *Step 2.*

Step 4. *Backtrack.* If $t = 1$ stop, otherwise set $t = t - 1$ and go to *Step 2*.

4.3 Simulated annealing metaheuristic (SAM)

Traditional heuristics, such as the two-phase and FB-SK heuristics, tend to converge to a local optimum within a restricted neighborhood of the problem's state space leaving neighborhoods containing potentially improved solutions unexplored. In such cases, metaheuristics, which are generalized procedures that orchestrate the search process to escape from local optima and perform a more robust search of the problem's feasible region (Hillier and Lieberman 2005), are attractive. A variety of metaheuristic techniques, including simulated annealing, tabu search and genetic algorithms, are implemented to solve difficult combinatorial problems. Boctor et al. (2004) propose a perturbation metaheuristic for CULSP. This research develops a simulated annealing metaheuristic for the problem.

The simulated annealing metaheuristic, introduced by Kirkpatrick et al. (1983), mimics physical annealing processes to escape from local optima solutions. The SAM escapes entrapment at local optima by applying transition probabilities, which define the probability of moving from the current solution to a neighboring candidate solution. A move's transition probability depends on the difference between the objective values of the current solution, $\hat{C}$, and the candidate (or neighborhood) solution, C', and the current temperature, θ_n . The transition probability, *Pr*, is:

$$Pr = Min\left[1, e^{(\hat{C}-C')/\theta_n}\right] \quad (4.11)$$

If C' < $\hat{C}$, the transition probability is 1 and we replace the current solution with the candidate solution and set $\hat{C} = C'$. When $C' > \hat{C}$, the candidate solution, C', is accepted with probability *Pr*.

The metaheuristic is initiated with a relatively high temperature, which increases the probability of accepting candidate solutions and escaping from the local optima early in the algorithm's implementation. However as the search continues, the temperature is reduced making it less likely to replace the current solution with an inferior candidate in the final iterations of the search. The algorithm variant applied in this research holds the temperature constant for three iterations before it is reduced. This increases the probability of jumping to new neighborhoods during each cooling cycle versus an algorithm in which the temperature is decreased after each transition opportunity.

The cooling schedule used in this experiment is,

$$\theta_n = \theta_{n-1} * 0.8 \quad (4.12)$$

where θ_{n-1} is the temperature from previous iteration. The search stops when the temperature reaches 1 or when the objective function value of the best found solution does not improve for ϕ successive iterations. During preliminary experiments, several values of ϕ and θ_o were tested with $\phi = 50$ and $\theta_o = 1000$ providing the best results. The steps of the SAM follow.

Step 1: Initialization. Set $n = 0$ and $\theta_o = 1000$. Apply the two-phase heuristic to obtain an initial problem solution. Set $\hat{C}$ equal to the objective function value of the heuristic solution. Set the best known solution $C_B = \hat{C}$ and the iteration counter, *count* = 1.

Step 2: Neighborhood generation. Randomly choose a value of $q \in \{1, 2, \ldots, T\}$. Alter the status of the joint setup decision variable in period q and reassign orders as necessary.

Step 3: Neighborhood Improvement. Attempt to improve this solution by applying the two-phase heuristic, while maintaining the status of the perturbed family setup in period

q. The resulting solution provides a new candidate solution and a candidate objective function value C'.

Step 4: Neighborhood search. Compute the transition probability and replace the current solution with the candidate solution if the probability Pr is greater than or equal to a randomly generated number between [0, 1]; otherwise reject the candidate solution. If $\hat{C} < C_B$ update the best known solution and set $C_B = \hat{C}$ and reset $count = 1$. Otherwise, set $count = count + 1$. Repeat Steps 2 - 4 three times.

Step 5: Update cooling temperature: Set $n = n + 1$. Update the temperature using (4.12).

Step 6: Termination. If $count \geq \phi$ or $\theta_n \leq 1$, stop and report the best found solution. Otherwise, go to step *2*.

4.4 Summary of benchmark heuristics for CULSP

This heuristics proposed in this research for solving the CULSP are compared against the FB-SK, Robinson and Gao's (RG) heuristic and perturbation metaheuristic. The FB-SK heuristic is the most effective standalone heuristic emerging from experiments in Boctor et al. (2004), while perturbation metaheuristic improved upon the performance of FB-SK procedure. The RG heuristic, while promising, has not been previously evaluated against other heuristics reported in the literature. The section briefly describes the FB-SK, perturbation and RG heuristics.

FB-SK heuristic

The FB-SK combines the Fogarty and Barringer (1987) heuristic with an improvement procedure suggested by Silver and Kelle (1988). The Fogarty and Barringer (FB) heuristic employ dynamic programming logic analogous to the Wagner and Whitin (1958) procedure. The method recognizes that the economic impact of the joint replenishment cost is equally shared across items only if all items are ordered in the same time period. Following this strategy, when a replenishment is made, it should exactly cover all item demands until the next replenishment period. Hence, the solution is primarily characterized by the periods experiencing a joint replenishment. Boctor et al. (2004) summarize the FB dynamic programming heuristic as follows where, f_t is the total replenishment cost for the first t

periods and c_{qt} is the total cost for replenishment in period *q*, which covers the demand for all items through period *t*.

$$f_t = \min\{f_{q-1} + c_{qt}\} \text{ for } t = 2, \ldots, T \text{ and } q \leq T$$

$$f_1 = S_1 Z_1 + \sum_{i=1}^{I} s_{i1} Y_{i1}$$

$$c_{qt} = S_q Z_q + \sum_{i=1}^{I} \left[s_{iq} Y_{iq} + \sum_{r=q+1}^{t} (r-q) h_i d_{ir} \right]$$

$$Z_q = 1 \text{ if } \sum_{i=1}^{I} \sum_{r=q}^{t} d_{ir} > 0 \text{ and } Z_q = 0 \text{ otherwise}$$

$$Y_{iq} = 1 \text{ if } \sum_{r=q}^{t} d_{ir} > 0 \text{ and } Y_{iq} = 0 \text{ otherwise}$$

The dynamic program starts by calculating f_1 and then iteratively calculates f_t for all $t = 2,\ldots, T$. The order schedule with the minimum value for f_T is the solution of the FB heuristic.

Silver and Kelle (1988) note that when the item setup cost is high relative to its holding cost, replenishing the item in only a subset of the scheduled replenishment periods may reduce costs. The SK improvement procedure successively considers each item in each scheduled replenishment period to see if cost savings are possible by shifting the item's production into an earlier scheduled replenishment period. The combined FB-SK procedure is shown to be the most effective known heuristic by Boctor et al. (2004).

RG heuristic

The RG heuristic applies a dual-ascent heuristic to solve the dual of the LP relaxation of the Robinson and Gao (1996) problem formulation. A primal feasible solution for CULSP is constructed from the dual LP solution using complementary slackness conditions. A dual adjustment procedure is then applied as necessary in an attempt to reduce any complementary slackness violations and improve upon the incumbent solution. The RG heuristic stops when the dual adjustment procedure fails to find improved solutions.

Perturbation metaheuristic (PM)

The perturbation metaheuristic contains four basic components: (1) the FB-SK heuristic, (2) a perturbation procedure to jump to other regions of the solution state space, (3) a greedy drop heuristic for eliminating joint setups, and (4) the SK improvement procedure. The PM procedure is as follows.

Step 1. Initial solution: Obtain an initial feasible solution using the FB-SK heuristic.

Step 2. Solution perturbation: Run the following procedure λ times. Randomly generate a time period, q ε {1, 2, …,T}. Alter the status of the joint setup decision variable in period q and reassign orders as necessary.

Step 3. Solution improvement: Apply a greedy drop procedure to eliminate any joint setup whose removal results in a cost savings. When no further cost savings are possible, run the SK improvement procedure. If the order schedule has lower cost than the current best solution, update the best known solution. Next, go to Step 2 until the stopping criterion is reached.

Stopping criterion: Steps 2 and 3 are applied until the best known solution doesn't improve for ϕ consecutive iterations.

As in Boctor et al. (2004) we implement the algorithm with $\lambda = 3$ and $\phi = 6T$.

4.5 Experimental design

The experimental design is modeled after Narayanan and Robinson (2006), which extends the test problem sets in Erenguc (1988), Robinson and Gao (1996) and Boctor et al. (2004). The experimental factors include the joint setup cost S_t, number of items *I,* planning horizon length *T,* and demand density *DD*.

The number of items is represented at four levels where, $I \in \{5, 10, 20, 40\}$ and the planning horizon length is taken from the set $T \in \{6, 12, 18, 24, 48\}$. Item demand per time period is randomly generated from a normal distribution with a mean of 50 units and a standard deviation of 20 units for odd-numbered items and a mean of 100 units and a standard deviation of 20 units for even-numbered items. $DD \in \{50\%, 100\%\}$, where DD = 50% indicates that each item has a demand occurrence in 50% of the time periods, with the exception of the first period, which has positive demand for all items. When DD = 100%, each item has positive demand in all time periods. The inventory holding cost is $1.00 per unit per period. Unit production costs are assumed to be constant in all time periods and hence set to zero in the experiments.

Erenguc (1988), Silver and Kelle (1988), Joneja (1990) and Robinson and Gao (1996) found that relatively higher joint setup costs increase the problem difficulty. Consequently, we study a variety of (item setup cost)/(joint setup cost) ratios. The item setup costs are drawn from a normal distribution with a mean of $60 and a standard deviation of $18. The setup cost varies across items, but is constant in all time periods for a specified item within a test problem. The joint setup cost, which is also constant for all time periods within a test problem, is drawn from a normal distribution with a mean $S_t \in \{\$60, \$120, \$480, \$960\}$ and a standard deviation of $36. This provides a mean setup cost ratio, $\sum_i s_{it} / S_t$, ranging from 0.3125 to 40.

We utilize a full factorial design with 160 different combinations of factor settings. For each combination of factors, ten test problems are randomly generated. Each test problem is solved by the five heuristics, two metaheuristics, and Xpress-MP version 2003F (Xpress Optimizer v14.24). Xpress-MP provides optimal solutions using the RG formulation described in Chapter III for benchmarking. The experiments are conducted on a personal computer running a Pentium® 4 processor at 1.9 GHz with the Windows 2000 Professional operating system.

4.6 Experimental results

The following notation is used to present the experimental results.

- FP-E: Forward-pass heuristic using the Eisenhut decision criterion
- FP-LV: Forward-pass heuristic using the LV decision criterion
- FB-SK: Fogarty-Barringer heuristic with the Silver-Kelle procedure
- RG: Robinson and Gao dual ascent and adjustment heuristic
- TP: Two-phase heuristic
- PM: Perturbation metaheuristic initialized by FB-SK

SAM: Simulated annealing metaheuristic initialized by TP

The results are summarized by demand density in Table 4.1, where each cell is associated with 800 test problems. Overall, the forward-pass heuristics perform slightly better when demand density is 100% versus 50%. However, the other procedures perform substantially better when demand density is 50%.

The FP-E heuristic performs slightly better than the FP-LV heuristic based on the average optimality gaps and the number of optimal solutions found. Preliminary computational experiments revealed the FP-E and FP-LV provide higher quality solutions than the Atkins and Iyogun (1988) and Iyogun (1991) forward-pass heuristics tested in Boctor et al. (2004). Hence, they were not included in the experiments.

Table 4.1: Experimental results for CULSP heuristic procedures

	Average Optimality Gap*		Maximum Optimality Gap*		Std. Dev of Optimality Gap*		No. of Optimal Solutions*	
	50% DD†	100% DD†	50% DD†	100% DD†	50% DD†	100% DD†	50% DD†	100% DD†
Stand-alone heuristics								
FP-E	1.64%	1.21%	14.11%	12.64%	1.90%	1.35%	75	94
FP-LV	1.76%	1.30%	14.11%	11.52%	2.12%	1.15%	40	30
FB-SK	0.51%	1.33%	12.43%	12.64%	0.85%	1.65%	257	245
RG	0.93%	2.97%	16.44%	35.43%	1.89%	6.06%	452	273
TP	0.34%	0.78%	8.20%	9.12%	0.87%	1.18%	457	145
Metaheuristics								
PM	0.46%	1.28%	6.83%	9.56%	0.69%	1.56%	271	246
SAM	0.08%	0.33%	3.46%	3.95%	0.27%	0.56%	579	261

† Each cell represents the average results for 800 test problems
* Optimality gap = (heuristic objective value – optimal objective value)/ optimal objective value

The performance of the TP heuristic is better than the FP-E, FP-LV, FB-SK and RG stand-alone heuristics based on the average, maximum, and standard deviation of the solution optimality gaps and the number of optimal solutions found for the 50% demand density. The FB-SK is the second best performing heuristic. A t-test† shows that there is

† $H_o : \mu_{FB\_SK} - \mu_{TP} \leq 0$; $H_a : \mu_{FB\_SK} - \mu_{TP} > 0$. Using the averages and standard deviations listed in Table 4.1, we compute the value of *t'* for *t*-test. t' > t (*p*-value : 0.001) for both demand densities, hence we reject H_0.

significant (p-value 0.001) evidence that TP performs better than FB-SK. While the RG heuristic found the most optimal solutions, RG solutions exhibit by far the largest standard deviation and maximum optimality gaps, especially when demand density is 100%. This highly variable performance makes RG the least suitable for industry application.

The SAM outperforms all the other procedures finding 840 optimal solutions for the 1600 test problems. The average, maximum, and standard deviations of the optimality gaps are extremely tight when compared to the other procedures and considering the difficulty of the combinatorial problems. The SAM is particularly effective on the 50% demand density problems.

The SAM improves the average TP optimality gap by 77.7% when *DD* = 50% and by 57.7% when *DD* = 100%. The analogous reduction in the FB-SK gaps by the PM is 9.8% and 3.8%, respectively. To better understand the relative effectiveness of the SAM versus the PM, we initialized the SAM using the FB-SK solution and initialized PM using the TP heuristic, and then ran both metaheuristics to evaluate the improvement in the initial solutions. The SAM improves upon the FB-SK initial solution on average by 72%, while PM only improves the initial TP solution by 40%. These results indicate the superiority of the SAM over the PM for this problem class. A t-test[‡] also shows that there is significant (p-value 0.001) evidence that SAM performs better than PM.

Table 4.2 presents the results by factor and setup cost ratio. The optimality gap of the best performing procedure is given in bold face for each factor setting. Performance tends to improve with an increase in the number of items for the FP-LV, RG, and TP heuristics, while the results are mixed for the other procedures. Increasing the length of the planning horizon tends to increase the optimality gaps for the RG, PM and SAM procedures, while the TP solution quality improves. The other procedures' solutions do not reveal a discernable pattern.

The FB-SK and PM procedures show declining solution quality with lower joint setup costs and higher setup cost ratios (i.e., lower joint setup costs relative to item setup

---

[‡] $H_o : \mu_{PM} - \mu_{SAM} \leq 0$; $H_a : \mu_{PM} - \mu_{SAM} > 0$. Using the averages and standard deviations listed in Table 4.1, we compute the value of t' for t-test. $t' > t$ (p-value : 0.001) for both demand densities, hence we reject H_0.

Table 4.2: Summary of average optimality gaps by experimental factors*

Experimental Factor		FP-E	FP-LV	FB-SK	RG	TP	PM	SAM
Items	5	2.79(1.61)	3.52(1.87)	0.62(1.05)	1.11(5.04)	0.60(1.00)	0.42(1.01)	**0.12(0.39)**
	10	1.69(0.96)	1.55(1.09)	0.56(1.29)	1.24(4.51)	0.54(1.16)	0.55(1.22)	**0.10(0.52)**
	20	1.24(1.03)	1.17(1.13)	0.35(1.19)	0.90(1.94)	0.18(0.56)	0.35(1.17)	**0.07(0.18)**
	40	0.84(1.26)	0.79(1.11)	0.51(1.79)	0.47(0.40)	0.04(0.42)	0.51(1.72)	**0.02(0.23)**
Joint Setup Cost	$60	1.15(1.55)	0.99(1.09)	0.75(2.08)	0.04(0.17)	0.04(0.23)	0.75(2.02)	**0.02(0.13)**
	$120	1.47(1.56)	1.32(1.14)	0.77(1.85)	0.11(0.27)	0.09(0.18)	0.76(1.78)	**0.03(0.12)**
	$480	1.51(0.94)	1.95(1.26)	0.25(1.10)	1.12(4.21)	0.39(0.97)	0.22(1.04)	**0.05(0.43)**
	$960	2.43(0.81)	2.76(1.71)	0.27(0.29)	2.45(7.23)	0.84(1.75)	**0.09(0.28)**	0.21(0.63)
Periods	6	1.49(0.94)	1.66(1.19)	0.56(1.20)	0.30(0.40)	0.36(0.91)	0.44(0.98)	**0.00(0.03)**
	12	1.57(1.20)	1.77(1.31)	0.46(1.32)	0.87(2.32)	0.34(0.77)	0.43(1.29)	**0.01(0.25)**
	18	1.83(1.28)	1.80(1.37)	0.50(1.36)	1.03(3.47)	0.35(0.77)	0.46(1.36)	**0.04(0.39)**
	24	1.67(1.31)	1.81(1.33)	0.54(1.39)	1.13(4.04)	0.33(0.74)	0.47(1.39)	**0.12(0.41)**
	48	1.62(1.34)	1.75(1.31)	0.48(1.38)	1.32(4.63)	0.32(0.73)	0.48(1.38)	**0.22(0.56)**
Setup Cost Ratio	0.3125	3.83(1.18)	5.55(2.62)	0.76(0.00)	2.25(8.74)	1.28(1.87)	**0.07(0.00)**	0.36(0.47)
	0.625	2.49(0.25)	3.37(1.11)	0.11(0.01)	2.32(11.76)	1.17(2.18)	**0.05(0.01)**	0.16(1.09)
	1.25	1.68(0.32)	1.67(1.22)	0.07(0.02)	2.41(5.78)	0.55(1.29)	**0.07(0.02)**	0.14(0.32)
	2.5	1.92(1.84)	1.75(1.89)	0.33(1.16)	0.95(0.75)	0.07(0.55)	0.33(1.15)	**0.03(0.23)**
	5	1.48(1.92)	1.15(1.26)	0.81(2.89)	0.07(0.17)	0.12(0.48)	0.80(2.75)	**0.04(0.24)**
	10	0.87(1.33)	0.86(0.82)	0.93(2.38)	0.03(0.18)	0.06(0.18)	0.93(2.24)	**0.03(0.14)**
	20	0.69(1.04)	0.66(0.82)	0.54(1.44)	0.01(0.22)	0.01(0.10)	0.54(1.43)	**0.01(0.09)**
	40	0.70(1.09)	0.72(0.81)	0.66(1.43)	0.01(0.18)	0.01(0.09)	0.65(1.41)	**0.01(0.08)**

*Table entries are for *DD* = 50%(*DD*= 100%)

costs). This result is as expected since the FB-SK and PM procedures are anchored on the assumptions that every item is setup in each joint replenishment period, which is more characteristic of optimal solutions for problems with higher joint setup costs and lower setup cost ratios. These results extend those in Boctor et al. (2004) to fully clarify the limitations of FB-based heuristics. The FP-LV, RG, TP, and SAM exhibit an opposite tendency where improved solution quality is associated with lower joint setup costs and higher setup cost ratios. As illustrated in the table, the PM outperforms the SAM at the highest joint setup cost setting and the three lowest setup cost ratios where the joint setup costs dominates the item setup costs.

A final observation from Table 4.2 relates to the relative improvement of the metaheuristics on their initial starting solutions. For the PM, the FB-SK solutions are noticeably improved when $I = 5$, $S_{t'} = \$960$, $T = 6$, and the setup cost ratio = 0.3125. At all other factor settings, the PM provides very little, if any, improvement over the FB-SK starting solution. In contrast, the SAM provides a substantially improved solution at every factor level.

The findings indicate that the forward-pass heuristics (FP-E and FP-LV) are capable of finding high quality solutions averaging approximately 1.42% and 1.53% from optimality, respectively. However, the new two-phase heuristic finds solutions with an average 0.56% optimality gap, which improves upon the 0.92% optimality gap associated with the FB-SK heuristic, the best known procedure in the prior literature. The Simulated Annealing Metaheuristic with a 0.2% optimality gap also improved upon the 0.87% optimality gap associated with the Perturbation Metaheuristic reported in earlier research.

Computational times for all of the stand alone heuristics average less than 0.05 CPU seconds. The SAM averages 0.18 CPU seconds with a maximum of 1.8 CPU seconds. Considering that SAM and TP solutions average 0.2% and 0.56% from optimality, they are both highly efficient and effective procedures for solving the CULSP.

CHAPTER V

FORMULATIONS FOR COORDINATED CAPACITATED LOT-SIZING PROBLEM (CCLSP)

In this chapter we describe six mathematical formulations for the coordinated capacitated lot-sizing problem (CCLSP) and evaluate their relative performance in general purpose optimization software. The formulations include those proposed by Federgruen et al. (FMT, 2004), Robinson and Lawrence (RL 2004), Gao and Robinson (GR 2004) and new extensions of GR and BLR1' formulations. The BLR1' refer to most efficient tight formulation for the uncapacitated problem in Narayanan and Robinson (2006).

5.1 FMT problem formulation

Federgruen et al. (2004) extend their network formulation for CULSP (Federgruen et al. 1994) to represent the capacitated problem. For $i = 1, \ldots, I$ and $t = 1, \ldots, T$, define, d_{it}, the demand for the item i in period t; s_{it}, setup cost for item i in period t; S_t , joint (family) setup cost in period t ; c_{it}, variable per unit order cost for item i in period t; h_{it}, the per unit inventory holding cost for item i in period t and P_t, rated capacity for period t. The aggregate demand in period t, is represented by D_t, where $D_t = \sum_{i=1}^{I} d_{it}$. Let I_t^0 be the minimum aggregate inventory at the end of period t, such that there exists a feasible replenishment schedule for the planning horizon from $t+1, \ldots, T$. These values are calculated by recursion starting from end period T and moving backwards by using the following definition, $I_t^0 = \left(D_{t+1}^0 - P_{t+1} + I_{t+1}^0\right)^+$ for all $t = 1, \ldots,$ T with $I_T^0 = 0$. The decision variables include: x_{it} , order size of item i in period t; I_{it} , ending inventory of item i in period t; $Y_{it} = 1$ if item i is replenished in period t and $Z_t = 1$ if a joint setup occurs in period t.

The FMT formulation is as follows,

$$Min\sum_{t=1}^{T} S_t Z_t + \sum_{i=1}^{I}\sum_{t=1}^{T} s_{it} Y_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} c_{it} x_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} h_{it} I_{it} \quad (5.1)$$

Subject to:

$$I_{it} = I_{it-1} + x_{it} - d_{it} \qquad (i = 1, \ldots, I;\ t = 1, \ldots, T) \quad (5.2)$$

$$x_{it} \le P_t Y_{it} \qquad (i = 1, \ldots, I;\ t = 1, \ldots, T) \quad (5.3)$$

$$\sum_{i=1}^{N} x_{it} \le P_t Z_t \qquad (t = 1, \ldots, T) \quad (5.4)$$

$$Y_{it} \le Z_t \qquad (i = 1, \ldots, I;\ t = 1, \ldots, T) \quad (5.5)$$

$$\sum_{i=1}^{I} I_{it} \ge I_t^0 \qquad (t = 1, \ldots, T) \quad (5.6)$$

$$X_{it} \ge 0,\ I_{it} \ge 0,\ Y_{it} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I;\ t = 1, \ldots, T) \quad (5.7)$$

$$Z_t = 0 \text{ or } 1 \qquad (t = 1, \ldots, T) \quad (5.8)$$

Constraint set (5.2) ensures that the demand is served. Constraint sets (5.3) and (5.5) prohibit replenishment unless the setup charges are incurred. Constraint set (5.4) represents the capacity constraints and constraint set (5.6) incorporates the aggregate inventory condition in the formulation. Constraints (5.7) and (5.8) force decision variables to take on feasible solution values.

5.2 RL problem formulation

Robinson and Lawrence (RL, 2004) extend the Robinson and Gao's (RG, 1996) uncapacitated formulation for the coordinated lot sizing problem. RG formulation for the CULSP is a tight fixed-charged arborescent network model which allows backorders. The RG formulation exploits the hierarchical linkages among the joint setup, item setup and assignment variables to constrain the setup variables to take on a value of 0 or 1 in the optimal solution of its LP relaxation. Let $i = 1,2, \ldots, I$, $t' = 1,2, \ldots, T$, and $t = 1,2, \ldots, T$, represent items, replenishment time periods, and demand time periods, respectively. Define:

$S_{t'}$, the joint setup cost in time t'; $s_{it'}$, setup cost for item i in period t'; $c_{it'}$ is the variable per unit order cost associated with replenishing item i in period t' and $h_{it't} = \sum_{r=t'}^{t-1} h_{ir}$, the per unit inventory holding cost for serving demand for item i in period t from a replenishment order in period t'. The total unit cost for supplying demand for item i in time t from a production in period t' is $C_{it't} = c_{it'} + h_{it't}$. The decision variables include: $Z_{t'} = 1$ if a joint setup occurs in period t', and 0 otherwise; $Y_{it'} = 1$ if item i is replenished in period t', and 0 otherwise; and $X_{it't}$ is the portion of demand for item i in period t that is served from a replenishment order in period t'. The RL's CCLSP formulation, assuming backorders are not allowed, is:

$$Min \sum_{t'=1}^{T} S_{t'} Z_{t'} + \sum_{i=1}^{I} \sum_{t'=1}^{T} s_{it'} Y_{it'} + \sum_{i=1}^{I} \sum_{t'=1}^{T-1} \sum_{t=t'}^{T} C_{it't} d_{it} X_{it't} \qquad (5.9)$$

Subject to

$$\sum_{t'=1}^{t} X_{it't} = 1 \qquad (i = 1, \ldots, I, t = 1, \ldots, T) \qquad (5.10)$$

$$Y_{it'} \le Z_{t'} \qquad (i = 1, \ldots, I, t' = 1, \ldots, T) \qquad (5.11)$$

$$X_{it't} \le Y_{it'} \qquad (i = 1, \ldots, I, t' = 1, \ldots, T, \; t = t', \ldots, T) \qquad (5.12)$$

$$\sum_{t=t'}^{T} \sum_{i=1}^{I} X_{it't} d_{it} \le P_{t'} \qquad (t' = 1, \ldots, T) \qquad (5.13)$$

$$Z_{t'} = 0 \text{ or } 1 \qquad (t' = 1, \ldots, T) \qquad (5.14)$$

$$Y_{it'} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I, t' = 1, \ldots, T) \qquad (5.15)$$

$$0 \le X_{it't} \le 1 \qquad (i = 1, \ldots, I, t' = 1, \ldots, T, t = t', \ldots, T) \qquad (5.16)$$

Constraint set (5.10) insures that each item's demand is satisfied in each period. Constraint set (5.11) prevents an item setup from occurring unless there is a joint setup, while constraint set (5.12) prohibits replenishment unless the item setup charge is incurred. Constraint set (5.13) represents the capacity constraint introduced by Robinson and Lawrence (2004). Constraints (5.14), (5.15), and (5.16) force decision variables to take on feasible solution values. The above model is extended to consider backorders by changing the summation in constraint (5.10) to consider $t' = 1, \ldots, T$; and altering the last summation in the objective function to include $t' = 1, \ldots, T$ and $t = 1, \ldots, T$; and defining constraint sets (5.13) and (5.16) to consider $X_{it't}$ from t = 1, …, T.

5.3 GR and GR^{ext} problem formulations

Gao and Robinson (GR 2004), propose replacing constraint set (5.13) with the following constraint,

$$\sum_{t=t'}^{T}\sum_{i=1}^{I} X_{it't} d_{it} \le P_{t'} Z_{t'} \quad (t' = 1, \ldots, T) \qquad (5.13')$$

where the binary decision variable $Z_{t'}$, for the joint setup is incorporated into the right-hand side of the constraint set. This formulation yields the convex envelop relaxation (E*(GR)*) as shown below, which provide a tighter LP relaxation than RL formulation's convex envelop (E*(RL)*) .

$$\mathrm{E}(GR) = \mathrm{Min}\ \sum_{t'=1}^{T} S_{t'}\ Max\{X_{it't}, \forall i = 1, ..., I, t = t', ..., T, \frac{1}{P_{t'}}\sum_{t=t'}^{T}\sum_{i=1}^{I} X_{it't} d_{it}\}$$

$$+\sum_{i=1}^{I}\sum_{t'=1}^{T} s_{it}\ Max\{X_{it't}, \forall i = 1, ..., I\} + \sum_{i=1}^{I}\sum_{t'=1}^{T-1}\sum_{t=t'}^{T} C_{it't} d_{it} X_{it't}$$

$$\mathrm{E}(RL) = \mathrm{Min}\ \sum_{t'=1}^{T} S_{t'}\ Max\{X_{it't}, \forall i = 1, ..., I, t = t', ..., T\} + \sum_{i=1}^{I}\sum_{t'=1}^{T} s_{it}\ Max\{X_{it't}, \forall i = 1, ..., I\}$$

$$+\sum_{i=1}^{I}\sum_{t'=1}^{T-1}\sum_{t=t'}^{T} C_{it't} d_{it} X_{it't}$$

The construction of convex envelop relaxation is based on the results in Denizel et al. (1996).

We include an additional inventory constraint similar to constraint set (5.6) in FMT to obtain GR^{ext} , extension of the tight GR formulation. The aggregate inventory constraint and its parameters are as follows,

$$\sum_{t'=1}^{q}\sum_{t=q+1}^{T}\sum_{i=1}^{I} X_{it't} d_{it} \ge I_q^0 \qquad (q = 1, \ldots, T) \qquad (5.17)$$

where I_q^0 is the minimum aggregate inventory at the end of period q , calculated as I_q^0 = $\left(D_{q+1} - P_{q+1} + I_{q+1}^0\right)^+$ with $I_T^0 = 0$ and $D_{q+1} = \sum_{i=1}^{I} d_{iq+1}$. Our preliminary analysis shows that this constraint reduces the computational requirements for solving the GR formulation using general purpose optimization software.

5.4 BLR1'cap and BLR1'cap-ext problem formulations

BLR1', as proposed by Narayanan and Robinson (2006) for the CULSP, views the problem as multiple Wagner and Whitin (1958) problems that are linked by a complicating joint setup decision variable. Since BLR1' is shown to be the most efficient formulation for solving the uncapacitated problem in general purpose optimization software, we explore its application in a capacitated environment.

Define w_{iqt} as the fraction of all the demand for item i from period t' to period t that is served from a replenishment order in period t' and $C_{it't}$ as the sum of the variable per unit order and inventory holding costs for producing item i in period t' and covering its demand from period t' through t , where $C_{it't} = \sum_{q=t'+1}^{t} (c_{it'} + \sum_{k=t'}^{q-1} h_{ik}) d_{iq})$. The binary decision variable $Y_{it'}$ is introduced into the original BLR1' formulation to decouple the item and family setup constraints. The capacitated BLR1' formulation (BLR1'cap) is as follows,

$$\text{Min } \sum_{t'=1}^{T} S_{t'} Z_{t'} + \sum_{t'=1}^{T} \sum_{i=1}^{I} s_{it'} Y_{it'} + \sum_{i=1}^{I} \sum_{t'=1}^{T} \sum_{t=t'}^{T} C_{it't} w_{it't} \qquad (5.18)$$

Subject to

$$\sum_{t'=1}^{r} \sum_{t=r}^{T} w_{it't} = 1 \qquad (i = 1, \ldots, I, r = 1, \ldots, T) \qquad (5.19)$$

$$\sum_{t=t'}^{T} w_{it't} \le Y_{it'} \qquad (i = 1, \ldots, I, t' = 1, \ldots, T) \qquad (5.20)$$

$$Y_{it'} \le Z_{t'} \qquad (i = 1, \ldots, I, t' = 1, \ldots, T) \qquad (5.21)$$

$$\sum_{i=1}^{I} \sum_{t=t'}^{T} w_{it't} \left(\sum_{q=t'}^{t} d_{iq} \right) \le P_{t'} Z_{t'} \qquad (t' = 1, \ldots, T) \qquad (5.22)$$

$$0 \leq w_{it't} \leq 1 \qquad (i = 1, \ldots, I, t' = 1, \ldots, T, t = 1, \ldots, T) \quad (5.23)$$

$$Y_{it'} = 0 \text{ or } 1 \qquad (i = 1, \ldots, I, t' = 1, \ldots, T) \quad (5.24)$$

$$Z_{t'} = 0 \text{ or } 1 \qquad (t' = 1, \ldots, T) \quad (5.25)$$

The unique and most effective modeling feature is the compact structure of constraint set (5.19), which insures that all demand is met. Constraint sets (5.20) and (5.21) make sure the appropriate setup costs are incurred when an item is produced. Constraint set (5.22) represents the capacity constraint. Constraints (5.23), (5.24) and (5.25) force decision variables to take on feasible solution values.

Like GR^{ext} , we include the aggregate inventory constraint (5.26) as shown below to obtain extended version, $BLR1'^{cap\text{-}ext}$.

$$\sum_{t'=1}^{q} \sum_{t=q+1}^{T} \sum_{i=1}^{I} w_{it't} \left(\sum_{r=t'+1}^{t} d_{ir} \right) \geq I_q^0 \qquad (q = 1, \ldots, T) \quad (5.26)$$

We evaluate this as a separate formulation in our experimental design in order to test the effectiveness of the additional constraint.

5.5 Experimental design

The experimental design is similar to Federgruen et al. (2004) and Narayanan and Robinson (2006). The base set of problems has $I = 10$ items and horizon of $T = 12$ periods. The demand, d_{it}, is assumed to be normally distributed and varies by item and time period. Odd numbered items have a mean demand of 50 units and a standard deviation of 20 units; even numbered items have a mean demand of 100 units and a standard deviation of 20 units. We consider two levels of demand density $DD \in \{0.50, 1.0\}$, where demand density is the fraction of time periods experiencing demand for an individual item. When $DD = 0.50$ only 50% of the periods for each item experience demand and mean demand in the normal distribution is doubled, so that the average demand over the horizon remains constant irrespective of the demand density. Unit production costs are assumed to be equal to zero, inventory holding cost per unit per time period is $1

Capacity utilization (CU), defined as the ratio of total demand divided by the total available capacity over the planning horizon, is evaluated at four levels where, CU $\in$ {0.2, 0.4, 0.6, 0.8}. The available capacity per time period, $P_{t'}$, is constant for all t'. For a specified value of CU, the value of $P_{t'}$ is calculated by first generating the test problem's demand stream, and then solving for $P_{t'} = \left(\sum_{i=1}^{I} \sum_{t=1}^{T} d_{it} \right) / (T * CU)$. For each item i, the minor setup cost, $s_{it'}$ is indirectly computed by first choosing the "Time Between Orders" (TBO) which is given by $\sqrt{2s_{it'}/hd}$, where h is the per unit inventory holding cost and d is the average demand for the item over the planning horizon. The TBO-values for the items are generated from a uniform distribution on the interval [2, 6]. The joint setup cost, $S_{t'}$ is also computed from the TBO, $\sqrt{2S_{t'}/hD}$, where D is the average total demand for the family over the planning horizon. The joint setup TBO-values are evaluated at three levels, major TBO $\in$ {low, medium and high}. The low TBO-values are generated from a uniform distribution on the interval [1, 3], when medium the interval is set at [2, 6] and the high TBO interval is [5, 10]. Within a test problem $S_{t'}$ and $s_{it'}$ are constant across all time periods.

We utilize a full factorial design, which results in 24 different combinations of factor settings i.e., 2 levels of DD, 4 levels of CU and 3 levels of major TBOs. For each combination of these factors, three test problems are randomly generated. Each test problem is solved using the six different formulations presented in this chapter. The experiments are conducted on a personal computer running a Pentium® 4 processor at 1.9 GHz with the Windows 2000 Professional operating system, and solved using Xpress-MP version 2005A (Xpress Optimizer v16.01.02), a state-of-the art general purpose optimization software. Each formulation is solved with a preset time limit of two hours. In the instances that could not be solved to optimality within two hours, the MIP gap at termination from Xpress-MP is noted. The MIP (end) gap, expressed as a percentage of the incumbent (LB) solution i.e. (best *integer* solution –LB)/LB, serves as a metric towards finding and verifying optimality.

5.6 Experimental results

The results are summarized by demand density in Table 5.1. The metrics used for evaluating the performance of the various capacitated formulations are LP gap, which is defined as (optimal objective value – objective value of the LP relaxation)/optimal objective value, MIP solution time, number of unsolved problems and the average MIP end gap of unsolved problems. Each cell in the table represents the average of 36 problems for the corresponding demand density. Only the GR formulation solves all the 72 problems to optimality, while one problem remains unsolved for $BLR1'^{cap\text{-}ext}$ and RL at the end of the two-hour pre-set time limit. FMT formulation could not solve 42% (30 out of 72) of the problems to optimality.

Even though the LP gap of the RL formulation is over 5 times weaker than the GR and $BLR1'^{cap}$ formulations, their MIPs solve relatively faster in an optimization software except for one outlier. This is because the LP of the RL formulation is much easier to solve than others. The effect of aggregate inventory constraint (a surrogate constraint for the MIP problem) is evident in $BLR1'^{cap}$ but not in GR formulation. The $BLR1'^{cap\text{-}ext}$ solves more problems to optimality than its counterpart, $BLR1'^{cap}$.

It is worth noting that GR, $BLR1'^{cap}$ and their extensions have equally tight LP relaxation, but their MIP solution times are different. GR formulations dominate the performance when the demand is lumpy (*DD*=0.5), and the $BLR1'^{cap}$ formulations dominate the performance in non-lumpy demand situations (*DD*=1.0).

Tables 5.2 and 5.3 provide the summary results by experimental factors; these tables provide further insight into the relative performance of these formulations with respect to *CU* and major TBOs, which is an indirect measure of joint setup costs. For the FMT formulation the LP relaxation becomes progressively tighter as major TBOs increases, whereas the exact opposite happens to the arborescent network formulations like RL and GR. When the joint setup costs are low (low major TBOs), the coordinated problems are generally easier to solve using RL, GR or $BLR1'^{cap}$ formulations.

Table 5.1: Summary of CCLSP formulations performance by demand density

	Average LP Gap		Average MIP time in seconds		No. of problems not solved to optimality		Average end (MIP) gap of unsolved problems	
	DD=0.5	*DD*=0.1	*DD*=0.5	*DD*=0.1	*DD*=0.5	*DD*=0.1	*DD*=0.5	*DD*=0.1
FMT	55.40%	55.04%	548.04*	6159.31*	2	28	3.25%	8.32%
RL	14.67%	17.80%	357.73*	102.77	1	0	0.83%	-
GR	3.85%	3.19%	465.98	96.46	0	0	-	-
GR^{ext}	3.85%	3.19%	554.28*	90.19	2	0	3.57%	-
$BLR1^{'cap}$	3.85%	3.19%	1157.35*	44.89	4	0	2.90%	-
$BLR1^{'cap\text{-}ext}$	3.85%	3.19%	553*	49.90	1	0	2.72%	-

* - indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours.

All formulations are sensitive to demand lumpiness (*DD*), the performance of the FMT deteriorates when we move from a lumpy to a non-lumpy situation, while the inverse happens in the case of other formulations. At *DD*=1.0, FMT. couldn't solve 28 of the 36 problems to optimality.

Expect for one outlier, the MIP solution times of the weaker RL outperforms the tighter GR under all experimental settings for *DD*=0.5. Even in *DD*=1.0 the MIP solution times of RL are very competitive and it does not reflect the vast difference in the LP gap. As stated earlier, this counter-intuitive result could be attributed to the relative easiness of LP problem of RL formulation. $BLR1^{'cap}$ and $BLR1^{'cap\text{-}ext}$ performs better than GR under all experimental settings for *DD*=1.0, the inverse is true for *DD*=0.5. The impact of the aggregate inventory constraint is still not apparent from this detailed summary, in some factor settings it aids both GR and BLR1' but the effect is not consistent.

The effect of *CU* is also not evident from this result. At lower *CU* the problems are to easier to solve, while at moderate to high *CU*, interaction effects with joint set-up costs sets-in and there is no clear evidence of its impact in performance. FMT is an exception to this inference.

Table 5.2: Summary results of CCSLP formulations by experimental factors for Demand Density = 0.5

Experimental Factors		Time in seconds						LP Gap		
CU	Major TBO	FMT	RL	GR	GR^{ext}	$BLR1^{\prime cap}$	$BLR1^{\prime cap\text{-}ext}$	FMT	RL	$GR/GR^{ext}/BLR1^{\prime cap}/BLR1^{\prime cap\text{-}ext}$
0.2	Low	2.48	0.29	0.37	0.36	0.51	0.51	87.55%	0.20%	0.20%
	Med	2.39	0.83	1.10	1.15	1.38	1.39	73.96%	1.31%	0.89%
	High	4.45	4.14	4.84	4.43	5.03	5.64	54.03%	10.06%	6.13%
0.4	Low	2.31	0.95	0.74	1.20	1.21	1.14	76.63%	1.09%	0.90%
	Med	3.77	2.94	3.53	5.67	4.71	6.37	56.28%	9.10%	2.14%
	High	6.19	9.16	10.52	14.94	19.73	16.46	32.68%	24.36%	2.97%
0.6	Low	3.42	3.85	4.23	4.36	5.42	6.05	68.17%	3.22%	1.76%
	Med	1226.56	3110.72*	2501.08	1458.79	3487.93*	2807.79	47.27%	18.69%	5.61%
	High	2721.08*	321.49	648.57	2587.11*	7240.57*	1149.02	28.92%	37.78%	9.01%
0.8	Low	13.10	7.36	14.06	9.48	21.80	27.06	61.46%	6.80%	3.18%
	Med	2571.13*	752.38	2304.38	2518.71*	2512.05*	2506.04*	40.35%	23.46%	5.64%
	High	29.21	78.69	98.36	45.16	587.86	117.20	23.62%	40.02%	7.79%
Average		548.84	357.73	465.98	554.28	1157.35	553.72	54.24%	14.67%	3.85%

* - indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours.

Table 5.3: Summary results of CCSLP formulations by experimental factors for Demand Density = 1.0

Experimental Factors		Average Time in Seconds						Average LP Gap		
CU	Major TBO	FMT	RL	GR	GR^{ext}	BLR1$^{'cap}$	BLR1$^{'cap\text{-}ext}$	FMT	RL	GR/GR^{ext}/BLR1$^{'cap}$/ BLR1$^{'cap\text{-}ext}$
0.2	Low	5064.94*	0.07	0.10	0.10	0.07	0.08	88.09%	0.00%	0.00%
	Med	6.34	2.55	3.08	3.82	1.77	2.08	74.33%	2.27%	1.31%
	High	5.41	15.01	14.93	19.85	9.19	13.51	54.31%	12.80%	7.32%
0.4	Low	7640.47*	12.23	16.35	13.47	8.47	8.78	78.07%	3.49%	1.21%
	Med	7394.77*	34.44	30.87	43.95	21.10	16.79	57.45%	12.81%	1.95%
	High	6974.41*	108.30	144.45	60.68	39.10	75.23	33.55%	28.35%	2.86%
0.6	Low	7604.91*	56.40	49.64	64.27	30.30	35.68	70.01%	7.90%	2.16%
	Med	7780.15*	121.72	120.91	87.77	62.05	69.80	48.50%	25.19%	4.97%
	High	8012.75*	427.18	413.53	425.19	87.82	99.39	28.78%	44.25%	7.53%
0.8	Low	7551.59*	143.06	89.12	104.33	70.75	79.64	64.42%	9.74%	2.33%
	Med	7885.45*	160.78	131.77	124.64	126.00	90.23	41.36%	25.86%	3.02%
	High	7990.53*	151.49	142.72	134.20	82.07	107.53	21.65%	40.91%	3.58%
Average		6159.31	102.77	96.46	90.19	44.89	49.90	55.04%	17.80%	3.19%

* - indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours.

The experimental design reveals that the tight GR formulation is still one of the most efficient CCLSP formulation for use in general purpose optimization software. The new BLR1$'^{cap}$ and its extensions is another set of tight CCLSP formulations but their computational performance is sensitive to the lumpiness of the demand stream. The FMT formulation which is used as a benchmark in Federgruen et al. (2004) is a poor choice to use in optimization software like Xpress-MP or CPLEX. It should also be noted that even though one formulation outperforms the rest in general purpose software, each one has a specialized structure which can be exploited to develop customized heuristics or optimization approaches. Federgruen et al. (2004) utilizes the network structure in the formulation to develop progressive interval heuristics, Robinson and Lawrence (2004) use Lagrangian based algorithms to develop specialized heuristics based on their formulation, and Gao and Robinson (2004) develop a Lagrangian/dual-ascent based heuristic to solve the capacitated problem. The mathematical structure of the new BLR1$'^{cap}$ and BLR1$'^{cap\text{-}ext}$ formulations are yet to be exploited and provide additional research opportunities. Finally, the aggregate inventory constraint is used only in the progressive interval heuristic and the effect of this surrogate constraint in the Lagrangian based heuristics is not known. This also presents another venue for research in CCSLP.

CHAPTER VI

HEURISTICS FOR COORDINATED CAPACITATED LOT-SIZING PROBLEM (CCLSP)

The CCLSP contains both the joint capacity constraints that complicate solution of the MCLSP and the family setup decision variables that complicate the mathematical structure of the CULSP. The resulting mathematical structure is NP-complete. Several optimization based heuristics are proposed for the CCLSP including Lagrangian relaxation (Robinson and Lawrence 2004), Lagrangian/dual-ascent (Gao and Robinson 2004) and new class of progressive interval heuristics (Federgruen et al. 2004). However the research findings highlight the difficulty of finding both good heuristic and optimal solutions for the CCLSP problem. The best known procedure is the expanding horizon (EH) progressive interval heuristics, but its performance rapidly deteriorates as the size of the problem increases. For example, it takes approximately 5.5 hours to solve a 25 item-10 period problem, whereas it requires only 30 seconds for a problem with 10 items and 10 periods. The current state of the art heuristic performance justifies the development of alternative heuristic approaches for this problem.

In this research, we move away from the optimization based techniques and consider construction and metaheuristic approaches to the problem. The construction heuristics were successfully applied to the MCLSP when requirements planning software was at its preliminary phase, but as the computing power increased literature has grown towards optimization based approaches. Construction heuristics are also proposed for the uncapacitated coordinated lot-sizing problem (Fogarty and Barringer 1987, Silver and Kelle 1988, Atkins and Iyogun 1988, Iyogun 1991 and Federgruen and Tzur 1991). More recently, Boctor et al. (2004) developed a metaheuristic for the uncapacitated problem. We have also developed and tested two new forward-pass heuristics, a two-phase construction heuristic and a simulated annealing metaheuristic for the uncapacitated problem (Chapter IV).

As a part of this research, we extended both the forward-pass and greedy-period construction heuristics for the uncapacitated problem to handle capacity constraints. The preliminary results indicate the superiority of the greedy-period approach, which directs our efforts in this area. Recognizing that CCLSP generalizes the ULSP, CLSP, MCLSP, and CULSP classes, the research objective is to develop heuristics for the CCLSP that also provides high quality solutions at minimal computational effort for all these problem classes. We extend the two-phase greedy heuristic to create a six-phase construction heuristic, develop a simulated annealing metaheuristic (SAM) based on the six-phase approach for the capacitated problem, and then evaluate their performance under a wide range of parameter settings.

6.1 Six-phase heuristic

The six-phase heuristic builds upon the basic concepts of the two-phase greedy heuristic and Dogramaci et al.'s (1981) construction heuristic for multi-item capacitated problem, by incorporating extensions and refinements necessary to consider the impact of the shared family setup on both cost and capacity. The heuristic contains three major subroutines that are implemented in six-phases. The three subroutines are:

Subroutine I: Cost minimizing left-shift. This procedure attempts to reduce costs by rescheduling production for individual items or the product family into earlier time periods (left-shifting) subject to aggregate capacity availability when considering only those time periods with a scheduled product family setup (i.e., open time periods).

Subroutine II: Feasibility seeking left-shift. The subroutine moves production into earlier time periods as necessary to guarantee capacity feasibility in each period, while minimizing the increase in cost. At the conclusion of this subroutine, the problem is capacity feasible.

Subroutine III: Cost minimizing right-shift. This procedure attempts to reduce schedule costs by right-shifting production as late as possible while still maintaining capacity feasibility.

The six-phase heuristic begins by verifying that the problem is "aggregate" capacity feasible, i.e., there is sufficient potential capacity to supply all the demand over the planning horizon without backordering. An initial production schedule is then established with lot-sizes $a_{it} = d_{it}$ for all *i* and *t*, where d_{it} is the demand of item *i* in time period *t* stated in capacity units. Next, the six-phase heuristic is implemented as follows:

Phase 1. Run Subroutine I to generate a lower cost production schedule if possible.

Phase 2. Run Subroutine II to insure that the production schedule is capacity feasible.

*Phase 3***.** Since the schedule changes during Phase 2 may generate new opportunities for cost reduction, Phase 2 is followed by another application of Subroutine I.

Phase 4. The application of Subroutine I in Phase 3 may destroy individual time period capacity feasibility. Hence, Subroutine II is invoked to guarantee feasibility.

Phase 5. Run Subroutine III in an attempt to decrease costs by moving production as late as possible in the planning horizon without violating capacity constraints.

Phase 6. Run Subroutine I to search for additional potential cost reductions. However in this final phase, only lot-size consolidations that do not violate individual time period capacity constraints are permitted. Hence at the conclusion of this phase, the heuristic terminates with a capacity feasible solution.

Subroutine I: Cost minimizing left-shift

The subroutine begins with a problem that is aggregate capacity feasible, but may violate capacity constraints in individual time periods. A cost reducing procedure iteratively left-shifts the cost minimizing lot-size(s) of either an individual item or a family of items from period *t* into period $t' < t$ until no further savings are possible. The procedure only considers left-shifting production into open periods (i.e., those with a scheduled family setup). The savings, $C_i(t', t)$, for left-shifting item *i*'s production from period *t* into t' is:

$$C_i(t', t) = (Y_{it'} - 1)s_{it'} + s_{it} - I_i(t', t) \text{ for } a_{it} > 0 \text{ and, 0 otherwise.}$$

where $Y_{it'} = 1$ if an order is scheduled for item *i* in period t', s_{it} is the item's setup cost in period *t*, the inventory carrying cost from period t' to period *t* is $I_i(t', t) = (t - t')h_i a_{it}$, h_i is the unit inventory carrying cost per period for item *i* and a_{it} is the current production quantity for item *i* in period *t*. The savings associated with re-scheduling the family of items from period *t* into period t' is

$$C(t^{'},t) = (Z_{t'} - 1)S_{t'} + S_t + \sum_{i=1}^{I} C_i(t^{'},t)$$

where, $Z_{t'} = 1$ if a family setup is scheduled in period $t^{'}$ and 0 otherwise, and S_t is the product family setup cost in period t.

The maximum quantity that can be left-shifted into period t', $E_{t'}$, is the unallocated capacity in open periods from 1 to t' less the capacity shortage in periods $t'+1$ to t-1. Defining P_j as the rated capacity in period j, the unallocated capacity in open period j is

$$\tilde{e}_j = (P_j - \sum_{i=1}^{I} a_{ij})Z_j$$

where, $\tilde{e}_j > 0$ indicates capacity is available and $\tilde{e}_j < 0$ indicates that the current production schedule exceeds the rated capacity of period j. The capacity shortage in period $t'+1$ to t-1 that must be supplied from period t' or earlier is

$$G(t'+1, t-1) = Min\{0, \tilde{e}_{t'+1}, \tilde{e}_{t'+1} + \tilde{e}_{t'+2}, \tilde{e}_{t'+1} + \tilde{e}_{t'+2} + \tilde{e}_{t'+3}, ..., \tilde{e}_{t'+1} + ... + \tilde{e}_{t-1}\}$$

where $G(t'+1, t-1) < 0$ signals a shortage. Therefore, the maximum quantity that can be left-shifted into period t' is

$$E_{t'} = Max\left\{0,\ \sum_{j=1}^{t'} \tilde{e}_j - \left|G(t'+1, t-1)\right|\right\}$$

The steps of the subroutine follow.

Steps of Subroutine I: Cost minimizing left-shift

Step 1. Compute unallocated capacities. Compute the unallocated capacity, e_t, in each period t for the current production schedule, where

$$e_t = P_t - \sum_{i=1}^{I} a_{it} \text{ , for } t = 1, 2 \ldots T$$

Step 2. Compute item and family savings. Calculate $C_i(t^{'},t)$ and $C(t^{'},t)$ for all i, $t' < t$, and t for which $E_{t'} > 0$. Each item lot-size with $C_i(t^{'},t) > 0$ and $a_{it} \le E_{t'}$ and each family of items with $C(t^{'},t) > 0$ and $\sum_{i=1}^{I} a_{it} \le E_{t'}$ are candidates for left-shifting from period

t into period t'. In addition, a product family cannot be left-shifted from period t if removing the capacity violates aggregate capacity feasibility when considering the opened production periods. Specifically, the following must hold

$$\sum_{j=1}^{t-1} \tilde{e}_j - \sum_{i=1}^{I} a_{it} + \sum_{j=t+1}^{T} \tilde{e}_j \geq 0$$

2a. *Item Saving Adjustment.* It is possible that shifting a production lot-size into period t' may violate the period's available capacity. In this case, some of the production quantity must be moved forward into a period(s) earlier than t' causing additional inventory holding costs. The quantity moved forward, $f_{t'}$, is

$$f_{t'} = \left| Min\{0, e_{t'} - a_{it}\} \right|$$

where a_{it} is the quantity being rescheduled from period t to period t'. The adjusted item cost savings is $AC_i(t',t) = C_i(t',t) - Cf_{t'}$, where $Cf_{t'}$ is the minimum potential increase in inventory holding cost (see Appendix B for details). The adjusted cost savings is the maximum potential savings obtained by left-shifting item i into period t'.

2b. *Family Saving Adjustment.* Rescheduling a family of items into an earlier time period may require two types of capacity related inventory cost adjustments.

Type 1 Cost Adjustment, CN_t. When rescheduling a family of items from period t to t', we eliminate the family setup and thereby the capacity in period t. Part of this capacity, N_t, may have been used to satisfy demand in periods greater than time t, which must now be supplied from a period(s) earlier than t, thereby increasing inventory holding costs. Specifically, N_t is expressed as:

$$N_t = Min\{Max\{0, \tilde{e}_t\}, \left|G(t+1,T)\right|\}$$

where, $Max\{0, \tilde{e}_t\}$ is the available capacity in period t that can be used to satisfy demand in periods t+1 to T, and $\left|G(t+1,T)\right|$ is the capacity shortage in periods t +1 to T that is currently supplied from production in period t or earlier. Mathematically,

$$G(t+1,T) = Min\{0, \tilde{e}_{t+1}, \tilde{e}_{t+1} + \tilde{e}_{t+2}, \tilde{e}_{t+1} + \tilde{e}_{t+2} + \tilde{e}_{t+3}, ..., \tilde{e}_{t+1} + ... + \tilde{e}_T\}.$$

Two cases are possible.

Case 1: $N_t \geq |G(t+1,T)|$. In this case, $|G(t+1,T)|$ was entirely supplied by period t. The computation of CN_t, the minimum possible increase in inventory costs, is detailed in Appendix B.

Case 2: $N_t < |G(t+1,T)|$. In this situation, only a part of $|G(t+1,T)|$ is supplied by period t with the remainder $|G(t+1,T)| - N_t$ units coming from an earlier period(s). In this case, we first temporarily adjust the available capacity as necessary in the open periods from 1 to t-1 to account for the quantity $|G(t+1,T)| - N_t$ (see Appendix C for details), and then compute the cost adjustment, CN_t, as described in Appendix B.

Type 2 Cost Adjustment, $CF_{t'}$. Left-shifting the product family into period t' may exceed the period's capacity. In this situation, a portion of the production must be moved into a period earlier than t' thereby, incurring additional inventory holding costs. The quantity moved forward is

$$F_{t'} = \left| Min\{0, e_{t'} - \sum_{i=1}^{I} a_{it}\} \right|.$$

The minimum possible increase in inventory holding costs for moving $F_{t'}$ units forward is $CF_{t'}$ (see Appendix B for calculations).

The adjusted family cost savings, $AC(t',t) = C(t',t) - CN_t - CF_{t'}$, is the maximum potential savings from left-shifting the product family from period t to t'.

Step 3. Left Shift Phase. If all $AC_i(t',t) \leq 0$ and $AC(t',t) \leq 0$, STOP, otherwise select the Maximum{ $AC_i(t',t)$, $AC(t',t)$ } for all i, $t' < t$, and t for left-shifting. If the maximum savings calls for left-shifting item i, update the order schedules by setting $a_{it'} = a_{it'} + a_{it}$ and $a_{it} = 0$. Otherwise, the maximum saving is associated with left-shifting a product family, so set $a_{it'} = a_{it'} + a_{it}$ and $a_{it} = 0$ for all i. Go to *Step 1.*

Subroutine II: Feasibility seeking left-shift

Subroutine II left-shifts production earlier in time as necessary to guarantee capacity feasibility in each time period while minimizing the increase in the schedule's cost.

Steps of Subroutine II: Feasibility seeking left-shift

Step 1. Initialize. Set $t = T$.

Step 2. Check individual time periods for feasibility. If the production in period t is within the capacity limit, i.e., $e_t \geq 0$, go to *Step 6.* Otherwise, continue.

Step 3. Insure sufficient capacity is available in earlier opened time periods. If sufficient capacity is available to cover the shortage in time t, $|e_t|$, then continue. Otherwise, schedule a family setup in the time period(s) immediately prior to t until sufficient capacity is available to cover the shortage.

Step 4. Compute the marginal cost of rescheduling. For each item i scheduled in period t, calculate the cost of producing the item in the immediate preceding open time period t'. The amount rescheduled, r_{it}, is either a_{it} or $|e_t|$ according to the following two cases.

Case 1. If $|e_t| > a_{it}$, then $r_{it} = a_{it}$. The marginal cost of rescheduling item i from period t to t' is $MC_i\ (t',t) = (t-t')h_i r_{it} + (1-Y_{it'})s_{it'} - s_{it}$

Case 2. If $|e_t| \leq a_{it}$, at least $|e_t|$ units must be transferred for feasibility. However, assuming sufficient capacity is available in earlier time periods, the entire lot size, a_{it}, could be rescheduled into period t' if it results in lower incremental cost. The value of r_{it} yielding the minimum cost in the following equation is rescheduled.

$$MC_i\ (t',t) = Min\left\{ \begin{array}{l} [(t-t')h_i r_{it} + (1-Y_{it'})s_{it'},\ where\ r_{it} = |e_t|], \\ \qquad [(t-t')h_i r_{it} + (1-Y_{it'})s_{it'} - s_{it},\ where\ r_{it} = a_{it}] \end{array} \right\}$$

Step5. Left-shift phase. Select the Minimum $\{ MC_i\ (t',t) \}$ for all i and $t' < t$ and reschedule it's production by setting $a_{it'} = a_{it'} + r_{it}$ and $a_{it} = a_{it} - r_{it}$. Next, update the available capacity in periods t' and t by setting $e_{t'} = e_{t'} - r_{it}$ and $e_t = e_t + r_{it}$. If $e_t <$ 0, go to *Step 4* otherwise go to *Step 6*.

Step 6. Roll back. If $t = 1$ stop, otherwise set $t = t - 1$ and go to *Step 2*.

Subroutine III: Cost minimizing right-shift

Subroutine III attempts to reduce costs by shifting production as late as possible subject to capacity availability. The procedure begins at time $t = T$ and iteratively works backward. For each period t with $e_t > 0$, the cost savings associated with shifting earlier production into period t from t' is calculated. The shift resulting in the maximum savings is implemented. The procedure continues shifting production into period t until $e_t = 0$ or further savings are not possible. The algorithm then moves to period t -1 and continues. A single item or a product family is candidate for right-shifting. The maximum quantity of item i that can be shifted from period t' to t, $v_{it'}$, is the minimum of the unallocated capacity in period t, e_t, and the quantity available to transfer out of period t' as detailed in the following equation.

$$v_{it'} = Min\left\{ e_{t,} \left(a_{it'} - \begin{bmatrix} Net\ requirement\ for \\ item\ i\ in\ period\ t' \end{bmatrix} - \begin{bmatrix} Net\ requirement\ for\ item\ i \\ from\ period\ t'+1\ to\ t-1 \end{bmatrix} \right) \right\}$$

The net requirement for item i in time period t' is $Max[(d_{it'} - \sum_{q=1}^{t'-1}(a_{iq} - d_{iq})),0]$. The procedures for calculating the net requirement for item i from period t' +1 to t -1 is detailed in Appendix A.

For $v_{it'} > 0$, the resulting cost savings is $H_i(t',t) = h_i v_{it'}(t - t') - s_{it} Y_{it} + s_{it'} X_{it'}$, where $X_{it'} = 1$ if the complete lot-size is shifted and 0, otherwise. For $v_{it'} = 0$, $H_i(t',t) = 0$. A family of items is feasible for right shifting when $\sum_{i=1}^{I} v_{it'} \le e_t$ with a cost savings of $H(t',t) = S_{t'} X_{t'} - S_t Z_t + \sum_{i=1}^{I} H_i(t',t)$, where $X_{t'} = 1$ if the whole product family is shifted from t' to t. The steps of the subroutine follow.

Steps of Subroutine III: Cost minimizing right-shift

Step 1. *Initialize.* Set $t = T$.

Step 2. *Check for capacity availability.* If $e_t \le 0$ go to *Step 5.*

Step 3. *Identify potential cost saving.* For i = 1, 2, …,I and t' = 1, 2, …, t-1 compute $v_{it'}$ and $H_i(t',t)$. If $\sum_{i=1}^{I} v_{it'} \le e_t$ then compute $H(t',t)$. If all $H_i(t',t) \le 0$ and $H(t',t) \le 0$ go to *Step 5.*

Step 4. *Right-shift phase.* Select the Maximum { $H_i(t',t)$, $H(t',t)$ } for all i and t' for right-shifting into period t. If the maximum savings is for right-shifting item i, update $a_{it'} = a_{it'} - v_{it'}$, $a_{it} = a_{it} + v_{it}$, $e_{t'} = e_{t'} + v_{it'}$ and $e_t = e_t - v_{it'}$. Otherwise, the maximum savings is associated with rescheduling a product family, so set $a_{it'} = a_{it'} - v_{it'}$ and $a_{it} = a_{it} + v_{it}$ for all i, $e_{t'} = e_{t'} + \sum_{i=1}^{I} v_{it'}$, and $e_t = e_t - \sum_{i=1}^{I} v_{it'}$. If $e_t \le 0$, go to *Step 5* otherwise, go to *Step 3.*

Step 5. *Roll back.* If t = 1 stop, otherwise set $t = t - 1$ and go to *Step 2*.

6.2 Simulated annealing metaheuristic for CCLSP

The CCLSP metaheuristic extends the simulated annealing metaheuristic (SAM) concepts in CULSP to consider limited capacity. The six-phase heuristic provides a feasible starting solution. Next we ensure capacity feasibility while generating new neighborhoods. Finally we use six-phase heuristic in the neighborhood improvement procedure. During preliminary experiments, several values of starting (θ_o : initial temperature) and stopping (ϕ : number of successive iterations without improvement) criterions were tested, with $\phi = 3T$ and θ_o = 1000 providing the best results. The steps for the CCLSP SAM follow.

Step 1: Initialization. Set $n=0$ and θ_o = 1000. Solve the six-phase heuristic to obtain an initial problem solution. Set $\hat{C}$ equal to the objective function value of the six-phase heuristic solution. Set $C_B = \hat{C}$ and the iteration counter, $count = 1$.

Step 2: Neighborhood generation. Randomly choose a value of t ε {1, 2, …,T} and attempt to change neighborhoods by perturbing the current solution as follows.

Case 1: If the current solution replenishes any items in period t, reschedule all the items in t into the preceding open replenishment period if there is sufficient aggregate capacity, otherwise generate another time period t for evaluation.

Case 2: If the current solution does not have any items scheduled in period t, then reschedule the plausible items in the preceding open replenishment period into period t.

Step 3: Neighborhood Improvement. Attempt to improve the perturbed solution generated in Case 1 or 2 by applying the six-phase heuristic, while maintaining the status of the perturbed joint setup in period t. The resulting solution provides a new candidate solution with an objective function value C'.

Step 4: Neighborhood search. Compute the transition probability Pr using equation (4.11). Replace the current solution with the candidate solution if the probability Pr is greater than or equal to a randomly generated number between [0, 1]; otherwise reject the candidate solution. If $\hat{C} < C_B$ update the best known solution and set = C_B = $\hat{C}$ and reset $count = 1$. Otherwise, set $count = count + 1$. Repeat Steps 2-4 five times.

Step 5: Update cooling temperature: Set $n=n+1$. Update the temperature using (4.12).

Step 6: Termination. If $count \geq 3T$ or $\theta_n \leq 1$, stop and report the best found solution. Otherwise, go to step 2.

6.3 Experimental design and results

To evaluate the performance of the two heuristics we conducted four different computational experiments. They are based on the experimental design of Robinson and Lawrence (RL, 2004), Gao and Robinson (GR, 2004), Federgruen et al. (FMT, 2004) and a new experimental design to study the effectiveness of the heuristic as a standalone procedure for solving a wide range of lot sizing problem classes as described in Chapter II.

6.3.1 Experimental design 1: based on RL

The experimental factors include number of items, capacity utilization and minor (item) setup cost. The number of items is taken from the set $N \in \{2, 4, 6, 10, 20, 30, 40\}$ and the planning horizon length is set constant at T=12 periods for all the problems generated in this design. The item demand is randomly generated from a uniform distribution on the interval [50, 150]. If the demand generated for an item at a particular time period is less than 60 units, it is set to zero representing a zero period demand. In this manner, demand lumpiness is incorporated into the problem set. Since it is based on a uniform distribution, on an average 90% of the time periods for any item have positive demand.

Minor setup costs are set at two levels, $s_{it} \in \{\$100, \$300\}$. The major setup costs range from $190 to $1920, with large values associated to problems with more items. This reduces the effect of minor setup costs on the problem. This approach is similar to the experimental design described in Erenguc (1988). The capacity utilization (*CU*) is tested at three levels, $CU \in \{0.05, 0.45, 0.85\}$. The *CU*=0.05 represents the uncapacitated problem, while *CU*=0.45 and *CU*=0.85 represent moderate and high capacity situations. Based on *CU* the available capacity P_t is computed as discussed in Section 5.5. For a particular problem, P_t is constant across all time periods.

We evaluate the performance of heuristics by considering all possible combinations of experimental factors. Ten random problems are generated for each of these combinations, resulting in 420 problems. Each problem is solved by both the six-phase and SAM, which are coded in C++. To obtain optimal problem solutions for performance benchmarking, each test problem is also solved with Xpress-MP Version 2005A (Xpress Optimizer Version 16.01.02), a state of the art optimization software package, using the GR formulation described in Section 5.3. All problems are solved on a personal computer running a Pentium® 4 processor at 1.9 gigahertz.

Experimental Results

Tables 6.1, 6.2 and 6.3 summarize the results by capacity utilization and experimental factors. Each cell in the table represents the average result of ten random problems. Overall SAM finds better solutions than both six-phase and RL heuristics. Six-phase performs better than RL in case of moderate and tight capacity situations, while the inverse is true for uncapacitated problems.

At *CU*=0.05, six-phase and SAM finds solutions with an average optimality gap of 0.81% and 0.05% respectively, compared to 0.44% of RL, while at *CU*=0.45, they find solutions with a gap of 1.48% and 0.46% compared to 3.91% of RL. For tightly capacitated problem, *CU*=0.85, RL could not obtain the benchmark solution for problems with more than 10 items. They used the weak RL formulation, discussed in Section 5.2, with an older version of the IBM OSL to obtain the optimal solutions. Since we use the tight GR formulation, we obtain optimal solutions for most of the test problems within the pre-set time limit of two hours. In case of problems which could not be solved to optimality, the best found integer solution was chosen as the benchmark. It should be noted that these unsolved problems had maximum MIP gap of only 0.45%, justifying the use of these solutions as a benchmark. At *CU* = 0.85, the six-phase and SAM finds solutions with an average optimality gap of 2.30% and 1.02%, compared to the gap of 4.72% reported by RL.

For SAM, the problems with higher minor setup cost ($300) are more difficult to solve than with lower setup cost. This corroborates the results in RL and is also true for six-phase heuristic, except for some uncapacitated instances. The problems also become difficult to solve as the *CU* increases. The impact of the number of items on the optimality gap is not clearly evident from this result, but the solution time required in solving increases with the number of items (size of the problem).

The Xpress-MP solution times increases exponentially as the capacity utilization increases, requiring more than two hours to obtain and verify optimality in some instances when *CU*=0.85, whereas the solution time of six-phase and SAM are fairly constant. The six-phase heuristic requires just milliseconds to solve the problems, while SAM requires an average 0.13 seconds to obtain the solutions. In contrast, the heuristic based on Lagrangian relaxation is not able to obtain solutions for tightly constrained (*CU*=0.85) problems with more than 10 items within their pre-set time limit of 100 minutes. Neither exact nor average comparisons of run-time could be made as these heuristics were solved in different machines, nevertheless the efficiency of six-phase and SAM is evident from these results.

Table 6.1: Summary results for the RL experimental design: $CU = 0.05$

Number of Items	Minor setup cost	Major setup cost	Average Optimality Gap[†]		Time for the heuristic (sec)		Xpress-MP time (sec) [‡]
			Six-phase[‡]	SAM[‡]	Six-phase[‡]	SAM[‡]	
2	100	190	1.18%	0.00%	0.00	0.02	0.03
2	300	190	0.71%	0.00%	0.00	0.02	0.04
4	100	310	1.33%	0.09%	0.00	0.03	0.07
4	300	310	1.07%	0.00%	0.00	0.03	0.07
6	100	360	1.14%	0.00%	0.00	0.04	0.17
6	300	360	0.23%	0.02%	0.00	0.04	0.19
10	100	620	0.42%	0.00%	0.00	0.06	0.13
10	300	620	0.62%	0.01%	0.00	0.07	0.32
20	100	1120	1.38%	0.00%	0.00	0.14	0.58
20	300	1120	0.77%	0.21%	0.00	0.14	1.51
30	100	1520	0.62%	0.00%	0.00	0.24	1.76
30	300	1520	0.65%	0.17%	0.00	0.23	5.72
40	100	1920	0.76%	0.00%	0.00	0.31	1.79
40	300	1920	0.50%	0.14%	0.00	0.35	11.10
Average			0.81%*	0.05%*	0.00	0.12	1.68

* Robinson and Lawrence (2004) reported an average gap of 0.44% and they use a different set of random seeds to generate the test problems

[†] (heuristic objective value – optimal objective value)/ optimal objective value

[‡] Each cell represents the average results for 10 test problems

Table 6.2: Summary results for the RL experimental design: *CU* = 0.45

Number of Items	Minor setup cost	Major setup cost	Average Optimality Gap†		Time for the heuristic (sec)		Xpress-MP time (sec) ‡
			Six-phase‡	SAM‡	Six-phase‡	SAM‡	
2	100	190	1.56%	0.00%	0.00	0.02	0.33
2	300	190	2.67%	0.44%	0.00	0.03	0.71
4	100	310	1.33%	0.00%	0.00	0.03	0.48
4	300	310	2.06%	0.74%	0.00	0.04	2.92
6	100	360	0.77%	0.00%	0.00	0.05	0.48
6	300	360	1.89%	1.11%	0.00	0.06	4.49
10	100	620	0.64%	0.00%	0.00	0.07	0.40
10	300	620	1.86%	0.94%	0.00	0.09	8.15
20	100	1120	1.37%	0.00%	0.00	0.17	1.34
20	300	1120	1.76%	1.14%	0.00	0.20	30.07
30	100	1520	0.62%	0.00%	0.00	0.26	2.10
30	300	1520	2.00%	1.08%	0.01	0.35	51.13
40	100	1920	0.76%	0.00%	0.00	0.43	3.80
40	300	1920	1.44%	0.98%	0.02	0.48	67.23
Average			1.48%*	0.46%*	0.00	0.16	12.40

* Robinson and Lawrence (2004) reported an average gap of 3.91% and they use a different set of random seeds to generate the test problems

† (heuristic objective value – optimal objective value)/ optimal objective value

‡ Each cell represents the average results for 10 test problems

Table 6.3: Summary results for the RL experimental design: *CU* = 0.85

Number of Items	Minor setup cost	Major setup cost	Average Optimality Gap†		Time for the heuristic (sec)		Xpress-MP time (sec) ‡
			Six-phase‡	SAM‡	Six-phase‡	SAM‡	
2	100	190	0.66%	0.27%	0.00	0.02	0.65
2	300	190	6.09%	1.61%	0.00	0.01	1.00
4	100	310	0.75%	0.39%	0.00	0.03	2.82
4	300	310	4.15%	2.60%	0.00	0.02	6.60
6	100	360	0.72%	0.42%	0.00	0.05	5.25
6	300	360	4.76%	1.87%	0.00	0.02	28.27
10	100	620	0.76%	0.28%	0.01	0.08	11.86
10	300	620	4.08%	2.09%	0.01	0.03	860.11
20	100	1120	0.23%	0.15%	0.01	0.20	34.62
20	300	1120	3.78%	1.65%	0.02	0.06	3418.14**
30	100	1520	0.15%	0.07%	0.03	0.30	134.83
30	300	1520	2.91%	1.58%	0.03	0.10	6014.60**
40	100	1920	0.05%	0.05%	0.04	0.50	151.73
40	300	1920	3.07%	1.25%	0.05	0.15	5508.31**
Average			2.30%*	1.02%*	0.01	0.11	1155.63

* Only a portion of this dataset (up to 10 item problems) was solved Robinson and Lawrence (2004) and they reported an average gap of 4.72% for the solved instances. Different set of random seeds were used to generate the test problems.

** indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours.

† (heuristic objective value – optimal objective value)/ optimal objective value

‡ Each cell represents the average results for 10 test problems

It is interesting to note the SAM runs slightly faster in tight capacity situations, this is directly related to number of feasible neighborhoods it can generate in a problem. In tightly capacitated problems, the number of feasible neighborhoods it can generate is far less than number of neighborhoods in uncapacitated or moderately capacitated instances.

Based on this experimental design, SAM improves the average solution gap of six-phase heuristic by 66% and is shown to be a better solution approach to the CCLSP than the Lagrangian relaxation based heuristic described in Robinson and Lawrence (2004).

6.3.2 Experimental design 2: based on GR

The experimental design follows Erenguc (1988) and Robinson and Gao (1996) with the necessary extensions to consider capacity. The experimental factors include the number of items, planning horizon length, family setup cost, and capacity utilization.

The number of items is represented at three levels where, $I \in \{10, 20, 40\}$. The planning horizon length is taken from the set, $T \in \{12, 18, 24\}$. In all test problems, demand, d_{it}, is assumed to be normally distributed and varies by item and time period. Odd numbered items have a mean demand of 50 units and a standard deviation of 20 units; even numbered items have a mean demand of 100 units and a standard deviation of 20 units. We assume demand occurs in 50% of the time periods. Without loss of generality, unit production costs are assumed to be equal to zero and inventory holding cost per unit per time period is $1.

Robinson and Lawrence (2004) find that the ratio of joint/family setup cost to item setup costs affect the run-time and quality of the heuristic solution. To study this factor, we draw the product family and item setup costs from normal distributions. The joint setup cost, $S_{t'}$, is represented at three different levels where the mean is an element of the set {$120, $480, $960} and the standard deviation is $36. Within a test problem $S_{t'}$ is constant across all time periods. Item setup costs, $s_{it'}$, are drawn from a distribution with a mean = $60 and a standard deviation = $18, where $s_{it'}$ varies by item but is constant across all time periods for a specific item within a test problem. The mean setup cost ratio per time period ranges from 0.05 to 1.6, where the setup cost ratio is $S_{t'} \Big/ \sum_{i=1}^{I} s_{it'}$.

Capacity utilization (CU) is evaluated at four levels where, CU ∈ {0.2, 0.4, 0.6, 0.8}. The available capacity per time period, $P_{t'}$, is constant for all t' and is computed as discussed in Section 5.5. Since backorders are not permitted in the model, problem data sets must be aggregate capacity feasible in each time period. That is $\sum_{t'=1}^{j} P_{t'} - \sum_{i=1}^{I} \sum_{t'=1}^{j} d_{it'} \geq 0$ for all j = 1, 2, …T. In a few problem instances with CU ≥ 0.8, aggregate capacity violations occur in the early time periods. These violations are remedied by increasing the available capacity in the associated time period(s).

For each combination of experimental factors, ten test problems are randomly generated resulting in a total of 1080 solved problems. Only five problems were generated in Gao and Robinson (2004), while we generate ten test problems to mitigate the effect of different random seeds. Each problem is solved by the six-phase heuristic, SAM and Xpress-MP version 2003F (Xpress Optimizer Version 14.24), a state-of-the-art optimization software package, using the tight GR formulation (Section 5.3). The study is conducted on a personal computer running a Pentium® 4 processor at 1.9 gigahertz.

Experimental results

The experimental results for the heuristics are encouraging. The metrics used are average optimality gap, defined by (heuristic solution value – optimal solution value)/optimal solution value) and heuristic run times. The computer used to run the GR's Lagrangian/dual ascent heuristic is a Pentium® 4 at 2.2 gigahertz which is similar to the one used in this study. Hence the CPU times for the problem solution are compared.

The average optimality gaps of six-phase, SAM and GR heuristic are 0.48%, 0.26% and 0.67%. The standard deviation of the optimality gaps for six-phase and SAM are 0.81% and 0.53%, with maximum gaps of 7.58% and 6.87% respectively. Six-phase finds optimal solutions for 29% of the problems (318 out of 1080), with 77% of the problems (834 out of 1080) lying below the average gap of GR heuristic. SAM finds optimal solutions for 43% of the problems (460 out of 1080), with 87% of the problems (938 out of 1080) lying below the average gap of GR heuristic. In comparison, GR finds optimal solutions for at least 47% of the problem, but the maximum gap and standard deviations are not reported. GR performance decreases as the capacity constraint becomes tighter, as evident from the higher average optimality gaps.

The computational requirements for the six-phase heuristic average 0.03 CPU seconds with a maximum of only 0.328 CPU seconds. SAM requires an average of 0.42 CPU seconds with a maximum of 2.515 CPU seconds. The GR heuristic requires an average of 1.05 CPU seconds; they do not report the maximum solution time. For comparison purposes, to find and verify the optimal solution Xpress-MP requires 51.7 CPU seconds on average, with a maximum of 9565 CPU seconds (2.65 hours).

Table 6.4 summarizes the results by experimental factor. The optimality gap of all the heuristics is positively correlated with the product family setup costs, and capacity utilization. SAM and GR results are also positively correlated with the number of time periods in the planning horizon. Heuristic performances are negatively correlated with the number of items. The number of items and the capacity utilization have the greatest impact on the quality of the heuristic solutions for SAM and GR, while the joint setup cost and number of items drives six-phase heuristic performance. SAM's performance is also influenced by the number of time periods to a greater extent when compared to other heuristics.

As expected, the computational requirements are positively correlated with the number of items and length of the planning horizon for all the heuristics. The requirements remain fairly constant with joint setup and capacity utilization for the six-phase and SAM heuristic. Increasing values of CU and joint setup cost tend to require more computational time for the GR heuristic, thereby demonstrating the limitations of Lagrangian/dual-ascent procedure. The six-phase heuristic's computational requirements are less than 2% of the time required by Xpress-MP to find optimal solutions under all factor level summaries, while SAM's requirements are slightly higher than 10% for some factor level summaries.

Table 6.4: Summary results for the GR experimental design

Experimental Factor	Average Optimality Gap*			Time for the heuristic (sec)			Xpress-MP time (sec)
	Six-phase	SAM	GR[†]	Six-phase	SAM	GR[†]	
I= 10	0.89%	**0.49%**	1.10%	0.01	0.15	0.67	51.68
I= 20	0.44%	**0.23%**	0.34%	0.02	0.32	0.89	90.90
I= 40	0.12%	**0.07%**	0.10%	0.06	0.80	0.43	12.55
T = 12	0.46%	**0.17%**	0.57%	0.01	0.10	0.30	3.77
T = 18	0.50%	**0.29%**	0.66%	0.02	0.35	0.86	19.95
T = 24	0.45%	**0.33%**	0.77%	0.06	0.81	1.98	131.41
S_t = 120	0.16%	**0.11%**	0.33%	0.03	0.42	0.05	2.66
S_t = 480	0.41%	**0.22%**	0.68%	0.03	0.42	0.33	18.72
S_t = 960	0.87%	**0.46%**	0.98%	0.02	0.42	2.76	133.75
CU= 0.2	0.31%	0.04%	**0.01%**	0.02	0.37	0.03	1.22
CU= 0.4	0.44%	**0.16%**	0.25%	0.02	0.42	0.11	4.39
CU= 0.6	0.51%	**0.34%**	1.04%	0.04	0.43	1.90	38.37
CU= 0.8	0.67%	**0.51%**	1.36%	0.03	0.46	2.15	162.85
Overall Average	0.48%	0.26%	0.67%	0.03	0.42	1.05	51.7

[*] Opt. Gap = (heuristic objective value- opt. objective value)/opt. objective value, the bold gaps indicate the best performing heuristic for the corresponding experimental factor.
[†] GR uses a different random seed to generate the test problems

SAM performs better than six-phase and GR heuristic at almost every level of experimental setting (indicated by bold entries in Table 6.4), except at CU = 0.20, where GR's Lagrangian / dual ascent is the best performing heuristic. SAM improves on the optimality gap of six-phase heuristic by an average of 46% for this experimental design. The amount of improvement decreases as *CU* and length of planning horizon (*T*) increases, while the inverse is true for joint setup cost.

Table 6.5 provides the average computational results for the ten randomly generated test problems associated with each combination of parameter settings. The majority of the lower optimality gaps for SAM are associated with lower *CU* and to some extent to the lower joint setup cost. Figure 6.1 indicates a two-way interaction between capacity utilization and the number of items. In general, higher quality solutions are associated with a greater numbers of items, with an increasing optimality gap moving to higher levels of capacity utilization. This result is explained by viewing a larger number of items as increased granularity, which permits the items to more effectively fit within a capacity constrained resource. Similar observations are made by Gao and Robinson (2004). The poorest heuristic performance for both six-phase and SAM is for problems with I = 10 and S_t = \$960, while in SAM it is also associated with the highest capacity utilization.

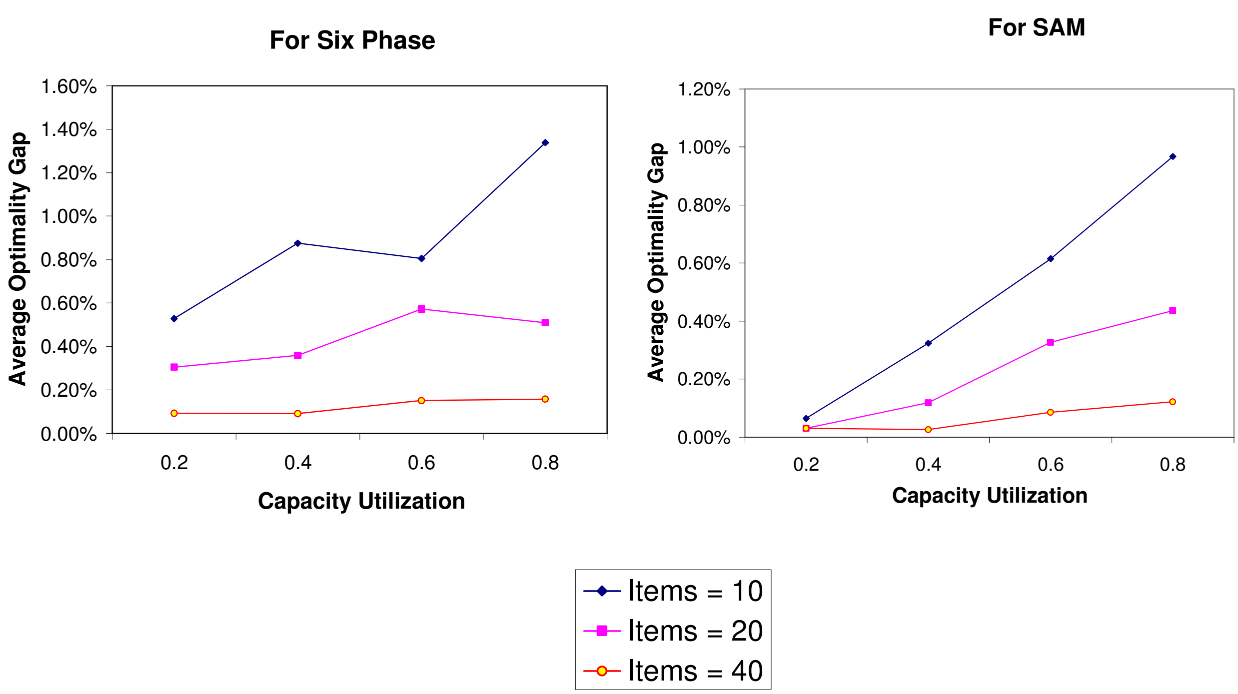


Figure 6.1: Effect of capacity utilization and number of items on heuristic optimality gaps

As the number of items increases, the effect of joint setup cost decreases, because this cost is shared among the items and its distinct influence on the cost structure diminishes.

Table 6.5: Expanded summary results‡ of six-phase and SAM heuristic for the GR experimental design

Experimental Factors			$S_t = 120$					$S_t = 480$					$S_t = 960$				
			Average Opt. Gap*		Heuristic time (sec)		MIP time (sec)†	Average Opt. Gap*		Heuristic time (sec)*		MIP time (sec)†	Average Opt. Gap		Heuristic time (sec)*		MIP time (sec)†
T	*CU*	*I*	Six-phase	SAM	Six-phase	SAM		Six-phase	SAM	Six-phase	SAM		Six-phase	SAM	Six-phase	SAM	
12	0.2	10	0.07%	0.04%	0.00	0.03	0.09	0.61%	0.00%	0.00	0.03	0.10	0.59%	0.00%	0.00	0.03	0.14
		20	0.02%	0.02%	0.00	0.07	0.19	0.10%	0.00%	0.00	0.07	0.36	0.93%	0.00%	0.00	0.07	0.32
		40	0.04%	0.01%	0.01	0.15	0.41	0.12%	0.03%	0.01	0.19	0.55	0.15%	0.03%	0.01	0.15	1.11
	0.4	10	0.07%	0.00%	0.00	0.04	0.08	0.63%	0.09%	0.00	0.04	1.97	2.35%	0.71%	0.00	0.03	4.37
		20	0.02%	0.02%	0.00	0.07	0.17	0.10%	0.00%	0.00	0.07	0.34	1.25%	0.43%	0.00	0.08	0.75
		40	0.04%	0.02%	0.02	0.18	0.40	0.12%	0.01%	0.02	0.17	0.51	0.15%	0.05%	0.01	0.18	1.09
	0.6	10	0.05%	0.05%	0.00	0.03	0.43	0.41%	0.31%	0.00	0.04	6.00	1.31%	0.50%	0.01	0.04	9.21
		20	0.02%	0.01%	0.01	0.08	0.17	0.50%	0.13%	0.01	0.11	4.06	0.84%	0.40%	0.00	0.09	11.55
		40	0.04%	0.01%	0.02	0.20	0.41	0.03%	0.01%	0.01	0.17	0.88	0.36%	0.18%	0.01	0.18	11.34
	0.8	10	0.61%	0.51%	0.00	0.03	1.73	0.99%	0.55%	0.00	0.03	10.13	2.21%	0.69%	0.00	0.03	13.86
		20	0.22%	0.10%	0.01	0.09	1.60	0.36%	0.26%	0.01	0.09	5.69	0.83%	0.72%	0.00	0.09	29.79
		40	0.11%	0.05%	0.01	0.25	2.17	0.08%	0.05%	0.02	0.23	2.48	0.20%	0.11%	0.01	0.19	11.25
18	0.2	10	0.07%	0.05%	0.00	0.10	0.20	0.80%	0.00%	0.00	0.11	0.35	1.07%	0.08%	0.01	0.12	0.46
		20	0.09%	0.01%	0.01	0.24	0.44	0.12%	0.03%	0.01	0.21	0.94	0.38%	0.08%	0.01	0.23	2.79
		40	0.03%	0.02%	0.03	0.60	1.00	0.06%	0.03%	0.04	0.61	1.35	0.18%	0.04%	0.02	0.61	2.68
	0.4	10	0.12%	0.03%	0.01	0.13	0.21	0.84%	0.25%	0.00	0.14	3.79	1.71%	0.74%	0.00	0.13	13.21
		20	0.09%	0.02%	0.01	0.22	0.42	0.12%	0.02%	0.01	0.29	1.43	0.54%	0.16%	0.01	0.25	5.95
		40	0.03%	0.02%	0.04	0.63	0.95	0.06%	0.02%	0.04	0.65	1.25	0.18%	0.01%	0.03	0.76	3.61
	0.6	10	0.16%	0.02%	0.01	0.15	1.02	1.03%	0.98%	0.01	0.12	21.67	1.53%	1.40%	0.01	0.11	55.40
		20	0.05%	0.05%	0.03	0.24	0.45	0.66%	0.26%	0.02	0.30	13.39	1.16%	0.77%	0.01	0.22	77.96
		40	0.03%	0.02%	0.09	0.63	1.06	0.04%	0.04%	0.08	0.60	2.04	0.45%	0.23%	0.08	0.77	32.38
	0.8	10	1.16%	0.80%	0.01	0.14	7.10	1.32%	1.01%	0.00	0.11	60.72	1.99%	1.62%	0.00	0.10	124.60
		20	0.34%	0.27%	0.02	0.33	6.63	0.55%	0.42%	0.02	0.27	29.38	0.58%	0.54%	0.01	0.22	182.68
		40	0.20%	0.17%	0.06	0.78	6.14	0.15%	0.12%	0.06	0.81	8.02	0.17%	0.12%	0.06	0.71	46.50

‡Each cell represents the average measure of ten random problems

* Opt. Gap = (heuristic objective value- opt. objective value)/opt. objective value

† Time for Xpress-MP to find and verify the optimal solution using the tight GR formulation

Table 6.5 (continued)

Experimental Factors			$S_t = 120$					$S_t = 480$					$S_t = 960$				
			Average Opt. Gap*		Heuristic time (sec)		MIP time (sec)†	Average Opt. Gap*		Heuristic time (sec)		MIP time (sec)†	Average Opt. Gap*		Heuristic time (sec)		MIP time (sec)†
T	*CU*	*I*	Six-phase	SAM	Six-phase	SAM		Six-phase	SAM	Six-phase	SAM		Six-phase	SAM	Six-phase	SAM	
24	0.2	10	0.17%	0.07%	0.01	0.30	0.42	0.59%	0.20%	0.01	0.26	0.69	0.79%	0.14%	0.01	0.25	1.09
		20	0.08%	0.02%	0.03	0.58	0.83	0.24%	0.09%	0.02	0.53	1.70	0.79%	0.02%	0.01	0.65	4.19
		40	0.02%	0.01%	0.08	1.28	1.92	0.04%	0.04%	0.08	1.24	2.60	0.18%	0.06%	0.04	1.36	6.13
	0.4	10	0.25%	0.06%	0.01	0.43	0.50	0.62%	0.24%	0.00	0.27	7.30	1.28%	0.80%	0.01	0.29	38.29
		20	0.08%	0.02%	0.03	0.58	0.87	0.19%	0.06%	0.02	0.61	2.76	0.85%	0.33%	0.02	0.64	17.96
		40	0.02%	0.01%	0.09	1.47	1.82	0.04%	0.02%	0.10	1.47	2.47	0.18%	0.07%	0.05	1.61	6.06
	0.6	10	0.13%	0.06%	0.02	0.30	2.06	0.61%	0.51%	0.02	0.29	122.15	2.00%	1.70%	0.02	0.25	304.28
		20	0.05%	0.04%	0.07	0.60	1.29	0.51%	0.34%	0.05	0.74	31.26	1.37%	0.94%	0.06	0.62	237.25
		40	0.02%	0.02%	0.24	1.37	1.91	0.02%	0.02%	0.16	1.50	3.61	0.37%	0.23%	0.10	1.75	82.77
	0.8	10	0.84%	0.68%	0.01	0.31	16.26	1.40%	1.33%	0.01	0.27	241.64	1.52%	1.52%	0.02	0.23	789.06
		20	0.39%	0.38%	0.04	0.72	21.75	0.50%	0.43%	0.06	0.66	66.91	0.82%	0.80%	0.04	0.58	2508.09
		40	0.18%	0.16%	0.14	1.90	12.51	0.14%	0.13%	0.16	1.79	13.50	0.19%	0.19%	0.14	1.59	176.78

‡ Each cell represents the average measure of ten random problems

* Opt. Gap = (heuristic objective value- opt. objective value)/opt. objective value

† Time for Xpress-MP to find and verify the optimal solution using the tight GR formulation

Hence, inventory costs, which are a smaller portion of total schedule costs at higher setup cost levels, are the major determinant of algorithm performance. Therefore, scheduling a family in a non-optimal time period, results in a lower percent optimality gap for higher values of family setup cost.

Finally, SAM improves the solution of six-phase heuristic at every treatment level (108 combinations of experimental factors) and its average gap and solution time is less than GR's Lagrangian/dual-ascent heuristic. Therefore based on this experimental design, SAM is a better heuristic for CCLSP.

6.3.3 Experimental design 2: based on FMT

Prior research (Maes and Van Wassenhove, 1988) suggests that problem difficulty is impacted by the natural time between orders (TBO). The above experiments consist of problems with relatively low TBOs. Hence, to verify the effect of TBO on six-phase and SAM, we conduct a fourth study based on the experimental design of Federgruen et al. (FMT, 2004).

The experimental factors in this design include the three levels of capacity utilization, CU $\in$ {0.5, 0.75, 0.9}, three levels of item TBOs ($\sqrt{2s_{it'}/hd}$) and three levels of family TBOs ($\sqrt{2S_{t'}/hD}$), where d is the average demand for the item and D is the average demand for the family over the planning horizon. The item and family TBOs are used to generate the respective setup costs, which are constant across all time periods for a test problem. The three levels of the TBOs are {low, medium and high}. The low TBO-values are generated from a uniform distribution on the interval [1, 3]. The medium TBO-values are generated from a uniform distribution on the interval [2, 6] and the high item TBO-values are generated from a uniform distribution on the interval [5, 10]. The available capacity $P_{t'}$ is computed from the *CU* as discussed in previous sections.

Individual demand values are randomly generated from a normal distribution with mean 100 and standard deviation of 10. The inventory holding cost per unit per period is \$1.00. The base set of problems in this study has 10 items and 15 periods. A full factorial design leads to 27 treatments; we randomly generate five test problems for each combination of factors, resulting in a total of 135 test problems. In previous experimental designs we generated ten test problems, while in this test we restrict to five because of its complexity, as most problems in this experiment requires more than an hour to obtain optimal solutions.

Each problem is solved by six-phase, SAM and Xpress-MP version 2003F (Xpress Optimizer Version 14.24). Due to the difficulty of obtaining optimal solutions we tested the capability of the tight GR and GR^{ext} formulations to find and verify optimal solutions for this experimental design. The GR^{ext} formulation was found to be the most efficient formulation for this dataset and hence it is used for benchmarking the performance.

Experimental results

The average optimality gap of SAM is 2.08%, taking less than 0.10 CPU seconds on average to solve test problems. The average optimality gap of six-phase is 9.92%, taking just 0.006 CPU seconds on average to solve this dataset. The standard deviation of the optimality gap for SAM was 1.97%. Sixty-five percent of the problems (88 out of 135) had gaps less than the average optimality gap of SAM. In comparison, the Expanding horizon (EH) heuristic of Federgruen et al. (2004) has an average gap of 1.2% and requires 16.6 CPU seconds on an average. Analogous time for the MIP software (Xpress-MP) to find and verify an optimal solution is 4251.45 CPU seconds. In the optimization software the problems were terminated after 2 hours of run time and the best integer solution obtained is used for calculating the optimality gap. Of the unsolved problems only 18 of the 135 had a MIP gap of more than 1% and the average MIP gap was 0.87%.

The EH heuristic of Federgruen et al. (2004) performs better than six-phase and SAM for the CCLSP in terms of optimality gaps for this set of small test problems. However the solution time increases significantly with the problem size, rendering the EH procedure ineffective for interesting sized problems. For instance, a 25 item-10 period problem, with CU of 0.75, medium item and family TBOs, requires a run time of 20,335 CPU seconds or 5.65 hours (Table 3 in Federgruen et al. 2004). In contrast, the same EH heuristic requires 180 seconds for a problem with 10 items and 25 periods. EH procedure does not provide a feasible approach for solving large sized problems commonly encountered in industry. On the other hand, SAM and the six-phase heuristic handle large problems within a reasonable amount of CPU time and with no degradation in heuristic performance. For example, SAM requires just three seconds to solve a problem with 40 items and 24 time periods (Section 6.3.2 and 6.3.3).

The computational study in Federgruen et al. (2004) considered a small set of test problems because of the computational limitation of EH heuristic and the inability of FMT (Section 5.1) to find optimal problem solutions for benchmarking. They run the weak FMT formulation for 6 hours to obtain a reasonable benchmark estimate, while we ran our tight GR^{ext} formulation for only 2 hours to obtain a better estimate for the same dataset. Federgruen et al. (2004) also report the performance of another heuristic called the strict partitioning (SP) heuristic. This heuristic is runs much faster than EH, requiring less than one CPU second to solve each test problem, but the average optimality gap of SP heuristic is 14.7%.

It is seen that even though SAM has higher average optimality gap than EH, it has its unique advantage in terms of computation requirements. To investigate the performance of SAM and six-phase we present Tables 6.6 and 6.7, which summarize the result for the two heuristics by experimental factors.

Table 6.6: Summary results for the FMT experimental design

Major TBO	Low			Medium			High		
	Six-phase	SAM	Xpress-MP‡	Six-phase	SAM	Xpress-MP‡	Six-phase	SAM	Xpress-MP‡
low item TBO†	1.72%	1.18%	-	1.09%	0.58%	-	0.38%	0.19%	-
CPU time (sec)	0.00	0.14	1938	0.01	0.12	2569	0.01	0.11	3346*
medium item TBO†	10.09%	2.98%	-	5.35%	1.59%	-	2.23%	0.63%	-
CPU time (sec)	0.01	0.10	3376	0.00	0.11	3843*	0.01	0.10	4285*
high item TBO†	29.99%	5.71%	-	20.81%	3.86%	-	11.04%	1.97%	-
CPU time (sec)‡	0.00	0.07	5983*	0.01	0.08	6012*	0.01	0.07	6914*

† - Each cell in the row represents Average Optimality gap ((heuristic objective value- optimal objective value)/optimality objective value) of five test problems
‡ - Optimal or Best integer solution obtained from Xpress-MP using GR^{ext} formulation
* - indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours, the average MIP gap of unsolved problems is 0.82%

The results indicate that problems with high item TBOs are relatively more difficult to solve with six-phase construction heuristic. These results are consistent with the observation made by Maes and Van Wassenhove (1988) that construction heuristics are not effective with problems having high item TBO. The SAM brings the gap of such problems down by as much as 80% on average.

Table 6.7 provides an expanded summary of the results by all three experimental factors and it sheds further light into the performance of the construction and metaheuristic for this experimental study.

The results indicate the heuristic's optimality gap is positively correlated with item TBO and negatively correlated with family (joint setup) TBO. The computational requirements of Xpress-MP are positively correlated with all three experimental factors. The worst case performance for both heuristics occurs at high item TBO levels. At this level the effect of family setup is inconsequential and it reduces to a multi-item problem rather than a coordinated lot sizing problem. SAM's maximum average optimality gap for a particular combination of experimental factors is 6.96%. It occurs at high item TBO, with low family TBO and medium capacity utilization, which is strongly reflective of a MCLSP.

SAM's performance is also affected by capacity utilization. The optimality gap increases as we move from low to high *CU*, the case of low item TBO is an exception to this. This effect of *CU* on metaheuristics is also seen in previous experimental designs and is attributed to the ability of generating feasible neighborhoods, which is inversely correlated with *CU*. Still the improvement of the metaheuristic over six-phase is substantial at high *CU* (approx. 52%).

Table 6.7: Expanded summary results for the FMT experimental design

Capacity utilization (*CU*)			0.5			0.75			0.90		
TBO Family			low	medium	high	low	medium	high	low	medium	high
Low Item TBO	Average Optimality Gap†	Six-phase	1.83%	1.44%	0.52%	1.99%	0.77%	0.27%	1.34%	1.06%	0.35%
		SAM	0.67%	0.48%	0.16%	1.65%	0.68%	0.27%	1.21%	0.58%	0.15%
	CPU Time (sec)	Six-phase	0.000	0.000	0.000	0.003	0.006	0.006	0.009	0.009	0.009
		SAM	0.24	0.20	0.20	0.13	0.11	0.10	0.04	0.04	0.03
		Xpress-MP	23	128	613	400	2014	2727	5391*	5564*	6698*
Medium Item TBO	Average Optimality Gap†	Six-phase	9.64%	5.52%	2.42%	11.59%	5.70%	2.31%	9.04%	4.82%	1.97%
		SAM	2.44%	1.44%	0.55%	2.86%	1.49%	0.57%	3.65%	1.85%	0.76%
	CPU Time (sec)	Six-phase	0.003	0.003	0.009	0.006	0.009	0.009	0.006	0.003	0.009
		SAM	0.18	0.21	0.19	0.09	0.10	0.09	0.03	0.02	0.02
		Xpress-MP	246	762	559	2516	3394*	4920*	7366*	7372*	7375*
High Item TBO	Average Optimality Gap†	Six-phase	38.02%	26.95%	14.52%	35.39%	23.31%	11.81%	16.55%	12.17%	6.78%
		SAM	3.84%	2.05%	1.07%	6.96%	4.48%	2.01%	6.34%	5.06%	2.83%
	CPU Time (sec)	Six-phase	0.003	0.000	0.006	0.003	0.006	0.009	0.006	0.006	0.009
		SAM	0.13	0.13	0.13	0.05	0.07	0.07	0.03	0.03	0.02
		Xpress-MP	3074*	3163*	5921*	7402*	7489*	7412*	7474*	7383*	7410*

† Average Optimality gap for the heuristic = (heuristic objective value- optimal objective value)/optimality objective value
* - indicates that one or more problem instances could not be solved to optimality within the pre-set time limit of 2 hours

6.3.4 Experimental design 4: All classes of dynamic demand lot-sizing problem

This experimental design also follows that in Erenguc (1988), Robinson and Gao (1996) and Gao and Robinson (2004) with the necessary extensions to consider a variety of dynamic demand lot-sizing problems. The test problems consider a wide range of parameter settings and are robust enough to permit evaluation of the heuristic's performance on the ULSP, CLSP, MULSP, MCLSP, CULSP, and CCLSP classes. The experimental factors include different levels of capacity utilization, family setup cost, number of items and planning horizon length.

We assume that demand for each item occurs in 50% of the time periods and randomly generate the periods experiencing demand. Individual demand values are randomly generated from a normal distribution with a mean (standard deviation) of 50(20) units for odd-numbered items and 100(20) units for even-numbered items. The inventory holding cost per unit per period is $1.00. The number of items is represented at five levels where, $I \in \{1, 5, 10, 20, 40\}$ and the planning horizon length is taken from the set $T \in \{12, 18, 24\}$.

Robinson and Lawrence (2004) and Robinson and Gao (1996) indicates that the setup cost ratio (family setup cost divided by the sum of the item setup costs) may impact the quality of the heuristic solutions. Consequently, we study a variety of setup cost ratios in the experiments. In all test problems, the item setup costs, $s_{it'}$ are drawn from a normal distribution with a mean of $60 and a standard deviation of $18.This setup cost varies across items, but is constant in all time periods for a specified item and test problem. The family setup cost, $S_{t'}$ which is constant in all time periods for a test problem, is drawn from a normal distribution with a mean of {$0, $60, $120, $480 or $960} and a standard deviation of $36. When the mean is $S_{t'}$ = $0.00, the standard deviation is set at $0.00. This provides mean setup cost ratios ranging from 0 to 16.

Capacity utilization (CU) is considered at seven levels, $CU \in \{0.05, 0.2, 0.4, 0.6, 0.8, 0.9\}$. This covers a range of capacity intensity, where $CU = 0.05$ approximates an uncapacitated environment and $CU = 0.9$ represents a heavily utilized resource. The resource's capacity per time period, $P_{t'}$ is calculated as discussed in Section 5.5. Backorders are not permitted in the model; hence necessary adjustments are made to make the problem aggregate capacity feasible in each period *t* (details are described in Section 6.3.2.). However, this adjustment is small in all cases and does not significantly alter the capacity utilization level.

The experimental design results in 450 combinations of experimental factors. For each factor combination, ten problems are randomly generated resulting in a total of 4500 test problems. The problems are solved by both the SAM and six-phase heuristic, which are coded in C++. Each test problem is also solved with Xpress-MP version 2003F (Xpress Optimizer Version 14.24), a state of the art optimization software package using the tight GR formulation, shown in Section 5.3. All computations are carried out in a personal computer running a Pentium® 4 processor at 1.9 gigahertz.

Experimental results

The experimental results for the SAM and six-phase heuristics indicate the capability of these heuristics as a one-stop solution approach for solving a variety of dynamic-demand problem classes. Table 6.8 provides the summary results for the 4500 test problems where the performance metric is the heuristic's optimality gap. The optimality gap for the SAM is 0.37%, corresponding to a 37% improvement over six-phase heuristic's optimality gap. The SAM found optimal solutions for 56% (2525 out of 4500) of the test problems versus only 44% (1980 out of 4500) for the six-phase heuristic. Seventy-seven percent of the SA heuristic solutions have optimality gaps lower than the average optimality gap of 0.37% and the standard deviation of the optimality gap is 0.84%. These results demonstrate that both SAM and six-phase are capable of consistently finding exceptionally high quality heuristic solutions.

In addition, the metaheuristic improves upon the quality of the initial solution provided by the six-phase heuristic at minimal computational cost. The computational resource requirements for the SAM average only 0.26 CPU seconds with a maximum of 3.06 CPU seconds. In contrast, the average requirement to find and verify optimal solutions using Xpress-MP's is 281.51 CPU seconds. However, Xpress-MP failed to find and verify an optimal solution for six test problems within 24 hours CPU time. At termination, the optimality gap for these problems averaged 0.15%. Hence, we utilized the best found solutions at that point to evaluate the heuristic's performance.

Table 6.8: Summary results for the 4500 test problems by heuristic procedures for experimental design 4

	Six-phase heuristic	SAM
Average optimality gap	0.59%	0.37%
Standard deviation of the optimality gap	1.17%	0.84%
Maximum optimality gap	11.96%	8.56%
Number of optimal solutions found	1980	2525
Number of solutions within the 0.37% of optimal	3020	3450
Average computational time in CPU seconds	0.016	0.260
Maximum computational time in CPU seconds	0.266	3.060

Summary of the results by experimental factors is provided in Table 6.9. As in experimental design 2, the optimality gaps of SAM and six-phase heuristics are positively correlated with the length of planning horizon and capacity utilization. With the exception of single-item problems, the gaps are negatively correlated with number of items.

The performance of the six-phase heuristic is positively correlated with family setup-cost (excluding when the family setup cost is $0.00), while SAM's performance is not significantly affected by family setup cost. The number of items and the capacity utilization have the greatest impact on the quality of the solutions found by each heuristic.

The impact of experimental factors on the run-time of the heuristics is as seen in the results of GR experimental design (Section 6.3.2). The SAM requires on average less than 10% of the computational resources required by Xpress-MP to find optimal solutions under most factor level settings. However, the SAM requires less than 1% of XpressMP's computational requirements for the more difficult problems to solve to optimality.

Table 6.9: Summary results by experimental factors for experimental design 4

Experimental Factors	Average Optimality Gap†		Improvement*	Time for the heuristic (sec)		Xpress-MP time (sec)
	Six-phase	SAM		Six-phase	SAM	
I = 1	0.28%	0.02%	92.86%	0.001	0.029	1.01
I = 5	1.30%	0.88%	32.31%	0.003	0.074	54.96
I = 10	0.76%	0.51%	32.89%	0.006	0.140	166.56
I = 20	0.45%	0.32%	28.89%	0.017	0.305	1029.75**
I = 40	0.17%	0.12%	29.41%	0.053	0.767	155.27
T = 12	0.51%	0.27%	47.06%	0.004	0.059	2.72
T = 18	0.61%	0.40%	34.43%	0.012	0.208	24.90
T = 24	0.66%	0.45%	31.82%	0.032	0.522	816.92**
S_t = 0	0.51%	0.41%	19.61%	0.017	0.271	148.50
S_t = 60	0.44%	0.34%	22.73%	0.017	0.267	196.68
S_t = 120	0.46%	0.31%	32.61%	0.016	0.265	176.51
S_t = 480	0.67%	0.35%	47.76%	0.016	0.259	284.72
S_t = 960	0.88%	0.46%	47.73%	0.013	0.252	601.15**
CU= 0.05	0.26%	0.02%	92.31%	0.012	0.245	0.65
CU= 0.2	0.33%	0.04%	87.88%	0.012	0.253	0.71
CU= 0.4	0.47%	0.15%	68.09%	0.014	0.268	3.03
CU= 0.6	0.47%	0.31%	34.04%	0.018	0.282	17.28
CU= 0.8	0.79%	0.62%	21.52%	0.022	0.314	67.69
CU= 0.9	1.22%	1.09%	10.66%	0.019	0.217	1599.72**

†- (heuristic objective value- optimal objective value)/optimality objective value
*- Improvement of the SAM optimality gap over Six-phase optimality gap
**-indicates that one or more problem instances could not be solved to optimality within the pre-set time limit

Table 6.9 also indicates the percent improvement of the SA solution over six-phase heuristic solution under all factor level summaries. Capacity utilization has the greatest impact on the effectiveness of the SAM over six-phase heuristic. The improvement drops from 92% to 11% when moving from uncapacitated to tightly capacitated problems. This finding is understood considering that in a tight capacity constrained environment, such as CU = 0.9, a setup must occur in almost every time period. Since the SAM's neighborhood transition scheme is based on perturbing the timing of the family replenishment time periods, there are not as many potential neighboring state spaces to explore for improvement. This finding highlights a general performance characteristic of metaheuristics, such as SAM, when applied to solve capacitated dynamic-demand lot-sizing problems. However, the metaheuristic's improvement is still substantial even at higher capacity utilization levels.

The improvement of SAM over six-phase heuristic is not affected by the number of items, with the exception of single-item problem. However, the relative improvement of the SAM's solutions increase at higher family setup cost levels. This also appears related to the number of possible neighborhood generation moves, where at higher values of the major setup cost there are fewer established family setups and hence more potential opportunities to jump to neighboring areas of the feasible region. The amount of improvement decreases as length of planning horizon (*T*) increases. It also holds true for the GR's experimental design.

Table 6.10: Heuristic performance by problem class

Problem Class	Average Optimality Gap†		Improvement*	Time for the heuristic (sec)		Xpress-MP time (sec)
	Six-phase	SAM		Six-phase	SAM	
ULSP	0.32%	0.01%	96.88%	0.001	0.031	0.031
CLSP	0.28%	0.02%	92.86%	0.001	0.029	1.013
MULSP	0.19%	0.01%	94.74%	0.008	0.151	0.234
MCLSP	0.42%	0.24%	42.86%	0.010	0.163	83.053
CULSP	0.26%	0.02%	92.31%	0.012	0.245	0.645
CCLSP	0.59%	0.37%	37.29%	0.016	0.263	281.512

†- (heuristic objective value- optimal objective value)/optimality objective value
*- Improvement of the SAM optimality gap over Six-phase optimality gap

Table 6.10 breaks the experimental results out by problem class. The six-phase heuristic finds high quality solution across all problem classes in less than 0.01 CPU seconds. But SAM still improves the optimality gap of the six-phase heuristic significantly, ranging from a 37.29% to a 96.88% improvement. As expected the largest percentage improvements are associated with the uncapacitated problem classes. The results indicate that SAM effectively and efficiently solves the most commonly encountered dynamic-demand lot-size problems. Its optimality gaps range from 0.01% to 0.37% across the problem classes.

The consistent high quality solutions across problem classes and the low computational requirements for obtaining them make these heuristics ideal candidates for industrial application within requirements planning systems.

6.4 Summary

The four computational experiments show that the six-phase heuristic performs better than the Lagrangian relaxation based heuristics (Robinson and Lawrence, 2004 and Gao and Robinson, 2004) for the capacitated coordinated lot-sizing problem, but its performance drops significantly as the item TBO level increases. On the other hand, the simulated annealing metaheuristic (SAM) not only improves the six-phase heuristic's performance but also provides a good estimate of the optimality gap at higher levels of TBO. The EH heuristic of Federgruen et al. (2004) provides the best known optimality gap for the capacity constrained coordinated problem but it is not practical to implement in a supply chain planning system, as the computational requirements of this heuristic is highly sensitive to the problem size. In contrast, SAM's CPU requirements are relatively invariant the problem size. We also tested the SAM on one of nation's leading direct (catalogue based) marketers dataset of 239 items and 26 periods and were able to obtain results in less than *14 seconds*. These results strongly suggest the potential application of SAM as a highly efficient and effective solver in logistics, operations, and supply chain planning software.

CHAPTER VII

EVALUATION OF COORDINATED LOT SIZING HEURISTICS UNDER ROLLING HORIZON

Our experimental design is derived from both the MPS rolling schedule and the coordinated lot-sizing literature, as there is no published research for this class of problem in the rolling horizon literature. The environmental factors are based on Robinson et al. (2006) and Federgruen et al. (2004) while the MPS design parameters are based on Zhao et al. (2001) and the performance metrics are based on the study by Sridharan et al. (1988) and Sahin et al. (2004). In this study, we also introduce additional improvements to the experimental factors and performance metrics.

7.1 Experimental design

The experimental design consists of four basic components; environmental factors, MPS design factors, coordinated uncapacitated lot-sizing heuristics and the simulation procedure.

7.1.1 Environmental factors

Four environmental factors, namely, number of items, major (family) TBO, minor TBO and demand lumpiness are considered for this computational study. The number of items is taken from the set $I \in \{4, 8, 12\}$. The demand generation follows Robinson et al. (2006) with necessary modifications to suit this computational study. Demand, d_{it}, is generated from a normal distribution and varies by item and time period. The even numbered items have a mean demand of 50 units and a standard deviation of 20 units and odd numbered items have a mean demand of 100 units and a standard deviation of 20 units. The demand density or lumpiness is tested at three levels, $DD \in \{0.50, 0.75, 1.0\}$. When $DD=0.50$ only 50% of the time periods experience demand, similarly at $DD=0.75$ only 75% of the periods experience demand, while the rest of time periods have zero demand. The mean of the normal distribution that generates the non-zero portion of the demand stream is adjusted, such that they retain the overall average demand. For example, at *DD=0.75*, the mean demand of normal distribution for odd and even numbered items are increased to 67 and 133, thereby maintaining the overall average at 50 and 100 units respectively.

The generation of time between orders (TBO) is based on Maes and Wassenhove (1988) and Federgruen et al. (2004). The major TBO $\left(\sqrt{2S_{t'}/hD}\right)$ is used to generate the joint setup cost $S_{t'}$, where D, is the average demand for the product family and h is the holding cost per unit per time period, which is set at $1.00. The item setup cost $s_{it'}$ is generated from the minor TBO $\left(\sqrt{2s_{it'}/hd}\right)$, where d represents the average demand for the item. Each TBO is evaluated at three levels, {low, medium, high}, whose values are taken from a uniform distribution on the intervals [1, 3], [2, 6] and [5, 10] respectively.

7.1.2 MPS design factors

We use two factors, planning horizon length and frozen interval length, to design our MPS scheduling policy. The planning horizon length (*PH*) is set as an integer multiple, $K \in \{2, 4, 8\}$ of natural order cycle, N, i.e. $PH = K*N$. The natural order cycle length, N, for the coordinated lot-sizing problem is calculated using the expression $\sqrt{2\left(S_{t'} + \sum_{i=1}^{I} s_{it'}\right)/hD}$ presented in Ballou (1998). The frozen interval length, n is defined as a portion F of planning horizon length, i.e. $n = F*PH$. It is evaluated at four levels, $F \in \{0.25, 0.5, 0.75, 1\}$. We assume re-planning is done at the end of the frozen interval which provides both lower schedule cost and stability (Sridharan et al. 1990, Zhao and Lee 1996, Zhao and Lam 1997).

7.1.3 Coordinated uncapacitated lot-sizing heuristics

The nine lot sizing procedures considered in this computational study are:

Four Forward pass heuristics

- FP-E: Forward-pass heuristic using the modified Eisenhut decision criterion
- FP-E-WR: FP-E with the right-shift improvement routine of two phase heuristic
- FP-LV: Forward-pass heuristic using the modified LV decision criterion

FP-LV-WR: FP-LV with the right-shift improvement routine of two phase heuristic

Three FB based heuristics

FB: Fogarty-Barringer heuristic

FB-SK Fogarty-Barringer heuristic with the Silver-Kelle procedure

PM: Boctor et al.'s perturbation metaheuristic initialized by FB-SK

Two Two-phase based heuristics

TP: Two-phase heuristic

SAM: Simulated annealing metaheuristic initialized by TP

The detailed descriptions of these heuristics are provided in Chapter IV.

7.1.4 Simulation procedure

We carry out a full factorial design, resulting in 972 combinations of experimental factors, which include both environmental and MPS design factors. For each combination we generate ten random problems. Each problem is then solved by the nine CULSP heuristics listed in Section 7.1.3, resulting in a total of 87,480 data points for analysis. All the heuristics were coded in C++ and the simulation study is conducted in a laptop running Pentium® M processor at 1.7 GHz. Prior research (Blackburn et al. 1986 and Sridharan et al. 1987) show that experimental run length of 300 time periods eliminates both initialization and termination effects; in this research we use a run length of 400 time periods. Figure 7.1 illustrates the rolling schedule policy for two successive planning cycles.

7.2 Performance metrics

In order to evaluate the performance of the heuristic with respect to total cost, we use a measure called *cost error*. In Sridharan et al. (1987) the cost error is measured as a percentage increase in the total schedule cost over an optimal cost, which was obtained by solving the problem over the entire set of demand data for the simulation experiment. Simpson (2001) argues that such a benchmark solution looses meaning since none of the lot-sizing heuristics under rolling schedule environment could reproduce that total schedule cost. Moreover the computational requirements for finding an optimal schedule for 400 time period multi-item problem is impractical due to the large problem size. Simpson (2001)

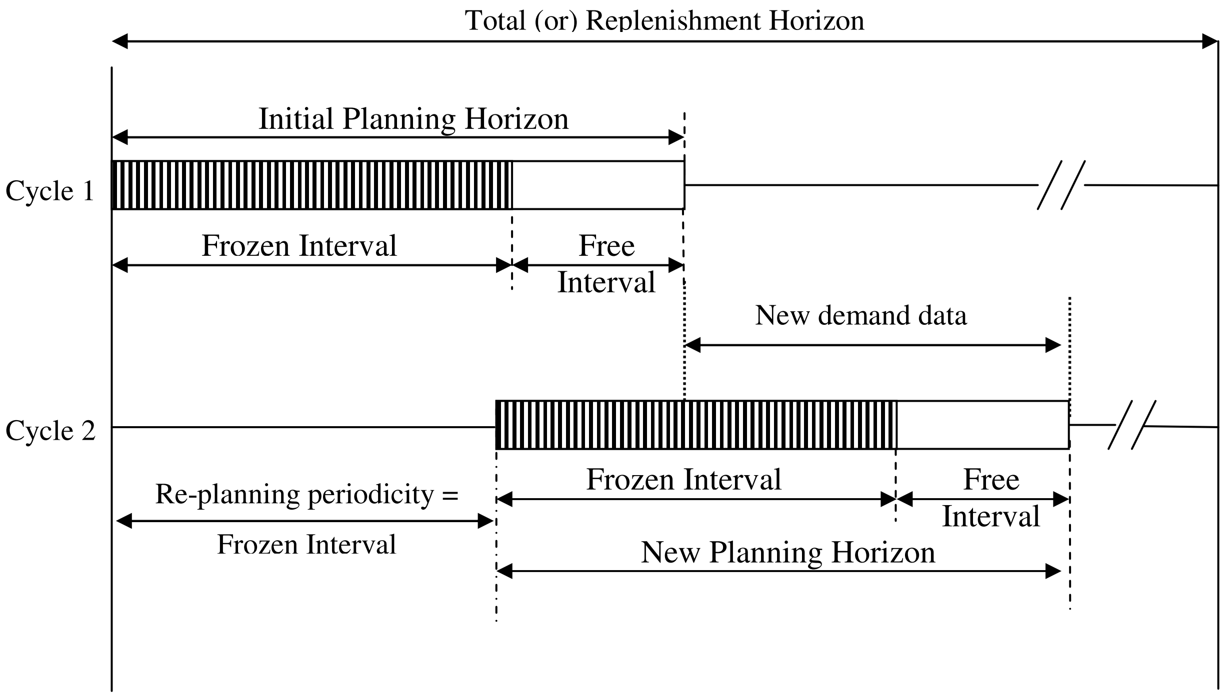


Figure 7.1: Illustration of MPS parameters for the simulation study involving coordinating lot sizing heuristics

suggests the use of constrained application of an optimal algorithm, in which the generated replenishment orders are not permitted to span a length of time longer than the planning horizon. Similarly, we attempted to recursively solve each planning horizon to optimality and apply that replenishment schedule policy in rolling schedule to obtain the total schedule cost for benchmarking. Unfortunately even such an attempt resulted in extreme computational requirements, especially in cases of high item TBOs. Therefore we use a relative cost measure to calculate cost error. The cost error for this experiment is calculated as $[(C_1-C_2)/C_2]*100$, where C_1 is the total cost of rolling schedule for a heuristic and C_2 is the lowest total cost rolling schedule recorded among the nine CULSP heuristics evaluated in this study.

Schedule instability is evaluated using three error measures. *Type 1* error measures system nervousness in terms of number of units rescheduled as a percentage of total number of units in all planning cycles. A similar measure was used in Sridharan et al. (1988). As in Sahin et al. (2004), we define *Type 2* error as a metric that tracks changes in the timing of replenishment orders (item setups) as a percentage of the maximum number of orders that can be executed over the simulation run. We define a new instability metric, *Type 3* error for this experiment. *Type 3* error tracks changes in the timing of joint (product family) setups as a percentage of the maximum number of joint setups that can be executed in a simulation run. All three error measures are calculated as follows,

$$\text{Type 1} = \left(\sum_{i=1}^{I} \sum_{j>1} \sum_{t=M_j}^{M_j+PH-1} \left| Q_{it}^{j} - Q_{it}^{j-1} \right| \right) \Big/ TQ$$

$$\text{Type 2} = \left(\sum_{i=1}^{I} \sum_{j>1} \sum_{t=M_j}^{M_j+PH-1} \left| Y_{it}^{j} - Y_{it}^{j-1} \right| \right) \Big/ U$$

$$\text{Type 3} = \left(\sum_{j>1} \sum_{t=M_j}^{M_j+PH-1} \left| Z_{t}^{j} - Z_{t}^{j-1} \right| \right) \Big/ V$$

where i is the index of item, j is the index of planning cycle, t is the time period, M_j is the starting period of planning cycle j, Q_{it}^{j} is the replenishment order scheduled for item i in time period t during planning cycle j, $Y_{it}^{j} = 1$ if an order is scheduled for item i in time t during planning cycle j, $Z_{t}^{j} = 1$ if a joint setup is scheduled in period t during planning cycle j, TQ is the total number of units in all planning cycles, U is the maximum number of item setups that can be executed over the simulation run and V is the maximum number of joint setups that can be executed in a simulation run.

7.3 Experimental results

The experimental results for the 87,480 test problems are discussed in two distinct parts: impact on cost error and impact on schedule instability.

7.3.1 Impact of experimental factors on cost error performance

Table 7.1 provides the summary result of 9,720 problems for each CULSP heuristic. The results indicate the dominance of perturbation metaheuristic (PM), which has an average gap of 0.37%. SAM ranks second with an average gap if 0.67%. We also see that the right-shift routine improves the performance of both forward pass heuristics. Among the stand alone construction heuristics, FB-SK performs better than FB and two-phase (TP) heuristic, but the performance FB and FB-SK are highly sensitive to demand lumpiness. It is interesting to note that the performance of forward pass and FB based heuristic improves with decrease in demand lumpiness, i.e. as demand density (*DD*) increase the cost error decreases, whereas the exact opposite happens for TP based heuristics. A similar result was shown for static horizon problem in Robinson et al. (2006).

The summary results by experimental factors for forward pass, FB and TP based heuristics are presented in Tables 7.2, 7.3 and 7.4 respectively.

Forward Pass heuristics

In forward pass heuristics without the right shift routine, the cost error drops as major TBO increases. The major TBO indirectly reflects the joint setup cost, hence the forward pass heuristics perform better when this cost is high. This contradicts the findings for forward pass heuristics in static horizon (Table 4.2). The same impact of the major TBO is seen in forward pass heuristics with right-shift routine, except for non-lumpy demand situation. In contrast, the cost errors increase with the minor TBO (item setup cost). This is due to the mechanics of forward pass heuristics; whose costs saving maneuvers are myopic in nature hence the item setups are inherently dominant over joint setup cost. There is a distinct interaction effect between number of items and demand density for forward pass heuristics. For most lumpy demand cases the cost error increase with the number of items but it decreases with the number of items for non-lumpy demand situation. The cost error of forward pass heuristic improves with *DD* across all experimental factors, expect for lower number of items.

Table 7.1: Summary results by demand density for the 9,720 problems: cost error

	Average Cost Error			Std. dev of Cost Error			Max Cost Error			No. of best solutions		
DD	50%	75%	100%	50%	75%	100%	50%	75%	100%	50%	75%	100%
Forward Pass heuristics												
FP-E	5.03%	4.35%	3.72%	5.15%	5.33%	6.13%	23.74%	28.09%	39.53%	260	326	272
FP-E-WR	4.00%	2.92%	2.30%	3.61%	3.72%	5.22%	21.18%	28.09%	39.53%	322	425	380
FP-LV	9.81%	6.24%	3.67%	7.68%	6.43%	4.96%	34.20%	30.63%	32.63%	48	30	116
FP-LV-WR	9.35%	4.94%	1.99%	6.97%	4.96%	4.18%	32.92%	30.60%	32.60%	48	117	675
FB based heuristics												
FB	3.49%	0.63%	0.41%	5.21%	0.99%	0.95%	35.86%	7.95%	7.31%	524	991	1668
FB-SK	2.90%	0.48%	0.36%	4.88%	0.88%	0.87%	31.13%	5.67%	6.27%	892	1429	1940
PM	0.44%	0.32%	0.36%	0.85%	0.73%	0.87%	5.91%	5.67%	6.26%	1790	2109	2073
Two phase based heuristics												
TP	1.56%	1.89%	2.15%	0.85%	0.86%	1.00%	5.41%	6.12%	7.72%	132	114	135
SAM	0.56%	0.70%	0.74%	0.76%	0.77%	0.85%	5.40%	5.67%	6.26%	1167	728	526

Table 7.2: Summary results of cost error by experimental factors: Forward pass heuristics

Experimental Factors		FP-E			FP-E-WR			FP-LV			FP-LV-WR		
DD		50%	75%	100%	50%	75%	100%	50%	75%	100%	50%	75%	100%
Item	4	4.11%	3.79%	4.03%	3.69%	3.38%	3.72%	9.07%	5.55%	3.46%	8.90%	5.07%	2.99%
	8	5.39%	4.47%	3.68%	4.21%	2.73%	1.93%	9.82%	6.02%	3.63%	9.29%	4.44%	1.62%
	12	5.60%	4.77%	3.43%	4.10%	2.64%	1.25%	10.55%	7.17%	3.91%	9.87%	5.32%	1.37%
Major TBO	Low	8.03%	7.19%	5.62%	5.78%	3.96%	2.26%	12.78%	9.58%	5.60%	11.81%	6.77%	2.10%
	Med	4.17%	3.52%	3.17%	3.37%	2.50%	2.29%	9.56%	5.58%	3.19%	9.18%	4.58%	1.89%
	High	2.89%	2.32%	2.36%	2.85%	2.29%	2.35%	7.10%	3.57%	2.21%	7.08%	3.48%	1.98%
Minor TBO	Low	2.69%	2.13%	1.82%	2.64%	1.95%	1.79%	4.18%	2.93%	1.96%	4.17%	2.70%	1.53%
	Med	4.39%	3.95%	3.59%	3.65%	2.78%	2.32%	8.34%	5.61%	3.49%	8.03%	4.57%	1.93%
	High	8.02%	6.96%	5.73%	5.70%	4.02%	2.79%	16.93%	10.19%	5.55%	15.87%	7.55%	2.52%
PH	2	4.37%	4.32%	4.49%	3.50%	3.07%	3.23%	9.43%	6.66%	4.29%	8.97%	5.40%	2.88%
	4	5.16%	4.32%	3.71%	4.06%	2.81%	2.21%	9.88%	5.86%	3.68%	9.41%	4.50%	1.88%
	8	5.56%	4.40%	2.94%	4.43%	2.87%	1.46%	10.14%	6.21%	3.03%	9.69%	4.93%	1.23%
F	0.25	4.58%	3.44%	2.42%	3.37%	1.69%	0.60%	8.92%	5.19%	2.50%	8.35%	3.47%	0.42%
	0.50	4.36%	3.31%	2.35%	3.25%	1.73%	0.67%	8.68%	5.06%	2.45%	8.18%	3.61%	0.51%
	0.75	4.05%	3.08%	2.28%	3.04%	1.75%	0.95%	8.81%	5.03%	2.36%	8.38%	3.85%	0.81%
	1.00	7.14%	7.54%	7.81%	6.33%	6.49%	6.99%	12.85%	9.69%	7.36%	12.51%	8.84%	6.23%

A counterintuitive result is seen for the impact of planning horizon length on cost error performance for forward pass heuristics. The traditional result, of longer planning horizon length leading to lower schedule cost, is true only for non-lumpy (*DD*=1.0) demand distribution, while there is no significant impact seen in lumpy demand cases. The impact of freezing horizon on cost error is as expected, but the effect is not prominent.

The right shift subroutine of two-phase heuristic consistently improves the performance of the corresponding forward pass heuristic for all experimental factors.

FB based heuristics

The relative ranking among the FB based heuristics does not change with respect to the rolling horizon environment; PM performs better than FB-SK, which in turn performs better than FB across all experimental factors. The most important driver of cost error seems to be demand density and its interaction with other design factors. PM's cost errors are fairly low across all experimental factors, with 0.88% being the maximum average cost error for shorter planning horizon at *DD*=1.0. Low number of items and high lumpy demand pattern produces the worst case measure for FB and FB-SK heuristic. PM improves the solution of FB-SK by an average of 92% for such cases. The cost error increases with the item and family TBOs for FB and FB-SK methods in lumpy demand cases. This is a counterintuitive result since FB based procedures are anchored on the assumptions that every item is setup in each joint replenishment period, which is more characteristic of optimal solutions for problems with higher joint setup costs (higher major TBOs). On the other hand such counterintuitive results are not seen in PM and non-lumpy demand cases, where the performance is rather invariant across the TBO values.

As expected, the longer planning horizon produces the least error measure. In contrast, the effect of frozen interval length is unexpected for PM and non-lumpy demand cases. In these situations the lowest cost error is obtained for longer frozen intervals.

Table 7.3: Summary results of cost error by experimental factors: FB based heuristics

Experimental Factors		FB			FB-SK			PM		
DD		50%	75%	100%	50%	75%	100%	50%	75%	100%
Item	4	8.60%	0.94%	0.36%	7.72%	0.88%	0.36%	0.55%	0.46%	0.36%
	8	1.09%	0.70%	0.75%	0.65%	0.47%	0.64%	0.48%	0.42%	0.64%
	12	0.77%	0.24%	0.11%	0.34%	0.08%	0.08%	0.30%	0.07%	0.08%
Major TBO	Low	2.71%	0.80%	0.48%	1.56%	0.43%	0.36%	0.76%	0.31%	0.35%
	Med	2.78%	0.53%	0.40%	2.39%	0.46%	0.38%	0.36%	0.32%	0.38%
	High	4.98%	0.55%	0.34%	4.75%	0.54%	0.34%	0.20%	0.33%	0.34%
Minor TBO	Low	2.63%	0.56%	0.46%	2.59%	0.49%	0.44%	0.28%	0.38%	0.44%
	Med	3.53%	0.72%	0.43%	3.00%	0.53%	0.36%	0.55%	0.36%	0.35%
	High	4.31%	0.59%	0.33%	3.11%	0.40%	0.29%	0.50%	0.21%	0.28%
PH	2	4.39%	1.22%	1.00%	3.73%	0.98%	0.88%	0.82%	0.76%	0.88%
	4	3.16%	0.40%	0.20%	2.62%	0.31%	0.19%	0.30%	0.16%	0.18%
	8	2.91%	0.25%	0.03%	2.36%	0.14%	0.02%	0.21%	0.04%	0.02%
F	0.25	3.00%	0.67%	0.58%	2.61%	0.54%	0.53%	0.51%	0.42%	0.53%
	0.50	3.08%	0.68%	0.59%	2.67%	0.57%	0.55%	0.52%	0.45%	0.54%
	0.75	3.50%	0.68%	0.40%	2.91%	0.52%	0.36%	0.54%	0.37%	0.35%
	1.00	4.37%	0.47%	0.06%	3.42%	0.27%	0.01%	0.20%	0.03%	0.00%

TP based heuristics

Most of the findings regarding the effect of environmental factors on TP based heuristics in static horizon holds true for rolling horizon environment. The cost error metric increases with demand density and major TBO (joint setup cost). The only surprising result is the effect of the number of items. In static horizon the cost error measure decreases with the increase in number of items while such a result is not seen in rolling schedules.

The MPS design policy parameters do not affect the performance of TP heuristic significantly, but on the other hand a counterintuitive result is seen for the frozen interval length in SAM. Lower schedule cost errors are produced for longer frozen intervals.

Table 7.4: Summary results of cost error by experimental factors: TP based heuristics

Experimental Factors		TP			SAM		
DD		50%	75%	100%	50%	75%	100%
Item	4	1.41%	1.61%	1.74%	0.71%	0.78%	0.61%
	8	1.61%	2.06%	2.48%	0.52%	0.76%	1.00%
	12	1.67%	2.00%	2.24%	0.45%	0.56%	0.61%
Major TBO	Low	1.27%	1.46%	1.91%	0.26%	0.41%	0.64%
	Med	1.63%	1.97%	2.25%	0.63%	0.76%	0.77%
	High	1.79%	2.24%	2.30%	0.80%	0.94%	0.81%
Minor TBO	Low	1.46%	1.85%	2.13%	0.66%	0.82%	0.82%
	Med	1.47%	1.89%	2.16%	0.58%	0.68%	0.72%
	High	1.76%	1.93%	2.17%	0.45%	0.60%	0.68%
PH	2	1.52%	1.76%	1.97%	0.60%	0.73%	0.87%
	4	1.55%	1.88%	2.29%	0.28%	0.36%	0.40%
	8	1.62%	2.04%	2.20%	0.81%	1.01%	0.95%
F	0.25	1.67%	2.05%	2.38%	0.64%	0.84%	0.95%
	0.50	1.69%	2.02%	2.29%	0.64%	0.85%	0.93%
	0.75	1.53%	1.84%	2.09%	0.64%	0.74%	0.72%
	1.00	1.35%	1.65%	1.87%	0.33%	0.38%	0.36%

Of all the results, the most important finding is the dominance of PM over SAM across all experimental factors in rolling schedule implementation. A t-test[§] further shows that there is significant (*p*-value 0.001) evidence that PM performs better than SAM. The only setting in which SAM edges PM is at lower joint setup cost, which was seen in static horizon experiments (Table 4.2). To verify this result we evaluate the heuristics under a different set of experimental factors, and their results are discussed in Section 7.4.

In order to understand the interaction effects of design factors on cost error metric, we analyze the results using the ANOVA procedure. To satisfy the assumptions of ANOVA, we apply the transformation of $1/x^2$ suggested by Yeo and Johnson (2000) on the dependent variable (cost error). The two best heuristics, PM and SAM, are considered for this analysis. The ANOVA results are presented in Table 7.5.

As illustrated in Table 7.5, the choice of heuristic followed by MPS design factors and number of items emerged as important main effects. The interaction effects such as PH*Heuristic, Major TBO*Heuristic, Item**PH*, *PH*F*, *DD**Heuristic have the dominant impact. Almost all of the 2-way and 3-way interactions were significant, but the F-ratio values for most of the interactions are low, indicating little impact on the cost error metric. The interesting finding from the ANOVA analysis is that, the performance of SAM and PM with respect to schedule cost are greatly affected by MPS design policies rather than environmental factors such as setup costs and demand lumpiness. Also the choice of heuristic plays a major role in determining the cost of rolling schedules for coordinated lot sizing problems, a result seen earlier in our analysis.

However, some of these ANOVA results might not apply to other procedures like forward pass and FB-SK heuristics. For these procedures, Tables 7.2 and 7.3 clearly show that the environmental factors such as items, *DD* and TBOs have greater impact on cost error than MPS design policies.

[§] $H_o : \mu_{SAM} - \mu_{PM} \leq 0$; $H_a : \mu_{SAM} - \mu_{PM} > 0$. Using the averages and standard deviations ($\overline{y_{SAM}} = 0.6678$, Std. $dev._{SAM}$= 0.799; $\overline{y_{PM}} = 0.3727$, Std. $dev._{PM}$= 0.824; sample size = 9720), we compute the value of *t*' for *t*-test. t' > t (*p*-value : 0.001), hence we reject H_0.

Table 7.5: ANOVA results for cost error †

Dependent Variables‡	DF	F	Sig
ITEM	2	492.32	0.000
MAJOR	2	139.03	0.000
MINOR	2	33.06	0.000
DD	2	35.83	0.000
PH	2	548.30	0.000
F	3	1184.27	0.000
HEURISTIC	1	6622.68	0.000
ITEM * MAJOR	4	38.31	0.000
ITEM * MINOR	4	12.83	0.000
ITEM * DD	4	13.02	0.000
ITEM * PH	4	772.79	0.000
ITEM * F	6	214.80	0.000
ITEM * HEURISTIC	2	121.90	0.000
MAJOR * MINOR	4	11.39	0.000
MAJOR * DD	4	24.64	0.000
MAJOR * PH	4	43.39	0.000
MAJOR * F	6	7.35	0.000
MAJOR * HEURISTIC	2	1222.26	0.000
MINOR * DD	4	16.54	0.000
MINOR * PH	4	44.27	0.000
MINOR * F	6	17.57	0.000
MINOR * HEURISTIC	2	104.90	0.000
DD * PH	4	8.24	0.000
DD * F	6	32.75	0.000
DD * HEURISTIC	2	370.44	0.000
PH * F	6	565.67	0.000
PH * HEURISTIC	2	4052.73	0.000
F * HEURISTIC	3	5.89	0.001
ITEM * MAJOR * F	8	34.41	0.000
ITEM * MINOR * DD	4	56.00	0.000
ITEM * MINOR * PH	8	16.05	0.000
ITEM * MINOR * F	8	17.28	0.000
ITEM * DD * F	8	52.57	0.000
ITEM * PH * F	4	34.12	0.000
ITEM * PH * HEURISTIC	12	99.28	0.000
ITEM * F * HEURISTIC	4	102.67	0.000
MAJOR * MINOR * F	8	23.57	0.000
MAJOR * DD * PH	4	52.56	0.000
MAJOR * PH * F	4	277.13	0.000
MAJOR * F * HEURISTIC	4	105.62	0.000
MINOR * PH * F	4	104.81	0.000
DD * PH * HEURISTIC	12	19.84	0.000
DD * F * HEURISTIC	4	19.93	0.000

Adjusted $R^2 = 0.672$, intercept was included in the model

† Based on Yeo and Johnson(2000) suggestions, inverse of square root transformation of cost error was considered to satisfy the assumptions of ANOVA.

‡ Heuristics considered are SAM and PM. All 3-way interactions were considered, only partial results of the 3-way interaction are presented in the table (with F > 15)

Simpson (1999) points out one disadvantage of using a relative cost measure in cost error calculation. If all the lot-sizing heuristics included in the study react the same way to a particular factor level, then the effect would go unnoticed in the analysis. We know that the CULSP heuristics included in our study react differently to environmental factors (Table 4.2), but we have no such evidence for MPS design policies. This could be the reason for some of the unexpected results with respect to planning horizon and frozen interval length. Hence there is a need to find a common benchmark, which is computationally reasonable, to unambiguously identify the effect of MPS design policies on cost error performance for coordinated lot-sizing heuristics. But the relative ranking among the heuristics will not be affected by the change in calculation of this metric.

7.3.2 Impact of experimental factors on schedule instability

Table 7.6 provides the summary results for schedule instability for the CULSP heuristics. Each cell in the table represents the average of 3,240 test problems. As expected, the forward pass heuristics produces the most stable rolling schedule, followed by forward pass procedures with right shift routine. The metaheuristics provide the most unstable schedules, especially SAM which has the worst schedule instability performance. An interesting result is the value of the *Type 3* metric for forward pass heuristics such as FP-E and FP-LV. The findings indicate that there are no joint setup reschedules in our entire simulation study for these procedures. In some situations, a firm may not have the ability to change a joint setup, such as when a truck or vessel is chartered for shipment, but can change the number of units loaded in them, in such scenarios forward pass heuristics could be the best procedure to adopt for its rolling schedule policies.

We consider only *Type 1* error measure for the rest of our analysis, since the correlation results in Table 7.7 shows it to be a good substitute for *Type 2* and *3* instability measures. *Type 2* has the highest correlation with other error measures, but it was not chosen as *Type 1* measure has more details and higher range than its counterparts.

Tables 7.8, 7.9 and 7.10 presents the expanded summary results of schedule instability for forward pass, FB and TP based heuristics by experimental factors.

Table 7.6: Summary results by demand density for the 9,720 problems: schedule instability

	Type 1 error[A]			Type 2 error[B]			Type 3 error[C]		
DD	50%	75%	100%	50%	75%	100%	50%	75%	100%
Forward Pass heuristics									
FP-E	16.89%	18.67%	20.54%	0.70%	0.41%	0.13%	0.00%	0.00%	0.00%
FP-E-WR	21.25%	27.93%	31.48%	1.53%	2.29%	2.27%	1.02%	2.08%	2.29%
FP-LV	13.37%	16.81%	19.68%	0.87%	0.51%	0.21%	0.00%	0.00%	0.00%
FP-LV-WR	15.47%	25.65%	33.50%	1.35%	2.46%	2.87%	0.58%	2.16%	2.82%
FB based heuristics									
FB	25.26%	26.89%	24.24%	2.66%	3.20%	2.98%	2.58%	3.19%	2.98%
FB-SK	25.33%	26.56%	24.24%	2.63%	3.09%	2.95%	2.62%	3.12%	2.96%
PM	26.60%	27.18%	24.68%	2.88%	3.21%	3.02%	2.97%	3.26%	3.04%
Two phase based heuristics									
TP	23.10%	25.16%	27.43%	2.18%	2.46%	2.71%	2.17%	2.46%	2.71%
SAM	33.56%	37.81%	40.87%	3.72%	4.47%	5.06%	4.02%	4.60%	5.12%

[A] Based on demand/unit change
[B] Based on item/minor setup change
[C] Based on family/joint setup change

Table 7.7: Correlation results for schedule instability measures

SI	Type 1	Type 2	Type 3
Type 1	1	0.855*	0.843*
Type 2		1	0.992*
Type 3			1

* Correlation is significant at the 0.01 level (2-tailed), sample size = 87,480

Forward pass heuristics

The results indicate the dominance of forward pass procedures over its counterparts with right shift subroutine in schedule stability. Non-lumpy (or) high density demand stream distinctly produce more unstable schedules, this is explained by the presence of increasing number of time periods having positive item demand. Schedule stability increases with the number of items only for the case of forward pass heuristics with the Eisenhut criterion. The major and minor TBOs affect the schedule stability for the forward pass heuristics with the right shift subroutine, while it is not true for its predecessors. The stability of rolling schedules is positively correlated with joint setups (major TBOs) and is negatively correlated with item setups (minor TBOs).

As expected, the stability of the rolling schedule increases with the length of frozen interval. On the other hand, we see a surprising result for the effect of planning horizon on schedule stability. Longer planning horizons produces the most stable schedules for forward pass heuristics; this contradicts the findings in literature for multi-item problems. Such an effect is not evident across all *DD* for heuristics with right-shift subroutine.

FB based heuristics

Among the environmental factors, the number of items has the greatest impact on schedule instability. The stability of rolling schedules decreases with an increase in number of items. The schedule stability also decreases with an increase in major and minor TBOs, but their

Table 7.8: Summary results of schedule instability (Type 1 error) by experimental factors: Forward pass heuristics

Experimental Factors		FP-E			FP-E-WR			FP-LV			FP-LV-WR		
DD		50%	75%	100%	50%	75%	100%	50%	75%	100%	50%	75%	100%
Item	4	18.70%	25.24%	30.78%	19.72%	27.03%	32.26%	16.62%	22.63%	27.02%	17.22%	24.67%	29.84%
	8	17.85%	17.75%	19.85%	22.65%	28.10%	32.73%	12.13%	13.26%	18.86%	14.08%	23.31%	34.36%
	12	14.13%	13.03%	10.98%	21.37%	28.65%	29.45%	11.37%	14.55%	13.18%	15.13%	28.96%	36.31%
Major TBO	Low	15.46%	17.59%	19.76%	25.07%	37.43%	43.55%	12.18%	16.08%	19.15%	16.40%	34.08%	44.23%
	Med	17.14%	18.80%	21.07%	20.39%	26.23%	29.93%	13.65%	17.31%	19.78%	15.66%	24.97%	32.77%
	High	18.07%	19.63%	20.77%	18.28%	20.11%	20.97%	14.29%	17.05%	20.12%	14.36%	17.90%	23.51%
Minor TBO	Low	16.38%	17.13%	19.35%	16.41%	17.62%	19.46%	13.83%	17.37%	18.54%	13.83%	17.78%	20.23%
	Med	16.96%	18.49%	20.50%	19.23%	25.65%	29.47%	13.72%	17.42%	19.62%	14.52%	23.44%	31.86%
	High	17.35%	20.40%	21.75%	28.11%	40.51%	45.51%	12.58%	15.66%	20.88%	18.06%	35.72%	48.42%
PH	2	23.86%	25.49%	28.92%	25.32%	26.89%	28.24%	20.84%	27.13%	27.95%	22.60%	29.88%	27.75%
	4	17.27%	19.65%	22.28%	22.28%	31.12%	36.74%	12.77%	14.95%	21.63%	15.19%	24.67%	39.63%
	8	9.56%	10.88%	10.41%	16.14%	25.76%	29.46%	6.52%	8.37%	9.46%	8.62%	22.39%	33.13%
F	0.25	30.64%	33.87%	36.89%	43.89%	62.22%	70.50%	25.79%	30.91%	35.63%	32.00%	57.33%	78.11%
	0.50	22.07%	24.08%	26.04%	25.75%	31.77%	34.88%	16.84%	21.27%	25.07%	18.59%	28.93%	36.25%
	0.75	14.87%	16.73%	19.21%	15.35%	17.71%	20.54%	10.86%	15.08%	18.03%	11.31%	16.33%	19.65%
	1.00	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 7.9: Summary results of schedule instability (Type 1 error) by experimental factors: FB based heuristics

Experimental Factors		FB			FB-SK			PM		
DD		50%	75%	100%	50%	75%	100%	50%	75%	100%
Item	4	19.85%	20.14%	18.86%	20.55%	20.10%	18.85%	21.71%	20.25%	18.91%
	8	26.67%	29.15%	25.22%	26.63%	28.74%	25.24%	28.11%	29.90%	26.11%
	12	29.27%	31.37%	28.64%	28.82%	30.85%	28.63%	29.97%	31.40%	29.00%
Major TBO	Low	23.53%	25.88%	22.32%	23.66%	25.13%	22.36%	25.67%	26.58%	23.61%
	Med	25.81%	27.02%	23.70%	25.72%	26.76%	23.67%	26.59%	27.29%	23.72%
	High	26.45%	27.76%	26.70%	26.62%	27.80%	26.70%	27.53%	27.68%	26.70%
Minor TBO	Low	24.31%	25.70%	22.63%	24.33%	25.53%	22.68%	25.01%	25.75%	22.76%
	Med	25.31%	27.65%	24.21%	25.32%	27.11%	24.24%	26.41%	28.08%	24.89%
	High	26.17%	27.31%	25.88%	26.35%	27.04%	25.82%	28.36%	27.72%	26.38%
PH	2	15.44%	9.98%	6.06%	15.19%	9.55%	6.16%	11.09%	9.07%	6.16%
	4	31.25%	29.19%	21.55%	31.25%	28.87%	21.50%	32.31%	30.07%	22.45%
	8	29.10%	41.50%	45.11%	29.56%	41.27%	45.07%	36.39%	42.41%	45.41%
F	0.25	64.32%	71.61%	63.56%	65.38%	71.09%	63.64%	71.95%	73.43%	64.99%
	0.50	28.83%	30.24%	28.54%	28.56%	29.66%	28.51%	28.58%	29.99%	28.87%
	0.75	7.90%	5.70%	4.86%	7.39%	5.49%	4.82%	5.86%	5.31%	4.85%
	1.00	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

effect is not prominent. The results for demand density do not reveal any discernable pattern.

Shorter planning horizons and longer frozen intervals produce more stable schedules for all FB based heuristics. Also we see a marked improvement in stability, when the frozen interval length is more than 50% of the planning horizon. This supports the results in multi-item literature (Sridharan et al. 1987).

TP based heuristics

Table 7.10 presents the expanded summary results of schedule instability for TP based heuristics by experimental factors. SAM produces the most unstable schedules for all test problems evaluated in this study. As in FB based heuristics, the number of items has the greatest impact on schedule stability. The schedule instability is positively correlated with all environmental factors, namely the number of items, demand density, major and minor TBOs. When compared to the two-phase heuristic, these effects are magnified in SAM.

The effects of frozen interval and planning horizon on schedule stability are expected with few surprises in the two-phase heuristics. For the most part, shorter planning horizon produces more stable schedules than longer ones, but this is not true for every level of PH in two-phase heuristic. Like FB based heuristics, we see a substantial improvement in schedule stability as the frozen interval approaches the length of planning horizon.

In order to identify the interaction effects of design factors on schedule stability we analyze the results using an ANOVA procedure. For dependent variable (Type 1 error measure of schedule instability), we apply the transformation of ln(x+1) suggested by Yeo and Johnson (2000) to satisfy the assumptions of ANOVA. PM and SAM are considered for this analysis, since they produce the least cost error measures. The ANOVA results are given in Table 7.11.

The major drivers of schedule instability are MPS design factors, such as PH and F, number of items and select interaction effects involving MPS design parameters. This shows that for heuristics such SAM and PM, the MPS design parameters have the major impact on

Table 7.10: Summary results of schedule instability (Type 1 error) by experimental factors: TP based heuristics

Experimental Factors		TP			SAM		
DD		50%	75%	100%	50%	75%	100%
Item	4	19.17%	19.94%	21.57%	23.94%	24.74%	26.63%
	8	25.30%	27.85%	30.49%	35.38%	40.28%	40.70%
	12	24.84%	27.68%	30.23%	41.36%	48.43%	55.26%
Major TBO	Low	21.69%	23.71%	26.38%	30.06%	32.71%	36.28%
	Med	23.13%	25.26%	27.56%	33.48%	37.88%	41.09%
	High	24.48%	26.50%	28.35%	37.14%	42.85%	45.23%
Minor TBO	Low	23.07%	25.05%	26.57%	31.51%	35.42%	38.47%
	Med	22.95%	25.27%	27.45%	33.22%	38.38%	40.60%
	High	23.29%	25.15%	28.25%	35.94%	39.63%	43.53%
PH	2	15.34%	14.17%	13.70%	12.06%	9.51%	6.37%
	4	30.61%	33.69%	36.96%	37.19%	39.20%	37.42%
	8	23.36%	27.62%	31.62%	51.43%	64.73%	78.82%
F	0.25	59.14%	66.24%	72.82%	94.47%	108.78%	119.56%
	0.50	26.14%	27.31%	29.98%	33.14%	36.50%	38.21%
	0.75	7.13%	7.08%	6.90%	6.63%	5.97%	5.69%
	1.00	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

both cost error and schedule instability. The number of items is the only main environmental factor that affects the metaheuristics on both performance metrics. The rest of the environmental factors come to play only in interaction effects. Almost all of the 2-way and 3-way interactions are significant, but most of their F-values indicate only marginal impact on the variance. Among the interactions, effects involving MPS design parameters and number of items such as, PH*F, Item*F, Item*PH, Item*PH*Heuristic provide the dominant effect on schedule instability. Even though the analysis was conducted for metaheuristics, similar results could be seen for the other CULSP heuristics in Tables 7.7 and 7.9.

Table 7.11: ANOVA results for schedule instability (Type 1 error) †

Dependent Variables‡	DF	F	Sig
ITEM	2	9807.58	0.000
MAJOR	2	13.52	0.000
MINOR	2	15.09	0.000
DD	2	620.99	0.000
PH	2	40433.19	0.000
F	3	120069.69	0.000
HEURISTIC	1	630.78	0.000
ITEM * MAJOR	4	37.23	0.000
ITEM * MINOR	4	3.86	0.004
ITEM * DD	4	61.08	0.000
ITEM * PH	4	3160.70	0.000
ITEM * F	6	1633.40	0.000
ITEM * HEURISTIC	2	69.21	0.000
MAJOR * MINOR	4	1.54	0.188
MAJOR * DD	4	74.92	0.000
MAJOR * PH	4	69.68	0.000
MAJOR * F	6	96.88	0.000
MAJOR * HEURISTIC	2	2.69	0.068
MINOR * DD	4	18.70	0.000
MINOR * PH	4	2.06	0.083
MINOR * F	6	89.28	0.000
MINOR * HEURISTIC	2	0.70	0.499
DD * PH	4	497.55	0.000
DD * F	6	103.12	0.000
DD * HEURISTIC	2	17.23	0.000
PH * F	6	5517.74	0.000
PH * HEURISTIC	2	161.20	0.000
F * HEURISTIC	3	156.40	0.000
ITEM * MAJOR * MINOR	6	31.48	0.000
ITEM * MAJOR * DD	8	22.91	0.000
ITEM * MAJOR * F	8	20.14	0.000
ITEM * MAJOR * HEURISTIC	12	20.05	0.000
ITEM * DD * F	8	58.48	0.000
ITEM * DD * HEURISTIC	12	25.59	0.000
ITEM * PH * HEURISTIC	12	1417.13	0.000
MAJOR * MINOR * DD	6	15.35	0.000
MAJOR * MINOR * HEURISTIC	12	19.94	0.000
MAJOR * DD * F	8	58.07	0.000
MAJOR * PH * HEURISTIC	12	15.68	0.000
DD * PH * HEURISTIC	12	118.15	0.000

Adjusted R^2 = 0.966, intercept was included in the model

† Based on Yeo and Johnson(2000) suggestions, ln(x+1) transformation of Type 1 error metric was considered to satisfy the assumptions of ANOVA.

‡ Heuristics considered are SAM and PM. All 3-way interactions were considered, only partial results of the 3-way interaction are presented in the table (with F > 15)

Finally figure 7.2 shows the relative trade-off between cost and instability of rolling schedules. FP-LV produces the most stable schedule, but has the worst performance with respect to cost error. In contrast, PM, which has the least measure of cost error, is at least 50% more unstable than FP-LV. Six of the nine CULSP heuristics have more or less same level of instability in their rolling schedules. Among them, the perturbation metaheuristic stands out as it produces the least cost schedules. SAM which ranks a close second on cost error produces the most unstable schedules for rolling horizon problems.

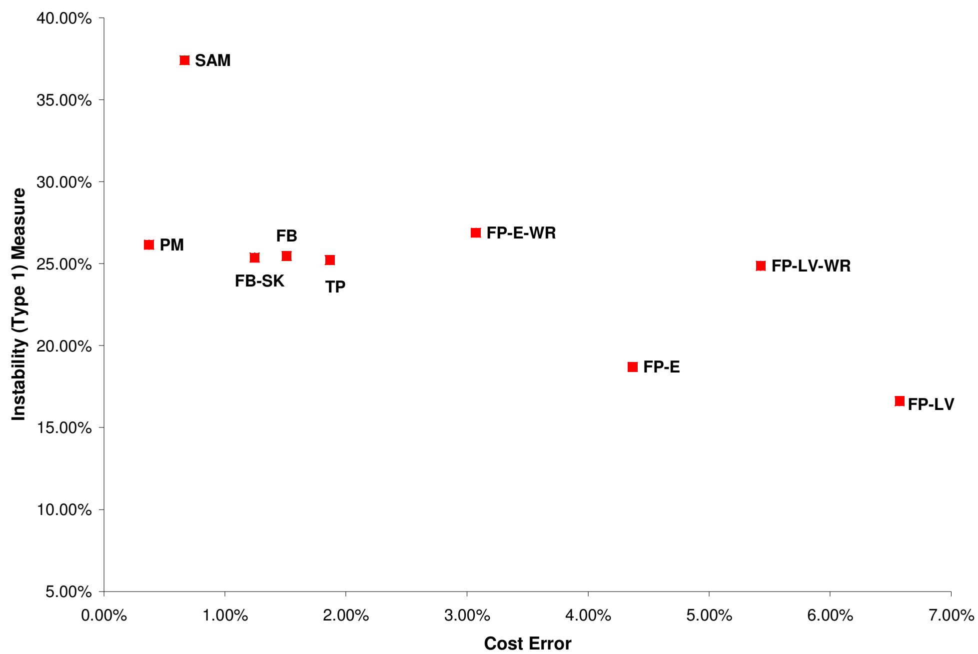


Figure 7.2: Cost error (vs) schedule instability

An important result of the computational study is that, PM outperforms SAM in rolling horizon environment. It is surprising since SAM was the best performing heuristic in static horizon (Table 4.2). To further verify this result we conduct another computational study involving these two metaheuristics using a different dataset which has a unique mix of all levels of item TBO.

7.4 Experimental design: ABC dataset

The four basic design components; environmental factors, MPS design factors, coordinated uncapacitated lot-sizing heuristics and the simulation procedure are described in this section.

7.4.1 Environmental factors

All test problems have 10 items. The first two items have a low TBO (A-items); the next three items 3-5 have a medium TBO (B-items) and the last five items (6-10) have a high TBO (C items). The item setup costs, $s_{it'}$, are drawn from a normal distribution with a mean of \$350 and a standard deviation of \$35. The setup cost varies across items, but is constant in all time periods for a specified item within a test problem. The ABC stratification is obtained by varying the average demand of the individual items. The average demand for the product family per time period is 1000 units. The first two items have an average demand of 400 units each, while the next three have an average demand of 50 units and the final five have an average demand of 5 units each. As in Robinson et al. (2006) we consider two levels of demand density, $DD \in \{0.50, 1.0\}$. The joint setup costs are obtained from the major TBO which is evaluated at three levels, major TBO $\in$ {low, medium, high}. Details regarding the generation of TBO values are explained in Section 7.1.1.

7.4.2 MPS design factors

Similar to the previous computational study, we consider three levels of planning horizon length, $PH \in \{2, 4, 8\}$ and four levels of frozen interval length, $F \in \{0.25, 0.5, 0.75, 1\}$. *PH* represents the integer multiples of natural cycle, whose computational details are presented in Section 7.1.2, while *F* represents the portion of planning horizon that is frozen.

7.4.3 Coordinated uncapacitated lot-sizing heuristics

SAM and PM are considered for this computational study

7.4.4 Simulation procedure

We utilize a full factorial design, resulting in 72 combinations of experimental factors. For each combination, ten test problems are randomly generated. To avoid initialization and termination effects, each test problem has a run length of 400 time periods. The study is conducted on a personal laptop running Pentium® M processor at 1.70Ghz.

7.5 Performance metrics

We use the cost error and *Type 1* schedule instability error measure defined in Section 7.2 for this analysis.

7.6 Experimental results: ABC dataset

Table 7.12 summarizes the experimental results for SAM and PM by experimental factors and performance metrics. SAM has the least cost solutions for 363 problems, while PM finds for 394 problems (out of 720).

Table 7.12: Summary results by experimental factors for ABC dataset: SAM vs. PM

Experimental Factors		Cost Error		Schedule Instability†	
		SAM	PM	SAM	PM
	Overall Average	0.36%	0.42%	39.72%	27.40%
Major TBO	Low	0.07%	1.14%	32.40%	29.04%
	Med	0.36%	0.12%	39.42%	26.53%
	High	0.66%	0.02%	47.34%	26.63%
PH	2	0.10%	0.28%	13.00%	13.35%
	4	0.30%	0.45%	45.53%	31.62%
	8	0.69%	0.54%	60.63%	37.23%
F	0.25	0.35%	0.50%	112.48%	73.52%
	0.50	0.38%	0.45%	37.88%	29.37%
	0.75	0.42%	0.40%	8.52%	6.71%
	1.00	0.25%	0.36%	0.00%	0.00%
DD	50%	0.33%	0.50%	30.12%	24.85%
	100%	0.40%	0.35%	49.32%	29.95%

†Type 1 measure is used

For this computational study, SAM performs marginally better than PM in terms of schedule cost error, but the impact of the environmental factors on the metaheuristics is evident in this result. As in static horizon (Chapter IV), SAM performs better at lower levels of the joint setup cost (low major TBOs), while PM performs better at higher cost levels. The cost error increases with demand lumpiness for PM, whereas it decreases for SAM.

The effect of MPS design parameters on cost error contradicts the results in the literature. Shorter planning horizon and longer frozen interval length produces the least cost error. The use of relative cost measure may explain some of this unexpected result with respect to MPS design parameters. (Simpson 1999)

Again SAM produces the most unstable rolling schedules. The environmental factors have no significant effect on schedule instability for PM, while this is not true for SAM, whose instability increases with major TBO and *DD*. The effect of MPS design parameters on schedule instability is as expected; shorter planning horizon and longer frozen intervals produce the more stable rolling schedules.

Based on the two computational studies, PM appears to be a better choice for implementation in rolling schedules. SAM outperforms PM only in special cases, but its unstable schedules potentially outweigh the benefit we obtain from marginal cost savings. Forward pass heuristics such as FP-E and FP-LV could find applications in unique situations, where changes in joint setups are discouraged in rolling schedules.

CHAPTER VIII
SUMMARY AND CONCLUSION

We developed new efficient mathematical formulations and heuristics for both uncapacitated and capacitated variants of the coordinated replenishment problem.

This research proposed two new formulations (BLR1' and BLR2') for the uncapacitated problem and evaluated their computational efficiency along with the tight formulation of Robinson and Gao (1996). All three formulations yield equally tight LP objective function values. The most efficiently solved formulation for the case with no backorders, BLR1', solves problems 8.34 times more efficiently than its associated weak formulation BLR1. The RG and BLR2' formulations can efficiently model and solve the backorder problem. BLR1' and BLR2' provide new tight formulations for CULSP and invite the development of algorithms which exploit their specialized mathematical structures, thereby providing an important avenue for future research.

For the CULSP, we have developed two new forward-pass heuristics, a two phase heuristic and a simulated annealing metaheuristic. The findings indicate that the two forward-pass heuristics (FP-E and FP-LV) are capable of finding high quality solutions averaging approximately 1.42% and 1.53% from optimality, respectively. However, the new two-phase heuristic finds solutions with an average 0.56% optimality gap, which improves upon the 0.92% optimality gap associated with the FB-SK heuristic, the best known procedure in the prior literature. The simulated annealing metaheuristic with a 0.2% optimality gap also improved upon the 0.87% optimality gap associated with the perturbation metaheuristic reported in earlier research. Overall, the two-phase heuristic and simulated annealing metaheuristic provides highly efficient and effective procedures for solving the combinatorial complex CULSP.

We developed a new mathematical representation of the capacitated variant of the coordinated lot-sizing problem. The new formulation and its extensions provide another set of tight CCLSP formulation but its performance is sensitive to the lumpiness of the demand stream. We find that the Gao and Robinson (2004) formulation is the most efficient CCLSP formulation for use in general purpose optimization software. Our result also indicates the classical formulation found in Federgruen et al. (2004), which is used as a benchmark in some capacitated coordinated literature, is a poor choice for use in optimization software like Xpress-MP or CPLEX.

CCLSP's mathematical complexity, which contains both complicating capacity constraints and joint setup costs, has thwarted past research efforts in their attempt to design effective heuristic and optimization-based approaches for the problem. In this research we develop a computationally efficient six-phase construction heuristic and simulated annealing metaheuristic (SAM) for the single-family CCLSP without backorders. The heuristic procedures integrate, synthesize and extend fundamental concepts from the literature for solving dynamic demand lot-size problems into a comprehensive algorithm capable of solving, in addition to the CCLSP, the ULSP, CLSP, MCLSP and CULSP classes. Over a wide range of parameter values, the six-phase heuristic finds solutions with an average optimality gap of 0.92% and in an average of 0.03 CPU seconds. But its performance drops significantly with the increase in item TBO. On the other hand SAM not only improves the solution of six-phase by an average of more than 50%, but also finds reasonable optimality gaps at high item TBOs. For a set of 6135 test problems, spanning all experimental designs in coordinated capacitated literature, the metaheuristic finds solutions with an average optimality gap of 0.43% and in an average time of 0.25 CPU seconds.

The high quality of the heuristic solutions and the computational efficiency of finding them, document the potential application of the six-phase heuristic and SAM in industrial settings. Considering that data is seldom 100% accurate in practice, questions the validity of investing the extra computational resources required to find optimal solutions when high quality heuristic solution can be consistently found in only a fraction of the time. We envision these heuristics being applied as a stand alone solver, embedded with requirements planning software, and as an upper bounding procedure for complex optimization based algorithms.

We evaluated alternative coordinated lot-size procedures in a rolling schedule environment. The simulation study indicates that, the perturbation metaheuristic (PM) is the most suitable heuristic for implementation in rolling schedules. SAM, which is the best performing CULSP heuristic in static horizon, outperforms PM only in special cases. SAM's unstable schedules outweigh the benefit we obtain from marginal cost savings. Forward pass heuristics such as FP-E and FP-LV could find applications in unique situations, where product family schedule stability is important. The research also calls for the development of tight lower bounds on the lowest possible cost schedule; so that the impact of MPS design policies, on the cost of coordinated rolling schedules policy, could be unambiguously identified. As we see in this study, the relative ranking among the uncapacitated heuristics change when we move from fixed horizon to rolling schedule environment. Hence this research also motivates the evaluation of capacitated coordinated lot-sizing rules in rolling schedule environment.

REFERENCES

Aggarwal, A., J.K. Park. 1993. Improved algorithms for economic lot sizing. *Operations Research* **41** 549-571.

Aksoy, Y., S.S. Erenguc. 1988. Multi-item inventory models with coordinated replenishments: A survey. *International Journal of Production Management* **8** 63-73.

Altay N. 2001. A cross decomposition algorithm for the capacitated dynamic demand coordinated replenishment problem. PhD Dissertation. Department of Information and Operations Management, Texas A&M University, College Station, TX.

Arkin, E., D. Joneja, R. Roundy. 1989. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters* **8** 61-66.

Armentano, V.A., P.M. Franca, M.B. de Toledo. 1999. A network flow model for the capacitated lot-sizing problem. *Omega* **27** 275-284.

Atkins, D., P. Iyogun. 1988. A heuristic with lower bound performance guarantee for the multi-product lot-size problem. *IIE Transactions* **20** 369–373.

Bahl, H.C., L.P. Ritzman, J. N. Gupta. 1987. Determining lot sizes and resource requirements: A review. *Operations Research* **35** 329-344.

Baker, K. R. 1977. An experimental study of the effectiveness of rolling schedules in production planning. *Decision Sciences* **8** 19-27

Baker, K.R., P. Dixon, M.J. Magazine, E.A. Silver. 1978. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. *Management Science* **24** 1710-1720.

Ballou, R. H. 1988. *Business Logistics Management*. 4th edition. Prentice Hall, Upper Saddle River, NJ.

Barany, I., T.J. Van Roy, L. A. Wolsey. 1984. Strong formulations for multi-item capacitated lot sizing. *Management Science* **30** 1255-1261.

Billington, P.J., J.O. McClain, L.J. Thomas. 1983. Mathematical programming approaches to capacity-constrained MRP systems: Review, formulation and problem reduction. *Management Science* **29** 1126-1141.

Bitran, G.R., H. Matsuo. 1986. The multi-item capacitated lot size problem: Error bounds of Manne's formulations. *Management Science* **32** 350-359.

Bitran, G.R., H.H. Yanasse. 1982. Computational complexity of the capacitated lot-size problem. *Management Science* **28** 1174-1186.

Blackburn, J.D., D. H. Kropp, R.A. Millen 1986. A comparison of strategies to dampen nervousness in MRP systems. *Management Science* **32** 413-429.

Blackburn, J. D., R.A. Millen. 1980. Heuristic lot-sizing performance in a rolling schedule environment. *Decision Sciences* **11** 691-701.

Blackburn, J. D., R.A. Millen. 1982a. The impact of a rolling schedule in a multilevel MRP system. *Journal of Operations Management* **2** 125-135.

Blackburn, J. D., R.A. Millen. 1982b. Improved heuristics for multistage requirements planning systems. *Management Science* **28** 44-56.

Boctor, F. F., G. Laporte, J. Renaud. 2004. Models and algorithms for the dynamic demand joint replenishment problem. *International Journal of Production Research* **42** 2667-2678.

Campbell, G.M., V.A. Mabert. 1991. Cyclical schedules for capacitated lot sizing with dynamic demands. *Management Science* **37** 409-427.

Chung, C., J. Flynn, C.M. Lin. 1994. An effective algorithm for the capacitated single item lot size problem. *European Journal of Operational Research* **75** 427-440.

Chung, C., C.M. Lin. 1988. An $O(T^2)$ algorithm for the NI/G/NI/ND capacitated lot size problem, *Management Science* **34** 420-426.

De Matteis, J. J., A.G. Mendoza. 1968. An economic lot-sizing technique: The part-period balancing algorithm. *IBM Systems Journal* **7** 30-46.

Denizel, M., S.S. Erenguc, H.D. Sherali. 1996. Convex envelope results and strong formulations for a class of mixed-integer programs. *Naval Research Logistics* **43** 503-518.

Diaby, M., H. C. Bahl, M.H. Karwan, S. Zionts. 1992a. A Lagrangian relaxation approach for very-large scale capacitated lot-sizing. *Management Science* **38** 1329-1353.

Diaby, M., H.C. Bahl, M.H. Karwan, S. Zionts. 1992b. Capacitated lot-sizing and scheduling by Lagrangian relaxation. *European Journal of Operational Research* **59** 444-458

Dixon, P. S., E.A. Silver. 1981. A heuristic solution procedure for the multi-item, single-level, limited capacity, lot sizing problem. *Journal of Operations Management* **2** 23-39.

Dogramaci, A., J.C. Panayiotopoulos, N.R. Adam. 1981. The dynamic lot sizing problem for multiple items under limited capacity. *AIIE Transactions* **13** 294-303.

Drexl, A., A. Kimms. 1997. Lot sizing and scheduling – Survey and extensions. *European Journal of Operational Research* **99** 221-235.

Dzielinski, B.P., R.E. Gomory. 1965. Optimal programming of lot-sizes, inventory and labor allocations. *Management Science*, **11** 874-890.

Eisenhut, P. S. 1975. A dynamic lot sizing algorithm with capacity constraints. *AIIE Transactions* **7** 170-176.

Eppen, G.D., R. K. Martin. 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research* **35** 832-848.

Erenguc, S. S. 1988. Multi-product dynamic lot-sizing model with coordinated replenishments. *Naval Research Logistics* **35** 1-22.

Erenguc, S.S., Y. Aksoy. 1990. A branch and bound algorithm for a single item nonconvex dynamic lot-sizing problem with capacity constraints. *Computers and Operations Research* **17** 199-210.

Erenguc, S.S., H.M. Mercan. 1990. A multifamily dynamic lot-sizing model with coordinated replenishments. *Naval Research Logistics* **37** 539-558.

Erlenkotter, D. 1978. A dual-based procedure for uncapacitated facility location. *Operations Research* **26** 992-1009.

Evans, J.R. 1985a. An Efficient implementation of the Wagner-Whitin algorithm for dynamic lot-sizing. *Journal of Operations Management* **5** 229-235.

Evans, J.R. 1985b. Network-based optimization algorithms for the capacitated multi-item lot sizing problem. *Computers and Industrial Engineering* **9** 297-305.

Federgruen A., J. Meissner, M. Tzur. 2004. Progressive Interval Heuristics for Multi-item Lot-sizing Problems. *Operations Research* (forthcoming).

Federgruen, A., M. Tzur. 1991. A simple forward algorithm to solve general dynamic lot sizing models with *n* periods in $0(n \log n)$ or $0(n)$ time. *Management Science* **37** 909-925.

Federgruen, A., M. Tzur. 1994. The joint replenishment problem with the time-varying costs and demands: Efficient, asymptotic and ε-optimal solutions. *Operations Research* **42** 1067–1086.

Florian, M., M. Klein. 1971. Deterministic production planning with concave costs and capacity constraints. *Management Science* **18** 12-20.

Florian, M., J.K. Lenstra, A.H.G. Rinnooy Kan. 1980. Deterministic production planning: Algorithms and complexity. *Management Science* **26** 669-679.

Fogarty, D., R. Barringer. 1987. Joint order release decisions under dependent demand. *Production and Inventory Management Journal* **28** 55–61.

Gao, L-L., N. Altay, E.P. Robinson. 2004. A comparative study of exact algorithms for the coordinated lot-sizing problem with dynamic demand. Working paper, Hofstra University, Long Island, NY.

Gao, L-L., E.P. Robinson. 2004. A Lagrangian/dual-ascent based heuristic for the capacitated coordinated replenishment with dynamic demand and backlogging. Working paper, Hofstra University, Long Island, NY.

Haseborg, F. 1982. On the optimality of joint ordering policies in a multi-product dynamic lot size model with individual and joint set-up costs. *European Journal of Operational Research*, **9** 47-55.

Hillier, F. S., G. J. Lieberman. 2005. *Introduction to Operations Research,* McGraw-Hill, NY, NY.

Hindi, K.S. 1995a. Algorithms for capacitated, multi-item lot-sizing without set-ups. *Journal of the Operational Research Society* **46** 465-472.

Hindi, K.S. 1995b. Computationally efficient solution of the multi-item capacitated lot-sizing problem. *Computers and Industrial Engineering* **28** 709-719.

Iyogun, P. 1991. Heuristic methods for the multi-product dynamic lot size problem. *Journal of Operational Research Society* **42** 889–894.

Joneja, D. 1990. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research* **38** 711-723.

Kao, E. P. C. 1979. A multi-product dynamic lot-size model with individual and joint setup costs. *Operations Research* **27** 279-289.

Karimi, B., S. M. T. Fatemi Ghomi, J. M. Wilson. 2003. The capacitated lot sizing problem: a review of models and algorithms. *Omega* **31** 365-378

Karni, R., Y. Roll. 1982. A heuristic algorithm for the multi-item lot-sizing problem with capacity constraints. *IIE Transactions* **14** 249-256.

Kirca, O. 1990. An efficient algorithm for the capacitated single item dynamic lot size problem. *European Journal of Operational Research* **45** 15-24.

Kirca, O. 1995. A primal-dual algorithm for the dynamic lot-sizing problem with joint setup costs. *Naval Research Logistics* **42** 791-806.

Kirkpatrick, S., C.D. Gelatt Jr., M.P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671-680.

Kleindorfer, P. R., E.F.P. Newson. 1975. A lower bounding structure for lot-size scheduling problems. *Operations Research* **23** 299-311.

Lambert, A., H. Luss. 1982. Production planning with time-dependent capacity bounds. *European Journal of Operational Research* **9** 275-280.

Lambrecht, M. R., H. Vanderveken. 1979. Heuristic procedures for the single operations, multi-item loading problem. *AIIE Transactions* **11** 319-326.

Lasdon, L.S., R.C. Terjung. 1968. An efficient algorithm for multi-item scheduling. *Operations Research* **19** 946-969.

Leung, J.M.Y., T. L. Magnanti, R. Vachani. 1989. Facets and algorithms for capacitated lot sizing. *Mathematical Programming* **45** 331-359.

Lotfi, V., Y. Yoon. 1994. An algorithm for the single-item capacitated lot-sizing problem with concave production and holding costs. *Journal of the Operational Research Society*, **45** 934-941.

Love, S.F. 1973. Bounded production and inventory models with piecewise concave costs. *Management Science* **20** 313-318.

Maes, J., L. Van Wassenhove. 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. *Journal of the Operational Research Society*, **39** 991-1004.

Manne, A.S. 1958. Programming of economic lot-sizes. *Management Science*, **4** 115-135.

Martin, R.K. 1987. Generating alternative mixed-integer programming models using variable redefinition. *Operations Research* **35** 820-831.

Millar, H.H., M. Yang. 1993. An application of Lagrangian decomposition to the capacitated multi-item lot sizing problem. *Computers and Operations Research* **20** 409-420.

Millar, H.H., M. Yang. 1994. Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. *International Journal of Production Economics* **34** 1-15.

Narayanan, A., E.P. Robinson. 2006. More on 'Models and algorithms for the dynamic demand joint replenishment problem.' *International Journal of Production Research* **44** 383-397.

Newson, E.F.P. 1975a. Multi-item lot size scheduling by heuristic part I: With fixed resources. *Management Science* **21** 1186-1193.

Newson, E.F.P. 1975b. Multi-item lot size scheduling by heuristic part II: With variable resources. *Management Science* **21** 1194-1203.

Pratsini, E. 2000. The capacitated dynamic lot size problem with variable technology. *Computers & Industrial Engineering* **38** 493-504.

Raghavan, P. 1993. The multi-item lot sizing problem with joint replenishments. PhD Dissertation. Leonard N. Stern School of Business, New York University.

Robinson, E. P., L-L. Gao. 1996. A dual ascent procedure for multi-product dynamic demand coordinated replenishment with backlogging. *Management Science* **42** 1556-1564.

Robinson, E. P., F.B. Lawrence. 2004. Coordinated capacitated lot-sizing problem with dynamic demand: A Lagrangian heuristic. *Decision Sciences* **35** 25-54.

Robinson, E.P., A. Narayanan, L-L. Gao. 2006. Effective heuristics for the dynamic demand joint replenishment problem. *Journal of Operational Research Society* (forthcoming)

Robinson, E.P., F. Sahin, L-L. Gao. 2005. Master production schedule time interval strategies in make-to-order supply chains. Working paper, Texas A&M University, College Station, TX.

Sahin F., E.P. Robinson, L-L.Gao. 2004. MPS policy and rolling schedules in a two-stage make-to-order supply chain. Working paper, University of Tennessee, Knoxville, TN.

Shapiro, R., D. Rosenfield, K.E. Stecke. 2002. *Baker Precision Instruments, Inc.* HBS-9-687-052, Harvard Business School Publishing, Boston, MA.

Silver, E. A. 1979. Coordinated replenishments of items under time varying demand: dynamic programming formulation. *Naval Research Logistics Quarterly* **26** 141-151.

Silver, E. A., P. Kelle. 1988. More on 'Joint order release decisions under dependent demand'. *Production and Inventory Management Journal* **29** 71–72.

Silver, E. A., H.C. Meal. 1973. A heuristic for selecting lot size quantities for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management Journal* **14** 64-74.

Simpson, N.C. 1999. Multiple level production planning in rolling horizon assembly environments. *European Journal of Operational Research* **114** 15-28.

Simpson, N.C. 2001. Questioning the relative virtues of dynamic lot sizing rules. *Computers and Operations Research* **28** 899-914.

Sridharan, V., W.L. Berry. 1990 Master production scheduling make-to-stock products: a framework for analysis. *International Journal of Production Research* **28** 541-558.

Sridharan, V., W.L. Berry, V. Udayabhanu. 1987. Freezing the MPS under rolling planning horizons. *Management Science* **33** 1137-1149.

Sridharan, V., W.L. Berry, V. Udayabhanu. 1988. Measuring master production schedule stability under rolling planning horizons. *Decision Sciences* **19** 147-166.

Stadtler H. 2000. Improved rolling schedules for the dynamic single-level lot-sizing problem. *Management Science* **46** 318-326.

Stowers C.L, U.S. Palekar 1997. Lot sizing with strong set-up interactions. *IIE Transactions* **29** 167-179.

Thizy, J.M., L. N. Van Wassenhove. 1985. Lagrangian relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation. *IIE Transactions* **17** 308-313.

Trigeiro, W.W. 1987. A dual-cost heuristic for the capacitated lot sizing problem. *IIE Transactions* **19** 67-72.

Trigeiro, W.W., L. J. Thomas, J.O. McClain, J.O. 1989. Capacitated lot sizing with setup times. *Management Science* **35** 353-366.

Van Nunen, J.A.E.E., J. Wessels. 1978. Multi-item lot size determination and scheduling under capacity constraints. *European Journal of Operational Research* **2** 36-41.

Veinott, A.F. Jr. 1969. Minimum concave-cost solution of Leontief substitution models of multi-facility inventory systems. *Operations Research* **17** 262-291.

Wagelmans, A., S. Van Hoesel, A. Kolen. 1992. Economic lot sizing: An *0*(*n* log *n*) algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research* **40** (Supp. No. 1) S145-156.

Wagner, H. M., T. M. Whitin. 1958. Dynamic version of economic lot size model. *Management Science* **5** 89-96.

Wolsey, L. 1989. Strong formulations for mixed integer programming: A survey. *Mathematical Programming* **45** 173-191.

Yeo, I., R.A. Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika* **87** 954-959.

Yeung J. H. Y., W. C. K. Wong, L. MA. 1998. Parameters affecting the effectiveness of MRP systems: a review. *International Journal of Production Research* **36** 313 – 332.

Zangwill, W.I. 1966a. A deterministic multi-period production scheduling model with backlogging. *Management Science* **13** 105-119.

Zangwill, W. I. 1966b. A deterministic multiproduct, multifacility production and inventory system. *Operations Research* **14** 486-507.

Zhao, X., K. Lam. 1997. Lot-sizing rules and freezing the master production schedule in material requirements planning systems. *International Journal of Production Economics* **53** 281-305.

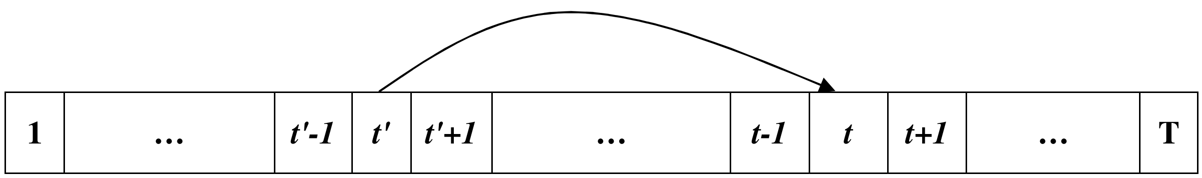
Zhao, X., T. S. Lee. 1996. Freezing the master production schedule in multilevel material requirements planning systems under deterministic demand. *Production Planning and Control* **7** 144-161.

Zhao, X., J. Xie, Q. Jiang. 2001. Lot-s-zing rule and freezing the master production schedule under capacity constraint and deterministic demand. *Production and Operations Management* **10** 45-67.

APPENDIX A

SUBROUTINE TO CALCULATE NET REQUIREMENT

The subroutine calculates the procedure for calculating the net requirement for item *i* from period $t'+1$ to period t-1.



Calculate Requirement (*i*, *t'*, *t*)

```
Set buffer = 0 and requirement = 0
For k = t'+1 to t-1
        If  d_ik > a_ik
                Case1 : buffer < d_ik - a_ik
                        requirement = requirement + d_ik -  a_ik – buffer
                        buffer = 0
                Case 2: buffer > d_ik - a_ik
                        buffer = buffer + a_ik - d_ik
        Else
                buffer = buffer + a_ik - d_ik
        End If
        k = k -1
Loop
Return requirement
```

APPENDIX B

SUBROUTINE TO CALCULATE COST ADJUSTMENT

This procedure calculates the incremental inventory carrying costs, *INCR*, associated with left-shifting quantity *LSQ* into an earlier time period labeled as *TIME*. The procedure receives as input for *LSQ* either $f_{t'}$, $F_{t'}$ or N_t, and returns as *INCR* the value $Cf_{t'}$, $CF_{t'}$, or CN_t, respectively. The procedure assumes the item with the lowest inventory holding cost is rescheduled earlier in time hence; the procedure returns the lowest possible incremental increase in costs.

<u>Procedure</u>

```
INCR = 0
While (LSQ > 0)
    INCR = INCR + min {h_i}*LSQ
    If Z_TIME =1
        If e_TIME > 0
            Case 1 : LSQ > e_TIME
                LSQ = LSQ - e_TIME ; set e_TIME = 0
            Case 2 : e_TIME > LSQ
                e_TIME = e_TIME - LSQ; set LSQ = 0
        Else
            The shortage |e_TIME| is accounted for in the immediate open
            period (i.e) Let TIME* be the immediate open period, then set
            e_TIME* = e_TIME* - |e_TIME| and e_TIME = 0
        End If
    End If
    TIME = TIME - 1
Loop
Return INCR
```

<u>Note :</u> This procedure is called as a subroutine and used as part of the cost calculation associated with left-shifting a candidate quantity into earlier time periods. All calculations that alter the value of available capacities, e_{TIME}, are valid only for the specific item and time period being evaluated when the procedure is called. The original values of e_{TIME} are restored before computing the cost adjustment for another period or item.

APPENDIX C

MODIFIED SUBROUTINE TO CALCULATE COST ADJUSTMENT

This procedure is similar to the one in Appendix B. However, the objective is to only update the values of e_{TIME} and *INCR* is not calculated. The procedure receives $|G(t+1,T)| - N_t$ as input for *LSQ*.

VITA

Arunachalam Narayanan
c/o Dr. E. Powell Robinson
Department of Information and Operations Management
Texas A&M University, College Station, TX 77843-4217

EDUCATION

- M.S. Industrial Engineering, Texas A&M University, College Station, TX. (May '02)
- B.E. Mechanical Engineering, Anna University, Guindy, Chennai, India. (May '00)

PUBLICATIONS

- "More on 'Models and algorithms for the dynamic-demand joint replenishment problem' " *International Journal of Production Research*, 2006, Vol. 44, 2, 383-397. (co-authored with Robinson E.P.)
- "Effective heuristics for the dynamic demand joint replenishment problem", *Journal of Operational Research Society*, published online April 19, 2006. (co-authored with Robinson E.P and L-L. Gao)

TEACHING EXPERIENCE

- Instructor, Department of Information and Operations Management, Texas A&M University, College Station, TX – Summer '03, '04, '05 and '06.

HONORS

- Dean's Award for Outstanding Research by a Doctoral student ('06)
- Dean's Award for Outstanding Teaching by a Doctoral student ('05)
- Undergraduate Gold Medalist for securing overall proficiency in academic skills
- High School Gold Medalist for overall proficiency

CERTIFICATIONS

- Professional Diploma in Network Centered Computing (May '99)
- Microsoft Certified Professional (MCP, Jun'98)