An Efficient Heuristic Algorithm for Capacitated Lot Sizing Problem with Overtime Decisions

Çagatay Iris[1] and Mehmet Mutlu Yenisey[1]

[1] Department of Industrial Engineering, Istanbul Technical University, 34367, Macka, Istanbul, Turkey
{ciris,yenisey}@itu.edu.tr

Abstract. Capacitated Lot Sizing Problem is a very important tactical level decision making problem that answers the questions of producing when and how many in dynamic demand environment. Solving Capacitated Lot Sizing Problem with Overtime decisions (CLSPO) and extensions derived from the fundamental structure optimally suffer from combinatorial nature of the problem. The aim of the study is to form a two-stage heuristic algorithm to solve related problem in polynomial time. In first part, characteristics of problem structure are presented. Dominance properties are presented to help algorithm obtain a bounded search area. Proposed algorithm directly utilizes such shortcoming. Performance of approach is tested by using different criteria. And finally, robustness test are applied to check how well algorithm performs against fluctuations in its data. Simulated annealing as improvement heuristic performs well for related problem. It is also observed that fluctuations of data directly affects performance outcome. Obtained results also reveal that performance of improvement heuristic highly depends on constructive heuristic. Algorithm is also applied to an industry case study to plan master production schedule with minimum costs.

Keywords: Production Planning; Lot Sizing with Overtime Decisions; Global Search; Simulated Annealing

1   Introduction

As the increase in competition, the importance of efficient planning has rapidly increased. Companies in this sense focus on how to reduce production cost by efficiently planning production systems. The difficulty in optimizing planning of production relies on two different tight constraints. Solutions found should be feasible with capacity of period and inventory balance between consecutive periods [1]. CLSP may be enlarged with different types of product structures, setup time, overtime, backlogging, lead times, time windows, planning horizon etc. The resulting combination of alternatives becomes more complicated. It is shown that finding a feasible solution becomes NP-complete whenever setup times are incorporated into model [2]. In most of studies, objective function of problem is formulated as a minimization of production, holding, setup, overtime and backlogging costs [3].

Traditional CLSP and extensions derived from main structure have attracted the attention of researchers over the last fifty years. Reference [4] clusters lot sizing problems regarding demand type and resource constraint. The fundamental aspects covered in this survey. However, it isn't sufficient to reflect problem characteristics by only focusing on problem structure. Reference [5] gives detailed information on solution strategies for CLSP problem. Heuristics and Metaheuristics are most common solution strategies. Simulated Annealing, Genetic Algorithms and Lagrangean Relaxation are apparently most popular techniques for related problem type. It is deduced that neighborhood structures proposed for CLSP apparently performs well for such heuristics. One of the most familiar researches in literature to ours was published in 2000. In the paper, an integrated lagrangean relaxation-simulated annealing approach was presented. However, related study lacks efficient search strategies and real-life implementation [2].

Our study aims at combine a very "easy to implement algorithm" which is lot-for-lot (LFL) and global search procedures to solve CLSPO. Lot-for-Lot heuristic which neglects capacity constraint while maintaining inventory balance equations is employed to find an initial solution. The results derived from this phase have been given as an input for improvement heuristic. Global search procedure is capable to improve given initial solution within the feasible solution. The philosophy maintained in improvement heuristic is transferring some amounts of lots from one period to another. Simulated annealing (SA) has been utilized not get stuck in local optimum.

2 Problem Structure and Dominance Properties

The paper focuses on CLSP in two further aspects. It assumes that two sources of capacity exist as regular time and overtime. Since, overtime and regular time has different costs, basic model should be extended. The related mathematical model is:

$$z = Min \ \Sigma t \Sigma i \ (c_{i,t} X_{i,t} + h_{i,t} I_{i,t} + os_t C_t + ov_t O_t) \tag{1}$$

$$s.t. \ I_{i,-1} + X_{i,t} - I_{i,t} = D_{i,t} \qquad \forall i, \forall t \tag{2}$$

$$\Sigma i \ cp_i X_{i,t} \leq C_t + O_t \qquad \forall t \tag{3}$$

$$C_t \leq Max C_t \qquad \forall t \tag{4}$$

$$O_t \leq Max O_t \qquad \forall t \tag{5}$$

$$X_{i,} \geq 0, \ I_{i,t} \geq 0, \ O_t \geq 0, \ C_t \geq 0 \qquad \forall i, t \tag{6}$$

The model covers multiple-items with single-level product structure to be planned over T periods. The objective (1) aims to minimize total cost of production, holding, regular and overtime. Constraint (2) is inventory balance equation for each product and period. Due to the fact that, there are two different resources, capacity limitations formulated by constraint (3) consist of production time with an upper bound of threshold of regular and overtime. Constraint (4) and (5) are control parameters to limit maximum available regular time and overtime. Constraint set (6) reflects non-negativity conditions.

It should be emphasized that lot sizing problem for single-level product structures doesn't require setup time as an individual parameter because most commonly, setup times do not change between periods. Hence, setup times are included in unit processing time of each product.

A tradeoff between cost of working with overtime and holding a unit of inventory in stock occurs in periods where demand result in idle capacity. It may be logical to produce more and hold inventory rather than using overtime in such peak periods [3]. Hence, proposed neighborhood strategy is based on transferring lots from one period to another regarding such tradeoffs. SA controls the amount tradeoff before taking a move, thus it is considered to be a proper metaheuristic for such neighborhood structures. More precisely, hierarchy of decisions to shift a lot to another period is determined by considering cost parameters of predecessor and successor period in SA [2]. In improvement heuristic, a move is generated by transferring a lot between periods in order to restore feasibility of capacity constraint or improve objective function. It depends totally on the constructive heuristic solution whether to improve objective function or attaining capacity feasibility. Since, LFL heuristic may produce capacity infeasible solution, a control mechanism is adopted.

There are three fundamental decisions that should be made: period to shift a lot from/to, item that will be shifted and amount of transfer. After determining the period and item that will be shifted, amount of lot that will be transferred may be determined. Transferring a lot between periods will result in a cost fluctuation by changing the indices of related cost parameter. Forming a dominance property on one-at-a-time lot transferring may be useful for problem structure. These properties help to limit search space, and results in high efficiency in search procedures.

Dominance properties: $Z = f(c_1, c_2, ..., c_n)$ is the value of measure (objective function in our case) that characterizes iteration S and that $Z' = f(c_1', c_2', ..., c_n')$ represents the value of the same measure under some different iterations S'. Then, dominance set is applied as long as condition $Z' \geq Z$ implies that $c_j' \geq c_j$ for some parameters j [6].

Property-1: Transferring an amount of Δ *(Delta)* from t_2 to t_1 is dominance property-1 as: Property-1.1: (*where $t_1 \leq t_2$ and t_2 is most overloaded period in respect to $C_{t2} + O_{t2}$);* Property-1.2:

$$proper\ for\ S = \{min\ \{c_{i,1} + h_{i,t1} - c_{i,t2} - h_{i,t2}\}\ \} \tag{7}$$

$$otherwise\ \{max\ X_{i,2} * cp_i\ otherwise\ rnd_i\} \tag{8}$$

Starting from the period with highest load and item with highest gap in total holding and production cost with previous period will form a dominance set for backward scheduling procedure. Here function of otherwise reflects whether the related production quantity is adequate. If $X_{i,t2}$ equals to zero, then there is no proper lot to be shifted from t_2 to t_1. Product with highest capacity consumption is selected to be shifted in such cases. If $X_{i,t2}$ still doesn't change, a random product may be selected for shifting procedure. Those factors that don't belong to dominant set will be clarified in algorithm explanations.

After determining the periods to transfer the lot in between and item to be transferred, amount that may be shifted may be calculated. There are two alternatives

on transferring a lot. Direction of transfer may be forward or backward. Amounts that may be transferred purely depend upon forward or backward procedure. As mentioned above, constructive heuristic is lot-for-lot, so there will be no inventory on hand at the start of each period in initial solution. Therefore, backward transferring of production lots will be sensible by accumulating on hand inventory for improvement heuristic.

Property-2: If decision is to transfer the lot in backward (which means transferring a lot from *t+1 (t2)* to *t (t1)*) direction, the maximum quantity that may be shifted depends upon the feasibility of the solution where the search is initiated and in all feasible solutions, the idea is to improve objective function. The maximum quantity of item *i* to be transferred is [7];

$$Delta\_max=max\{0,min \{X_{i,t2},Q_{t1}\}\} \tag{9}$$

The maximum quantity that may be shifted from *t2* to *t1* is limited to the amount of product that is produced in *t2* and the available production capacity in *t1*. The related Q_{t1} is calculated by dividing available time in *(C_{t1}+O_{t1})* to unit processing time of determined item *i* which is cp_i.

However, if decision is to transfer the lot in forward (which means transferring a lot from *t (t1)* to *t+1 (t2)*) direction, the maximum quantity that may be shifted depends upon the performance of backward shifting and inventory on-hand in analyzed period *t1*. The maximum quantity of item *i* to be transferred is [7];

$$Delta\_max=max\{0,min \{I_{i,t1},Q_{t2}\}\} \tag{10}$$

The maximum quantity that may be shifted from *t1* to *t2* is limited to the amount of product that is on hand in *t1* and the available production capacity in *t2*. The limitation here is directly related to the initial performance of lot-for-lot schedule. The quantity that will be shifted is determined by using random integer generators between *0* and *Delta_max* values. [7]

3 Heuristic Algorithm

As mentioned above, algorithm consists of two consecutive stages. In the first, there is an "easy to implement" constructive heuristic which is lot-for-lot technique. Solution set obtained in first phase (Step 0) are inputs for improvement heuristic. Lot-for-lot technique results in zero on-hand inventories as initial solution, so related improvement algorithm starts with an accumulation strategy with backward shifting. The following pseudo code gives information about the flow of algorithm:

Step 0: Obtain an initial solution regarding demand by using LFL heuristic, and calculate all of parameters.

Step 1: Find periods where lot transfer take place by dominance property-1.1

Step 2: After determining periods, identify product that will be backward shifted by using holding and production cost savings via dominance property-1.2

Step 3: Calculate maximum possible amount of lot transfer by dominance property-2, determine lot size to be shifted randomly

Step 4: Transfer required production lot between periods, control overall cost change in objective function. If the change in cost is negative, then apply move. Otherwise, calculate simulated annealing parameters

Step 5: If move is rejected, initialize forward shifting procedure, apply step 2 and 3 iteratively, and transfer an amount of production lot to forward periods

The parameters which are underlined will be changed systematically to understand robustness of proposed algorithm. These variations should give information about how well algorithm performs under different circumstances.

Once algorithm obtains an initial solution, improvement heuristic is applied for n times. The number of n is determined as a function of actual stopping criteria. Improvement heuristic initiates backward shifting of lots with aforementioned dominance properties. Some nervousnesses may reveal in this phase. First nervousness of Dominance property-1 considers whether most loaded period (regular and overtime) $t2$ is 1. If $t2$ is equal to one, there will be no convenient period prior to $t2$. Therefore, a random t should be assigned to update $t1$ and $t2$. The step taken by this command may not yield a better solution, but this will help to search for global optimum without getting stuck in local optimum. After that, available capacity for period $t1$ is calculated, and then maximum quantity of item i which can be shifted without violating feasibility of available capacity is calculated.

Exact amount of transfer is determined by using dominance property-2. Whenever a step is taken, the value of objective function is updated. If there is an improvement in the objective function, new values are set as solution. Otherwise, ($\Phi_{cost} \geq 0$), PA, tSA values are calculated according to geometric cooling schedule formulated in [2]-[8]. If the related parameter is greater than threshold value, the move is accepted. Otherwise, lots are transferred from period t to $t+m$ *(where m is a random integer between 1 and T-1)* to overwhelm problems derived from backward shifting. The adjustments made by forward shifting are directly applied without assessment.

4   Computational Tests

Computational tests are executed on a personal computer with Intelcore i5, 2.53 GHz, 64 byte. Performance of algorithm is checked with two classes of problem sets. Class A problems are used to control optimality, while Class B is intended to reflect characteristics of algorithm for bigger problem instances [1]. The related problem type is not studied in pervious literature. Hence, data are produced with special pre-determined characteristics.

The tests of optimality (Class A) consist of 4 items, 4 periods. These set of data are used to understand how well the proposed algorithm approaches to optimal solution. Class A consists of 20 different data sets. Demands of items are normally distributed with a mean of 140 units and a standard deviation of 60. The factor h/ov is held in two levels as 1.1 and 0.9. To reflect different types of products in the data set, 4 different levels of c/h are imposed in each period. Item1 has a c/h with a mean of 12 and a standard deviation of 8. Item1 represents products with higher production cost respect to holding cost. Item2 has a c/h ratio with mean 14 and a standard deviation of 12.

These types of products have huge deviations in each period in c/h level. That means for item2, there is a great fluctuation in its data for each period. For item3, the ratio is normally distributed with a mean of 7 and standard deviation of 3. Item4 has a c/h with a mean of 2 and standard deviation of 1. These types of products represent the items with high holding costs. The performance of algorithm is evaluated by using 5 different performance measures in Table 1. (Z is the objective function value of given alternative (H: Heuristic, O: Optimal, C: Lot-for-Lot))

Table 1. Results for Class A problems and deviation from optimality

	Mean	Std. Dev.	Parameters
GAP_1	3,58	2,85	$[(Z_H-Z_O)/Z_O]*100$ (distance of heuristic from optimal)
GAP_2	18,09	17,16	$[(Z_C-Z_O)/Z_O]*100$ (distance of constructive from optimal)
GAP_3	47,43	26,3	$[(Z_C-Z_H)/Z_C]*100$ (distance taken from constructive to heuristic solution)
GAP_4	54,80	44,44	$[(Z_H-Z_O)/(Z_C-Z_O)]*100$ (distance by heuristic solution to constructive solution)
GAP_5	4494 \$	3707	$[Z_H - Z_O]$ (cost gap of heuristic from optimal solution)

The related data set is analyzed in detail and it is found out that algorithm performs quite well in CLSPO. It is also understood that solution set cannot be improved if constructive heuristic yields a GAP_2 value lower than 8 percent. Due to the fact that, lot-for-lot doesn't guarantee a capacity feasible initial solution, performance criteria based on constructive heuristic may yield some misunderstandings. Overall heuristic performance highly depends on characteristics of initial solution. Starting with a good initial solution may not yield a proper ending result. The algorithm approximately takes 1 CPU second to solve the problem. For Class A type problems total numbers of iterations have not exceeded 50 in each run. The number of optimal solution obtained by using algorithm is one for this data set.

Larger instance data is collected from a propeller shaft production company, and two different sub-classes are formed. In Class B1, 20 periods (5 months in company's MRP system) are planned for 7 different basic products. In Class B2, 35 different items have been scheduled in 6 production periods. Results obtained are evaluated with two different characteristics. One of them is actual stopping criterion and other is CPU seconds. For Class B1, algorithm mostly stopped by maximum number of iteration criterion which is fixed at 5000 for each run to facilitate a fair comparison. Approximate time to obtain a result is 8-15 CPU seconds for Class B1. Company plans to schedule its production based on %70 of its product mix. Hence, other class contains more products. Most of runs in Class B2 stopped by losing forward move capability because of high number of products within short planning horizon. For this set of problem, approximate solution time is 10-20 CPU seconds. These results show that algorithm performs quite well in big problem sets as well. Another critical performance criterion is parametric robustness of algorithm. Given high fluctuations

in data with item2, results obtained have low standard deviations in optimal solutions. Further analysis will be made regarding parameter dependent tests about robustness in next phase.

5 Robustness Tests

The performance of heuristic also depends on robustness of structure. In proposed CLSP heuristic, there are three sources of variability. One of most influential variability is demand pattern of products. Standard deviation of demand (STD) is widely used to test this aspect. Hence, we set three levels of demand deviation (Scenario 1-2-3). The ratio of average production cost to average holding cost is another source (c/h, Scenario 4-7). There are also some other sources of volatility which depends on metaheuristic used by algorithm. Determining the amount of lot to transfer backward or forward between periods is key source of algorithmic variability. The one which is uniform(both), right triangular to Delta_max (forward-backward), right triangular to zero (both shifting direction), and lastly right skewed right triangular for forward shifting and left skewed right triangular for backward shifting (Scenario 8-11). Algorithm simulation of different levels for these characteristics is listed in Table.2.

Table 2. Robustness Results for Class A problems

Sce. (S)	Deman. Mean (STD)	C/H Mean (STD)	Min COST	S	Delta Pattern	Dem. STD	C/H Mean (STD	Min COST
1	140(70)	7(3)	114628	8	delta_unif_unif	70	7(3)	114628
2	140(140)	7(3)	122992	9	delta_triatria_max	210	7(3)	140774
3	140(210)	7(3)	140390	10	delta_tria_tria_min	70	7(3)	116956
4	140(70)	1(2)	42486	11	tria_min_tria_max	210	7(3)	140771
5	140(70)	7(7)	84513					
6	140(70)	7(4)	99732					
7	140(70)	7(1)	155347					

It can be understood that increasing values of coefficient of variation (CV) in demand pattern results in higher overall costs. The first three scenarios also give information about computational efficiency of algorithm. Whenever standard deviation of demand is increased time required to obtain a solution also increases.

Scenario 1,5,6,7 have "c/h ratio" fixed at seven while standard deviation is changed. It can be interpreted as; decreasing values of standard deviation in c/h, results in higher overall cost. This result is quite interesting, although given fluctuation in parameters is increased while cost associated is decreased. The underlying reason is capability of algorithm which is a function of dominance properties decreases with lower values of standard deviation.

Scenarios 8-11 show us that there is not any specific best performing conditions for Delta pattern. It is also deduced that for scenario 8&10, algorithm performs better. It is found out that fluctuation of input-algorithmic variables directly affect obtained results. These validations help us to understand characteristics of problem better.

6   Conclusion

In this chapter, a two-stage heuristic algorithm is presented to solve capacitated lot sizing problem with overtime decisions. Results obtained indicate that algorithm is quite satisfactory in respect to optimality tests and computational time. Further robustness test shows that algorithm is very stable against fluctuations in its parameters.

As long as the production planning department has an access to real-time inventory data, the production plan generated by algorithm may be frequently updated or the study may be used as an MPS tool to execute in weekly basis (planning horizon) for such kinds of facilities that mostly have single-level items with continuous production flow. Hence, proposed heuristic procedure may be used as a part of MRP planning tool of facilities where continuous production take place. Future studies will focus on applicability of algorithm to multi-level product structures and optimization of simulated annealing parameters for this problem type.

References

1. Tempelmeier, H., and Derstroff, M.: A Lagrangean-based Heuristic for Dynamic Multilevel Multiitem Constrained Lotsizing with Setup Times. Management Science, Vol. 42, No.5, pp. 738-757. (1996)
2. Ozdamar, L., Barbarosoğlu, G.: An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. International Journal of Production Economics, (68), pp. 319-331. (2000)
3. Ozdamar, L., Bozyel MA.: The capacitated lot sizing problem with overtime decisions and setup times. IIE Transactions, 32, 1043-1057. (2000)
4. Bahl, H.C., Ritzman, L.P., and Gupta, J.N.D.: Determining lot sizes and resource requirements: A review, Operations Research, Vol. 35, no.3, pp. 329-345. (1987)
5. Jans, R., and Degraeve Z.: Modelling Industrial Lot Sizing Problems: a Review. International Journal of Production Research, Vol. 46, no.6, pp. 1619-1643. (2008)
6. Baker, K.R.: Introduction to Sequencing and Scheduling. John Wiley and Sons, USA. (1974)
7. Barbarosoğlu, G., Özdamar, L.: Analysis of solution space-dependent performance of simulated annealing: the case of the multi-level capacitated lot sizing problem. Computers & Operations Research, Vol. 27, pp. 895-903. (2000)
8. Tang, O.: Simulated annealing in lot sizing problems. International Journal of Production Economics, (88), pp. 173-181. (2004)