IIE Transactions

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uiie20

Complexity of the ELSP with general cyclic schedules

GUILLERMO GALLEGO [a] & DONG XIAO SHAW [b]

[a] Department of Industrial Engineering and Operations Research , Columbia University , New York, NY, 10027, USA

[b] School of Industrial Engineering, Purdue University , West Lafayette, IN, 47907, USA

Published online: 13 Jul 2010.

IIE Transactions (1997) **29**, 109–113

Complexity of the ELSP with general cyclic schedules

GUILLERMO GALLEGO[1] and DONG XIAO SHAW[2]

[1]*Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA*
[2]*School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA*

We show that the Economic Lot Schedule Problem (ELSP) is NP-hard in the strong sense under General Cyclic Schedules (GSC), Zero-Inventory Cyclic Schedules (ZICS), Time-Invariant Cyclic Schedules (TICS), Lot-Invariant Cyclic Schedules (LICS), and Basic Period Cyclic Schedules (BPCS).

1. Introduction

We show that the Economic Lot Schedule Problem (ELSP) is NP-hard in the strong sense under General Cyclic Schedules (GSC), Zero-Inventory Cyclic Schedules (ZICS), Time-Invariant Cyclic Schedules (TICS), Lot-Invariant Cyclic Schedules (LICS), and Basic Period Cyclic Schedules (BPCSs). The last result strengthens Hsu's [1] weak NP-complete result for BPCS.

The ELSP is that of finding a multi-item single-machine production schedule to minimize the long-run average cost of holding inventories and performing setups. The problem arises when economies of scale dictate the use of a single-machine to manufacture a set of items instead of a dedicated machine for each item in the set.

In the ELSP literature it is typically assumed that the rates of demand and production are constant, but item-dependent, that setup costs and setup times are item-dependent but sequence-independent, and that the inventory holding costs are charged at item-dependent linear time-weighted rates. What makes the ELSP difficult is that production of the items must be synchronized to avoid scheduling two items to use the facility at the same time. This is known as the synchronization constraint.

Research on the ELSP has focused on cyclic schedules, that is on schedules that repeat after a finite time interval. The smallest such interval, say T, is called the schedule *cycle length*. The reason researchers confine themselves to the realm of cyclic schedules is that non-cyclic schedules are very difficult to describe, let alone evaluate. See Schweitzer and Silver [2] for a description of the mathematical pitfalls of the ELSP.

The class of Common Cycle Cyclic Schedules (CCCSs) restricts all the item's cycle times to be equal [3]. CCCSs always satisfy the synchronization constraint, so it is very easy to find optimal CCCS. Although the CCCS approach works well if the items' independent cycle times are not too different [4], CCCS can perform abysmally when an item with high setup and low holding cost is produced as frequently as an item with the opposite characteristics. Even when setup costs and holding costs are identical across the items, it is often economical to produce high-demand items more frequently.

The class of BPCSs, popular for nearly three decades, allows different items to have different cycle times. However, it constrains the cyclic schedules to satisfy the following three properties: (i) Zero-Inventory (ZI) ordering, (ii) Time-Invariant (TI) ordering and (iii) Lot-Invariant (LI) ordering. The ZI property, also known as the Zero-Switch Rule, states that production for an item should not start until its inventory reaches zero. The TI property states that there is a constant time lag between any two consecutive production runs for an item. The LI property states that each item is produced in lots of constant size.

It is easy to see that if we impose the ZI and the TI properties together then the LI property holds. By the same token, if we impose the ZI and the LI properties together then the TI property holds. BPCS are often described somewhat differently, but the above definition is equivalent and more useful for our purposes. The classical definition states that the items are allowed to have different cycle times, but restricts them to be an integer multiple of a *basic period* long enough to accommodate the production of all the items. Moreover, all the lots of an item are required to be of the same size and the zero-inventory rule is imposed. See Doll and Whybark [5] for one of the best heuristics in this class and Elmaghraby [6] for a review of the literature and some interesting extensions. The drawback of BPCS is that it is difficult to find an executable timetable (i.e., starting and finishing times for the lots that satisfy the synchronization constraint) for BPCS for a given set of multiples of

0740-817X

the basic period. This is mainly because too many restrictions are imposed on the class of BPCSs. In fact, Hsu [1] showed that the problem of finding an executable BPCS based on given multipliers is NP-complete in the weak sense.

Dobson [7] considered a larger set of cyclic schedules by requiring only the ZI property. The cyclic schedules considered by Dobson are sometimes called time-varying, but they could as well have been called lot-varying without fully describing them. For that reason we prefer to call them simply Zero Inventory Cyclic Schedules (ZICS). Extending the ideas of Maxwell [8] and Delporte and Thomas [9], Dobson showed that given any production sequence (i.e., the order in which the items are produced in the cycle) it is always possible (given enough capacity) to find an executable timetable based on the production sequence. The timetable is obtained by quadratic programming.

Hodgson and Nuttle [10] considered the class of cyclic schedules where only the LI property is retained. We call them Lot Invariant Cyclic Schedules (LICS). Given a sequence, he shows that an executable timetable can be found by linear programming. It is also possible to consider the class of schedules where only the TI property is preserved. We call the latter the Time-Invariant Cyclic Schedule (TICS). Again, feasible timetables based on a sequence can be found by linear programming.

General cyclic schedules (GCSs) do not impose any of the above properties other than the existence of a finite cycle length T. To our knowledge no one has studied GCSs.

The heuristics for ZICS, LICS, and TICS avoid the difficulty of obtaining executable schedules associated with the BPCS. This has led to a distinction between those authors that constrain themselves to BPCS and those that do not because it seems to change the very essence of the problem, e.g., whether it is NP-complete to find an executable timetable based on production frequencies.

It is known that a problem restricted to a given set of solutions can be NP-complete whereas the same problem over a larger set of solutions may be polynomially solvable. For instance, the Asymmetric Traveling-Salesmen Problem is a restricted version of the Assignment problem, yet the former is NP-complete whereas the latter is polynomially solvable.

The reasons stated in the previous two paragraphs lead us to wonder whether the ELSP remains difficult over a larger class than the BPCS, for instance over the ZICS, LICS, TICS or GCS. In this paper we answer this question in the affirmative by showing that the ELSP is NP-hard in the strong sense under the sets ZICS, LICS, TICS and GCS. At the same time we strengthen Hsu's result and establish that the ELSP is strongly NP-complete over the class of BPCS. This negative result means that, unless P = NP, it is not possible to find a pseudo-polynomial algorithm to solve the ELSP under any of the classes discussed above, and research efforts should therefore be aimed at finding better heuristics.

2. NP-hardness results

Let i be the item index and let m be the number of items: We assume without loss of generality that all of the demand rates are 1. Let

p_i = production rate of item i;
h_i = holding cost of each unit of item i for a unit of time;
s_i = setup time of item i;
K_i = set up cost of item i.

Then, $\kappa = 1 - \sum_{i=1}^{m} 1/p_i$ is the proportion of time available for set up which we assume to be greater than zero.

Let n be the number of production runs in the cycle. Because each product must be produced at least once in the cycle we must have $n \geq m$. The class of ZI cyclic schedules can be characterized by three n-dimensional vectors $f = (f^j)_{j=1}^n$, $t = (t^j)_{j=1}^n$ and $u = (u^j)_{j=1}^n$, where f is the production sequence, $t \geq 0$ is the vector of production run times and $u \geq 0$ is the vector of idle times. Each production run time is just long enough to ensure that the ZI property holds. Each item must have at least one position in the production sequence, so $n \geq m$. We adopt Dobson's notation [7], and use the subscript i to refer to the items and the superscript j to refer to the positions in the schedule's sequence, for instance $p^j = p_{f^j}$. Dobson [7] shows that an executable ZI schedule exists if and only if $\kappa > 0$.

In this paper we consider GCSs so we relax the ZI property. When a production run of item f^j at position j begins, there may be some *leftover inventory* $w^j \geq 0$ of item f^j. To be consistent, we express this inventory in time units, that is, we define $v^j = w^j/(p^j - 1) \geq 0$, which represents the time required to accumulate w^j units of item f^j. Recall that the normalized production rate of f^j is p^j and that its demand rate is 1. We call v^j the *accumulated inventory time* at position j. Then we can characterize GCS by four n-dimension vectors:

$$f = (f^j)_{j=1}^n, \; t = (t^j)_{j=1}^n, \; u = (u^j)_{j=1}^n, \text{ and } v = (v^j)_{j=1}^n.$$

A cycle starts with the set-up time of s^1 of item f^1, followed by a production run of length t^1 and by an idle period of length u^1. Just after the setup but before the production run starts, the accumulated inventory time of item f^1 is v^1. The schedule proceeds with the second item in the sequence in the same fashion until the sequence is exhausted. At that point the cycle is completed. The inventory of each item at the end of the cycle should be identical with its initial inventory. For a schedule to be executable each production run of an item should be long enough to satisfy demand until the next production run of that item. It is easy to prove that an executable GCS schedule exists if and only if $\kappa > 0$.

Let $c(f, t, u, v)$ denote the average cost per unit time of an executable GCS characterized by (f, t, u, v); then

$$c(f,t,u,v) = \frac{1}{T}\sum_{j=1}^{n}\left(H^j(p^j)^2((t^j+v^j)^2-(v^j)^2)+K^j\right), \tag{1}$$

where $H^j = \frac{1}{2}h^j(1-1/p^j)$ and $T = \sum_{j=1}^{n}(t^j+s^j+u^j)$. Let $T^j \equiv t^j + (p^j - 1)t^j = p^j t^j$ denote the jth-run cycle length. Because demand equals supply over a cycle, we have

$$T = \sum_{f^j=i} T^j, \quad i = 1,2,\ldots,m. \tag{2}$$

The feasibility version of the ELSP over the class of GCS is: given an instance of the ELSP and an integer c^*, do there exist vectors f, t, u, and v such that $c(f, t, u, v) \le c^*$?

If the number $n \ge m$ of production runs in the cycle is given as input data, then the feasibility version of the ELSP belongs to the class NP of decision problems that can be solved in polynomial time by a nondeterministic algorithm. However, if as above, we do not specify n as part of the input data, then n can be exponentially large and the feasibility problem no longer belongs to the class NP. Any decision problem, whether in NP or not, to which we can transform a binary (unary) NP-complete problem will have the property that it cannot be solved by a polynomial (pseudo-polynomial) algorithm unless P = NP. The term NP-hard is used to distinguish difficult problems that do not belong to the class NP [11].

To prove that the ELSP over the class GCS is NP-hard in the strong sense will need the three-partition problem.

Three-Partition: Given integers $a_1, a_2, \ldots, a_{3m}$ and b such that $\frac{1}{4}b < a_i < \frac{1}{2}b$ and $\sum_{i=1}^{3m} a_i = mb$. Does there exist a partition $B_1, B_2, \ldots, B_m$ of the set $\{1, 2, \ldots, 3m\}$ such that

$$\sum_{i\in B_k} a_i = b, \quad k = 1,2,\ldots,m?$$

The three-partition problem is well-known to be NP-complete in the strong sense; that is, even when the input length is measured by $\sum_{i=1}^{3m} a_i$; see Garey and Johnson [12] for the details.

Theorem. *The ELSP under GCS is strongly NP-hard.*

Proof: Given a three-partition problem, consider the following instance of ELSP with $3m + 1$ items, and

$$c^* = 1,$$

$$p_0 = m, \quad s_0 = \frac{1}{m^2}, \quad K_0 = \frac{1}{m^2}, \quad H_0 = 1,$$

$$s_i = \frac{a_i}{2mb}\left(1-\frac{2}{m}\right), \quad p_i = s_i^{-1}, \quad K_i = H_i = s_i,$$
$$i = 1,2,\ldots,3m. \tag{3}$$

All the data of this instance are rational and can be scaled to integer by the multiple $2m^2b$; the summations of all the scaled integral data are bounded by the polynomial of $\sum_{i=1}^{3m} a_i$.

We first assume that the three-partition problem is feasible, and show that there exists a (f, t, u, v) such that $c(f, t, u, v) \le c^*$. Let α_k, β_k and γ_k denote the elements of B_k, where $B_k, k = 1, 2, \ldots, m$, is a feasible solution of the three-partition problem. Now construct the vectors f, t, s, u, and v as follows:

$$(f^j)_{j=1}^{4m} = (0,\alpha_1,\beta_1,\gamma_1,0,\alpha_2,\beta_2,\gamma_2,\ldots,0,\alpha_m,\beta_m,\gamma_m), \tag{4}$$
$$t^j = s^j, \quad u^j = 0, \quad \text{and} \quad v^j = 0, \qquad j = 1,2,\ldots,4m.$$

Within a cycle, item $i(\neq 0)$ is produced only once, whereas item 0 is produced m times. So the vectors f, s, t, u, and v define an executable schedule.

By the definition of t, s and u the cycle length is

$$T = \sum_{j=1}^{4m}(s^j+t^j) = m(s_0+t_0) + \sum_{i=1}^{3m}(s_i+t_i) = m\frac{1}{m} = 1.$$

Since $p_0 s_0 = 1/m$ and $p_i s_i = 1$, $i \neq 0$, it follows from (1) and $T = 1$ that

$$\begin{aligned} c(f,t,u,v) &= \frac{1}{T}\sum_{j=1}^{4m}\left(H^j(p^jt^j)^2+K^j\right) \\ &= \sum_{i=0}^{3m}\sum_{f_j=i}\left(H_i(p^jt^j)^2+K_i\right) \\ &= m\left(H_0(p_0s_0)^2+K_0\right)+\sum_{i\neq 0}(H_i+K_i) \\ &= \frac{2}{m}+\sum_{k=1}^{m}\sum_{i\in B_k}\frac{2a_i}{2mb}\left(1-\frac{2}{m}\right) = 1 = c^*, \end{aligned}$$

so the schedule is feasible with respect to c^*.

To establish the converse part, we assume that there is a feasible schedule with average cost at most c^* and use the well-known fact that, for any integer p and real positive number T,

$$\min_y\left\{\sum_{k=1}^{p}(y_k)^2, \text{ subject to} \sum_{k=1}^{p} y_k = T\right\} = \frac{T^2}{p}, \tag{5}$$

at $y_k = T/p$, $k = 1, 2, \ldots, p$.

Combining (2) and (5) we obtain

$$\sum_{f^j=i}(T^j)^2 \ge \frac{T^2}{x_i},$$

where x_i is the number of times item i is produced in the cycle. That is $x_i = |\{j \mid f^j = i\}|$. This justifies the second inequality used below to obtain a lower bound on the average cost. For any executable schedule characterized by f, t, u and v, we have

$$\begin{aligned}
c(f,t,u,v) &= \frac{1}{T}\sum_{j=1}^{n}\left(H^j(p^j)^2((t^j+v^j)^2-(v^j)^2)+K^j\right)\\
&\geq \frac{1}{T}\sum_{j=1}^{n}\left(H^j(p^jt^j)^2+K^j\right)\\
&= \frac{1}{T}\sum_{i=0}^{3m}\sum_{f_j=i}\left(H_i(T^j)^2+K_i\right)\\
&\geq \frac{1}{T}\sum_{i=0}^{3m}\left(H_iT^2/x_i+K_ix_i\right)\\
&= \sum_{i=0}^{3m}\left(H_i(T/x_i)+K_i(x_i/T)\right);\\
&\geq H_0/m+mK_0+\sum_{i=1}^{3m}(H_i+K_i)\\
&= \frac{2}{m}+2\sum_{i=1}^{3m}s_i = 1 = c^*.
\end{aligned} \tag{6}$$

The first inequality follows from the non-negativity of v. The third inequality follows from calculus and results on the EOQ quantities. We conclude that for all executable schedules $c(f,t,u,v) \geq c^* = 1$. By assumption there exist an executable schedule such that $c(f,t,u,v) \leq c^* = 1$, so for that schedule, the three inequalities in (6) must hold as equalities.

It follows from the first inequality that $v^j = 0$, so the schedule satisfies the ZI property; from (5) and the second inequality we have at once that $T^j = T/x_i$, $\forall\, j$ such that $f^j = i$, so the schedule satisfies the LI property and hence the TI property follows. Thus the schedule is a BPCS. From the third equality we know that $T/x_0 = 1/m$, and that $T/x_i = 1$ for $i \neq 0$, so $x_0 = mx_i$, $i = 1, \ldots, 3m$. The x_i must be integers and by definition the cycle length is the smallest time needed for the schedule to repeat so we conclude that $x_0 = m$, $x_i = 1$ for $i \neq 0$ and that the cycle length $T = 1$. Therefore, for j such that $f^j = 0$,

$$T^j = \frac{1}{m},\ t^j = \frac{T^j}{p_0} = \frac{1}{mp_0} = s_0, \tag{7}$$

and for j such that $f^j = i \neq 0$,

$$T^j = 1,\ t^j = T^j/p_i = s_i.$$

And consequently,

$$\begin{aligned}
T &= \sum_{j=1}^{n}(s^j+t^j+u^j) \geq \sum_{j=1}^{n}(s^j+t^j)\\
&= \sum_{i=0}^{3m}\sum_{f^j=i}(s_i+t^j) = \sum_{i=0}^{3m}x_i(2s_i)\\
&= T\left(\frac{2}{m}+2\sum_{i=1}^{3m}s_i\right) = T,
\end{aligned} \tag{8}$$

where the last equality follows from the definition of the data. Hence there is no idle time in the schedule so $u = 0$.

Let τ^k denote the epoch of the kth set up of item 0 within a cycle. From (7) we know that item 0 is setup every $1/m$ units of time. Then at epoch $\tau' \equiv \tau^k + s_0 + t_0$ item 0 will leave the machine, and at epoch $\tau^{k+1} \equiv \tau^k + 1/m$ the machine will set up again for item 0.

Because the machine has no idle time, the interval $[\tau', \tau^{k+1})$ is devoted to the setup and production of items other than 0. Let B_k denote these items; then,

$$\sum_{i\in B_k}(s_i+t_i) = \tau^{k+1}-\tau' = \frac{1}{m}-s_0-t_0 = \frac{1}{m}\left(1-\frac{2}{m}\right).$$

Using the definition of the s_i and the fact that $t_i = s_i$, $i \neq 0$ we obtain

$$2\sum_{i\in B_k}s_i = \frac{1}{mb}\left(1-\frac{2}{m}\right)\sum_{i\in B_k}a_i = \frac{1}{m}\left(1-\frac{2}{m}\right),$$

and conclude that

$$\sum_{i\in B_k}a_i = b, \quad k = 1, 2, \ldots, m.$$

Since $x_i = 1$ for $i \neq 0$, these items are produced exactly once within a period so $\{B_k, k = 1, \ldots, m\}$ is a partition of $\{1, 2, \ldots, 3m\}$. We conclude that the three-partition problem is feasible, and consequently, conclude our proof. ■

As a consequence of the above proof, we know that in that particular instance, the three-partition problem has a feasible solution if and only if the feasibility version of the ELSP has an executable schedule satisfying the cost constraint. We saw that such an executable schedule exists if and only if it is a ZICS and a TICS and a LICS and a BPCS. Therefore the same reduction shows that the ELSP is strongly NP-hard over all these classes of cyclic schedules.

References

[1] Hsu, W.L. (1983) On the general feasibility of scheduling lot sizes of several products on one machine. *Management Science*, **29**, 93–105.

[2] Schweitzer, P.J. and Silver, E.A. (1983) Mathematical pitfalls in the one machine multi-product lot scheduling problem. *Operations Research*, **31**, 401–405.

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

[4] Gallego, G. and Queyranne, M. (1995) Inventory coordination and pricing decisions: analysis of a simple class of heuristics, in Sciomachen, A. (ed.), *Optimization in Industry, vol.3 (Mathematical Programming and Modeling Techniques in Practice)*, Wiley, New York, pp. 83–107

[5] Doll, C. and Whybark, D. (1973) An iterative procedure for the single machine multi product lot scheduling problem. *Management Science*, **20**, 50–55.
[6] Elmaghraby, S.E. (1978) The economic lot scheduling problem (ELSP): review and extensions. *Management Science*, **24**, 587–598.
[7] Dobson, G. (1987) The economic lot scheduling problem: achieving feasibility using time-varying lot sizes. *Operations Research*, **35**, 764–771.
[8] Maxwell, W.L. (1964) The scheduling of economic lot sizes. *Naval Research Logistics Quarterly*, **11**, 98–124.
[9] Delporte, C.M. and Thomas, L.J. (1978) Lot sizing and sequencing for *N* products on one facility. *Management Science*, **23**, 1070–1079.
[10] Hodgson, T. and Nuttle, H. (1986) A note on linear programming and the single machine lot size scheduling problem. *International Journal of Production Research*, **24**, 939–943.
[11] Garey, M. and Johnson, D.S. (1979) *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York.

Acknowledgment

The research of the first author was supported in part by research grant the National Science Foundation Award SBR–94–10146, and that of the second author by the National Science Foundation CAREER Award under Grant No. DMI - 9625320.

Biographies

Guillermo Gallego is Associate Professor of Industrial Engineering and Operations Research at Columbia University. His interests are in production planning, inventroy control and yield management. He holds a graduate degree in Mathematics from UCSD, and M.Sc. and Ph.D. degrees in Operations Research and Industrial Engineering from Cornell University. He serves as Associate Editor for *IIE Transactions, Management Science, Naval Research Logistics* and *Operations Research*.

Dong Shaw is an Assistant Professor in the School of Industrial Engineering, Purdue University. He obtained his Ph.D. degree in Industrial Engineering and Operations Research from Columbia University in 1991, his M.S degree in Management from Shanghai Jiao Tong University, China, in 1984, and his B.S. degree in Mathematics from Fudan University, China, in 1982.