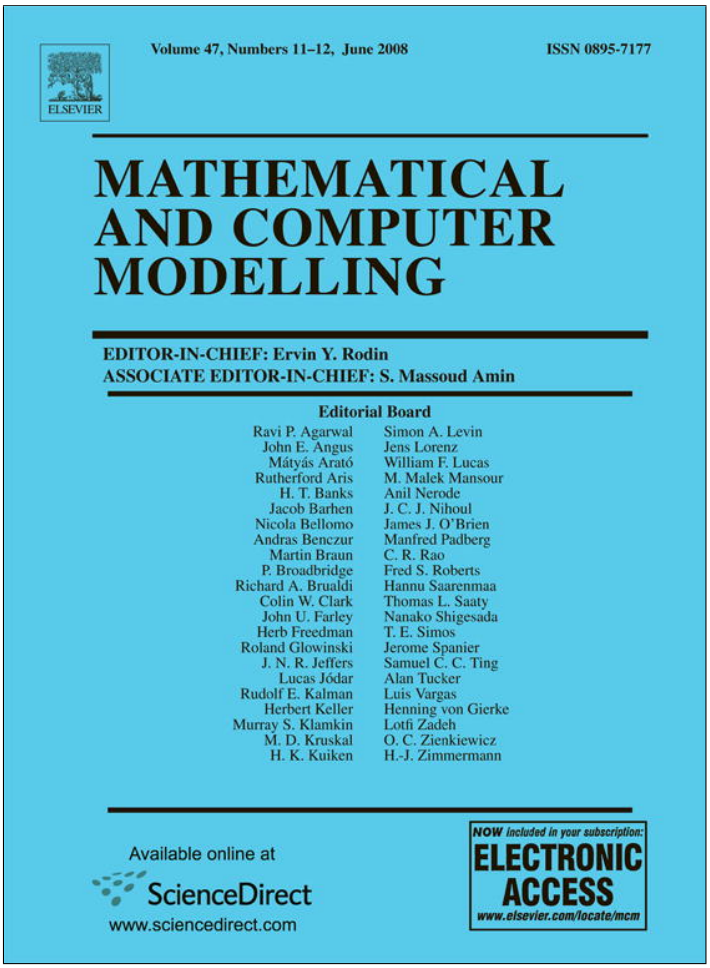
2008

A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand

Li-Lian Gao, *Hofstra University*
Nezih Altay, *University of Richmond*
Powell E Robinson, *Texas A & M University - College Station*



Available online at www.sciencedirect.com
ScienceDirect
Mathematical and Computer Modelling 47 (2008) 1254–1263
MATHEMATICAL AND COMPUTER MODELLING
www.elsevier.com/locate/mcm

A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand

Li-Lian Gao[a], Nezih Altay[b], E. Powell Robinson[c,*]

[a] *Hofstra University, Zarb School of Business, Department of Management, Entrepreneurship and General Business, Hempstead, NY 11549-1000, United States*
[b] *University of Richmond, Robins School of Business, Management Systems Department, Richmond, VA 23173, United States*
[c] *Texas A&M University, Mays College of Business, INFO Department, 322 Wehner Building, College Station, TX 77843-4217, United States*

Received 7 September 2004; received in revised form 1 December 2006; accepted 7 February 2007

Abstract

In the coordinated lot-size problem, a major setup cost is incurred when at least one member of a product family is produced and a minor setup cost for each different item produced. This research consolidates the various modeling and algorithmic approaches reported in the literature for the coordinated replenishment problem with deterministic dynamic demand. For the two most effective approaches, we conducted extensive computational experiments investigating the quality of the lower bound associated with the model's linear programming relaxation and the computational efficiency of the algorithmic approaches when used to find heuristic and optimal solutions. Our findings indicate the superiority of the plant location type problem formulation over the traditional approach that views the problem as multiple single-item Wagner and Whitin problems that are coupled by major setup costs. Broader implications of the research suggest that other classes of deterministic dynamic demand lot-size problems may also be more effectively modeled and solved by adapting plant location type models and algorithms.

Keywords: Inventory modeling; Lot-sizing; Uncapacitated; Mixed-integer-programming; Computational experiments

1. Introduction

The dynamic-demand coordinated replenishment lot-sizing problem (CRP) assumes that a major (joint) setup is shared across a product family. A secondary minor setup cost is incurred for each different item replenished. Silver [1] provides several examples of coordinated replenishment including scheduling a packaging line to produce various sizes and types of containers, purchasing multiple items from a common supplier, and shipping items that share a common mode of transportation. Robinson and Lawrence [2] show two cases of the CRP in the production of industrial lubricants and shipment of vaccines from a manufacturer's warehouse to distribution centers. The objective is to minimize the total setup and inventory holding costs while serving all the customer demands.

While there is considerable research proposing alternative modeling and algorithmic approaches for the CRP with deterministic dynamic demand, the research findings are not adequately consolidated in the literature such that the

* Corresponding author.
E-mail addresses: Li-lian.Gao@Hofstra.edu (L.-L. Gao), naltay@richmond.edu (N. Altay), probinson@cgsb.tamu.edu (E.P. Robinson).

0895-7177/$ - see front matter
doi:10.1016/j.mcm.2007.02.035

relative benefits and drawbacks of various solution approaches are identified. The objective of this research is two-fold. First, we consolidate the literature and research findings on the coordinated replenishment problem (CRP) with deterministic dynamic demand. Second, we conduct extensive computational experiments to evaluate the performance of the two most promising modeling and algorithmic approaches in the literature. Each of these approaches assumes a distinct modeling and algorithmic perspective, but their relative effectiveness is not addressed in the current literature. Finally, the conclusions and implications of the research are presented.

2. Literature survey

Wagner and Whitin's (WW) [3] seminal research on the dynamic-demand lot-size problem optimizes the replenishment schedule for a single item assuming discrete dynamic demand and an unlimited product supply in each replenishment period. System costs include a fixed ordering cost and per-unit inventory holding costs. WW present a dynamic programming algorithm of $O(T^2)$, where T is the number of time periods in the planning horizon. Due to its significance as a planning model in industry and as a sub-problem in other complex planning problems, several researchers propose more efficient implementation schemes for the basic WW lot-size problem [4–7].

The CRP with deterministic dynamic-demand generalizes the WW lot-size problem to consider a multi-item product family that shares a joint setup cost. Arkin, Joneja and Roundy [8] prove the problem class is *NP*-complete. Aksoy and Erenguc [9] review the broader coordinated replenishment literature considering constant, stochastic and dynamic-demand patterns. Traditional approaches for solving CRP utilize dynamic programming algorithms. Zangwill [10] solves an acyclic network formulation of the problem with a dynamic programming algorithm that uses time period as the stage variable and the starting inventory level of the products as the state variable. Zangwill shows that, there exists an optimal replenishment policy in which an item is ordered if and only if there is no remnant inventory carried into the ordering period for the item. Hence, each period's demand is sourced from a single replenishment period. This single sourcing property is exploited in almost every solution strategy developed for the CRP.

Veinott [11] proposes Leontief substitution models to evaluate the 2^{T-1} feasible ordering schedules for the major setup time periods. For a given major setup schedule, the problem decomposes into multiple WW lot-size problems that can be efficiently solved. Kalymon [12] uses a similar approach for arborescent inventory systems, where each item has only one immediate predecessor. His implicit enumeration algorithm identifies a relatively small subset of all possible major setups to include in an optimal schedule, and appears to be more efficient than Veinott's procedure.

Kao [13] presents an alternative acyclic network representation of the problem. Although the formulation results in fewer nodes and arcs compared to Zangwill's approach, the size of the state space is still a limiting factor for efficient solution by dynamic programming. Erenguc [14] shows that for a 12-period planning horizon increasing the number of items to more than three significantly increases the algorithm's computational requirements. Silver [15] concludes that pure dynamic programming approaches are suitable only for relatively small values of T (number of periods) and K (number of items) and suggests that research should focus on developing effective heuristic procedures. Haseborg [16] proposes a variable reduction strategy in an effort to reduce the number of time periods considered by the dynamic programming algorithm. However, the largest test problems only contain 10 periods and 4 items.

Atkins and Iyogun [17] utilize an extension of the Silver–Meal heuristic to get good lower bounds on the optimal solution of CRP. Time requirements for constructing the lower bound are linear in the number of products and quadratic in the number of time periods. Joneja [18] analyzes the worst-case performance of Kao's [13] multi-pass heuristic, and proposes an alternative "cost-covering" single-pass heuristic, which has a uniformly bounded worst-case performance. Joneja also suggests a mixed-integer-programming (MIP) of CRP whose linear programming (LP) relaxation provides a tight lower bound. Finally, Chung and Mercan [19] develop a polynomial-time heuristic procedure based on a forward dynamic programming algorithm. Problems with 12 periods and 20 items are solved on an average to 1.34% of optimality approximately ten times faster than finding an optimal solution in [14].

Erenguc [14] deviates from the pure dynamic programming approaches traditionally employed in the earlier literature and proposes a hybrid dynamic programming and B&B method. The approach, based on Veinott's [11] "major setup pattern" concepts, branches on major setup time periods and solves a series of WW type single-item lot-size problems at each node of the B&B tree. This approach requires considerably less memory than pure dynamic programming approaches and solution times are approximately linear in the number of items. However, computational requirements increase rapidly with an increase in the length of the planning horizon and cost structures with higher

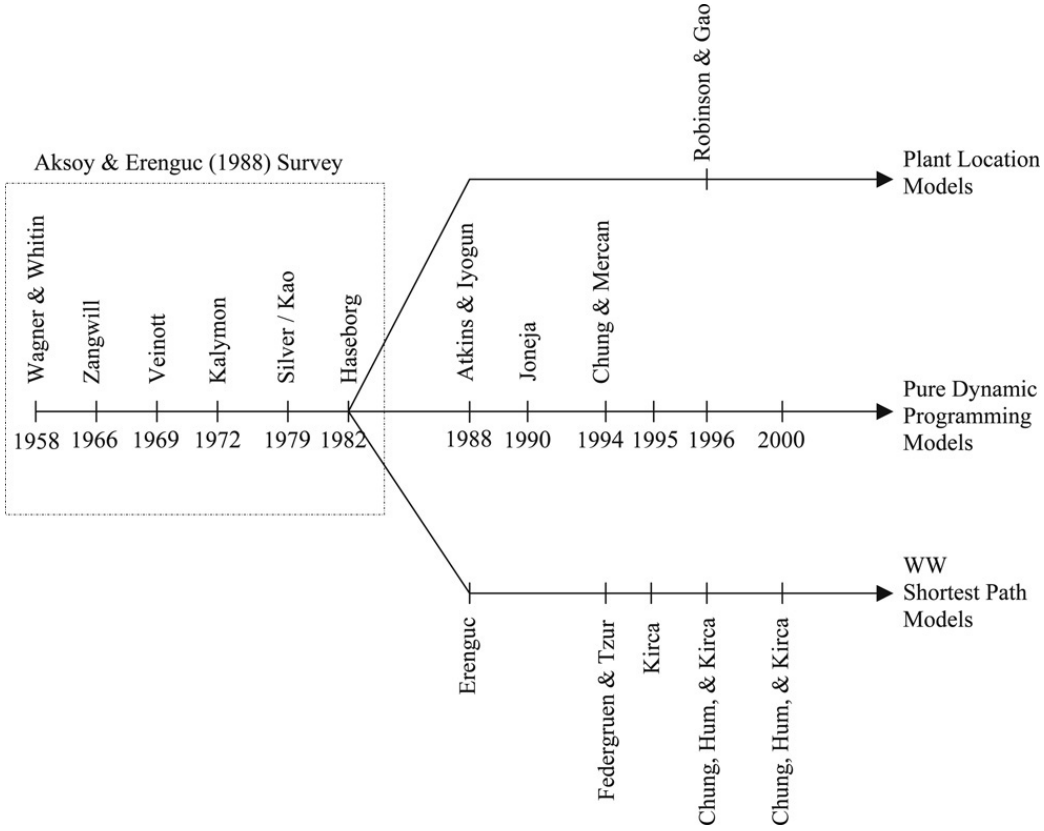


Fig. 1. CRP research timeline.

(total fixed cost)/(total cost) ratios at optimality. Problems with 12 time periods and 20 products are solved to optimality, which exceed the largest sized problems solved to optimality with pure dynamic programming methods.

Federgruen and Tzur [20] build upon Erenguc's [14] approach by developing a partitioning heuristic that decomposes the planning horizon into several smaller time intervals, which are solved as sub-problems in a B&B procedure. The partitioning heuristic solves problems with 30 periods and 10 items to within 0.15% of optimality in approximately one second of the central processing unit (CPU) time, while finding an optimal solution requires more than 1000 CPU seconds.

Kirca [21] proposes a primal-dual heuristic to solve the dual of the LP relaxation of Joneja's [18] MIP formulation to obtain a tight lower bound for use in a B&B procedure. The dual problem is a resource allocation problem in which major setup costs are allocated to maximize the value of the individual item replenishment schedules in the dual problem. The heuristic procedure iterates between solving a primal problem and a series of WW lot-sizing problems. The procedure terminates in a finite number of iterations and any remaining duality gap is resolved by a B&B method with primary branching on the major setup time periods. The procedure solves problems with 24 time periods and 50 products optimally. It is the most efficient known approach that exploits the fact that the CRP is basically a series of independent single-item WW lot-size problems that are coupled by major setup costs. Chung, Hum and Kirca [22,23] extend this approach to consider coordinated replenishment problems with incremental quantity discounts.

Robinson and Gao [24] depart the earlier WW-based formulations of the CRP and propose a plant location type MIP model. They formulate the problem as an arborescent structured fixed charge network programming problem and provide a B&B procedure utilizing the dual ascent, adjustment, and primal construction concepts initially proposed by Erlenkotter [25] for the simple plant location problem. Computational results indicate that the procedures are substantially more efficient than those in Erenguc [14].

Fig. 1 classifies the research according to the modeling and algorithmic methodologies employed. Kirca's [21] approach, which formulates the MIP as multiple WW lot-sizing problem that are coupled by major setup costs, and Robinson and Gao's [24] plant location approach are the two most promising strategies for solving the CRP. However, since both computational studies utilized different computers and test problems, the relative efficiency of the two

methods in finding good heuristic solutions at node 0 of the B&B tree and verifying optimal problem solutions are unknown. Furthermore, there has been no formal comparison of the tightness of the associated LP relaxations or integrality properties of the two CRP formulations. We resolve these gaps in the literature. Our results complement those in [26–28], which provide comparative studies of alternative heuristics, models and grouping strategies for joint replenishment problems assuming stationary, stochastic demand. The following sections describe the problem formulations and solution algorithms in greater detail.

3. WW shortest path formulation and primal-dual algorithm

Kirca [21] exploits the strong formulation of the joint lot-size problem, for which the linear programming relaxation is known to provide tight lower bounds (see, [8] and [18]). The formulation considering N products and T periods recognizes that the lot-size problem for the individual items can be formulated as WW lot-sizing problems and solved as shortest path network programming problems. The nodes of the shortest path model represent time periods and an arc between any pair of nodes t and $k(t < k)$ corresponds to a sub-plan for a given item in which production is scheduled in time t to meet the total demand from time t to $k-1$. Hence, a path through the network from time 1 to time $T' = T + 1$ provides a production schedule for a given item covering all demand for the T planning horizon. Additional constraints are imposed to ensure that a major setup cost is incurred in each time period whenever at least one item is ordered.

Demand for item i in period t is d_{it}, $(i = 1, \ldots, N; t = 1, \ldots, T)$ and D_{itk} is the total demand for item i in periods t through $k-1$. Cost parameters include: S_t, major setup cost in period t $(t = 1, \ldots, T)$; s_{it}, minor setup cost for item i in period t $(i = 1, \ldots, N; t = 1, \ldots, T)$; p_{it}, per- unit procurement cost for one unit of item i in period t $(i = 1, \ldots, N; t = 1, \ldots, T)$; and h_{it}, holding cost for one unit of item i in period t $(i = 1, \ldots, N; t = 1, \ldots, T)$. The total cost of the sub-plan from period t through period $k-1$ for item i is $C_{itk} = s_{it} + p_{it} D_{itk} + \sum_{r=t+1}^{k-1} h_{i,r-1} D_{irk}$. The decision variable $Z_{itk} = 1$ if D_{itk} units of item i are ordered in period t, and 0 otherwise. $Y_t = 1$ if a major setup is scheduled in period t, and 0 otherwise. The mixed-integer-programming (MIP) formulation is stated as Problem $P1$:

$$P1 \quad v(P1) = \text{Min} \sum_{t=1}^{T} S_t Y_t + \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=t+1}^{T'} C_{itk} Z_{itk}. \tag{1}$$

Subject to:

$$\sum_{k=1}^{t-1} Z_{ikt} - \sum_{k=t+1}^{T'} Z_{itk} = 0 \quad i = 1, \ldots, N; t = 2, \ldots, T \tag{2}$$

$$\sum_{k=1}^{T} Z_{ikT'} = 1 \quad i = 1, \ldots, N \tag{3}$$

$$Y_t - \sum_{k=t+1}^{T'} Z_{itk} \geq 0 \quad i = 1, \ldots, N; t = 1, \ldots, T \tag{4}$$

$$Z_{itk} \geq 0, \text{integer} \quad i = 1, \ldots, N;\ t = 1, \ldots, T;\ k = t+1, \ldots, T' \tag{5}$$

$$Y_t \text{integer} \quad t = 1, \ldots, T. \tag{6}$$

The above formulation consists of T integer variables (at any optimal solution, integrality of Z_{itk} is implied by WW single sourcing property), $NT[(T+1)/2]$ continuous decision variables, and $N(2T+1)$ structural constraints. Constraints (2) and (3) enforce conservation of flow at each node of the shortest path problems. Constraint set (4) ensures that a major setup cost is incurred in every period in which at least one item is ordered.

Kirca [21] recognizes the computational difficulty of solving the linear programming (LP) relaxation of $P1$ to optimality and proposes a primal-dual algorithm (PDA) to find a good heuristic solution to the dual of the LP relaxation for use in lower bounding in a B&B procedure. The PDA attempts to allocate the major setup costs at each period among items such that the objective function values of the resulting N shortest paths are maximized. This is

accomplished by iteratively solving N shortest path problems for a given allocation of major setup costs, constructing a feasible primal solution from the value of the dual variables (this also provides an upper bound on $P1$), and adjusting the value of the fixed cost allocations until further improvement in the dual solution (sum of the shortest path problem objective functions) is not possible. In this manner, the PDA exploits the computational efficiency of solving the WW lot-size problems. The PDA yields a monotonic increase in the dual objective function value each iteration and terminates in a finite number of iterations. The B&B procedure is similar to the one proposed by Erenguc [14] except that the primary branching decision selects the period with the greatest complementary slackness violation. The reader is referred to Kirca [21] for algorithm details and schemes for improving its computational efficiency.

Kirca's computational experiments indicate that greater computational resources are required to solve problems with longer planning horizons, more items, lower average time between orders (TBO), and lower ratios of major setup cost to the minor setup costs (MSR).

4. Plant location formulation and dual-ascent-based algorithm

The coordinated replenishment problem has similar mathematical structure to the single-warehouse, multiple-retailer problem [11] under the assumptions of no inventory storage at the warehouse. Robinson and Gao [24] adapt the multi-activity plant location problem formulation in Klincewicz and Luss [29] to the joint lot-size problem. In addition to the earlier notation, y_{it} is a binary decision variable where $y_{it} = 1$ if item i is setup in time t, and 0 otherwise. X_{itk} is the fraction of demand in period k for product i that is supplied from an order in period t. The total variable cost for supplying the total demand for item i in period k from product ordered in period t is defined as $\bar{C}_{itk} = d_{ik}\,(p_{it} + h_{itk})$, where h_{ikt} is the per-unit holding cost for carrying an inventory of item i from period t to period k. The MIP formulation is stated as Problem $P2$ below for the case with no backorders allowed. Additional features of the formulation, including the representation of backorders, are discussed in Robinson and Gao [24].

$$P2 \quad v(P2) = \text{Min} \sum_{i=1}^{T} S_t Y_t + \sum_{i=1}^{N}\sum_{t=1}^{T} s_{it} y_{it} + \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=t}^{T} \bar{C}_{itk} X_{itk}. \tag{7}$$

Subject to:

$$\sum_{t=1}^{k} X_{itk} = 1 \quad i = 1, \ldots, N;\ k = 1, \ldots, T \tag{8}$$

$$-Y_t + y_{it} \le 0 \quad i = 1, \ldots, N;\ t = 1, \ldots, T \tag{9}$$

$$-y_{it} + X_{itk} \le 0 \quad i = 1, \ldots, N;\ t = 1, \ldots, T;\ k = t, \ldots, T \tag{10}$$

$$0 \le Y_t \le 1 \text{ and integer} \quad t = 1, \ldots, T \tag{11}$$

$$0 \le y_{it} \le 1 \text{ and integer} \quad i = 1, \ldots, N;\ t = 1, \ldots, T \tag{12}$$

$$0 \le X_{ikt} \le 1 \quad i = 1, \ldots, N;\ t = 1, \ldots, T;\ k = t, \ldots, T. \tag{13}$$

The formulation requires $T + NT$ integer variables, $NT[(T+1)/2]$ continuous variables, and $NT[2 + (T+1)/2]$ structural constraints. Constraint set (8) ensures that the demand for each item in each period is served. Constraint sets (9) and (10) prevent replenishment unless the appropriate setup costs are incurred.

Consequent to the results in Denizel, Erenguc, and Sherali [30], $P2$ is the strong MIP formulation yielding the linear programming relaxation equivalent to the convex envelope relaxation of the problem. The LP relaxation provides a tight lower bound on $P2$ due to the disaggregate representation of the variable upper bound constraints (9) and (10), the hierarchical linkages among decision variables in the variable upper bound constraints, and the single sourcing property of an optimal solution in which $X_{itk} \in \{0, 1\}$ naturally. However, solving the LP relaxation of $P1$ is computationally expensive due to the large number of variables and constraints and the frequency of degenerate pivots. Hence, Robinson and Gao [24] attack the dual of the LP relaxation of $P2$ with a primal-dual algorithm to obtain a lower bound and a feasible upper bound for $P2$ utilizing the complementary slackness conditions.

The procedures build upon the dual ascent, adjustment, and primal construction concepts in Erlenkotter [25] for the simple plant location problem, Klincewicz and Luss [29] for the multi-activity facility location problem, and Gao and Robinson [31] for the two-echelon facility location problem. Complementary slackness violations guide the variable

selection decision within the branch and bound procedure. The reader is referred to [24] for a full description of the algorithm including the application of the planning horizon property to reduce the number of decision variables.

As in [21], problems with longer planning horizons and more items require greater computational resources. However, solution times are not sensitive to the ratio of the major to minor setup costs (MSR). While backorders cannot be represented in $P1$ without destroying the formulation's special mathematical structure and computational efficiency, $P2$ can consider backorders without any adverse impact on the problem structure. Robinson and Gao [24] report that solving problems with backorders requires approximately 33% more CPU time than problems without backorders.

The mathematical models of $P1$ and $P2$ and associated solution procedures represent two distinct philosophies for modeling and solving the joint lot-size problem. While $P1$ exploits the computational efficiency of iteratively solving the WW lot-size problems, $P2$ exploits the hierarchical linkages among decision variables, to allow the dual of the LP relaxation to be solved easily. However, neither the relative tightness of the respective LP relaxations, nor the computational efficiency of solving the models using off-the-shelf general-purpose optimizers versus the proposed primal-dual algorithms have not yet been explored. Given the potential application of the joint lot-size models as advanced planning procedures within enterprise resource planning systems, it is important to better understand the computational performance of these modeling and solution approaches for this problem class. In the following sections, we describe computational experiments, which compare the performance of these two approaches and provide the findings.

5. Computational study

We followed the experimental design and data generation schemes described in [21]. The experimental factors include the number of time periods (12, 24, 36, and 48), the number of items (5, 10, 20, and 40), the ratio of the major to the minor setup costs (MSR = 0.3, 0.6, and 0.9), and the time between orders (TBO $= 1.5, 2.5$, and 4.5). Details for randomly generating the major and minor setup costs based on given MSR and TBO values are described in [21]. Unit production costs and unit inventory holding costs per time period are set to 0 and 1 respectively for all items and all periods. The demand for each item is normally distributed with a mean of 200 units and a standard deviation of either 70 or 20 units per time period. The standard deviation of demand was randomly assigned to items with an equal probability of occurrence. Demand is assumed to occur in every time period. For each combination of the experimental factors five random problems were generated. The computational experiments were conducted on a Pentium 4/2.2 GHz desktop PC. The test computer codes, written in FORTRAN, are the same as the ones evaluated in [23,24].

6. Experimental results

The computational experiments enable comparative analysis of the problem formulations and alternative solution approaches for $P1$ and $P2$. We evaluate the quality of the proposed formulations in terms of the tightness and integrality properties of their respective LP relaxations and the computational requirements for finding and verifying optimal solutions using Xpress-MP, a general-purpose optimization software package [32]. The effectiveness of both primal-dual algorithms in finding both heuristic and optimal problem solutions is also studied.

We initiated the experiments by solving both the formulations of the 720 test problems to optimality with Xpress-MP, which utilizes the LP relaxation as a lower bound (LB) in a B&B procedure. Our initial interest concerned the relative quality of the LB and the frequency of integer/optimal solutions identified by the LP relaxation. An unexpected, although not totally surprising, result is that the LB provided by LP relaxation of $P1$ and $P2$ are identical, where either LP relaxation provided integer/optimal solutions for 99.44% (716/720) of test problems with a worst optimality gap of 0.022%. This result is attributed to disaggregate variable upper bound constraints (i.e. (4), (9) and (10)) in each formulation, which tightly constrain the integer variables to take on values of 0 or 1 in the solution of the LP relaxation.

The average and maximum central processing unit (CPU) times for solving the LP relaxation and verifying an optimal solution are summarized in Table 1. The average/maximum CPU times for solving the LP relaxation for $P1$ and $P2$ are 0.934/15.98 and 0.561/11.06 s, respectively. Analogous CPU times for solving the MIP to optimality for $P1$ and $P2$ are 1.02/16.50 and 0.809/13.72 s, respectively. On an average solving $P2$ with Xpress-MP requires only

Table 1
Computational results for Xpress-MP

		$P1$		$P2$	
		LP time	MIP time	LP time	MIP time
Periods	12	0.052	0.067	0.049	0.077
	24	0.322	0.367	0.242	0.349
	36	1.042	1.134	0.658	0.940
	48	2.321	2.512	1.294	1.871
Items	2	0.065	0.088	0.069	0.113
	10	0.197	0.235	0.181	0.274
	20	0.698	0.778	0.490	0.738
	40	2.778	2.979	1.503	2.112
MSR	0.3	0.723	0.808	0.727	0.992
	0.6	0.899	0.986	0.577	0.825
	0.9	1.180	1.266	0.378	0.611
TBO	1.5	0.444	0.525	0.302	0.530
	2.5	0.855	0.942	0.447	0.687
	4.5	1.504	1.594	0.933	1.211
Average		0.934	1.020	0.561	0.809
Maximum		15.98	16.50	11.06	13.72

All times are in CPU seconds.

60.1% and 79.3% of the computational resources required by $P1$ when finding LP and optimal solutions, respectively. This finding is counter-intuitive considering that $P2$ contains both more integer variables and structural constraints than $P1$, while the number of continuous variables is equal. In addition, $P2$ is relatively more effective for larger problems, higher ratios of major setup fixed costs to minor setup fixed costs (MSR) and longer time between orders (TBO). An additional finding is that $P2$'s solution time decreases with higher values of MSR, while $P1$'s solution resource requirements increase.

The second phase of the analysis evaluated the relative performance of the two approaches when used as heuristics. The comparative data is drawn from each algorithm's status at node 0 of the B&B tree. Performance metrics include the percent of the UB solutions that are optimal, the percent of solutions that are verified as optimal at node 0 (i.e. UB $=$ LB), and the guaranteed worst-case heuristic performance or duality gap (i.e. (UB $-$ LB)/UB). The overall average results for $P1(P2)$ are: percent of UB solutions that are optimal are 81.39% (35.83%); percent of solutions with UB $=$ LB are 4.03% (32.22%); average duality gaps are 0.55% (1.72%); and the maximum duality gaps are 2.44% (14.54%). These results, as presented in Table 2, indicate the superiority of $P1$ over $P2$ in finding good heuristic solutions at node 0 of the B&B tree. However, $P2$ is superior in verifying optimal solutions. Both approaches find good heuristic solutions in most cases.

For $P1$ the UB gap is only 0.014%, while it is 1.654% for $P2$. Furthermore, the UB gaps are fairly consistent for $P1$, while the UB gaps for $P2$ increase with the number of periods, items and lower MSR values. However, $P2$ finds tighter LB solutions, which is important for fathoming nodes in a B&B method. The average LB gaps are 0.534% and 0.146% respectively for $P1$ and $P2$. Finally, node 0 solution times average 0.113 and 0.005 CPU seconds respectively for $P1$ and $P2$.

As illustrated in Table 2, $P2$ finds and verifies optimal solutions on an average in 0.023 CPU seconds compared to 41.452 CPU seconds for $P1$. Hence, $P2$ requires only approximately 0.055% of the computational resources required by $P1$ for solving the test problems. $P2$'s superior performance is at least partially related to the ability of the dual ascent and adjustment procedures to efficiently find tight LB solutions to the MIP problem, which facilitates node fathoming. The average number of B&B nodes evaluated by $P2$ is 8, while on an average $P1$ evaluates 15 534 nodes.

Table 2 also shows that $P1$ finds problems with higher values of MSR and TBO to be harder to solve. $P2$ solution times decrease with higher MSR and lower TBO values. As expected, solution times increase with the number of periods and items for both algorithms. However, the impact of increasing the number of items and time periods is much more dramatic on $P1$. Specifically, as indicated in Figs. 2 and 3 solution times for $P1$ seem to rise exponentially

Table 2
Computational results for $P1$ and $P2$

		$P1$					$P2$				
		Node 0 performance			Optimal time[c]	B&B nodes	Node 0 performance			Optimal time[c]	B&B nodes
		UB gap[a] (%)	LB gap[b] (%)	Time[c]			UB gap[a] (%)	LB gap[b] (%)	Time[c]		
Periods	12	0.02	0.46	0.002	0.005	24	0.99	0.16	0.001	0.002	2
	24	0.01	0.55	0.027	0.199	237	1.52	0.14	0.003	0.008	5
	36	0.01	0.57	0.112	14.178	6 776	2.01	0.16	0.005	0.021	10
	48	0.01	0.56	0.310	151.426	55 098	2.09	0.13	0.010	0.062	16
Items	5	0.01	0.35	0.010	0.129	295	1.21	0.11	0.000	0.002	5
	10	0.02	0.47	0.032	2.813	3 000	1.64	0.13	0.002	0.007	7
	20	0.02	0.59	0.094	18.460	10 008	1.81	0.18	0.004	0.020	10
	40	0.01	0.73	0.315	144.406	48 831	1.96	0.16	0.013	0.063	12
MSR	0.3	0.04	0.69	0.127	98.899	39 601	2.15	0.17	0.006	0.035	13
	0.6	0.00	0.49	0.114	22.085	5 990	1.54	0.12	0.004	0.021	7
	0.9	0.00	0.42	0.098	3.372	1 010	1.27	0.15	0.004	0.013	5
TBO	1.5	0.03	0.75	0.051	118.294	44 427	1.02	0.01	0.004	0.011	6
	2.5	0.01	0.58	0.154	5.835	2 102	2.13	0.11	0.004	0.016	8
	4.5	0.00	0.28	0.133	0.227	72	1.80	0.31	0.007	0.043	11
Average		0.01	0.53	0.113	41.452	15 534	1.65	0.15	0.005	0.023	8
Maximum		0.69	2.30	1.530	8714.56	2600 031	17.02	2.08	0.090	1.780	143

[a] (UB − Optimal)/Optimal.
[b] (Optimal − LB)/Optimal.
[c] In CPU seconds.

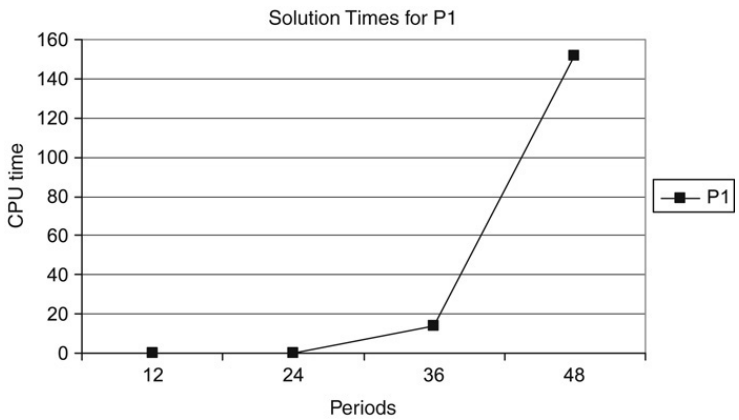


Fig. 2. Solution times for $P1$ with respect to the number of items considered.

with increase in problem size, while $P2$ times appear linear. Due to the large difference in their scale solution times for $P1$ are shown in Fig. 2 separately from $P2$ and Xpress-MP runs. Increasing the number of items produces a similar picture. This is important when considering the potential application of the algorithms within a requirements planning software package and for solving large scale problems quickly.

Finally, it should be noted that $P2$ is computationally more efficient than Xpress-MP for all of the test problems by a factor of 35 times on an average. $P1$ only outperforms Xpress-MP on problem sets with 12 and 24 time periods and a TBO of 4.5. On an average, Xpress-MP solves the shortest path formulation of the joint lot-size problem 40 times faster than $P1$.

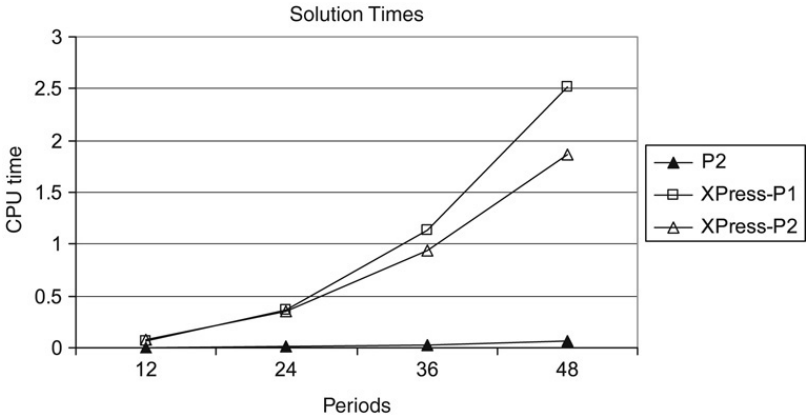


Fig. 3. Solution times of $P2$ and Xpress-MP runs with respect to number of items.

7. Conclusion

Coordinated dynamic-demand lot-sizing problems are commonly encountered when planning multiple item procurement, production and distribution operations. While early research focused on dynamic programming formulations and solution methodologies, more recent approaches focus on solving MIP formulations with tight LP relaxations through the development of specialized primal-dual algorithms. This research studied the two most efficient modeling and algorithmic approaches that emerged for the CRP during the past twenty years. One approach, $P1$, models the problem as a series of single-item lot-size problems that are coupled by a variable upper bound side-constraint that incorporates major setup considerations. The other approach, $P2$, draws upon modeling and algorithmic approaches more often associated with facility location problems.

An extensive computational study revealed that the values of the LP relaxations are identical for the two problem formulations. While unexpected, this finding can be explained by the use of disaggregate variable upper bound constraints in both formulations. However, each algorithmic approach has associated advantages and disadvantages. $P2$ is very efficient both as a heuristic and optimization approach for solving large sized problems. Hence, its potential application in requirements planning software is encouraging. However, its application as a heuristic is compromised by its relatively weak upper bound that is found at node 0 of the branch and bound tree. Improving the quality of this upper bound might provide significant improvements in both the heuristic and optimization effectiveness of the Robinson and Gao algorithm. The better quality of $P1$'s upper bound provides incentive for exploring methods for tightening $P2$'s upper bound.

Overall findings strongly suggest that the plant location perspective taken by $P2$ is superior to the traditional single-item lot-sizing approach taken by $P1$. Where $P1$ found high quality heuristic solutions at node 0 of the branch and bound tree, $P2$ due to its tight lower bound, solved the identical problems to optimality more efficiently. When compared against state-of-the-art general-purpose optimization software $P2$ solved test problems 35 times more efficiently. More importantly, the effect of problem size on the computational requirements is approximately linear for $P2$ while $P1$ solution times increase exponentially.

Each solution approach has its shortcomings and could benefit from additional research. Particularly, improving the upper bounding procedures of $P2$ or the lower bounding procedure of $P1$ could increase the computational efficiency of the solution approach of interest.

From a modeling perspective $P2$ provides a platform more readily capable of incorporating backorders and other more general problem representations such as capacity constraints. This is enabled by the arborescent network structure of $P2$ which allows backorders to be incorporated as additional arcs and directly incorporated into the algorithm. In contrast, the addition of backorders destroys the favorable mathematical properties of the $P1$ model and requires significant reworking of the solution approach. Capacity constraints can be appended to either model, but compromise the mathematical structure of the problem by eliminating the natural single sourcing property of an optimal solution. Evaluating Lagrangean relaxation approaches for the capacitated problem is a promising research area. Due to its highly efficient solution method, relaxing the capacity constraint and applying the Robinson and Gao solution methodology for the resulting CRP sub-problems may prove effective.

Based on the findings of this research, additional study of the plant location approach to modeling and solving the various classes of dynamic-demand lot-size problems appears well-justified.

References

[1] E.A. Silver, Coordinated replenishment of items under time-varying demand: Dynamic programming formulation, Nav. Res. Log. Quart. 26 (1) (1979) 141–151.

[2] E.P. Robinson, F.B. Lawrence, Coordinated capacitated lot-sizing problem with dynamic demand: A Lagrangean heuristic, Decision Sci. 35 (1) (2004) 25–54.

[3] H.M. Wagner, T.M. Whitin, Dynamic version of the economic lot size model, Manage. Sci. 5 (1) (1958) 89–96.

[4] J.R. Evans, An efficient implementation of the Wagner–Whitin algorithm for dynamic lot-sizing, J. Oper. Manag. 5 (2) (1985) 229–235.

[5] A. Federgruen, M. Tzur, A simple forward algorithm to solve general dynamic lot sizing models with n periods in O(nlog n) or O(n) time, Manage. Sci. 37 (8) (1991) 909–925.

[6] A. Wagelmans, S. van Hoesel, A. Kolen, Economic lot sizing: An O(nlog n) algorithm that runs in linear time in the Wagner–Whitin case, Oper. Res. 40 (S1) (1992) S145–S156.

[7] A. Aggarwal, J.K. Park, Improved algorithms for economic lot size problems, Oper. Res. 41 (3) (1993) 549–571.

[8] E. Arkin, D. Joneja, R. Roundy, Computational complexity of uncapacitated multi-echelon production planning problems, Oper. Res. Lett. 8 (1989) 61–66.

[9] Y. Aksoy, S.S. Erenguc, Multi-item inventory models with coordinated replenishments: A survey, Int. J. Oper. Prod. Man. 8 (1) (1988) 63–73.

[10] W.I. Zangwill, A deterministic multiproduct, multi-facility production and inventory model, Oper. Res. 14 (1966) 486–507.

[11] A.F. Veinott, Minimum concave-cost solution of Leontief substitution models of multi-facility inventory systems, Oper. Res. 17 (1969) 262–291.

[12] B.A. Kalymon, A decomposition algorithm for arborescence inventory systems, Oper. Res. 20 (1972) 860–874.

[13] E.P.C. Kao, A multi-product dynamic lot-size model with individual and joint set-up costs, Oper. Res. 27 (2) (1979) 279–289.

[14] S.S. Erenguc, Multi-product dynamic lot-sizing model with coordinated replenishments, Nav. Res. Log. 35 (1988) 1–22.

[15] E.A. Silver, A simple method of determining order quantities in joint replenishments under deterministic demand, Manage. Sci. 22 (12) (1976) 1351–1361.

[16] F. Haseborg, On the optimality of joint ordering policies in a multi-product dynamic lot size model with individual and joint set-up costs, Eur. J. Oper. Res. 9 (1982) 47–55.

[17] D.R. Atkins, P.O. Iyogun, A heuristic with lower bound performance guarantee for the multi-product dynamic lot-size problem, IIE Trans. 20 (4) (1988) 369–373.

[18] D. Joneja, The joint replenishment problem: New heuristics and worst case performance bounds, Oper. Res. 38 (4) (1990) 711–723.

[19] C. Chung, H.M. Mercan, Heuristic procedure for a multiproduct dynamic lot-sizing problem with coordinated replenishments, Nav. Res. Log. 41 (1994) 273–286.

[20] A. Federgruen, M. Tzur, The joint replenishment problem with time-varying costs and demands: Efficient, asymptotic and ε-optimal solutions, Oper. Res. 42 (6) (1994) 1067–1086.

[21] O. Kirca, A primal–dual algorithm for the dynamic lot-sizing problem with joint set-up costs, Nav. Res. Log. 42 (1995) 791–806.

[22] C. Chung, S. Hum, O. Kirca, The coordinated replenishment dynamic lot-sizing problem with quantity discounts, Eur. J. Oper. Res. 94 (1996) 122–133.

[23] C. Chung, S. Hum, O. Kirca, An optimal procedure for the coordinated replenishment dynamic lot-sizing problem with quantity discounts, Nav. Res. Log. 94 (8) (2000) 686–695.

[24] E.P. Robinson, L.L. Gao, A dual ascent procedure for multiproduct dynamic demand coordinated replenishment with backlogging, Manage. Sci. 42 (11) (1996) 1–9.

[25] D. Erlenkotter, A dual-based procedure for uncapacitated facility location, Oper. Res. 26 (6) (1978) 992–1009.

[26] B. Golany, A. Lev-Er, Comparative analysis of multi-item joint replenishment inventory models, Int. J. Prod. Res. 30 (8) (1992) 1791–1801.

[27] M.J.G. van Eijs, R.M.J. Heuts, J.P.C. Kleijnen, Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs, Eur. J. Oper. Res. 59 (1992) 405–412.

[28] M. Ben-Daya, M. Hariga, Comparative study of heuristics for the joint replenishment problem, Omega-Int. J. Manage. S. 23 (3) (1995) 341–344.

[29] F.G. Klincewicz, H. Luss, A dual-based algorithm for multiproduct uncapacitated facility location, Trans. Sci. 21 (1987) 198–206.

[30] M. Denizel, S.S. Erenguc, H.D. Sherali, Convex envelope results and strong formulations for a class of mixed-integer-programs, Nav. Res Log. 43 (1996) 503–518.

[31] L.L. Gao, E.P. Robinson, A dual-based optimization procedure for the two-echelon uncapacitated facility location problem, Nav. Res. Log. 39 (1992) 191–212.

[32] Xpress-MP, Dash Optimization Inc.