Technical Memorandum Number 827

The Multiple-Family ELSP with Fill Rates

by

S.M. Karalli
College of Business Administration, California State University
karallis@csus.edu

A. D. Flowers
Weatherhead School of Management, Case Western Reserve University,
dale.flowers@case.edu,

September 2010

Department of Operations
Weatherhead School of Management
Case Western Reserve University
330 Peter B. Lewis Building
Cleveland, Ohio 44106

The Multiple-Family ELSP with Fill Rates

S.M. Karalli*
College of Business Administration, California State University, Sacramento,
6000 J Street, Sacramento, CA 95819
Email: karallis@csus.edu

A.D. Flowers
Department of Operations, Weatherhead School of Management,
Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106
Email: dale.flowers@case.edu

September 13, 2010

Abstract

We revisit the Multiple Family Economic Lot scheduling Problem with safety stocks (MFELSP-SS) [1]. In this technical note we focus our attention on safety stock maintained to meet fill rate requirements, again considering them explicitly in the formulation.

Differences between service level and fill rate criteria are discussed. A solution procedure for the MFELSP with fill rates (MFELSP-FR) is presented for this model.

Subject classifications:
inventory/production: approximations/heuristics, multi-item, multifamily, safety stock;
production/scheduling: applications, approximations/heuristics, cyclic schedule
Area of review: Manufacturing, Service, Supply Chain Operations

*Corresponding author

1 Background

Consider an item, whose demand is normally distributed with mean $d = 1000$ units per period and standard deviation of demand $\sigma = 400$ units per period, is scheduled for production every t periods. Safety stock must be maintained in order to meet a predetermined service criteria. Figure 1 depicts the behavior of safety stock as a function of time when held (a) to hedge against stockouts, with service level requirements of $SL = 95\%$, $SL = 97\%$, and $SL = 98\%$ (dashed curves), and (b) to meet fill-rate requirements of $f = 95\%$, $f = 97\%$, and $f = 98\%$ (solid curves).

Service Level Safety Stock. Safety stock held to hedge against stockouts varies with time according to the expression $400z\sqrt{t}$ where z is the standard normal variate (or safety factor) corresponding to the required service level (one minus the probability that a stock out will occur) in the time interval $t \in (0, 30)$.

Fill-Rate Safety Stock When considering the fill rate, one seeks to determine a product's expected excess demand, which cannot be filled from inventory so that (1) below holds, where the LHS, f, is the required fill rate for the item under consideration.

$$f = \frac{q}{q + e} \tag{1}$$

The quantity on hand, q, at the beginning of the inventory cycle id defined in (2). The right hand side of (2) is the sum of the mean demand between production runs and the safety stock required to meet the required fill rate, f.

$$q = dt + z\sigma\sqrt{t} \tag{2}$$

The expected shortage, e, is the demand expected to exceed q given $z\sigma\sqrt{t}$ units of safety stock, is defined in (3). The denominator in (1) is the expected demand given q units were in stock.

$$e = E(z)\sigma\sqrt{t} \tag{3}$$

where, $E\left(z\right)$, given in (4), is the partial expectation evaluated at a value of z that satisfies the identity in (1).

$$E(z) = \frac{1}{2\pi}\int_{z}^{\infty}\left(x-z\right)e^{-\frac{x^2}{2}}\,dx \tag{4}$$

We combine (1), (2), (3), and (4) to yield (5), below.

$$E\left(z\right) - \left(\frac{1-f}{f}\right)\left(\frac{dt}{\sigma\sqrt{t}}+z\right) = 0 \tag{5}$$

We can now express t in terms of z as follows in (6) below

$$\sqrt{t} = \left[\left(\frac{f}{1-f}\right)E(z) - z\right]\left(\frac{\sigma}{d}\right) \tag{6}$$

Property 1. An increase (decrease) in t requires a decrease (increase) in z to maitain the identitly in (1).

Proof. Using (1) and (3), we can write (7) below.

$$\frac{dt + z\sigma\sqrt{t}}{dt + z\sigma\sqrt{t} + E(z)\sigma\sqrt{t}} = f \tag{7}$$

Because $\frac{\partial E(z)}{\partial z} < 0$ [2], maintaining the identity in (7) requires a reduction in the value of z to raise the value of the denominator of the LHS of **??** when t, in the numerator, increases. $\square$

Property 2. The safety stock function, $ss = z\sigma\sqrt{t}$, increases with t, reaches a maximum point and then decreases.

Proof. By Property 1 z is a decreasing function of t for any given fill-rate, ss

continues to rise until $z = 0$ where $t_0 = \left[\left(\frac{\sigma}{d}\right) \left(\frac{f}{1-f}\right) \right]^2 \frac{1}{2\pi}$. With $z < 0$ to the right of t_0, ss is decreasing $\forall t > t_0$. $\square$

Property 1 can be seen on the 95%, 97%, and 98% fill-rate curves in Figure 1. The hollow circles on these curves are points where the value of z is zero, that is, where the identity in (8) holds. Safety stock requirements drop below zero as shown on the 95% and 97% fill-rate curves to the right of the filled circles, where the order quantity, q, required to satisfy f is smaller than the mean demand the replenishment cycle.

$$\sqrt{t} = \frac{1}{\sqrt{2\pi}} \left(\frac{\sigma}{d}\right) \left(\frac{f}{1-f}\right) \tag{8}$$

Fill Rate Safety Factors. Property 3 of the fill-rate safety factor will be useful in the development of an efficient solution algorithm.

Property 3. (The Fill-Rate Safety Factor Upper Bound)
$\forall\, f \in (0,1)\,, \exists!\, z_0^f \ni \left(\frac{f}{1-f}\right) E\left(z_0^f\right) - z_0^f = 0$ (see (7)).

Proof. When $z = z_0^f$, the value of t is 0. Values of $z \geq z_0^f$ imply $\sqrt{t} \leq 0$ and are therefore not in the domain of values of z for f. $\square$

Property 3 is illustrated in Figure 2. Safety factor curves for $f = 90\%$ (dark gray), 95% (gray), and 98% (black) displayed for an item with $d = 1000$ and $\sigma = 400$. When $t = 0$, $z_0^{90\%} = 0.901, z_0^{95\%} = 1.159$, and $z_0^{98\%} = 1.485$.

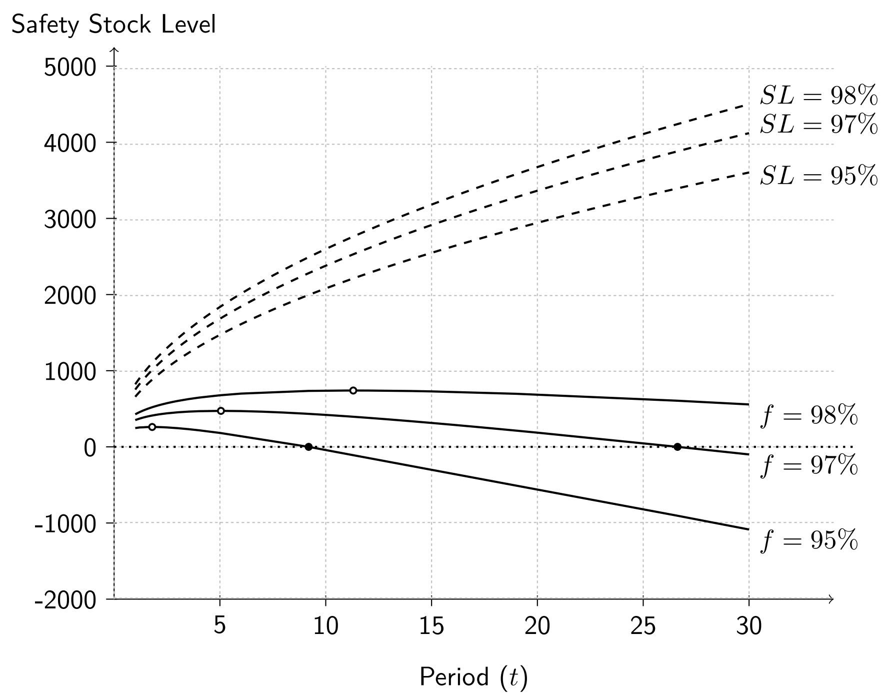

Figure 1: Requirements of (a) service level safety stock (dashed curves), and (b) fill-rate safety stock (solid curves) as a function of time for an item with $d = 1000$ units per period, and $\sigma = 400$ units per period.

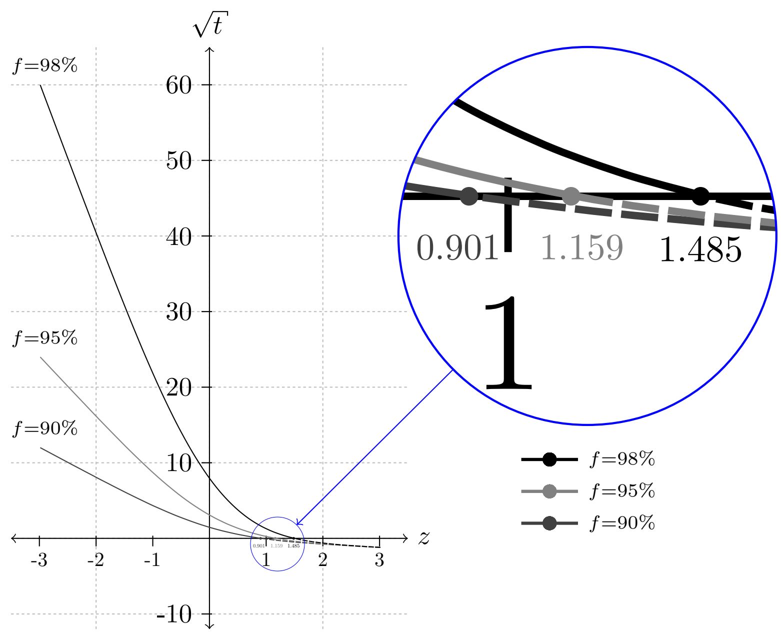

Figure 2: Upperbound of the safety factor, z, determined exclusesively by the fill rate, f.

2 The Family Planning Problem with Fill Rates (FPP-FR)

The following parameters are inputs to the FPP-FR:

S_i	Setup time for family i, $i \in \mathbb{N}$
s_{ij}	Setup time for the j^{th} item in family i, $(i,j) \in \mathbb{N} \times \mathbb{N}$
A_i	Setup cost for family i
a_{ij}	Setup cost for item j in family i
d_{ij}	Demand mean for item j in family i
σ_{ij}	Demand standard deviation for item j in family i
p_{ij}	Production rate for item j in family i
ρ_{ij}	$= \frac{d_{ij}}{p_{ij}}$, production capacity needed by item j in family i
ρ	$= \sum_{i\in\mathbb{N}} \sum_{j\in\mathbb{N}} \frac{d_{ij}}{p_{ij}}$, total production capacity required
h_{ij}	Inventory carry cost for item j in family i
f_{ij}	Required fill rate for item j in family i
z_{ij}	Standard deviation from $\mathbf{N}(0,1)$ corresponding to f_{ij}

The decision variables of the problem are:

T	Length of basic period
K_i	Multiplier of family i
k_{ij}	Multiplier for item j in family i

The following notation will appear in the discussion:

$C(\cdot)$	Cost function evaluated at$(\cdot)$
$G(\cdot)$	Feasibility constraint (see the formulation below)
$F(\cdot)$	Fill rate identity constraint (see the formulation below)
$\mathbb{N}$	Set of subscripts of i or j from 1 to N
$\mathbf{K}$	$N-$vector of K_is

$\mathbf{k}$ $\quad N \times N-$matrix of k_{ij}s

$\mathbf{k}_{i\cdot}$ $\quad$ Vector of all item multipliers in family i

$\mathbb{P}$ $\quad = \{2^p : p \in \mathbb{Z}_+\}$, set of integer powers of two.

T_{ij} $\quad = TK_i k_{ij}$

Problem Formulation The firm's objective is to find $(T, \mathbf{K}, \mathbf{k})$ so as to minimize the total average cost, given in (9).

$$C(T, \mathbf{K}, \mathbf{k}) = \sum_{i \in \mathbb{N}} \left(\frac{A_i}{TK_i k_{ij}} + \sum_{j \in \mathbb{N}} \frac{a_{ij}}{TK_i k_{ij}} + b_{ij} TK_i k_{ij} + g_{ij} \sqrt{TK_i k_{ij}} \right) \tag{9}$$

$$G(T, \mathbf{K}, \mathbf{k}) = \sum_{i \in \mathbb{N}} \frac{S_i}{K_i} + \sum_{i \in \mathbb{N}} \sum_{j \in \mathbb{N}} \frac{s_{ij}}{K_i k_{ij}} - (1 - \rho) T \leq 0 \tag{10}$$

$$F(f_{ij}, d_{ij}, \sigma_{ij}, T_{ij}) = \sqrt{T_{ij}} - \left(\frac{\sigma_{ij}}{d_{ij}} \right) \left[E(z_{ij}) \left(\frac{f_{ij}}{1 - f_{ij}} \right) - z_{ij} \right] = 0 \tag{11}$$

Problem M

The objective of Problem M is to find $(T, \mathbf{K}, \mathbf{k}, \mathbf{z})$ so as to

$$\begin{aligned}
\text{Minimize} \quad & (9) \\
\text{Subject to} \quad & (10) \\
& (11); && \forall (i, j) \in \mathbb{N} \times \mathbb{N} \\
& K_i \in \mathbb{P}; && \forall j \in \mathbb{N} && (12) \\
& K_{ij} \in \mathbb{P}; && \forall (i, j) \in \mathbb{N} \times \mathbb{N} && (13) \\
& T \geq 0 && && (14)
\end{aligned}$$

where,

$$b_{ij} = \frac{1}{2} h_{ij} d_{ij} (1 - \rho) \tag{15}$$

$$g_{ij} = \begin{cases} h_{ij} z_{ij} \sigma_{ij} & \text{if } z_{ij} \geq 0 \\ \frac{1}{2} h_{ij} z_{ij} \sigma_{ij} & \text{otherwise} \end{cases} \tag{16}$$

Constraint (10) is a necessary feasibility condition. To ensure that the safety stock quantities are set to levels that satisfy the required fill rate for each item, we require that the identity in (11) be enforced by making it a constraint. The Constraints in (12) and (13) require that the family and item multipliers be integer-powers-of-two (IPOT). While a strictly positive value of T is sought, the strict inequality can be written as a weak one as shown in (14) without loss of accuracy.

The continuous relaxation and solution proporties are analagous to [1]. We now proceed to the solution procedure.

3 The Multiple Family Algorithm

The steps for solving Problem F are

1. solve Problem S n^2 times; for each item $(i, j) \in \mathbb{N} \times \mathbb{N}$ run
   - Algorithm 1, and
   - Algorithm 2
2. employ Algorithm 3 n^2 times; for each family $i \in \mathbb{N}$
   - find the KKT points of the continuous relaxation of Problem F
   - select the KKT point, $m^\star$, with the lowest objective value, $c_i^{m^\star}(x_i^\star, \mathbf{y}^\star)$.
     - $\sum_{i \in \mathbb{N}} c_i^{m^\star}(x_i^\star, \mathbf{y}^\star)$ is a lower bound for Problem F.
3. employing Algorithm 4, execute the multi-family roundoff algorithm to
   - compute n^2 trial values of $(T, \mathbf{K}, \mathbf{k})$
   - select the cost minimizing $(T^\star, \mathbf{K}^\star, \mathbf{k}^\star)$

For a tolerance of $\epsilon = 0.0001$, Algorithm 1 usually requires five to seven iterations to provide an answer. For a tolerance of $l = 0.0001$, Algorithm 2 will solve problem S in n iterations, where n is the smallest integer satisfying the inequality in (17) [3].

$$(0.618)^{n-1} \leq \frac{l}{b-a} \tag{17}$$

For fill-rates of 92% and higher, $n = 24$. Below 92%, $n = 23$. The partial expectation in (4) can be expressed as follows [2]:

$$E(z) = \phi(z) - zF(z) \tag{18}$$

where,

$\phi(z)$	Standard normal p.d.f. evaluated at z
$\Phi(z)$	Area under the standard normal curve to the left of z
$F(z)$	Area under the standard normal curve to the right of z; that is $F(z) = 1 - \Phi(z)$

The Newton-Raphson Method employs $f(z)$ and $f'(z)$ using (19) and (20).

$$f(z) = \left[\left(\frac{f}{1-f}\right)E(z) - z\right]\left(\frac{\sigma}{d}\right) - \sqrt{t} \tag{19}$$

$$f'(z) = \left[-\left(\frac{f}{1-f}\right)F(z) - 1\right]\left(\frac{\sigma}{d}\right) \tag{20}$$

Algorithm 1 Procedure to find z_t^f using the Newton-Raphson Method

Require:

f, d, σ //item preperties, subcripts supressed

t //to find z_0^f set $t = 0$

Initialize

$\epsilon \leftarrow 0.0001$

$z_0 \leftarrow 3$

Begin Newton-Raphson Method

$z_1 \leftarrow z_0 - \frac{f(z_0)}{f'(z_0)}$

while $|z_1 - z_0| > \epsilon$ **do**

  $z_0 \leftarrow z_1$

  $z_1 \leftarrow z_0 - \frac{f(z_0)}{f'(z_0)}$

end while

$z_t^f \leftarrow z_1$

return z_t^f

End

Algorithm 2 Solution Procedure for the Single Item Problem

Require:

$\beta \leftarrow \frac{\sqrt{5}-1}{2} \qquad \alpha \leftarrow 1-\beta \qquad \varepsilon \leftarrow 0.0001 \qquad \left(z_a^0, z_b^0\right) \leftarrow \left(-3, z_0^f\right)$

Equations

1: $z^\star \leftarrow \frac{\left(z_a^k + z_b^k\right)}{2}$

2: $t^\star \leftarrow \left(\frac{\sigma}{d}\right)^2 \left[E(z^\star)\left(\frac{f}{1-f}\right) - z^\star\right]^2$

BEGIN

3: $\left(z_a^1, z_b^1\right) = \left(z_a^0, z_b^0\right)$

4: **for** $k = 1$ to 24 **do**

5: $\quad \delta^k \leftarrow z_b^k - z_a^k$

$\quad \left(\lambda^k, \mu^k\right) \leftarrow \left(z_b^k + \alpha\delta^k, z_b^k + \beta\delta^k\right)$

$\quad \delta^k \leftarrow \mu^k - \lambda^k$

6: **if** $\delta^k \leq \varepsilon$ **then**

7: **return** $z^\star$ and $t^\star$ //Using the equations in statements 1 & 2

Stop: EXIT algorithm

8: **else**

9: $t_\lambda^k \leftarrow \left(\frac{\sigma}{d}\right)^2 \left[E(\lambda^k)\left(\frac{f}{1-f}\right) - \lambda^k\right]^2 \qquad t_\mu^k \leftarrow \left(\frac{\sigma}{d}\right)^2 \left[E(\mu^k)\left(\frac{f}{1-f}\right) - \mu^k\right]^2$

$c\left(\lambda^k\right) \leftarrow \frac{s}{t_\lambda^k} + \frac{1}{2} h d t_\lambda^k + g_\lambda^k \sqrt{t_\lambda^k} \qquad c\left(\mu^k\right) \leftarrow \frac{s}{t_\mu^k} + \frac{1}{2} h d t_\mu^k + g_\mu^k \sqrt{t_\mu^k}$

//use (16) to compute g_λ^k //use (16) to compute g_μ^k

10: $\left(z_a^k, z_b^k\right) \leftarrow \begin{cases} \left(\lambda^k, z_b^k\right) & \text{if } c\left(\lambda^k\right) > c\left(\mu^k\right) \\ \left(z_a^k, \mu^k\right) & \text{otherwise} \end{cases}$

11: **end if**

12: **end for**

13: **return** $z^\star$ and $t^\star$ //Using the equations in statement 1 & 2

Stop: EXIT algorithm

Algorithm 3 Find and Test KKT Points

1: **Initialize**

$\forall j \in \mathbb{N}$, $y_j \leftarrow t_j$

reindex $\mathbf{y}$ so that $y_1 \leq y_2 \leq y_n$

set $\mathbb{M} = \{m :$ the solution produced in iteration m is feasible$\} = \emptyset$

2: **Functions**

3: $c_m(t, \mathbf{y}) = \dfrac{\bar{a}_m}{t} + \bar{b}_m t + \displaystyle\sum_{j=m+1}^{n} \left(\frac{a_j}{y_j} + b_j y_j \right) + \sum_{j \in \mathbb{N}} \left(g_j \sqrt{y_j} \right)$

4: **for** $m = 1$ to n **do**

5:     $\bar{a}_m = A + a_1 + \cdots + a_m$

6:     $\bar{b}_m = b_1 + \cdots + b_m$

7: **end for**

8: Compute $\left(\bar{t}_1, \bar{z}_1\right)$ //Using Algorithm 1 and Algorithm 2

9: **if** $\bar{t}_1 \geq y_2$ **then**

10:     Constraint (11) in [1] is not satisfied; infeasible solution, not a KKT point.

11: **else**

12:     $t \leftarrow \bar{y}_1 \leftarrow \bar{t}_1$

13:     $\mathbb{M} \leftarrow \{1\}$

14:     $\mathbb{C} \leftarrow c_1(t, \mathbf{y})$ //use the function in statement 3

15: **end if**

16: **for** $m = 2$ to n **do**

17:     Create dummy item m_0 by duplicating item m and making one change:

18:         $a_m^0 = \bar{a}_m$

19:     Compute $\left(t_m^0, z_m^0\right)$ for the dummy item //Using Algorithm 1 and Algorithm 2

20:     Compute $\left(\bar{t}_m, \bar{z}_1, \ldots, \bar{z}_m\right)$ //Using Algorithm 2 and Algorithm 5

21:     **if** $\bar{t}_m < y_{m+1}$ **then**

22:         **for** $j = 1$ to m **do**

23:             $t \leftarrow \bar{y}_j \leftarrow \bar{t}_j$

24:         **end for**

25:         $\mathbb{M} \leftarrow \{m\}$

26:         $\mathbb{C} \leftarrow c_m(t, \mathbf{y})$ //use the function in statement 3

27:         //When $\bar{t}_m >= y_{m+1}$, Constraint (11) in [1] is not satisfied; infeasible solution, not a KKT point.

28:     **end if**

29: **end for**

30: **return** $m^\star = \operatorname*{argmin}_{m \in \mathbb{M}} \mathbb{C} = \{c_m(t, \mathbf{y}) : (t, \mathbf{y}) \text{ is a KKT point}\}$

Stop: EXIT algorithm

Algorithm 4 Multi-Family Roundoff Algorithm

Data
  $(t^\star, \mathbf{y}^\star)$
Initialize
  $\gamma \leftarrow t^\star$ //Begin Roundoff Procedure
for $(i, j) \in \mathbb{N} \times \mathbb{N}$ **do**
  Find t^{ij} and $\pi^{ij} \in \mathbb{Z}_+ \ni y_{ij} = t^{ij} 2^{\pi^{ij}}$ and $t^{ij} \in [\gamma, 2\gamma)$
end for
Add a subscript h to t^{ij} and $\pi^{ij} \ni t_h^{ij} \leq t_{h+1}^{ij}$
for $\phi = 1$ to n^2 **do**
  $\pi_\phi^{ij} \leftarrow \begin{cases} \pi_h^{ij} - 1 & \text{if } h \leq \phi \\ \pi_h^{ij} & \text{otherwise} \end{cases}$
  $k_\phi^{ij} \leftarrow 2^{\pi_{ij}^{\phi}}$
  for $j = 1$ to n **do**
    Compute z_{ij}^{ϕ} using Algorithm 1
  end for
  $t_\phi^\star = \underset{t_\phi}{\operatorname{argmin}} = c\left(t_\phi^\star, k_\phi\right)$
end for

Algorithm 5 Golden Section Search along t

Require:

$\left(t_a^0, t_b^0\right) \leftarrow \left(t_m^0, y_1\right)$ $\quad \delta^0 = t_b^0 - t_a^0$ $\quad \varepsilon \leftarrow 0.0001$

$\omega = 1$ $\quad \beta \leftarrow \frac{\sqrt{5}-1}{2}$ $\quad \alpha \leftarrow 1 - \beta$

Compute ω, the required number of iterations

while $0.618^{(\omega-1)} > \frac{\varepsilon}{\delta^0}$ **do**
  $\omega \leftarrow \omega + 1$
end while
$\left(t_a^1, t_b^1\right) \leftarrow \left(t_a^0, t_b^0\right)$
for $k = 1$ to ω **do**
  $\delta^k \leftarrow t_b^k - t_a^k$
  $\left(\lambda^k, \mu^k\right) \leftarrow \left(t_a^k + \delta^k \alpha, t_b^k + \delta^k \beta\right)$
  $\delta^k \leftarrow \mu^k - \lambda^k$
  if $\delta^k \leq \varepsilon$ **then**
    $\left(t_a^{k+1}, t_b^{k+1}\right) \leftarrow \left(\lambda^k, \mu^k\right)$
  else
    $c^{\lambda^k} \leftarrow \frac{\bar{a}_m}{\lambda} + \bar{b}_m \lambda$
    $c^{\mu^k} \leftarrow \frac{\bar{a}_m}{\mu} + \bar{b}_m \mu$
    for $j = 1$ to m **do**
      compute $z^f_{\lambda^k}$ using Algorithm 1 $\quad$ compute $z^f_{\mu^k}$ using Algorithm 1
      $c^{\lambda^k} \leftarrow c^{\lambda^k} + g_j^{\lambda^k} \sqrt{\lambda^k}$ $\quad$ $c^{\mu^k} \leftarrow c^{\mu^k} + g_j^{\mu^k} \sqrt{\mu^k}$
    end for
    $\left(t_a^{k+1}, t_b^{k+1}\right) \leftarrow \begin{cases} \left(\lambda^k, z_b^k\right) & \text{if } c_\lambda^k > c_\mu^k \\ \left(z_a^k, \mu^k\right) & \text{otherwise} \end{cases}$
  end if
end for
$t_m^\star = (t_a^{k+1} + t_b^{k+1})/2$
$c_m^\star = \frac{\bar{a}_m}{t_m^\star} + \bar{b}_m t_m^\star$
for $j = 1$ to m **do**
  compute $z^f_{t_m^\star}$ using Algorithm 1
  $c_m^\star \leftarrow c_m^\star + g_j^\star \sqrt{t_m^\star}$
end for
return $t_m^\star$ and $c_m^\star$

4 Example

Algorithm F is illustrated with an example. In this example, a facility produces five families, each with five items. The time unit was arbitrarily chosen to be one week. The data were generated from the uniformly distributed parameters in Table 1.

Table 1: Contributions of the Literature and this Paper

Parameter	**Distribution**
Family setup time (weeks)	$U(0.015, 0.025)$
Family setup cost (\$)	$U(500, 1000)$
Item setup time (weeks)	$U(0.0042, 0.0125)$
Item setup cost (\$)	$U(100, 500)$
Item holding cost (\$)	$U(0.10, 1.25)$
Item demand mean (units)	$U(1000, 2, 500)$
Item demand standard deviation (% of demand mean)	$U(0.60, 0.90)$
Item production rate (units/week)	$U(50, 000, 200, 000)$
Item fill rate (%)	$U(0.95, 0.9999)$

Example 1.

$$\mathbf{S} = \begin{bmatrix} 0.0243 & 0.0249 & 0.0210 & 0.0236 & 0.0182 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \$967 & \$643 & \$634 & \$712 & \$559 \end{bmatrix}$$

$$\mathbf{s} = \begin{bmatrix} 0.0110 & 0.0047 & 0.0106 & 0.0044 & 0.0121 \\ 0.0120 & 0.0086 & 0.0060 & 0.0118 & 0.0125 \\ 0.0109 & 0.0123 & 0.0078 & 0.0118 & 0.0118 \\ 0.0095 & 0.0056 & 0.0104 & 0.0117 & 0.0114 \\ 0.0118 & 0.0113 & 0.0104 & 0.0081 & 0.0045 \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} \$286 & \$389 & \$398 & \$477 & \$163 \\ \$368 & \$292 & \$229 & \$328 & \$302 \\ \$330 & \$475 & \$499 & \$340 & \$489 \\ \$180 & \$181 & \$296 & \$204 & \$294 \\ \$194 & \$306 & \$396 & \$171 & \$379 \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} \$0.17 & \$0.25 & \$0.39 & \$1.09 & \$1.22 \\ \$0.48 & \$0.54 & \$0.67 & \$0.51 & \$0.22 \\ \$1.19 & \$0.46 & \$0.25 & \$1.22 & \$0.15 \\ \$1.03 & \$0.16 & \$1.06 & \$0.52 & \$0.20 \\ \$0.71 & \$1.25 & \$0.54 & \$0.50 & \$0.53 \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} 1,084 & 1,629 & 1,999 & 1,248 & 1,267 \\ 1,140 & 1,708 & 2,190 & 2,342 & 2,383 \\ 1,471 & 2,180 & 1,541 & 1,251 & 1,362 \\ 1,156 & 1,049 & 1,141 & 1,711 & 2,317 \\ 1,655 & 2,392 & 2,213 & 1,378 & 1,689 \end{bmatrix}$$

$$\sigma = \begin{bmatrix} 912.94 & 1,128.57 & 1,384.11 & 1,057.80 & 1,021.33 \\ 746.81 & 1,430.45 & 1,869.60 & 1,796.31 & 2,143.03 \\ 915.11 & 1,613.20 & 1,273.69 & 870.07 & 895.79 \\ 845.27 & 695.17 & 1,017.89 & 1,402.85 & 1,762.08 \\ 1,091.80 & 1,783.18 & 1,791.56 & 954.95 & 1,023.16 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} 77,992 & 193,190 & 96,706 & 148,305 & 136,860 \\ 85,259 & 156,193 & 116,702 & 57,720 & 146,190 \\ 62,443 & 179,797 & 118,258 & 92,120 & 192,021 \\ 168,194 & 166,679 & 141,044 & 73,565 & 59,973 \\ 72,447 & 187,138 & 64,699 & 142,463 & 60,164 \end{bmatrix}$$

$$\mathbf{f} = \begin{bmatrix} 0.9507 & 0.9848 & 0.9796 & 0.9780 & 0.9611 \\ 0.9891 & 0.9663 & 0.9929 & 0.9990 & 0.9536 \\ 0.9901 & 0.9971 & 0.9590 & 0.9553 & 0.9514 \\ 0.9820 & 0.9917 & 0.9638 & 0.9846 & 0.9513 \\ 0.9954 & 0.9989 & 0.9612 & 0.9804 & 0.9885 \end{bmatrix}$$

Computation results are shown using two procedures (a) We first use algorithm F as given in §3, next (b) we apply the multi-family algorithm to the corresponding deterministic problem with demand means used as demand rates. Once the solution is obtained, we add safety stocks and compute their costs.

Table 2: Computation results of the Fill Rate Algorithm

Algorithm	(1)*	(2)**
Total average cost	TC=46, 904.71	49, 481.40
% Difference		5.49%
Total average Family setup cost	\$6, 435.15	\$3, 771.36
Total average Item setup cost	\$11, 003.05	\$7, 863.53
Total average working stock holding cost	\$8, 038.05	\$11, 634.89
Total average safety stock holding cost	\$21, 428.46	\$26, 211.62
Basic period length	$T* = 0.323$	0.932
Family multipliers	$\mathbf{K}^\star = \begin{bmatrix} 2 & 1 & 2 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Item multipliers	$\mathbf{k}^\star = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 \\ 4 & 2 & 2 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$

*(1) Multi-family algorithm
**(2) Deterministic multi-family with safety stocks computed at the end

Using algorithm F, which considers safety stock costs explicitly, resulted in cost savings of 5.49% over algorithm (2). This represents annual savings of \$133, 988.33 over algorithm (2). The average cost savings over 50 randomly generated problems with the distribution of parameters given above were 4.27% over algorithm (2). This represents annual dollar savings of $\$139,075.87$. The solution performance was, on average, 0.9997% above the lower bound, with a standard deviation of 0.5571% and a maximum of 2.2565%.

Algorithm F generated solutions whose basic period length was, on average, 62.16% smaller than that of algorithm (2), consistent with the trends in practice whereby reductions in working stock and safety stock levels and their associated costs drive lot sizes to be smaller and production to be more frequent.

5 Summary

In this technical note, we have extended the Multiple Family Economic Lot Scheduling Problem to explicitly include the consideration of safety stocks for a fill rate criterion. We exploited a number of properties of the problem to develop an efficient solution procedure. The form of our solution is a basic period cyclic schedule, with item multipliers restricted to integer-powers-of-two. We solved a representative set of sample problems using our procedure, and compared the results to a solution procedure that ignored the safety stock costs. Our results exhibited total costs typically 4.27% lower than the alternative procedure.

There are a number of directions for future research on the MFELSP-SS for fill rates. There is compelling reason to believe that, unlike the multi-family algorithm–which solves the MFELSP-SS for service level, algorithm F may very well produce solutions that favor longer basic periods over algorithm (2). This is because the safety factor is a decreasing function of the cycle time. Consequently, safety stock levels begin to decrease with increasing cycle time. Computational studies to investigate this and the impact of other parameters on the algorithm's performance would be useful.

Another useful study would be a computational study that compares the performance of the algorithm for the service level safety stocks and the one for fill rates.

References

[1] S. M. Karalli and A. D. Flowers. The multiple-family elsp with safety stocks. *Operations Research*, 54(3):523–531, 2006.

[2] R. G. Brown. *Decision rules for inventory management*. Holt, Rinehart and Winston New York, 1967.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. Wiley-Interscience, 2006.