A Hybrid Algorithm for Solving the Economic Lot and Delivery Scheduling Problem in the Common Cycle Case

Suquan Ju Jens Clausen

Informatics and Mathematical Modelling
Technical University of Denmark
2800 Kgs. Lyngby, Denmark

9th June 2004

Abstract

The ELDSP problem is a combined lot sizing and sequencing problem. A supplier produces and delivers components of different component types to a consumer in batches. The task is to determine the cycle time, i.e. that time between deliveries, which minimizes the total cost per time unit. This includes the determination of the production sequence of the component types within each cycle.

We investigate the computational behavior of two published algorithms, a heuristic and an optimal algorithm. With large number of component types, the optimal algorithm has long running times. We devise a hybrid algorithm, which is both optimal and efficient.

1 Introduction

The ELDSP problem is a combined lot sizing and sequencing problem. A supplier produces and delivers components of different component types to a consumer in batches. The amount of each component type delivered in a batch is equal to the demand of the consumer for that component type in the time period between deliveries. Holding costs are present, both at the supplier and the consumer side. Each delivery has a fixed cost, and there are both set-up times and costs.

As usual in lot sizing problem, one wants to find that time between deliveries, which minimizes the average cost per time unit, the cycle time. Since the set-up times and holding costs may vary between component types, finding the actual cost of a given cycle time involves also the determination of the production sequence for the component types. Since there are exponentially many sequences, this problem at first glance seems hard.

Hahm and Yano introduced a heuristic for the problem in 1995. The heuristic is remarkably accurate regarding solution quality. In 2003, Jensen and Khouja devised an algorithm for the ELDSP problem, which solves the problem to optimality. They also compared this against the H&Y heuristic, both with respect to quality and running time. However, their experiments were limited regarding to the number of possible component types.

In the current paper we perform a thorough computational investigation of the two algorithms. We have modified the problem generator used by both Hahm and Yano and Jensen and Khouja, since this for larger number of component types turned out to generate problems with trivial optimal solutions. The investigation shows that the running time of the optimal algorithms becomes excessive for large number of component types, and we therefore construct an algorithm, which is a hybrid between the two previous algorithms. The algorithm uses the H&Y heuristic as a preprocessor to the K&J algorithm, thereby maintaining the optimality guarantee and at the same time obtaining a running less than three times that of the H&Y heuristic.

The paper is organised as follows: In Section 2, we give the basic formulation of the problem and describe the two key components of all solution methods, namely procedures to find the optimal production sequence given the cycle time, and to find the optimal cycle time given the production sequence. Section 3 briefly introduces previous solution methods, and Section 4 describes our hybrid algorithm. Section 5, which is main contribution of the paper, describes the problem generator and gives the computational results from extensive testing of the algorithms. Section 6 concludes the paper.

2 The Problem

We consider the standard formulation of the ELDSP-problem as given in [6]. A consumer uses a set of components $\{1, \ldots, J\}$ all produced by the same supplier. The demand per time unit for each component D_i is known. The components are produced and delivered in batches. and there is a fixed delivery cost A per batch. For each specific component $i \in \{1, \ldots, J\}$ there is a fixed setup time s_i and a production time per unit, p_i. Furthermore, there is a setup cost, S_i. When the required number of units of component i has been produced, these have to be held in inventory unitl all components of the batch have been produced. The holding cost per unit for i is h_i per time unit. Likewise, the consumer has to pay an inventory cost for holding components during the period of production; again the holding cost per unit for i is h_i per time unit.

The ELDSP problem is now to decide the so-called *cycle time* denoted T. T is the time interval between deliveries and should be determined such that the overall costs of production, inventory (both at supplier and consumer), and shipping is as small as possible. The cost is expressed as *average cost per time unit* and denoted TC. Note that for a fixed T, the number of units of component i is calculated as D_iT, and the problem becomes that of finding the best sequence q among all possible sequences C of components (permutations of $\{1, \ldots, J\}$). This sequence depends only on the inventory costs at the supplier - the transportation cost and setup cost is a constant for each batch, and for fixed T, the inventory cost at the assembly facility is also a constant.

Figure 1 is an example of the inventory levels of three components both at the supplier and the assembly facility.

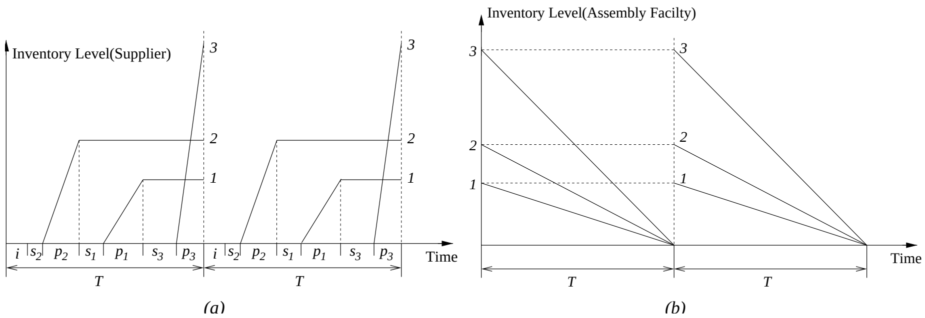

Figure 1: Inventory levels of three components in two production cycles: (a) supplier; (b) assembly facilty.

2.1 Mathematical Formulation of ELDSP

During the cycle time T, components necessary for the next delivery must be produced. The time needed to set up and produce all components for one delivery is $\sum_{i=1}^{J}(s_i + p_i D_i T)$, which must be less than or equal to the cycle time T. Therefore, to be feasible, T must satisfy

$$\sum_{i=1}^{J}(s_i + p_i D_i T) \leq T \Rightarrow T \geq \frac{\sum_{i=1}^{J} s_i}{1 - \sum_{i=1}^{J} p_i D_i} \equiv \tau_{min} \tag{1}$$

There is no production at the consumer (assembly facility). Thus all components needed in a period of length T must be delivered at the beginning of the period. On average the inventory level at the consumer for a component is one half of the delivery, and the inventory holding cost of component i is thus $\frac{1}{2}TD_i h_i$. The total inventory holding cost of all components per time unit at the assembly facility is therefore:

$$\sum_{i=1}^{J} \frac{1}{2} T D_i h_i = \frac{1}{2} T \sum_{i=1}^{J} D_i h_i$$

The average inventory holding cost for units of component i at the supplier depends upon the accumulation during the production time of component i itself and the elapsed time for setup and production of components produced after i but before delivery. The inventory holding cost per time unit at the supplier is therefore:

$$\begin{aligned}
&\sum_{i=1}^{J} D_{q[i]} h_{q[i]} \left\{ \frac{1}{2} T D_{q[i]} p_{q[i]} + \sum_{j=i+1}^{J} (T D_{q[j]} p_{q[j]} + s_{q[j]}) \right\} \\
&= \frac{1}{2} T \sum_{i=1}^{J} D_i^2 p_i h_i + \sum_{i=1}^{J} \left\{ D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} (T D_{q[j]} p_{q[j]} + s_{q[j]}) \right\} \\
&= \frac{1}{2} T \sum_{i=1}^{J} D_i^2 p_i h_i + Z_1(q) + Z_2(q) \cdot T
\end{aligned}$$

where

$$Z_1(q) = \sum_{i=1}^{J} D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} s_{q[j]}$$
$$Z_2(q) = \sum_{i=1}^{J} D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} D_{q[j]} p_{q[j]}$$

The average cost per time unit TC is the sum of production setup costs at the supplier, inventory holding costs at the supplier, inventory holding costs at the assembly facility, and transportation costs:

$$\begin{aligned} TC &= \frac{\sum_{i=1}^{J} S_i}{T} + \frac{1}{2} T \sum_{i=1}^{J} D_i^2 p_i h_i + Z_1(q) + Z_2(q) T + \frac{1}{2} T \sum_{i=1}^{J} D_i h_i + \frac{A}{T} \\ &= \frac{S+A}{T} + \alpha T + Z_1(q) + Z_2(q) T + \beta T \qquad (2) \end{aligned}$$

with $S = \sum_{i=1}^{J} S_i$, $\alpha = \frac{1}{2} \sum_{i=1}^{J} D_i h_i (1 - p_i D_i)$, and $\beta = \sum_{i=1}^{J} D_i^2 p_i h_i$.

TC must be minimized subject to the constraint:

$$T \geq \tau_{min}$$

2.2 The case of a fixed production sequence

Consider now the situation with a fixed production sequence, q. Differentiating TC as given by 2 with respect to T gives:

$$\frac{\partial TC}{\partial T} = -\frac{S+A}{T^2} + \alpha + \beta + Z_2(q)$$

Since $\frac{\partial^2 TC}{\partial T^2} = \frac{S+A}{2T^3}$ is positive for $T > 0$, TC is a convex function of T for fixed q. Denote the value of T which minimizes TC for sequence q by $T^*(q)$. Then $T^*(q)$ can be found at a unique stationary point provided this is feasible:

$$\frac{\partial TC}{\partial T} = 0 \Leftrightarrow T = \sqrt{\frac{S+A}{\alpha + \beta + Z_2(q)}} \equiv T^*(q) \qquad (3)$$

Define $\tau_{max} = \sqrt{\frac{S+A}{\alpha+\beta}}$. Note that $\alpha and \beta$ do not depend in the particular sequence q in question, and that $Z_2(q) > 0$ for any q. Therefore, for any q the optimal cycle time $T^*(q)$ must satisfy $T^*(q) < \tau_{max}$ if it is found at a stationary point of TC. Any feasible cycle time satisfies $T \geq \tau_{min}$. Therefore, if $\tau_{min} > \tau_{max}$, the optimal production cycle time is τ_{min}.

2.3 The case of a fixed cycle time

We now consider the problem of finding the best production sequence for a given cycle time T. The only sequence-dependent parts of the objective function TC (2) is $Z_1(q) + Z_2(q)T$. In order to minimize TC, we must therefore minimize:

$$\begin{aligned} & Z_1(q) + Z_2(q)T \\ = & \sum_{i=1}^{J} D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} s_{q[j]} + T \sum_{i=1}^{J} D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} D_{q[j]} p_{q[j]} \\ = & \sum_{i=1}^{J} \left\{ D_{q[i]} h_{q[i]} \sum_{j=i+1}^{J} (T D_{q[j]} p_{q[j]} + s_{q[j]}) \right\} \end{aligned}$$

According to theorem 2.4 of [1], the value is minimized when the components are arranged in non-increasing order of $\frac{TD_{q[i]}p_{q[i]}+s_{q[i]}}{D_{q[i]}h_{q[i]}}$. We therefore determine the optimal solution by finding a processing sequence satisfying:

$$\frac{TD_{q[1]}p_{q[1]} + s_{q[1]}}{D_{q[1]}h_{q[1]}} \geq \frac{TD_{q[2]}p_{q[2]} + s_{q[2]}}{D_{q[2]}h_{q[2]}} \geq \ldots \geq \frac{TD_{q[J]}p_{q[J]} + s_{q[J]}}{D_{q[J]}h_{q[J]}} \tag{4}$$

3 Previous Solution Methods

Solving the ELDSP-problem requres the simultaneous identification of the optimal cycle length and the corresponding production sequence. In the previous sections, the optimal cycle time for a given production sequence q and the optimal production sequence for a given cycle time were determined.

3.1 Hahm and Yano's heuristic algorithm

The heuristic of Hahm and Yano [6] iterates between finding the optimal value of T for a given q, and finding the optimal sequence given T using the methods previously described. The algorithm starts with $T = \tau_{max}$, and the termination criterion is that the optimal sequence does not change or that the optimal cycle time equals τ_{min}.

In Figure 2, the two possible results are illustrated: Either the optimal solution is identified or the algorithm terminates in a local optimum.

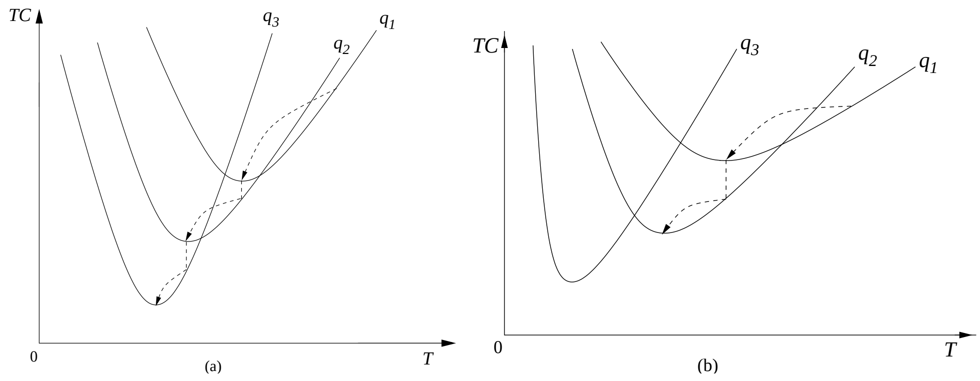

Figure 2: Performance of Hahm and Yano's heuristic algorithm: (a) terminated at a global optimal solution. (b) terminated at a local optimal solution.

Hahm and Yano test their algorithm on 72 randomly generated problems, cf. [6]. The algorithm is able to find the global optimal solution in all of these. Nevertheless, there exist examples, where only a locally optimal solution is found.

3.2 Jensen and Khouja's polynomial time algorithm

In [14], Jensen and Khouja developed a polynomial time algorithm for the ELDSP-problem. The J&K algorithm divide the range of feasible cycle times into a number of intervals such that the optimal sequence in each interval is unique. The optimal sequence in each interval can be found by (4). The optimal values for TC from each interval are compared, and the minimum of these is then the optimal value of TC with corresponding optimal cycle time and optimal production sequence. For a small number of components, the running time of the algorithm is short. For a large number of components, the increase observed in the running time suggests the search for a more efficient algorithm. Running times for the J&K are reported in Table 3 and Table 4.

4 The Hybrid Method

Since H&Y's algorithm does not always generate the optimal solutions and the running time of J&K's algorithm is large for a large number of components, an obvious idea is to combine these two methods into a hybrid algorithm. First, H&Y's algorithm is used twice. One solution (q_l, T_l, TC_l) is found by setting the starting point at τ_{min} and working with increasing values of T, and another (q_r, T_r, TC_r) is found from the starting point τ_{max} working with decreasing values of T. If these two solutions are equal, i.e. $T_l = T_r$, then the optimal solution is found, cf [6]. Otherwise, the intervals between T_l and T_r corresponding to unique production sequences are calculated. If the the smallest of the optimal solution values for these intervals is less than both TC_l and TC_r, the corresponding optimal cycle time and production sequence is chosen as the optimal solution, otherwise, the smaller value of TC_l and TC_r is the optimal solution.

Figure 3 illustrates the behavior of the hybrid method. The search range for T covers $[\tau_{min}, \tau_{max}]$, and thus the global optimal solution is found.

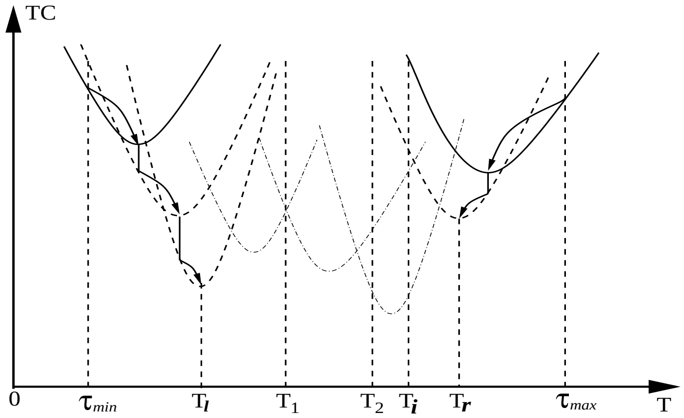

Figure 3: The behavior of the hybrid method.

5 Test Results

H&Y's heuristic algorithm and J&K's polynomial time algorithm have previously been tested only for number of components up to 9. In this section, the algorithms are run for larger number of components and compared both with respect to solution time and solution quality.

5.1 The Test Problem Generator

For an ELDSP-problem satisfying $\tau_{min} \geq \tau_{max}$, the optimal sequence is uniquely determined by τ_{min} and both the H&Y and the J&K algorithm stops at the very beginning before iterating. In order to gain insight and compare the running times of the two algorithms on a more fair basis, the problems generated should satisfy $\tau_{min} < \tau_{max}$ to make each algorithm complete also the iterative part. Hahm and Yano can not generate problems which make their algorithm fail cf. [6]. We have

Step 1: Calculate τ_{min} and τ_{max}.

Step 2: If $\tau_{min} \geq \tau_{max}$, then τ_{min} is optimal,
$T_{opt} = \tau_{min}$, find q_{opt} by Eq.(4),
$TC_{opt} = TC^*(q_{opt}, \tau_{min})$,
go to **Step 5**.

Step 3: $T_l = \tau_{min}$, $T_r = \tau_{max}$.
$stop = 0$.
do {
  for a given T_r, calculate q_r (Eq.(4)), $T^*(q_r)$ (Eq.(3)),
  if $T^*(q_r) \neq T_r \wedge T^*(q_r) \geq \tau_{min}$
    $T_r = T^*(q_r), TC_r = TC^*(q_r, T^*(q_r))$
  else if $T^*(q_r) \neq T_r \wedge T^*(q_r) < \tau_{min}$
    $T_{opt} = \tau_{min}$, find q_{opt} by Eq.(4)
    $TC_{opt} = TC^*(q_{opt}, \tau_{min})$
    go to **Step 5**.
  else
    stop = 1;
} while (stop = 0);

$stop = 0$.
do {
  for a given T_l, calculate q_l (Eq.(4)), $T^*(q_l)$ (Eq.(3)),
  if $T^*(q_l) \neq T_l \wedge T^*(q_l) \geq \tau_{min}$
    $T_l = T^*(q_l), TC_l = TC^*(q_l, T^*(q_l))$
  else if $T^*(q_l) \neq T_l \wedge T^*(q_l) < \tau_{min}$
    $T_l = \tau_{min}$, find q_l by Eq.(4)
    $TC_l = TC^*(q_l, \tau_{min}), stop = 1$.
  else
    stop = 1;
} while (stop = 0);

Step 4: If ($TC_l = TC_r$)
  $q_{opt} = q_l, T_{opt} = T_l, TC_{opt} = TC_l$
else
  for each pair of components i and j, $i \neq j$ solve $\frac{TD_i p_i + s_i}{D_i h_i} = \frac{TD_j p_j + s_j}{D_j h_j}$.
  Store the values within $[T_l, T_r]$ into $M = [T_l, T_1, T_2, \ldots, T_i, \ldots, T_r]$.
  Sort M in increasing order. m = size(M).
  $TC_{best} = \infty$
  for ($i = 1; i < m; i = i + 1$){
    for $T_m = \frac{1}{2}(M_i + M_{i+1})$, calculate $q_m, T^*(q_m), TC(q_m, T^*(q_m))$
    if ($T^*(q_m) \in [M_i, M_{i+1}] \wedge TC(q_m, T^*(q_m)) < TC_{best}$)
      $q_{best} = q_m, T_{best} = T^*(q_m), TC_{best} = TC(q_m, T^*(q_m))$
  }
  if $TC_{best} < min(TC_l, TC_r)$
    $q_{opt} = q_{best}, T_{opt} = T_{best}, TC_{opt} = TC_{best}$
  else
    if ($TC_l > TC_r$)
      $q_{opt} = q_r, T_{opt} = T_r, TC_{opt} = TC_r$,
    else
      $q_{opt} = q_l, T_{opt} = T_l, TC_{opt} = TC_l$,

Step 5: Return $q_{opt}, T_{opt}, TC_{opt}$.

Table 1: The hybrid algorithm.

first used the same generation rules as [14] to generate 100,000 problems. See Table 2.

J	Number of problems (J&K's generator)	Number of problems satisfying $\tau_{min} < \tau_{max}$	Number of problems (New generator)	Number of problems satisfying $\tau_{min} < \tau_{max}$
2	100000	98722	100000	99454
3	100000	98084	100000	99737
5	100000	91909	100000	99933
7	100000	73382	100000	99976
10	100000	30809	100000	99997
20	100000	33	100000	100000
30	100000	0	100000	100000
50	100000	0	100000	100000
100	100000	0	100000	100000

Table 2: Problems compared using different generation rules.

The left part of Table 2 shows that J&K's generator can not generate problems satisfying $\tau_{min} < \tau_{max}$ especially if the number of components exceeds 10. From the expressions $\tau_{min} = \frac{\sum_{i=1}^{J} s_i}{1-\sum_{i=1}^{J} p_i D_i}$ and $\tau_{max} = \sqrt{\frac{\sum_{i=1}^{J} S_i + A}{\alpha+\beta}}$ it is clear that both values increase with increasing number of components. However, the rate of increase for τ_{min} is expected to be much larger than the rate for τ_{max} if the drawing ranges for the problem parameters are kept constant. Thus, situations with $\tau_{min} > \tau_{max}$ will occur more frequently for larger number of components. This deficit in the generator used by J&K can be resolved by decreasing s_i and increasing S_i respectively while drawing for larger number of components. In our generator, S_i is drawn from a uniform distribution $U[0, \sqrt{J}]$ rahter than from $U[0, 1]$, and s_i and is drawn from a uniform distribution $U[0, \frac{0.25}{\sqrt{J}}]$ rather than from $U[0, 0.25]$. Otherwise, the features of the problem generator are kept unchanged. Once again 100,000 problems are generated now using the new generation rules. The last column of Table 2 shows that the new generation rules perform well especially for large number of components. In the following, all problems are hence generated using the new generation rules.

5.2 Efficiency of the algorithms

Table 3 shows the results from the solution of 100,000 problems all generated using the new generation rules with number of components ranging from 2 up to 100, and maintaining the condition $\tau_{min} < \tau_{max}$. The running time for each algorithm and each component number is given as the total running time in seconds of all problems with that number of components. When the number of components is small, the differences in running times are small. However, when the number of components increases, the differences becomes more and more profound. For the 100 components case, the total running time of J&K's algorithm is more than 20 times those of the other two algorithms.

The average number of intervals investigated in J&K's algorithm increases quickly when the number of components increase, while the average number of intervals calculated when $T_l \neq T_r$ in the hybrid algorithm is very stable. This accounts for the steep increase in the running time of J&K algorithm.

J	No. of problems with $\tau_{min} < \tau_{max}$	Running time of H&Y's algorithm	J&K's algorithm		Hybrid algorithm	
			Running time	Average no. of intervals	Running time	Average no. of intervals when $T_l \neq T_r$
2	100000	0.137	0.148	1.119	0.309	2.000
3	100000	0.227	0.260	1.424	0.479	2.015
5	100000	0.511	0.705	2.282	1.231	2.004
7	100000	0.962	1.833	3.291	2.136	2.033
10	100000	1.725	6.078	5.064	4.685	2.027
20	100000	6.978	43.514	12.740	16.447	2.027
30	100000	14.533	160.275	23.058	37.697	2.047
50	100000	37.679	868.8	50.515	98.827	2.034
100	100000	125.603	8091.5	152.677	319.429	2.042

Table 3: Running times in seconds for different algorithms.

In Table 4, the running time of each algorithm is compared for component numbers larger than 100. Due to time limitations only one problem is generated and solved for each problem size. Nevertheless, it is evident that J&K's algorithm is not efficient for larger number of components. The running time of the hybrid algorithm is two to three times larger than that of H&Y's heuristic.

5.3 Making the H&Y heuristic fail

Table 5 displays the number of failures of H&Y's algorithm. In the vast majority of the 100,000 problems generated, H&Y's algorithm finds the globally optimal solution. The fourth column in Table 5 is the number of problems passed on to **Step 4** of the hybrid algorithm. The last two columns indicate that H&Y's algorithm fails when T_l is found optimal or a new sequence is found optimal between $[T_l, T_r]$.

J	No. of problems with $\tau_{min} < \tau_{max}$	Running time of H&Y's algorithm	J&K's algorithm		Hybrid algorithm	
			Running time	No. of intervals	Running time	No. of intervals when $T_l \neq T_r$
100	1	0.001	0.054	104	0.003	0
200	1	0.005	0.852	425	0.013	0
500	1	0.031	27.645	1580	0.081	0
1000	1	0.123	382.451	6257	0.402	0
2000	1	0.638	1.14 hs	14720	2.395	0
5000	1	6.562	12.4 hs	26833	13.877	0
10000	1	28.630	$>$ 12.4 hs	$>$26833	72.373	0

Table 4: Running times for number of components larger than 100.

Moreover, the cases when a new sequence between $[T_l, T_r]$ gives rise to the global optimum are rare.

J	No. of problems satisfying $\tau_{min} < \tau_{max}$	Hybrid algorithm: No. of problem satisfying				
		H&Y's algorithm fails	$T_l \neq T_r$	T_r is optimal	T_l is optimal	A new q is found optimal
2	100000	21	30	9	21	0
3	100000	66	127	61	66	0
5	100000	203	414	211	203	0
7	100000	266	563	297	264	2
10	100000	359	721	362	358	1
20	100000	331	697	366	327	4
30	100000	308	614	306	305	3
50	100000	233	442	209	232	1
100	100000	163	313	150	163	0

Table 5: Statistical results of the hybrid algorithm.

5.4 An example

Table 6 displays the parameter values of a 10 component ELDSP problem. The problem is solved once with each of the three algorithms described above. The results are shown in Table 7. J&K's algorithm and the hybrid algorithm find the globally optimal solution, while H&Y's algorithm is trapped in a local optimum.

Comparing the objective function values of TCs as given in Table 7, the difference between the global optimum and the local optimum is only 0.09%.

i	S_i	s_i	p_i	h_i	D_i
1	1.86048	0.0627592	0.169927	0.575915	0.0550249
2	1.28404	0.325193	0.0496841	0.955138	0.899564
3	0.588602	0.00670489	0.0496841	0.78338	0.0011597
4	1.80885	0.0705763	0.0901517	0.95291	0.484787
5	0.197166	0.0645518	0.753441	0.00778222	0.00747703
6	2.49116	0.0197262	0.158788	0.962798	0.532823
7	0.518634	0.0185681	0.328471	0.294168	0.234291
8	0.372038	0.00069727	0.103854	0.596454	0.753502
9	2.66912	0.00392305	0.145878	0.139225	0.236702
10	2.34833	0.0776865	0.0960417	0.28254	0.880032
$A =$ 1.97607					

Table 6: Parameter values of 10 components.

This example also illustrates the case for the hybrid algorithm, where a new circuit time between $[T_l, T_r]$ gives rise to the optimal solution. In Figure 4, curve l is the final sequence found from the side of lower bound ($\tau_{min} = 1.20959$). Curve r is the final sequence found from the side of upper bound ($\tau_{max} = 3.37365$). Obviously, the objective function values from curve l and curve r are not equal, so the hybrid algorithm investigates the intervals between T_l and T_r. In this case, three intervals are found, and a new sequence m is found optimal in one of these.

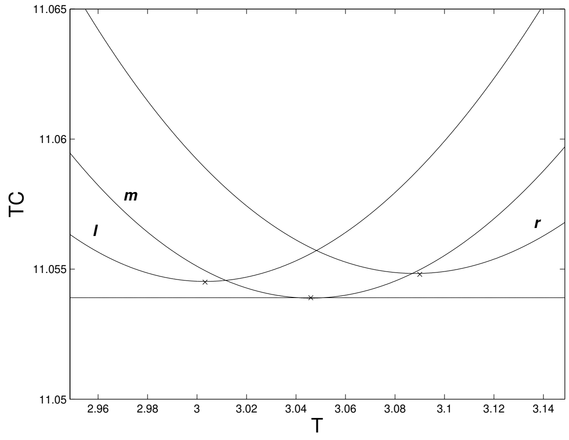

Figure 4: An example of 10 components.

	T	TC	q
H&Y's algorithm	3.08817	11.0548	5 3 7 9 1 10 6 8 2 4
J&K's algorithm	3.04597	11.0539	5 3 7 9 1 10 6 2 8 4
The hybrid algorithm	3.04597	11.0539	5 3 7 9 1 10 6 2 8 4

Table 7: Results of 10-component example.

6 Conclusion

In this paper we have reported a thorough computational investigation of two published solution methods for the ELDSP problem, a heuristic and an optimal algorithm, and a third hybrid method. The hybrid method has running times only a factor three larger than the heuristic, but is still an optimal algorithm. An immediate topic for further research is the generalisation of the methods described to more general cases of the ELDSP problem.

References

[1] Baker, Kenneth R. (1974) *Introduction to Sequencing and Scheduling*, John Wiley & Sons, Inc. New York. pp. 20-22.

[2] Dobson, G. (1987) The economic lot scheduling problem: achieving feasibility using time-varying lot sizes. *Operations Research*, **35**(5), 764-771.

[3] Gallego, G. and Shaw, D.X. (1997) Complexity of the ELSP with general cyclic schedules *IIE Transactions*, **29**(2), 109-113.

[4] Garey, M.R. and Johnson, D.S. (1979) *Computers and intractability. A guide to the theory of NP-completeness*, W H Freeman & Co., San Francisco.

[5] Hahm, J. and Yano, C.A. (1992) The economic lot and delivery scheduling problem: the single item case. *Internatinal Journal of Production Economics*, **28**, 235-252.

[6] Hahm, J. and Yano, C.A. (1995) The economic lot and delivery scheduling problem: the common cycle case. *IIE Transactions*, **27**, 113-125.

[7] Hahm, J. and Yano, C.A. (1995) The economic lot and delivery scheduling problem: models for nested schedules. *IIE Transactions*, **27**, 126-139.

[8] Hahm, J. and Yano, C.A. (1995) The economic lot and delivery scheduling problem: powers of two policies. *Transportation Science*, **29**(3), 222-241.

[9] Hall, Randolph W. (1996) On the integration of production and distribution: economic order and production quantity implications. *Transportation Research Part B: Methodological*, **30**(5), 387-403.

[10] Hanssmann, F. (1962) *Operations Research in Production and Inventory Control*, John Wiley & Sons, New York, pp. 158-160.

[11] Khouja, M. (2000) The economic lot and delivery scheduling problem: common cycle, rework, and variable production rate. *IIE Transactions*, **32**(8), 715-725.

[12] Khouja, M. (1997) The scheduling of economic lot sizes on volume flexible production systems. *International Journal of Production Economics*, **48**(1), 73-86.

[13] Khoury, B.N., Abboud, N.E. and Tannous, M.M. (2001) The common cycle approach to the ELSP problem with insufficient capacity. *International Journal of Production Economics*, **73**(2), 189-199.

[14] Mikkel T. Jensen and Moutaz Khouja (2003) An optimal polynomial time algorithm for the common cycle economic lot and delivery scheduling problem. *European Journal of Operational Research.*

[15] Wolsey, L.A. (1998) *Integer Programming*, John Wiley & Sons, Inc.

[16] Yu, Gang (1997) Robust economic order quantity models. *European Journal of Operational Research*, **100**(3), 482-493.