SIMULATED ANNEALING AND JOINT MANUFACTURING BATCH-SIZING

Ruhul SARKER

*School of Computer Science, The University of New South Wales,
ADFA, Canberra, Australia.*

Xin YAO

*School of Computer Science, The University of Birmingham,
Edgbaston, Birmingham, UK.*

Abstract: We address an important problem of a manufacturing system. The system procures raw materials from outside suppliers in a lot and processes them to produce finished goods. It proposes an ordering policy for raw materials to meet the requirements of a production facility. In return, this facility has to deliver finished products demanded by external buyers at fixed time intervals. First, a general cost model is developed considering both raw materials and finished products. Then this model is used to develop a simulated annealing approach to determining an optimal ordering policy for procurement of raw materials and also for the manufacturing batch size to minimize the total cost for meeting customer demands in time. The solutions obtained were compared with those of traditional approaches. Numerical examples are presented.

Keywords: Inventory, procurement, periodic delivery, optimum order quantity, heuristic, simulated annealing.

1. INTRODUCTION

We discuss a manufacturing system where the manufacturer uses raw material, received from an outside supplier, so to produce a finished product. Traditionally the economic lot size for raw material purchase and manufacturing batch size are determined separately. However, when the raw material is used in production, its ordering quantities are dependent on the batch quantity of the product. Therefore, it is undesirable to separate the problem of economic purchase of raw materials from economic batch quantity. As a result, one should determine the optimum production

batch size of a product and the ordering quantities of associated raw materials together. This could be done by treating production and purchasing as components of an integrated system, minimizing the total cost of the system.

Consider the raw material inventory system of an organization. Since the amount of raw material used for a product is given, the amount of raw material to be used in a batch of known quantity is also known. The raw materials are consumed at a given rate only during the production up-time, and not throughout the whole cycle time. The ordering policy of a raw material can either be based on its economic ordering quantity (EOQ) or on the requirement of raw material in a lot of economic production quantity (EPQ). The latter reflects the dependent relationship between raw material requirement and production quantity. The relationship between a product and its raw material has been the basic consideration for the model developed in this paper. The problem is also considered from the point of view of the benefit to the manufacturing firm. The proposed systematic approach supports the fact that the optimum purchasing-production policy should be determined by considering an integrated system, rather than by considering several independent systems [1, 2, 3].

It would be possible to consider either a continuous or periodic product supply policy. In this research we consider a fixed, known quantity supply at a fixed interval of time. However, the supply of raw material will be taken in a lot where the lot size is a decision variable.

The problem considered in this paper has the following attributes:
- a finite production-rate environment uses raw materials from outside suppliers,
- only the product lot sizing and its associated raw material supply quantities are under consideration,
- the supply policy for the product is to deliver an equal quantity at fixed intervals, and
- the product cannot be delivered until the whole lot is finished and quality certification is complete.

The above situation exists in many industries. The objective is to determine the optimum batch size of the product and the ordering quantities of raw materials minimizing the overall system cost. To simplify the model, we consider that a single raw material purchase provides stock for several production runs.

In the literature, there are different models for dependent lot sizing. The supply patterns of raw material considered in those models are (a) Lot for lot and (b) Multiple lot for lot. The *multiple lot for lot* is further classified into (i) single raw material purchase for several production runs and (ii) several raw material purchases for a single production run. The delivery policy of the finished product is classified into (a) continuous and (b) periodic supply. In periodic supply, a lot can either be supplied in a single shipment or in several equal shipments.

Sarker et al. [1, 2 , 3] developed a model operating under continuous supply at a constant rate for both raw material delivery policies. In the *lot-for-lot* system, the ordering quantity of raw material is assumed to be equal to the raw material required for one production run. The raw material that is replenished at the beginning of a *finished product inventory cycle* will be fully consumed at the end of the *production run*. It is assumed that the length of a production run is always less than the finished

product inventory cycle time. In *multiple lot for lot*, it is assumed that the ordering quantity of raw material is n times the quantity required for one production run, where n is an integer.

An ordering policy for raw materials to meet the requirements of a production facility under a fixed quantity, periodic delivery policy has been developed by Sarker and Parija [4], Jamal and Sarker [5], Golhar and Sarker [6] and Sarker and Golhar [7]. They considered that the manufacturer is allowed to place only one order for raw material per *finished product inventory cycle*. In this case, a fixed quantity of finished goods (say x units) is to be delivered to the customer at the end of every L units of time (fixed interval). This delivery pattern forces inventory build-up in a saw-tooth fashion during the production up-time. The on-hand inventory depletes sharply at regular intervals during the production down time until the end of the cycle. The latter one forms a stair case pattern. Recently, Sarker and Parija [8] developed another model of several purchases of raw material for a single production run under a periodic delivery policy.

In this paper, we consider a problem for determining the ordering policy for raw materials to meet the requirements of a production facility under a fixed quantity, periodic delivery policy. In this case the product cannot be delivered until the whole lot is produced because of the quality assurance requirement. We analyze a system where a single raw material purchase is utilized for several production runs. This system is economic when the ordering cost of raw material purchase is much higher than the setup cost of a production run. Figure 1 shows the finished product inventory system. The product is produced at a constant rate during the production up-time period t. The delivery of finished product starts as soon as the level of accumulated finish product inventory reaches the manufacturing batch quantity, Q_p. A fixed quantity of finished product (x) is delivered to the customers at the end of every L units of time. This forms a staircase pattern during the production down time of the cycle. To fulfill the demand on time, there may be an overlapping of two consecutive *finished product inventory cycles*. The duration or amount of overlapping depends on the production rate, demand rate and other relevant cost data. The shaded area in Figure 1 shows the overlapping of inventory depletion of a *finished product inventory cycle* with the inventory accumulation of the next cycle. Figure 2 presents the raw material inventory level of the system. At the beginning, the raw material of quantity nrQ_p is replenished for the purpose of providing stock for n production runs. The raw material is depleted at a constant rate during the production up-time of the *finished product inventory cycle*. The level of raw material inventory remains constant during the production down-time.

In this paper, the cost factors considered are ordering/setup, holding and material costs. We have the total cost equation of the system with respect to the production quantity and the equation as an unconstrained optimization problem. Two heuristics have been proposed in this paper to solve this optimization problem. The first is a simple heuristic, which is much simpler than other existing methods. The solution obtained using this approach may be suboptimal because of the dependent relationship and the nature of the solution method. The second method is based on simulated annealing (SA) [9,10,11], which is a powerful stochastic search algorithm that can be applied to complex and nonconvex optimization problems. The purposes of

designing simulated annealing are (i) to explore its use in solving batch-sizing problems and (ii) to compare its solution to those obtained by other methods.

Following this general introduction, the paper presents a brief exposition of simulated annealing. Then the mathematical formulation of the coordinated policies is developed. In the following section, the solution approaches are provided. Subsequently, numerical examples and analysis are given. The conclusions are provided in the final section.
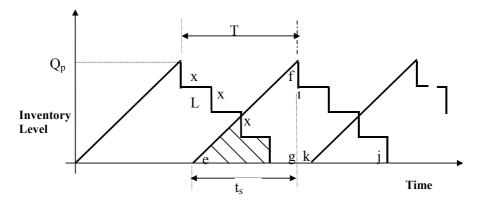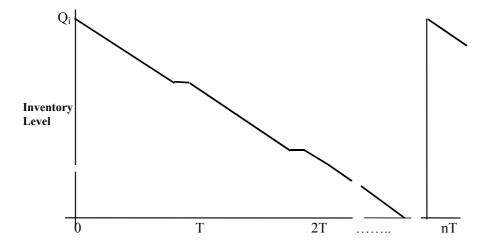

Figure 1: Finished product inventory level

Figure 2: Raw material inventory system

2. SIMULATED ANNEALING

Simulated annealing (SA) [9,10,11] is a powerful stochastic search method applicable to a wide range of problems for which little prior knowledge is available. It can produce high-quality solutions for hard combinatorial optimization [12].

The basic idea of SA comes from condensed matter physics. It is well-known that in condensed matter physics a good way to find minimum energy states, called ground states, of complex systems, such as solids, is to use the annealing technique, in which the system (solid) is first heated to some high temperature and then slowly cooled down. The system (solid) will reach a ground state if the cooling rate around the freezing point of the system is sufficiently slow. This process can be simulated on computers via abstract models, such as systems of interacting particles with many degrees of freedom.

At each step of the simulation, a new state of the system is generated from the current state by giving a random displacement to a randomly selected particle. The new state will be accepted as the current one if the energy of the new state is no greater than that of the current state; otherwise, it will only be accepted with probability

$$e^{-\frac{E_{new\_state} - E_{current\_state}}{T}}$$

where E stands for the energy of the system and T is the temperature. This step can be repeated as many times as necessary with a slow decrease of temperature in order to find a minimum energy state. Of course, only finite steps are taken in practical simulations. This simulation procedure was proposed by N. Metropolis *et al.* [9] and is called the *Metropolis* procedure [10].

SA has been applied to numerous problems in operations research and industrial engineering [18-25], such as cell formation [18], scheduling with resource constraints [19], machine conditioning [20], scheduling with multi-level product structure [21], lot sizing [23-24], and guillotine cutting [25]. A good survey of SA applications can be found in [22]. However, none of these papers discusses the problem considered in this paper. It is well-known that SA works well for some problems, but not for all. It is interesting to investigate whether SA can be applied effectively to the manufacturing batch-sizing problem described in this paper and whether SA is robust (i.e., not sensitive to different parameter settings) for this problem.

3. MATHEMATICAL FORMULATION

A general cost model is developed considering the major points of both the supplier of raw materials and the buyer of finished products. This model will be used to determine an optimal ordering policy for the procurement of raw materials, and the manufacturing batch size by minimizing the total cost for meeting equal shipments of the finished product, at fixed intervals, to the buyers.

In order to find an economic order quantity (EOQ) for the raw materials and an economic production quantity (EPQ) for the production run, it is customary to consider the cost components: (i) raw material inventory carrying cost; (ii) finished

goods inventory carrying cost; (iii) raw material ordering cost; and (iv) manufacturing setup cost.

To develop the model, the assumptions have been made: (i) the production rate is finite and constant; (ii) the production capacity is greater than the demand; (iii) no shortages are permitted; (iv) the time horizon is infinite; and (v) a fixed quantity of the product is delivered after a fixed interval of time.

The notation used in developing the cost functions is shown below:

$A_i =$ ordering cost of raw material

$A_p =$ setup cost for a product ($/setup)

$D_i =$ demand of raw material for the product in a year, $D_i = rD_p$

$D_p =$ demand rate of a product, units per year

$H_i =$ annual inventory holding cost for raw material, $/unit/year

$H_p =$ annual inventory holding cost, $/unit/year

$L =$ time between successive shipments $= x / D_p$

$m =$ number of full shipments during the cycle time $= T / L = Q_p / x$

$n =$ number of production run to use one lot of raw material

$P_p =$ production rate, units per year (here, $P_p > D_p$)

$PR_i =$ price of raw material

$Q_i =$ ordering quantity of raw material $= nrQ_p$

$Q_i^* =$ optimum ordering quantity of raw material

$Q_p =$ production lot size

$r =$ amount/quantity of raw material required in producing one unit of a product

$t =$ production up-time in years in a cycle of length T

$T =$ cycle time measured in years $= Q_p / D_p$, where $T > t$

$x =$ shipment quantity to customer at a regular interval (units/shipment)

3.1. Finished Product Inventory

Since the product cannot be delivered to the customers until the whole lot is completed and quality certification is ready, there is a continuous build-up of finished product inventory, at the rate equal to production, during the production up-time of a given lot (Figure 1: line *ef*). The delivery of the finished product is permitted during the production down time of that lot only. In order to fulfill the demand in time, the production of a new lot may be started before finishing the delivery of the previous lot.

From Figure 1, *We can write:*

$$t = Q_p / P_p \quad \text{and} \quad Q_p / D_p = mL ,$$

and the finished product inventory in a cycle $=$ Area ($efg + ijk$)

$$\text{Area } (ijk) = x(m-1)L + x(m-2)L + \cdots + x2L + xL = x\frac{(m-1)m}{2}L .$$

So, the finished product inventory in a cycle $= \frac{1}{2}Q_p t + x\frac{(m-1)}{2}mL$

Hence, the average finished product inventory in a cycle

$$= \frac{\frac{1}{2}Q_p t + \frac{x(m-1)}{2}mL}{T}$$

$$= \frac{1}{2}Q_p\frac{t}{T} + x\frac{m-1}{2}\frac{mL}{T}$$

$$= \frac{1}{2}Q_p\frac{D_p}{P_p} + \frac{mx}{2} - \frac{x}{2}$$

$$= \frac{1}{2}Q_p\left(\frac{D_p}{P_p} + 1\right) - \frac{x}{2} \tag{1}$$

3.2. Raw Material Inventory

The raw material inventory is shown in Figure 2. In this case, the inventory for raw material in a cycle of nT periods is

$$= \frac{1}{n}[\frac{1}{2}Q_i t + Q_i T + \frac{1}{2}Q_i t + Q_i 2T + \frac{1}{2}Q_i t + \cdots + Q_i(n-1)T + \frac{1}{2}Q_i t]$$

$$= \frac{1}{2}Q_i t + \frac{(n-1)}{2}Q_i T$$

For raw material the average inventory per cycle

$$= \frac{\left[\frac{1}{2}Q_i t + \frac{(n-1)}{2}Q_i T\right]}{nT}$$

$$= \frac{1}{2}rQ_p\left(\frac{D_p}{P_p} + n - 1\right) \tag{2}$$

Then the total Cost Function for a year can be represented as follows:
Total cost of the system,

$$TC_1 = \frac{D_p}{Q_p}\left(A_p + \frac{A_i}{n}\right) + \frac{Q_p}{2}(kk) - \frac{x}{2}H_p \tag{3}$$

where, $kk = \left(\dfrac{D_p}{P_p} + 1\right)H_p + rH_i\left(\dfrac{D_p}{P_p} + n - 1\right) = CC + n\, r\, H_i$,

where, $CC = \left(\dfrac{D_p}{P_p} + 1\right)H_p + rH_i\left(\dfrac{D_p}{P_p} - 1\right)$

Since, $Q_p = mx$, then TC_1 can be rearranged as follows:

$$TC_2 = \frac{D_p}{mx}\left(A_p + \frac{A_i}{n}\right) + \frac{mx}{2}(kk) - \frac{x}{2}H_p \tag{4}$$

TC_2 is a nonlinear function with integer variable m and n . The behavior of this function is discussed in the next section. Our objective is to determine the optimal m and n while minimizing the total cost.

4. SOLUTION APPROACH

The total cost of the system can be expressed either as a function of continuous variable Q_p and an integer variable n (equation 3) or as a function of the integer variables m and n (equation 4). TC_1 and TC_2 are no differentiable since they contain integer variables. As such, a closed form solution for Q_p cannot be obtained. However, it can be shown that TC_1 is a piecewise convex function of Q_p . Sarker and Parija [4] examined and plotted a similar function. Efficient algorithms may be applied to solve this problem by using a discrete optimization technique. An algorithm has been proposed by Moinzadeh and Aggarwal [13] to obtain a global minimum for such a function. Although it may not be efficient, a simple procedure is developed here to obtain an optimal or near-optimal solution in this paper.

We consider TC_2 to determine the optimal solutions. The reasons for choosing TC_2 , instead of TC_1 , are:

    (i)   the range of Q_p is too large as compared to m ,

    (ii) Q_p is a dependent variable of m ,

    (iii) the solution may not guarantee an integer value of m , and

    (iv) in reality, Q_p is integer too.

With a search algorithm, the number of iterations will be much lower with a function of m and n than that of Q_p and n . The function TC_2 can be plotted connecting the values of TC_2 for integer n and m . For any given value of m , the connected function of n looks like a convex function. This similarity to a convex function is also true for the connected function of m , for any given value of n .

In the following section, we develop two approaches to optimizing the function TC_2 .

4.1. Heuristic Based on Traditional Optimization

The properties of the function motivated us to develop a conventional optimization based approach. This approach is well accepted in the OR/MS literature in regard to finding the near optimal solution (Silver, Pyke and Peterson [14] and Joglekar and Tharthare [15]). Relaxing the requirement that m and n are integer and allowing them to be continuous, and then differentiating TC_2 with respect to m and equating to zero, we get

$$m^* = \sqrt{\frac{2D_p\left(A_p + \frac{A_i}{n_i}\right)}{x^2(kk)}} \tag{5}$$

Substituting m^* in Equation (4) we get the annual total cost as

$$TC_2 = \frac{D_p}{x}\left(A_p + \frac{A_i}{n}\right)\sqrt{\frac{x^2(kk)}{2D_p\left(A_p + \frac{A_i}{n}\right)}} + \frac{x}{2}(kk)\sqrt{\frac{2D_p\left(A_p + \frac{A_i}{n}\right)}{x^2(kk)}} \cdot \frac{x}{2}H_p$$

$$= \sqrt{2D_p\left(A_p + \frac{A_i}{n}\right)[(CC) + n\,r\,H_i]} - \frac{x}{2}H_p \tag{6}$$

Differentiating the modified TC_2 (equation 6) with respect to n and equating to zero, we get

$$n^* = \sqrt{\frac{A_i(CC)}{A_p r H_i}} \tag{7}$$

Both m and n should be integers. However the values obtained from equation (5) and (7) may not be integer, as we solved the relaxed function. In such a case, the neighbouring integer point (values of m and n) is sorted which incurs the minimum cost. The complete heuristic algorithm is presented below:

Algorithm: finding batch size.

Step 0. Initialize and store $D_p, P_p, A_p, A_i, H_p, H_i, r$ and x.

Step 1. Compute m^* using (equation 5).

Step 2. Compute n^* and TC_2 using (7) and (6) respectively.

If both m^* and n^* are integers, then calculate Q_p^* and go to Step 6. Otherwise go to Step 3.

Step 3. If m^* is integer and n^* is not, then compute TC_2 using $n = \lfloor n^* \rfloor$ and $\lceil n^* \rceil$. Choose the n that gives minimum TC_2. Calculate Q_p^* and go to Step 6. Otherwise go to Step 4.

Step 4. If n^* is integer and m^* is not, then compute TC_2 using $m = \lfloor m^* \rfloor$ and $\lceil m^* \rceil$. Choose the m that gives minimum TC_2. Calculate Q_p^* and go to Step 6. Otherwise go to Step 5.

Step 5. If both n^* and m^* are non-integers, then compute TC_2 for all four combinations of n^* and m^*:

Option 1: $\lfloor m \rfloor \lfloor n^* \rfloor$ Option 2: $\lfloor m \rfloor \lceil n^* \rceil$

Option 3: $\lceil m \rceil \lfloor n^* \rfloor$ Option 4: $\lceil m \rceil \lceil n^* \rceil$

Choose n and m that gives the minimum TC_2. Calculate Q_p^* and go to Step 6.

Step 6. Stop

Numerical Example

A numerical example is provided to show the applicability of the model developed in this research. The data for the example are as follows:

$$D_p = 4 \times 10^6 \text{ units}$$
$$P_p = 5 \times 10^6 \text{ units}$$
$$A_p = \$50.000 \text{ per setup}$$
$$A_i = \$3,000 \text{ per order}$$
$$H_p = \$1.20 \text{ per unit per year}$$
$$H_i = \$1.00 \text{ per unit per year}$$
$$x = 1,000 \text{ units.}$$

Solution: using traditional optimization based heuristic

Let $r = 1.00$
From Equation (5), the number of full shipments during the cycle time $m = 14.11$
From Equation (7), the number of lot size $n = 10.84$
From Equation (6), the total cost $TC = \$182,914.30$
To find the integer value of m and n, we try for $m = 14$ and 15 and $n = 10$ and 11

m	n	Total Cost ($)	Remarks
14	10	183,120.00	Highest
	11	**182,327.80**	Lowest total cost
15	10	182,433.30	In between
	11	182,660.60	In between

So the number of full shipments during the cycle time is $m = 14$, the number of lots is $n = 11$ and the lot size $= 14,000$ units.

4.2. Simulated Annealing Approach

The classical SA (CSA) [10] starts with an initial configuration generated at random. At each step, it selects the next solution Y from the neighbourhood N_X of the current solution X. The next solution will be accepted as the current one if its cost is no greater than that of the current solution; otherwise, it will only be accepted with probability

$$e^{-\frac{C_Y - C_X}{T}}$$

Where C_Y represents the cost (to be minimized) of solution Y and C_X is the cost of solution X. This procedure is repeated with a slow decrease of the control parameter T, called temperature, until a sufficiently good solution has been found. CSA can be summarized by the following algorithm:

```
select initial solution X at random;
select initial temperature T ;
REPEAT
  REPEAT
    randomly select Y from N_X with uniform distribution;
    IF C_Y ≤ C_X
      THEN accept Y as the new solution
      ELSE accept Y as the new solution with probability e^{-(C_Y - C_X)/T}
  UNTIL  inner-loop stop criterion' is satisfied
  decrease temperature T
UNTIL  outer-loop stop criterion' is satisfied
```

One of the major tasks in developing an SA-based approach is to define a suitable neighborhood function, i.e., to define a suitable generation function that guarantees every feasible solution in the search space can be reached. In this paper, the neighbors of a given solution $X = (m, n)$ are $\{(m + i, n + j) \mid i, j = 1, -1\}$ in our algorithm. The acceptance probability used in our algorithm is the same as that defined in the above CSA.

The initial solution was generated uniformly at random within a user-specified range, e.g., for m and n between 1 and 1000. The initial temperature was generated using the following procedure: (1) Generate 100 solutions at random; (2) Find the solutions with the minimum cost and with the maximum cost; (3) The initial temperature was set at twice the difference between the maximum and minimum cost. The program we developed allows its user to adjust the initial temperature for each run as well.

Two cooling schedules were tested in our algorithms. Both performed well, i.e., enabled the algorithm to find the global optimum consistently. The first cooling schedule is $T = \rho * T$, where ρ is a parameter that can be adjusted by the user. The optimal setting of ρ is highly problem-dependent. Different problems require different values. The common practice to find a near-optimal ρ is to start with a relatively smaller value, such as 0.85, and than gradually increase it to a large value close to 1.0, such as 0.99. This trial-and-error process can be very tedious. A better alternative is to design a robust SA algorithm which is not sensitive to the parameter setting. In other words, the performance of the SA algorithm changes little when a different value of ρ is used. The SA algorithm used in this paper is very robust in this sense. There is no need to tune ρ as long as it is within a reasonable range, i.e., 0.85 to 0.99, which is the range used by many SA users and researcher. To show the robustness of the SA algorithm, we ran the SA algorithm for $\rho = 0.85$ to 0.99 and obtained the global optimum in all cases using the given numerical example.

The second cooling schedule used in our algorithm is $T = \dfrac{T}{1 + \beta T}$, where β is

a parameter that can be adjusted by the user. Similar to the case of ρ, the optimal value of β depends on the problem. There is no universally optimal β which is best for all problems. A trial-and-error process for finding a good β value has to be used in practice. However, if an SA algorithm is robust, there will be little need for a lengthy trial-and-error process. Robustness is one of the criteria in designing our SA algorithm. To test how robust our algorithm is, we ran it with $\beta = 0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 3.0$ and 4.0 in our experimental studies. Thirty runs were conducted for each parameter setting. The SA algorithm was able to find the global optimum at $m = 14, n = 11$ consistently for all runs. None of our runs took more than a few seconds. These experiments have shown that our SA algorithm worked well with a wide range of parameter settings. There is no need to have a time-consuming parameter-tuning process. A default value of $\beta = 1.0$ can be used for our problem.

The inner loop stop criterion used in our algorithm was determined by the number of iterations for which the temperature T was kept the same. It can be set by the user easily. The outer loop stop criterion was determined by two factors. One was the maximum number of iterations set by the user. The other was the lowest temperature. That is, the SA algorithm stops when its temperature is below this value.

Further improvement to our SA algorithm was made after the above success. We modified the range of i and j in the neighbourhood definition. The neighbourhood of $X = (m,n)$ now becomes $\{(m+i, n+j) \mid i, j$ in range $[-range, +range]\}$, where i and

j are generated uniformly at random within [−*range*,+*range*], and *range* is a user-defined parameter that depends on the initial ranges of m and n. This improvement enables our algorithm to converge much faster, since larger jumps are made possible through this neighbourhood. Our previous work has shown the benefit of having a large neighbourhood size [16]. The optimal solution can be found in less than a second.

5. ANALYSIS AND DISCUSSION

It is clear from previous studies that the joint product batch size and raw material ordering policy is more cost-effective than the separate policy [1, 2, 3, 17]. In the joint batch sizing problem, we consider that the raw material of a single purchase will provide stock to several production runs. This is true when an ordering cost of raw material purchase is much higher compared to the setup cost of a production run. In the example problem, if the ordering cost is $25 or less, it is not economic to use single raw material purchase for more than one production run.

The behavior of the function encouraged us to develop a simple, calculus-based, heuristic to solve the problem. This approach is generally well accepted in the OR/MS literature. However, there is no guarantee that such a heuristic will produce a global optimum. So we designed an SA algorithm. Although there is no guarantee that SA will find an exact global optimum in finite time, the solution obtained by our SA algorithm for the example problem has been found consistently excellent. In order to evaluate the quality of solutions, an enumeration algorithm was used to search for the exact global optimum (several hours on a PC) for the example problem. This result confirmed that the solution found consistently by SA and the simple heuristic method is indeed the global optimal solution.

6. CONCLUSIONS

A model for a two-stage batch environment has been proposed in this paper. This model takes a finite-rate of production into account and jointly determines batch-sizes for the product and order-sizes for the associated raw materials. To avoid complexity in formulation, the variable m and n are assumed to be integers for all cases. However, these are relaxed in the solution approach that is performed. As a result, it is not expected that the solution will be globally optimal. This method can be considered as a simple heuristic to solve the stated problem with little computational effort and an acceptable level of solution quality. To improve the quality of the solution, we developed an SA algorithm, which can produce high-quality solutions reliably and efficiently. The SA algorithm does not require the cost function to be differentiable or even continuous. It can deal with functions with multiple local optima. A distinct feature of our SA algorithm is its robustness against different parameter settings. The algorithm worked very well under a wide range of parameter values. There was no need to tune parameters such as ρ and β using a lengthy trial-and-error process. The experimental results using an example problem show that the SA algorithm would be a very useful tool in attacking other batch-sizing problems.

The model developed in this paper can be applied in real life situations where (i) a finite production-rate environment uses raw materials taken from outside suppliers, (ii) the supply policy is to deliver equal quantities at fixed intervals, and (iii) the product cannot be delivered until the whole lot is finished and quality certification is ready. This situation may arise in the chemical and pharmaceutical industries under certain conditions.

REFERENCES

[1]   Sarker, R.A., Karim, A.N.M., and Azad, S., "Integrated inventory system: cases of product-rawmaterials and producer-wholesalers", *37th Annual Convention of the Institution of Engineers Bangladesh,* Rajshahi, Bangladesh, 1993.

[2]   Sarker, R.A., Karim, A.N.M., and Azad, S., "Two cases of integrated inventory", *Journal of the Institution of Engineers*, *Bangladesh*, 21 (4) (1995) 45-52.

[3]   Sarker, R.A., Karim, A.N.M., and Haque, A.F.M.A., "An optimal batch size for a production system operating under a continuous supply/demand*", International Journal of Industrial Engineering*, 2 (3) (1995) 189-198.

[4]   Sarker, B. R., and Parija, G.R., "An optimal batch size for a production system operating under a fixed-quantity, periodic delivery policy", *Journal of the Operational Research Society*, 45 (8) (1994) 891-900.

[5]   Jamal, A.M.M., and Sarker, B.R., "An optimal batch size for a production system operating under a just-in-time delivery system", *International Journal of Production Economics*, 32 (2) (1993) 255-260.

[6]   Golhar, D.Y., and Sarker, B. R., "Economic manufacturing quantity in a just-in-time deliver system", *International Journal of Production Research*, 30 (5) (1992) 961- 972.

[7]   Sarker, B. R., and Golhar, D.Y., "A reply to 'A note to "Economic manufacturing quantity in a just-in-time delivery system"'", *International Journal of Production Research*, 31 (11) (1993) 27-49.

[8]   Sarker, B. R., and Parija, G.R., "Optimal batch size and raw material ordering policy for a production system with a fixed-interval, lumpy demand delivery system", *European Journal of Operational Research*, 89 (1996) 593-608.

[9]   Metropolis, N, Rosenbluth, A., Rosenbluth, M., and Teller, E., "Equations of state calculations by fast computing machines", *Journal of Chemical Physics,* 21 (1953) 1087-1091.

[10] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M.P. "Optimization by simulated annealing", *Science*, 220 (1983) 671-680.

[11] Yao, X., "A new simulated annealing algorithm", *International Journal of Computer Mathematics*, 56 (1995) 161-168.

[12] Yao, X., "Call routing by simulated annealing", *International Journal of Electronics*, 79 (4) (1995) 379-387.

[13] Moinzadeh, K., and Aggarwal, P., "Order expedition in multi-level production inventory system", Paper presented at *TIMS/ORSA Joint National Meeting*, Las Vegas, NV, USA, May 7-9, 1990.

[14] Silver, E.A., Pyke, D. F., and Peterson, R., *Inventory Management and Production Planning and Scheduling, (3rd ed.)*, John Wiley & Sons, New York, 1998.

[15] Joglekar, P., and Tharthare, S., "The individually responsible and rational decision approach to economic lot sizes for one vendor and many purchasers", *Decision Sciences*, 21 (1990) 492-506.

[16] Yao, X., "Simulated annealing with extended neighbourhood size," *International Journal of Computer Mathematics*, 41 (1991).

[17] Goyal, S. K., "An integrated inventory model for a single product system", *Operational Research Quarterly*, 28 (1977) 539-545.

[18] Adil, G.K., Rajamni, D., and Strong, D., "Assignment allocation and simulated annealing algorithms for cell formulation", *IIE Transactions*, (1997) 53-67.

[19] Gemmill, D.D., and Tsai, Y.-W., "Using a simulated annealing algorithm to schedule activities of resource-constrained projects", *Project Management Journal*, (1997) 8-20.

[20] Khan, Z., Prasad, B., and Singh, T., "Machining condition optimization by genetic algorithms and simulated annealing", *Computers & Operations Research*, (1967) 647-657.

[21] Kim, J.-U., and Kim, Y.-D., " Simulated annealing and genetic algorithms for scheduling products with multi-level product structure", *Computers & Operations Research*, (1996) 857-868.

[22] Koulaman, C., Antony, S.R., and Jean, R., "A survey of simulated annealing applications to operations research problems," *Omega*, 22 (1994) 41-56.

[23] Kuik, R., and Salomon, M., "Multi-level lot sizing problems: evaluation of a simulated annealing heuristics", *European Journal of Operational Research*, 45 (1990) 25-37.

[24] Kuik, R., Salomon, M., Wassenhove, L.N., and Maes, J., "Linear programming, simulated annealing and tabu search heuristics for lotsizing in bottleneck assembly systems", *IIE Transactions*, 25 (1993) 62-72.

[25] Parada, V., Sepulveda, M., and Solar, M., "Solution for the constrained guilotine cutting problem by simulated annealing", *Computers & Operations Research*, (1998) 37-47.