

**A SURROGATE OBJECTIVE FOR UTILITY
WORK IN PACED ASSEMBLY LINES**

AHMET BOLAT
Mechanical Engineering Department
King Saud University
Riyadh, Saudi Arabia

CANDACE ARAI YANO
Department of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-32

October 1991

A SURROGATE OBJECTIVE FOR UTILITY WORK IN PACED ASSEMBLY LINES

Abstract

In this note, we introduce a surrogate objective for utility work at a single station on a paced assembly line. We show that it is asymptotically equal to utility work as the number of jobs increases, and provide expressions for the worst case difference between the two objectives. We also derive closed form expressions for the surrogate objective when a simple sequencing procedure, which provides optimal solutions with respect to utility work under certain conditions, is applied. This circumvents the need to solve dynamic programs in instances where only the value of the objective function is needed, such as in heuristics for multi-station problems.

1 Introduction

In a companion paper (Bolat and Yano 1991), we addressed the problem of determining a schedule to minimize utility work at a single station on a paced assembly line, where the station offers two types of operations. Such single station problems are solved many times in the context of heuristics for the multi-station version of the problem (cf. Bolat and Yano 1990, Yano and Rachamadugu 1991). Moreover, practical applications often require hundreds or thousands of jobs to be scheduled. Consequently, reduction of computation time for the single-station problem may have a significant impact on overall solution times for the multi-station problem. Such reductions may also allow heuristic procedures to consider a greater number of alternatives, thereby improving solution quality.

In this note, we introduce and analyze a surrogate objective for utility work which leads to scheduling problems that are much easier to solve optimally. Perhaps more importantly, our analysis provides a means to compare utility work and the surrogate objective with other similar objectives reported in the literature. We also discuss how the choice of objectives affects characteristics of the resulting schedules.

The motivation for our problem, a detailed problem statement, and a literature review appear in Bolat and Yano (1991), and are not repeated here. For the convenience of the reader, preliminary notation appears in Table 1. Other notation will be defined as needed.

In the next section, we introduce the surrogate objective. The subsequent section provides formulas for the value of the surrogate objective when a particular simple sequencing procedure, which is optimal under certain conditions, is used.

2 Surrogate Objective Function

The development of our surrogate objective was motivated by industry practice. We observed that a commonly used rule-of-thumb to aid in smoothing the flow of work to a station is to require that no more than k out of each l consecutive jobs have the relevant option. Such rules are often referred to as spacing constraints. Sometimes it is not possible to ensure that these constraints are strictly satisfied. The question then becomes one of how to translate violations of these constraints into an objective function that captures the consequences of the violations.

Discussions with scheduling experts at two major automobile companies convinced us that utility work was a reasonably accurate representation of both companies' true objectives. Utility work can be defined as the amount of work (measured in processing time) that would remain undone if the workers proceed at their normal pace and remain within the boundaries of their respective stations. The challenge, then, was to relate spacing constraints and utility work in a systematic way.

In this section, we propose a surrogate objective function which is to minimize the total number of jobs with the option in excess of the k allowed jobs in each consecutive subsequence of l jobs. If we can show that this objective function is closely related to total utility work then we can save a considerable amount of computation by using this new function because it deals only with the combinatorial (sequencing) aspects of the problem and not the detailed time schedule.

Let O and B denote jobs requiring the optional and basic operation, respectively. In Bolat and Yano (1991) we have shown that if there exist integer values of k and m satisfying

$$k = \frac{L - 1}{o - 1} \quad (1)$$

and

$$m = \frac{L - 1}{1 - b}, \quad (2)$$

then a repeating sequence of k O jobs followed by m B jobs has no idle time and no utility work. Also, the system regenerates (returns to its original condition) at the end of each of cycle of $k + m$ jobs. Consequently, if these conditions are satisfied, an optimal sequence can be constructed as follow. Initially fill N positions in the sequence, alternating k O jobs and m B jobs. Define G as the number of repetitive subsequences and r as the number of slots left over at the end of the G subsequences. Each repetitive subsequence has length $l = k + m$, so $G = \lfloor \frac{N}{l} \rfloor$ and $r = N - Gl$. Define $H^* = Gk + \min(k, r)$. It is easy to show that H^* is the maximum number of O jobs that can be scheduled without incurring utility work, and is exactly the number of O jobs in the sequence described above.

The number of O jobs that must be scheduled is referred to as H . If the initial repetitive sequence contains more O jobs than required (i.e., $H^* > H$), an appropriate number of O jobs, arbitrarily selected, are modified to B jobs. (It should be evident that the modified sequence has no utility work.) On the other hand, if the initial sequence has fewer O jobs than required (i.e., $H^* < H$), B jobs must be changed to O jobs until an appropriate number have been modified. To minimize the utility work incurred by these modified jobs, certain rules need to be satisfied in selecting jobs to be modified (see Bolat and Yano 1991). However, utility work computations are simplified (by maximizing the number of regeneration cycles), while maintaining optimality, if the $H - H^*$ "excess" O jobs are scheduled as late as possible in the sequence. Thus, it is useful to think of a specific modification of the initial sequence in which B jobs are changed to O jobs starting at the end of the sequence. We have formalized this procedure in an algorithm called ALGSEQ. (A more detailed discussion and proofs appear in Bolat and Yano, 1991.)

Consider an example with $o = 2.00$ time units, $b = 0.25$ time units, and $L = 4.00$ positions. We would then have $k = 3$, $m = 4$ and $l = 7$. Consider a sequence in which the

pattern of 3 O jobs and 4 B jobs is repeated indefinitely. Observe that in this sequence, every subsequence of 7 consecutive jobs has 3 O jobs and 4 B jobs. (Throughout the remainder of the paper, the term "subsequence" always refers to a subsequence of consecutive jobs.) If one of the B jobs somewhere in the middle of the repeating pattern were an O job instead, then the optimal schedule would have $o - b = 1.75$ time units of utility work. Moreover, each subsequence of 7 consecutive jobs that includes that modified job would have $k + 1 = 4$ O jobs. In other words, when $H = H^* + 1$, the optimal sequence yields $o - b$ time units utility work and l subsequences (of length l) with one excess O job. We now formalize this relationship.

We first introduce some additional notation. Let full window t be the subsequence of l positions $t - l + 1, t - l + 2, \dots, t$ where $l \leq t \leq N$, and partial window t be the subsequence of t positions $1, 2, \dots, t$ where $1 \leq t \leq l - 1$ or the subsequence of $N + l - t$ positions $t - l + 1, t - l + 2, \dots, N$ where $N + 1 \leq t \leq N + l - 1$. Notice that each job appears in l windows with the exception of jobs near the beginning or end of the schedule where the windows are shorter than l . Let the *unit load* of window t , UL_t , be the number of jobs requiring the optional operation in window t , i.e.,

$$UL_t = \begin{cases} \sum_{h=1}^t X_h & t = 1, \dots, l - 1, \\ \sum_{h=t-l+1}^t X_h & t = l, \dots, N, \\ \sum_{h=t-l+1}^N X_h & t = N + 1, \dots, N + l - 1. \end{cases} \quad (3)$$

Finally, let the *unit violations* in window t , V_t , be the number of O jobs in excess of the k allowed jobs, i.e.,

$$V_t = UL_t + E_t - k, \quad t = k + 1, \dots, N + m - 1 \quad (4)$$

where $E_t = [k - UL_t]^+$, which can be defined as the slack in window t . We omit the partial

windows whose lengths are k positions or less, since these windows cannot have more than k O jobs.

The definition of violations here is different from the one used in the existing literature. Previous researchers have counted the number of constraints that are violated or the number (or equivalently, fraction) of jobs violating constraints. The number of jobs violating constraints depends on one's interpretation of the term. We will interpret it as the number of jobs with the option contained in any violated constraint. That is, the number of jobs violating constraints equals

$$\sum_h X_h \delta_h, \quad (5)$$

where $\delta_h = 1$ if $UL_t > m$ for any $t \in \{h - l + 1, \dots, h\}$ and zero otherwise.

These measures do not capture the extent to which the constraints are violated, a factor which is captured in our metric. To illustrate this point, consider the previous example. With one excess O job, l different spacing constraints would be violated. In this instance, the direct correspondence is maintained between unit violations and both the number of violated constraints and the number of jobs violating constraints. If we now increase the number of excess O jobs, however, these metrics behave differently depending on the position of the second excess O job. First consider the situation in which the second excess O job is at least l positions away from the first. In this case, the second excess O jobs increases both unit violations and the number of violated constraints by l , and the proportionality between the two metrics is once again maintained. If the second excess O job is placed at least $2l - 1$ positions from the first, then the number of jobs violating constraints increases by l because the two sets of jobs violating constraints have no jobs in common. On the other hand, if the second excess O job is greater than l positions but less than $2l - 1$ positions away from the first, the number of jobs violating constraints will increase by fewer than l .

Now consider placing the second excess O job adjacent to the first one. This increases unit violations by l . On the other hand, it may not increase the number of violated constraints at

all, and in any case, by at most one. It increases the number of jobs violating constraints by at most 2 (the modified job plus possibly one of the jobs $l - 1$ positions away, if it happens to be a job with the option). Thus, counting the number of violated constraints or the number of jobs violating constraints tends to encourage concentrating excess work in a small portion of the sequence, and clearly does not cause an increase in the objective function that is proportional to the incremental amount of excess work. As a consequence, our metric is more closely related to total utility work.

3 Formulas for Unit Violations

Let $H' = H - H^*$ be the number of excess O jobs. The following theorem gives total unit violations for the sequence \mathbf{S} which is produced by ALGSEQ.

Theorem 1: The total unit violations in \mathbf{S} is given by the following formulas:

$$\sum_{t=k+1}^{N+m-1} V_t = \begin{cases} 0 & \text{if } H \leq H^* \\ H'l & \text{if } H > H^*, r = k \\ H'(H' + r) & \text{if } H > H^*, r < k, H' < \min(k - r, m) \\ H'l - m(k - r) & \text{if } H > H^*, r < k, H' \geq \min(k - r, m) \\ H'l - H'(r - H') & \text{if } H > H^*, r > k, H' < \min(r - k, k) \\ H'l - k(r - k) & \text{if } H > H^*, r > k, H' \geq \min(r - k, k) \end{cases} \quad (6)$$

Proof: See Appendix 1. ■

Notice that in all cases, total unit violations is bounded by $H'l$.

Earlier in the paper, we showed that in some sequences, each excess O job creates l unit violations and $a - b$ utility work. Therefore, we propose the following surrogate objective function for total utility work: unit violations are summed over all windows then multiplied by a weight $\vartheta = \frac{a-b}{l}$. The following corollaries establish relationships between two objective

functions in the solutions obtained by ALGSEQ when there exist integral k and m satisfying equations (1) and (2).

Corollary 1: In \mathbf{S} , total utility work is equal to total weighted violations with weight $\vartheta = \frac{o-b}{l}$ when (i) $r = k$, or (ii) $r < k$ if $H' \geq k - r$, or (iii) $r > k$ if $H' \geq k$.

Proof: The proof is trivial when $r = k$ because total weighted unit violations is equal to $H'l(\frac{o-b}{l}) = H'(o-b)$ which is the total utility work.

When $r < k$, the difference between two objective values is

$$\vartheta \sum_{t=k+1}^{N+m-1} V_t - \sum_{i=1}^N U_i = H'(o-b) - m(1-b) + r(o-1) - [H'(o-b) - \frac{m}{l}(o-b)(k-r)], \quad (7)$$

where U_i is the utility work incurred by the job in the i th position in the sequence and

$$\sum_{i=1}^N U_i = \begin{cases} 0 & \text{if } H \leq H^* \\ H'(o-b) & \text{if } H > H^*, r = k \\ 0 & \text{if } H > H^*, r < k, H^e \geq H' \\ H'(o-b) - (k-r)(o-1) & \text{if } H > H^*, r < k, H^e < H' < m \\ H'(o-b) - m(1-b) + r(o-1) & \text{if } H > H^*, r < k, H' \geq m \\ 0 & \text{if } H > H^*, r > k, H^f \geq H' \\ H'(o-b) - (r-k)(1-b) & \text{if } H > H^*, r > k, H^f < H' \end{cases} \quad (8)$$

where $H^e = \lfloor (k-r)\frac{m}{l} \rfloor$ and $H^f = \lfloor (r-k)\frac{k}{l} \rfloor$, which are values that help us to determine whether the final full and fractional cycles have utility work. (See Bolat 1988 for detailed derivations of the expressions in (8).) By using equations (1) and (2) we can show that the following equalities are true.

$$\frac{o-b}{l} = \frac{L-1}{km} \quad (9)$$

$$1-b = \frac{L-1}{m} \quad (10)$$

$$o-1 = \frac{L-1}{k}. \quad (11)$$

After substituting these into equation (7) it is easy to see that the difference is

$$= (L - 1)\left(-1 + \frac{r}{k} + \frac{k - r}{k}\right) = 0. \quad (12)$$

Finally, the case $r > k$ will be considered. Notice the assumption $H' \geq m$ implies that $H' > H^f$. Therefore, the difference is now equal to:

$$\vartheta \sum_{t=k+1}^{N+m-1} V_t - \sum_{i=1}^N U_i = H'(o - b) - (r - k)(1 - b) - [H'(o - b) - \frac{k}{l}(o - b)(r - k)]. \quad (13)$$

After substituting (7) and (8) into (11) we can show that

$$\vartheta \sum_{t=k+1}^{N+m-1} V_t - \sum_{i=1}^N U_i = (L - 1)\left(-\frac{r - k}{m} + \frac{r - k}{m}\right) = 0. \quad \blacksquare \quad (14)$$

Thus we have shown that if there exist integral k and m satisfying equations (1) and (2) then under certain conditions, the two measures give the same result in a solution obtained by ALGSEQ. The next corollary completes the analysis.

Corollary 2: If there exists integral k and m satisfying (1) and (2) and the two objective values of S are not equal, then the utility work per job is asymptotically equal to weighted violations per job.

Proof: By using equations (3), (4), (6), and (8), one can show that the maximum difference between total utility work and weighted unit violations for the cases not covered in the previous corollary is given by

$$\vartheta \sum_{t=k+1}^{N+m-1} V_t - \sum_{i=1}^N U_i = \begin{cases} \frac{L-1}{l}(k - r)\frac{m+r}{l} & \text{if } r < k, 0 \leq H' < (k - r)\frac{m}{l} \\ \frac{L-1}{l}(k - r) & \text{if } r < k, (k - r)\frac{m}{l} \leq H' < (k - r) \\ \frac{L-1}{l}(r - k)\frac{l+k-r}{l} & \text{if } r > k, 0 \leq H' < (r - k)\frac{k}{l} \\ \frac{L-1}{l}(r - k)\frac{l+k-r}{m} & \text{if } r > k, (r - k)\frac{k}{l} \leq H' < k \end{cases} \quad (15)$$

This implies that the maximum difference between two measures is bounded by $\frac{L-1}{l}(k - r)$ if $r < k$ and $\frac{L-1}{l}(k - r)\frac{l+k-r}{m}$ if $r > k$. Hence, in any case, the difference *per job* can not be greater than $\frac{L-1}{N}$. Therefore, for sufficiently large N this fraction goes to zero. \blacksquare

If we relax the assumption about the integrality of k and m , i.e., if we assume instead that these parameters are determined by

$$k = \lfloor \frac{L-1}{o-1} \rfloor \quad (16)$$

and

$$m = \lceil \frac{k(o-1)}{1-b} \rceil. \quad (17)$$

then the value of total weighted violations in \mathbf{S} is not affected, and consequently, ALGSEQ still produces optimal solutions with respect to the surrogate objective. However, the total utility work changes. In Bolat and Yano (1991) we have shown that the maximum error in the amount of utility work per cycle is bounded by $(1-b)$. This error bound represents the maximum amount of potentially avoidable idle time per cycle in a schedule where k and m are defined by (16) and (17) respectively. The actual error approaches this bound when it is barely infeasible to use parameters k and $m-1$, so one needs to use k and m , leading to slightly less than $(1-b)$ units of idle time at the end of the cycle. Since total number of cycles, G , is bounded by N/l , the difference between two measures is bounded by $(L-1) + (1-b)\frac{N}{l}$. Therefore the maximum difference per job goes to $\frac{1-b}{l}$ as N goes to infinity. In practice, however, the actual error is much smaller. Indeed, once k and m have been specified, the maximum error in the amount of utility work per cycle can be computed as $l - (ko + mb)$, which is the difference between the time available and the time required for k O jobs and m B jobs. The maximum error per job, therefore, goes to $1 - (ko + mb)/l$, which is equivalent to the fraction of the time the worker is idle during a cycle with k O jobs and m B jobs.

One interesting result of our analyses is that for a range of parameter values, these two objective values are equal. If not, the difference per job is asymptotically equal either to zero or to a small value which is a simple function of the problem parameters. This implies that if the correct spacing parameters are selected and the jobs are sequenced optimally according

to the spacing rules, a very good sequence will result. We believe that this is the reason underlying the satisfactory performance of these spacing rules in practice.

One very important feature of our surrogate function is that it does not impose any restrictions on the spacing parameters. Moreover, ALGSEQ always produces optimal sequences with respect to the new function (see Bolat 1988). On the other hand, much more complicated procedures, such as dynamic programs, may be needed to find optimal solutions for the objective of total utility work. Another important benefit of employing the surrogate objective function is that it deals with only the combinatorial aspects of the problem and an accurate approximation of utility work can be obtained by multiplying the surrogate objective value by an appropriate weight. Thus, most of the computations involve integers rather than real values. Indeed, unit violations can be expressed in closed-form as a function of some values that can be computed easily from the problem data. This can result in a considerable amount of computational savings, thereby making the multiple-station problem more tractable.

4 Acknowledgments

This work was supported in part by a research contract from a major U.S. automobile company to the University of Michigan.

References

- Bolat, A., 1988. "Generalized Mixed Model Assembly Line Sequencing Problem." Unpublished Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.
- Bolat, A. and C.A. Yano, 1991. "Scheduling Algorithms to Minimize Utility Work at a Single Station on a Paced Assembly Line," Technical Report 88-7, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI (revised October 1991).
- Yano, C.A. and A. Bolat, 1990. "Survey, Development, and Application of Algorithms for Sequencing Paced Assembly Lines," *Journal of Manufacturing and Operations Management*, Vol. 2, No. 3, pp. 172-198.
- Yano, C.A. and R. Rachamadugu, 1991. "Sequencing to Minimize Work Overload in Assembly Lines with Product Options." *Management Science*, Vol 37, No. 5, pp. 572-586.

Table 1

o	=	time required to complete an optional operation,
b	=	time required to complete a basic operation,
L	=	length of the station expressed in terms of number of jobs, = sojourn time of a job in the station,
N	=	total number of jobs,
H	=	number of jobs requiring the optional operation,
U_h	=	utility work for the job in position h ,
X_h	=	$\begin{cases} 1 & \text{if a job with the optional operation is assigned to slot } h, \\ 0 & \text{otherwise..} \end{cases}$

Appendix

Proof of Theorem 1:

Case 1: $H \leq H^*$.

By the construction of \mathbf{S} , the number of O jobs in each window is less than or equal to k . Hence, by definition, the number of unit violation in each window is zero.

Before we consider cases with $H > H^*$, we will prove that if $H = H^*$ and $r = k$, then the slack term E_t is zero in every full window of \mathbf{S} . We have already shown that the number of unit violation in each window t , V_t , is zero. Therefore, by using equation (4), we can write that

$$\sum_{t=k+1}^{N+m-1} E_t = (N + m - k - 1)k - \sum_{t=k+1}^{N+m-1} UL_t. \quad (18)$$

By applying equation (3) for windows $k + 1, \dots, N + m - 1$, we can write that

$$\begin{aligned} \sum_{t=k+1}^{N+m-1} UL_t &= mX_1 + (m+1)X_2 + \dots + (l-1)X_k + \\ &\quad l(X_{k+1} + \dots + X_{N-k}) + \\ &\quad (l-1)X_{N-k+1} + \dots + mX_N \end{aligned} \quad (19)$$

Observe that ALGSEQ places O jobs into the following positions: $1, 2, \dots, k; l+1, \dots, l+k; \dots (G-1)l+1, \dots, (G-1)l+k; N-k+1, \dots, N$. Therefore,

$$\sum_{t=k+1}^{N+m-1} UL_t = m + (m+1) + \dots + (l-1) + l[(G-1)k] + (l-1) + \dots + m \quad (20)$$

$$= k(2m + k - 1) + kl(G-1). \quad (21)$$

By using (18) and (20), the fact that in this case $N = Gl + k$, and $l = k + m$, we can show that

$$\sum_{t=k+1}^{N+m-1} E_t = (Gl + k + m - k - 1)k - [k(l + m - 1) + kl(G-1)] = 0. \quad (22)$$

Now we will consider the remaining cases.

Case 2: $r = k$.

We have shown that $E_t = 0$ when $H = H^*$; hence, it must also be true when $H > H^*$ because each additional O job either creates some unit violations or eliminates some slacks. Therefore, by using equation (4), we can write that

$$\sum_{t=k+1}^{N+m-1} V_t = \sum_{t=k+1}^{N+m-1} (UL_t - k). \quad (23)$$

Moreover we have shown that

$$\begin{aligned} \sum_{t=k+1}^{N+m-1} UL_t = k(Gl + m - 1) + l & (X_{k+1} + \dots + X_l + X_{l+k+1} + \dots + X_{2l} \\ & + \dots + X_{(G-1)l+k+1} + \dots + X_{Gl}) \end{aligned} \quad (24)$$

where the X 's on the right hand side reflect the assignment of the set of jobs still available after the first H^* jobs have been assigned. Equation (24) implies that each O job from this set creates l unit violations if it is placed into any available position. Since there are only H' of them and $N = Gl + k$, by substitution into (23) we obtain

$$\sum_{t=k+1}^{N+m-1} V_t = H'l. \quad (25)$$

Case 3: $r < k$ and $H' < m$.

We initially assume that $G > 1$. The first window that has more than k jobs requiring the optional operation starts at position $N - (r + H' + l) + 2$, i.e., window $N - (r + H') + 1$. From this window on, there will be unit violations. Namely, windows $N - (r + H') + 1$, $N - (r + H') + 2, \dots, N - r$ have $1, 2, \dots, H'$ unit violations respectively; windows from $N - r + 1$ to N have H' unit violations in each; windows from $N + 1$, $N + 2, \dots, N - r + k$ have $H' - 1, H' - 2, \dots, H' + r - k$ unit violations respectively; windows from $N - r + k + 1$ to $N + l - H' - r$ have $(H' + r - k)^+$ unit violations in each; and finally windows from $N + l - H' - r + 1, N + l - H' - r + 2, \dots, N + m - 1$ have $H' + r - k - 1, H' + r - k - 2, \dots, 1$

unit violations respectively. Therefore, the total unit violations can be written as

$$\sum_{t=k+1}^{N+m-1} V_t = \frac{H'(H'+1)}{2} + rH' + (m-H')(H'+r-k) + \frac{H'(H'-1)}{2} \quad (26)$$

$$= H'l - m(k-r). \quad (27)$$

If $H' + r \leq k$, the first, second, third and fifth groups of windows and their unit violations are still as above. The fourth group of windows has zero unit violations in this case. Therefore, total unit violations is given by

$$\sum_{t=k+1}^{N+m-1} V_t = \frac{H'(H'+1)}{2} + rH' + \frac{H'(H'-1)}{2} = H'(H'+r) \quad (28)$$

We can show that in the case $G = 1$ total unit violations in the first group is still $1 + 2 + \dots + H'$. Hence, all the previous results hold in this case. The case $G = 0$ is not possible because of the assumption that $k < r$.

Case 4: $r < k$ and $H' \geq m$.

G must be greater than or equal to 1 by assumption. Equation (27) implies that total unit violations for the excess jobs placed in the last full cycle is equal to

$$ml - m(k-r) = m(m+r). \quad (29)$$

Moreover, ALGSEQ spreads the remaining $H' - m$ jobs over the $G - 1$ cycles arbitrarily. In case 2 we have shown that each one of these jobs creates l unit violations. Therefore, total unit violations is given by

$$\sum_{t=k+1}^{N+m-1} V_t = (H' - m)l + m(m+r) = H'l - m(k-r). \quad (30)$$

Case 5: $r > k$ and $H' < r - k$.

Without loss of generality we can assume that $G \geq 1$. The first window that has more than k jobs requiring the optional operation starts at position $N - (H' + l) + 2$, i.e., window $N - H' + 1$. From this window on, there will be unit violations. Namely, windows $N - H' + 1$,

$N - H' + 2, \dots, N$ have $1, 2, \dots, H'$ unit violations respectively; windows from $N + 1$ to $N + l - r$ have H' unit violations in each; windows from $N + l - r + 1, N + l - r + 2, \dots, N + l - r + k$ have $H' - 1, H' - 2, \dots, H' - k$ unit violations respectively; windows from $N + l - r + k + 1$ to $N + l - H'$ have $(H' - k)^+$ unit violations in each; and finally, windows from $N + l - H' + 1, N + l - H' + 2, \dots, N + m - 1$ have $H' - k - 1, H' - k - 2, \dots, 1$ unit violations respectively. Therefore, the total unit violations can be written as

$$\sum_{t=k+1}^{N+m-1} V_t = \frac{H'(H'+1)}{2} + (l-r)H' + [r - (H'+k)](H'-k) + \frac{H'(H'-1)}{2} \quad (31)$$

$$= H'l + k(k-r). \quad (32)$$

When $H' \leq k$, the unit violations indicated above for the first, second, third and fifth groups of windows are still true, but there are no unit violations for the fourth group of windows. Hence, total unit violations is

$$\sum_{t=k+1}^{N+m-1} V_t = H'(H'+l-r). \quad (33)$$

One can show that in the case $G = 0$ total unit violations in the first (and only) group will be $1 + 2 + \dots + H'$. Hence all the results also will be true in this case.

Case 6: $r > k$ and $H' \geq r - k$.

G must be greater than 0 by assumption. Equation (32) implies that total unit violations for the additional jobs placed in the last full cycle is equal to

$$(r-k)l + k(k-r) = rl - k(m+r). \quad (34)$$

Moreover, ALGSEQ spreads the remaining $H' - (r - k)$ jobs over the $G - 1$ cycles arbitrarily. In case 2 we have shown that each one of these jobs creates l unit violations. Therefore, total unit violations will be

$$\sum_{t=k+1}^{N+m-1} V_t = [H' - (r - k)]l + rl - k(m+r) = H'l - k(r - k). \quad \blacksquare \quad (35)$$