Coordinating n Parallel Stochastic Activities by

an Exact Generalization of the Newsvendor Model

by

Dan Trietsch

and

Francisco Quiroga

The University of
Auckland Business School
ISOM, Old Choral Hall
7 Symonds Street
Auckland, New Zealand
++649-373-7599 ext 87381
fax: ++649-373-7430
d.trietsch@auckland.ac.nz

August 2004

Revised

July 2005

---

1 Corresponding author

2 This paper had been submitted to European Journal of Operational Research before the paper "Optimal Feeding Buffers for Projects or Batch Supply Chains by an Exact Generalization of the Newsvendor Result" (by Trietsch) was submitted to International Journal of Production Research. The EJOR referees agreed that this paper is correct (in terms of mathematical derivations and citations), but disagreed whether it deserves to be archived there. The editor, generously, allowed us a further revision (to make the paper publishable as a technical note). However, by that time the more advanced paper had been accepted by IJPR, so--after further discussion with the EJOR editor--we decided to decline the revision.
Coordinating $n$ Parallel Stochastic Activities by

an Exact Generalization of the Newsvendor Model

by

Dan Trietsch

Francisco Quiroga

Abstract

The problem of coordinating $n$ parallel stochastic inputs, all of which are necessary before we can deliver the output, and with linear input-earliness and output-tardiness costs, has been independently addressed by several authors since the late eighties. They provided an exact solution and bounds, one of which is an effective heuristic for systems with a high service level (but not otherwise). We consolidate the former results and enhance them by providing tighter bounds and studying the constrained version of the model (where inputs must be non-negative) in detail. Such constraints may lead to low service level. Indeed, our main practical contribution is solving cases with low service level effectively. We also interpret the existing optimal solution as an exact generalization of the newsvendor result.

Keywords: Project management and scheduling, uncertainty modeling, feeding buffers, stochastic scheduling
We consolidate and augment the *assembly coordination model* (ACM), whose basic parts have been introduced independently, between 1988 and 1993, by three sets of authors [10,8,2,11]. We provide new bounds and study the case where inputs must be non-negative in detail. We also highlight a new economical insight: the previous optimal result can be interpreted as an exact generalization of the classical newsvendor result.

The ACM concerns scheduling of purchase orders for parts required for batch assembly operations or projects. Let an assembly require n statistically independent parallel inputs composing a simple assembly tree (with the assembly at the root). Let $Y_i$, with mean $\mu_i$, denote the lead time of input i, where $F_i(y)$ is its CDF and, assuming continuous distributions, $f_i(y)$ is the corresponding density function (i=1,...,n). The assembly must start at a given due date, $G$ (a mnemonic for "gate"), and is subject to a tardiness penalty cost of $C \geq 0$ per time unit (C is a mnemonic for "cost"). Let input i be ordered at time $g_i \geq 0$ (i=1,...,n) and let $g=\{g_i\}$. Once input i arrives (at time $g_i+Y_i \geq g_i \geq 0$), it incurs holding costs at the rate of $c_i \geq 0$, but such that $\sum_{i=1,...,n} c_i > 0$, until completion or the due date, whichever is later. For convenience, we may refer to G as $g_{n+1}$ and to C as $c_{n+1}$, so they augment the vectors $\{g_i\}$ and $\{c_i\}$ (but $g$ only denotes the first n elements of $\{g_i\}$), and let $Y_{n+1}=0$. Define T as the completion time, then $T=\max_{i=1,...,n+1} (g_i+Y_i)$ and the assembly tardiness is $(T-G)$. The objective is to minimize the expected total cost, $Z$

$$Z = \mathbb{E} \left[ \sum_{i=1}^{n+1} c_i (T-g_i) \right] - \sum_{i=1}^{n} c_i \mu_i$$

The last element reflects the assumption that holding costs apply only after an input actually arrives. Because it is constant, this element does not influence the optimal solution. Essentially, for optimization purposes, it does not matter if the costs start when
we order, upon receipt, at any convex combination of the above, or at an increasing rate proportional to the order’s progress. The model is continuous, but directly extendable to discrete periods.

Historically, the constraints $g_i \geq 0$ only appeared in [10] and [11], perhaps because theirs was a project scheduling scenario where such constraints are often binding (the others assumed a repetitive batch assembly scenario--i.e., a supply chain model). We refer to the model with (without) these non-negativity constraints as constrained (unconstrained). The previous formulations did not allow $c_i = 0$, but we include it for generality. Similarly, [2] did not allow $C=0$. [13] generalized the ACM to scheduling flights at hub airports, where sequencing decisions are solved heuristically along with the determination of optimal safety times between arrivals and departures. This is analogous to supply chains with many inputs (supplies) feeding a central node that combines various combinations of inputs to divergent outputs (products). For example, break-bulk/make-bulk depots. [7] addressed the basic ACM, but with a step tardiness penalty function (a case included by [13] as well). [15,16] studied structures with serial operations, and [17,18] dealt with two operations in parallel that follow or precede a single operation. While all these sources focus on lead-time performance, isomorphic structures focus on throughput [5,14], where the most constrained node determines the capacity. Similarly, the stochastic element may involve quantities (stochastic demand); e.g., in refurbishing projects where precise part requirements are not known in advance. [12] proved that the unconstrained ACM is convex. They considered stochastic demand and stochastic lead times together and noted that their convexity proof holds for either demand or lead time, but not both together. By addressing demand and lead time separately, however, good approximate results were obtained. [1] used the ACM as a building block for an approximate solution.
of a more complex assembly tree, and reported results within 1% of the optimum.

In the basic model we assume that assembly is instantaneous. So, to prevent assembly tardiness, all inputs must arrive by G. Since [10,11] focused on project applications, they allowed each input its own "latest finish" due date. Because the model is static, however, this merely shifts ordering times such that the safety buffers that the model yields remain unchanged. This extends the model to assemblies or projects with more complex structures than a simple assembly tree. For the results to be valid, however, the variance of assembly operations must be negligible relative to the lead time variance. Based on practical experience, however, [10,11] claimed that this was often the case. Specifically, in the environment for which their model was originally developed--a large avionics firm--purchasing lead times had standard deviations (s.d.) often measured in months while assembly s.d. (once all inputs were in place) was measured in hours or days. To illustrate, if the s.d. of assembly is less then 20% of the lead time s.d. (i.e., the ratio between a day and a week) then the s.d. of the whole project is only about 2% higher than that of the lead time alone. Finally, within this context, G-(g_i+u_i) is nowadays also known as the feeding buffer of input i [6]. With this in mind, the ACM and the other contemporaneous lead-time models we cited are the bellwethers of the more recent focus on feeding buffers in project management.

In Section 2 we consolidate previous results for the ACM (excluding those that are superseded in this paper). In section 3, we provide tighter bounds than formerly available, including constrained cases. Section 4 interprets the optimal solution as an exact generalization of the newsvendor model with n inputs. While it does not alleviate the need for a numerical search for the optimal ordering times, it holds even if we don’t know the distributions and may be used for control of stochastic systems where such activities occur
repetitively. It also enables effective solution by simulation of both constrained and unconstrained cases, as discussed in Section 5. Some observations based on numerical evidence are discussed in Section 6. Section 7 is the conclusion.

2. Consolidating the Existing Results

The CDF of the maximum of \( n \) independent random variables is given by the product of their CDFs and the expected value of a non-negative random variable is the area above the CDF, so the expected tardiness penalty is given by

\[
E((T-G)^+) = C \int_{G}^{\infty} \left( 1 - \prod_{i=1}^{n} F_i(t-g_i) \right) dt
\]

Assuming no constraints on the ordering times of any input, [2] showed that \( G-g_i \) must be non-negative; i.e., we should not postpone any order so much that the assembly will be delayed with certainty. This particular result requires \( C>0 \), however, as we discuss later.

They then showed that the expected holding cost of input \( i \) is

\[
c_i \left( \int_{G}^{\infty} \left( 1 - \prod_{k=i}^{n} F_k(t-g_k) \right) dt - G - g_i - \mu_i \right) = c_i \left( E((T-G)^+) - G - g_i - \mu_i \right) ; \quad i=1,\ldots,n
\]

By Jensen’s inequality, \( E(\max\{g_i+Y_i\}) \geq \max\{E(g_i+Y_i)\} \), i.e., \( G+E((T-G)^+) \geq \max_i \{\mu_i + g_i\} \) so these costs are non-negative. Define \( s=\Sigma_{i=1,\ldots,n+1} c_i \) (i.e., \( C+\Sigma_{i=1,\ldots,n} c_i \)), and \( Z \) is given by

\[
Z = \sum_{i=1}^{n} c_i (G-g_i-\mu_i) + s \int_{G}^{\infty} \left( 1 - \prod_{i=1}^{n} F_i(t-g_i) \right) dt \tag{1}
\]

The first part of the expression is the total cost of the feeding buffers. The expected holding costs beyond \( G \) are included in the second part. Taking partial derivatives by \( g_i \),
\[ \frac{\partial Z}{\partial g_i} = -c_i + s \int_0^\infty f_i(t-g_i) \prod_{k \neq i} F_k(t-g_k) \, dt \quad ; \quad i = 1, \ldots, n \] (2)

We use stars to denote optimal values, e.g., \( g_i^* \) (Similarly, we will use the superscripts L and U to denote lower- and upper bounds.) To obtain \( Z^* \), Eq. 2 should be set to zero for all \( i \), except where the partial derivative is positive for \( g_i = 0 \), in which case \( g_i^* = 0 \) (by constraint). \cite{8} and \cite{2} showed that setting Eq. 2 to zero yields

\[ c_i = s \Pr\{g_i + Y_i \geq \max_{k=1, \ldots, n+1} \{g_k + Y_k\}\}. \]

Because the problem is convex (see \cite{12} and the proof of Theorem 1 below), this is sufficient for optimality.

Utilizing these results directly requires repetitive computation of the integrals. Thus, bounds that do not require integration (except, possibly, to evaluate \( F_i(y) \)), are useful. In this paper we ignore bounds that violate this condition. Define the service level of input \( i \), \( SL_i \) \((i = 1, \ldots, n)\), as the probability it will arrive by the due date; i.e., \( SL_i = SL_i(g_i) = F_i(G-g_i) \). Let \( R_i = R_i(g_i) = 1 - SL_i \) denote the risk of input \( i \) \((i = 1, \ldots, n)\); e.g., \( R_i(0) = 1 - SL_i(0) = 1 - F_i(G) \) is the risk when the input is ordered immediately. The probability all inputs will be in time is the assembly service level, \( SL \), and, by independence, \( SL = \prod_{i=1, \ldots, n} SL_i \).

\( R = 1 - SL \) is the assembly risk. By setting \( \prod_{k \neq i} F_k = 1 \), which increases the integral in Eq. 2, we obtain an upper bound on the service level of input \( i \)

\[ SL_i^L \leq SL_i^U = 1 - \frac{c_i}{s} \Rightarrow R_i^L \leq R_i^U = \frac{c_i}{s} \quad ; \quad i = 1, \ldots, n \] (3)

\cite{10,11} observed that this bound also holds for the constrained case. \cite{2} described an iterative algorithm that starts at \( g_i^L = \arg\{F_i(G-g_i) = SL_i^U\} \), i.e., \( g_i^L = G - F_i^{-1}(SL_i^U) \), and increases gates one by one iteratively, converging to the optimal solution. Although they only considered the unconstrained case, it can be shown that this applies in the constrained case too (as we discuss later).
For the unconstrained case, \([8,2]\) obtained a classical newsvendor result,

\[
\text{SL}^* = \frac{C}{s}
\]

(4)

A similar expression was obtained by \([13]\) for each aircraft in his model. \([2]\) proposed using \(\text{SL}^*\) as a lower bound on \(\text{SL}^*_i\). \([8]\) noted that this result does not require stochastic independence (an observation for which he credited D.R. Smith). To see this, define a block of inputs as any subset of inputs where we are not concerned with individual criticalities but rather with the combined criticality of all inputs in the block. Blocks are adjusted by shifting all the activities within them by the same increment. Here, consider a block that includes all the physical inputs \((i=1,...,n)\). Assume all \(g^*_i\) are strictly positive and advance (postpone) all the orders by an infinitesimal amount, \(\varepsilon\). As a result, with probability \(\text{SL}^*\) we lose (gain) \(\varepsilon(s-C)\) and with the complementary probability, \(R^*\), we gain (lose) \(\varepsilon C\). Balancing the two proves the result for the unconstrained case. To see that all \(g^*_i\) must be unconstrained for this to hold, consider that if, for some \(i\), \(g^*_i=0\) due to a constraint, then postponing the inputs as a block exacerbates the problem and it is simply impossible to advance them as a block.

We now consider the \(C=0\) case. Here, the optimal ordering times can be postponed as a block indefinitely without changing the objective function. Furthermore, \(\text{SL}^* = 0\) because if we allow the project to be ready before \(G\), we incur holding costs for no benefit. Therefore, for \(i\) such that \(c_i > 0\), one optimal solution satisfies \(\min\{g_i\} = G\). In contrast, if \(c_i = 0\) then \(g^*_i = 0\) (to minimize the risk that such inputs will delay the project).
3. New Bounds

In this section we first develop tighter lower bounds for unconstrained cases. Then we discuss the constrained case, for which the bounds are less tight. When the upper and the lower bounds are close enough, either of them can serve as a heuristic solution.

**Bounds for the unconstrained case**

Let \( SL^U = \Pi_{i=1}^{n} SL^U_i \), then our first lower bound is based on the inequality

\[
SL^*_i \geq SL^L_i = \frac{SL^U_i \cdot SL^*_i}{SL^U} \geq \frac{C}{s \prod_{k \neq i} SL^U_k} \geq SL^* \quad ; \quad i, k = 1, ..., n
\]

where, for each \( i \), we use the product of the upper bounds of the other inputs to increase \([2]’s bound, SL^*\). By substituting \( SL^*/SL^*_i \) for \( \prod_{k \neq i} F_k \) in the partial derivative (Eq. 2), a substitution that decreases the integral for \( n \geq 2 \), we obtain another lower bound,

\[
SL^*_i \geq SL^L_i = \frac{C}{C + c_i} \cdot SL^* = \frac{s}{C - c_i} \geq SL^* \quad ; \quad i = 1, ..., n
\]

When \( C >> s - C \), both bounds are tight. Examples exist where the first is better, and others where the second is better. So we define a composite lower bound by,

\[
SL^L_i = \max \left( \frac{C}{C + c_i}, \frac{SL^U_i C}{SL^U s} \right) \quad ; \quad i = 1, ..., n \quad (5)
\]

We need the independence assumption to derive the tighter bounds. However, using [8]’s observation that \( SL^* \) does not require statistical independence, it follows that the former bound given by [2], \( SL^* \), does not require statistical independence either. So \( SL^* = C/s \) remains the tightest lower bound for \( SL^*_i \) with dependence. It cannot be tightened because with perfect positive dependence it applies as an equality.
**The constrained case**

We now derive bounds on the assembly service level and lower bounds for each input for the constrained case. These bounds are not guaranteed to be meaningful, however (because in constrained systems $SL^* = 0$ is possible). Let $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n) \geq 0$ be a vector denoting the respective Lagrange multipliers. For $i=1,...,n$, define $v_i$ by $v_i = c_i + \lambda_i$ (where $v$ is a mnemonic for "value" and $\lambda_i$ is the difference between the marginal economic value of an input, $v_i$, and its marginal cost, $c_i$). Let $V = C - \Sigma \lambda_i$, and we will also use $v_{n+1}$ as an alias for $V$, so $\Sigma_{i=1,...,n+1} v_i = \Sigma_{i=1,...,n+1} c_i = s$. The Lagrangian is essentially the unconstrained objective function minus $\Sigma \lambda_i g_i$, or, equivalently,

$$L(g, \lambda) = \sum_{i=1}^{n} v_i (G_i - g_i - \mu_i) - \sum_{i=1}^{n} \lambda_i (G_i - \mu_i) + s \int_{G} \left( 1 - \prod_{i=1}^{n} F_i(t - g_i) \right) dt$$

where, to satisfy the Karush-Kuhn-Tucker (KKT) necessary conditions, $(v_i^* - c_i) g_i^* = 0$.

Because $\lambda_i \geq 0$, $V^* \leq C$. We are concerned with the Lagrangian as a function of $g$, so for the purpose of optimization we can safely ignore the element $-\Sigma \lambda_i (G_i - \mu_i)$. Therefore, the Lagrangian is isomorphic to Eq. 1, with $v_i$ replacing $c_i$ ($i=1,...,n+1$). By the optimality condition presented in the next section it can be shown that $SL^* = V^*/s$, which implies $V^* \geq 0$ (where $V^* = 0$ necessarily if the probability of meeting the due date is 0 even if all inputs are ordered immediately). For $V^* > 0$, if we replace $c_i$ by $v_i^*$ where they are different (for $i=1,...,n+1$), all the results we obtained for the unconstrained case apply for the Lagrangian. However, by the KKT conditions, if $V^* = 0$ but $C > 0$ there must exist at least one input, say $i$, with $g_i^* = 0$. This is in contrast to the unconstrained $C = 0$ case where meeting the due date has no economic importance and we could postpone all activities as a block indefinitely. So when $SL^* = 0$ due to the constraint $g \geq 0$ (i.e., $C > V^* = 0$), the Lagrangian solution is different from the unconstrained solution with $c_i = v_i^*$ because it
must include at least one immediate order.

Upper bounds for $v_i^*$ are given by $v_i^* \leq v_i^{U} = \max\{c_i, R_i(0)s\}; i=1,...,n$. They are based on the unconstrained lower bounds on $R_i$, because if we set $v_i/s = R_i(0)$, then we know that the earliest possible $g_i^*$ (with $v_i^*$ replacing $c_i$) is 0, and therefore the optimal unconstrained ordering time is non-negative. If $v_i^U$ induces a strictly positive $g_i$, then it must be decreased towards $v_i^*$ until either $g_i^* = 0$ or we find that $v_i^* = c_i$ (i.e., "value"="cost"). After calculating $v_i^U$, we obtain $V^L = \max\{0, s - \sum_{i=1,...,n} v_i^U\}$. If $v_i^U = c_i$ then $v_i^* = c_i$, so when we look for lower bounds on $v_i^*$ (for $i=1,...,n$) we can focus exclusively on those inputs for which $v_i^U > c_i$.

Because we cannot do better than order immediately,

$$SL_i^U = \min\left\{SL_i(0), 1 - \frac{c_i}{s}\right\}, \quad i=1,...,n$$

If $SL_i(0) = 0$ for some $i$, then $V^* = V^L = 0$. Else, we can identify a strictly positive new lower bound by generalizing Eq. 5. To this end, utilizing the observation that $V$ is analogous to $C$ and $v_i^*$ to $c_i$, note that

$$SL_i^* \geq \frac{V^*}{V^* + v_i^*} \geq \frac{V^L}{V^L + v_i^*} \geq \frac{V^L}{V^L + v_i^{U}} \quad \text{and} \quad SL_i^* \geq \frac{V^L}{s}$$

Therefore,

$$SL_i^* \geq SL_i^L = \max\left\{\frac{V^L}{V^L + v_i^{U}}, \frac{SL_i^U V^L}{SL_i^{US}}\right\}$$

We can also identify a lower bound on $v_i^*$, $v_i^L$, as follows,

$$SL_i(0) \geq \frac{V^L}{V^L + v_i^*} \rightarrow v_i^* \geq V^L \frac{1 - SL_i(0)}{SL_i(0)} \rightarrow v_i^L = \max\left\{c_i, V^L \frac{1 - SL_i(0)}{SL_i(0)}\right\}; \quad V^L > 0$$

where the condition prevents division by zero (recall that if $SL_i(0) = 0$ for any $i$ then $V^* = 0$).
Finally, to clarify the interpretation of $v_i^*$ as an economic value, look at the partial derivative of the Lagrangian by $v_i$. If $g_i^* = 0$ then $\lambda_i$ is the economic amount by which $c_i$ should be increased to set the partial derivative to zero. This yields

$$v_i^* = s \int_{G}^{\infty} f_i(t-g_i^*) \prod_{k \neq i} F_k(t-g_k^*) dt \quad i=1,\ldots,n$$

(6)

4. The Generalized Newsvendor Optimality Condition

From the start, [10,11] presented the ACM as a generalization of the newsvendor model, but they failed to see that the (well known and elegant) newsvendor result can be generalized too (except for the approximation $SL_i^* \approx SL_i^U$). [2] and [8]--their derivations of $SL^*$ notwithstanding--did not address the newsvendor model connection explicitly. With hindsight, however, their mathematical insight that setting Eq. 2 to zero yields $c_i = s \Pr\{g_i + Y_i \geq \max_{k=1,\ldots,n+1} \{g_k + Y_k\} \}$ could have been used to show that the ACM optimal result is actually an exact generalization of the newsvendor result for $n$ parallel stochastic inputs. In this section we discuss this exact generalization. For this purpose we treat the due date as an additional input. We focus on the criticality of input $i$ ($i=1,\ldots,n+1$), defined as the probability, $p_i$, that input $i$ will arrive last. Criticality, also known as criticality index, is a well known concept in project management [3; p. 277], where the critical path is defined as the longest chain of activities. $p_i$ and $R_i$ are related but not identical: $R_i$ depends only on $G-g_i$ while $p_i$ depends on all other ordering times too. The assembly is on time if and only if the due date is critical, so $p_{n+1}^* = SL^* = C/s = c_{n+1}/s$. We generalize this result for $p_i^*$ ($i=1,\ldots,n$), in a form that includes the constrained case.

**Theorem 1:** For $i=1,\ldots,n+1$, $p_i^* = v_i^*/s$ is a necessary and sufficient optimality condition for the ACM.
Proof: For i=1,...,n, necessity applies because the integral in Eq. 6 gives the criticality: $f_i$ measures the probability that input i will arrive at time x and $\prod_{k \neq i} F_k$ is the probability all the other inputs will have arrived before, making input i critical. For i=n+1, necessity applies because the sum of all criticalities must be 1. Sufficiency is assured because the model is convex: utilizing the convexity of the max function, [12] proved convexity for the unconstrained case, but the constraints are convex so our case is convex too. QED

The proof can be extended to discrete distributions by clarifying the definition of criticality: if two or more activities are critical at the same time, allocate the criticality among them in proportion to $c_i$. Recall that, for $i \leq n$, $v_i^* > c_i$ only if $g_i^* = 0$; so if $g_i^* > 0$, $p_i^* = c_i/s$. Thus the optimal criticality of unconstrained inputs is $c_i/s$ while constrained inputs can have higher criticalities, at the expense of the criticality of the due date, so $SL^* \leq C/s$.

Now consider management policies that set specific SL targets, denoted by $SL^T$; e.g., $SL^T = 95\%$. One interpretation is that this is equivalent to setting $C = \sum_{i=1}^{n} c_i SL^T/(1-\Sigma_{i=1}^{n} c_i)$; e.g., $SL^T = 95\%$ implies $C = 19 \sum_{i=1}^{n} c_i$. A more strict interpretation may call for iteratively increasing C until $SL^T$ can actually be achieved--which may or may not be possible. The latter is different from the first only in cases that are constrained for the first interpretation and essentially calls for $V^* = \sum_{i=1}^{n} v_i^* SL^T/(1-\Sigma_{i=1}^{n} v_i^*)$ where $v_i^*$ are determined such that $V^*/(V^*+\Sigma_{i=1}^{n} v_i^*) = SL^T$, and also implies a higher s. Under both interpretations, such policies indirectly determine the tardiness penalty. Often managers specify high SL targets without realizing that by this they are implicitly setting very high C levels (and even more so under the second interpretation). Studying our model may help convince managers to set C directly instead or, at least, settle for the first interpretation and do so explicitly.
5. **An Efficient Simulation-Based Solution**

For the unconstrained case, [2] showed that if we start a numerical search from \( g_i^L \) as a basis and optimize gates one by one iteratively, then \( g_i \) can only increase towards \( g_i^* \). To see this consider that increasing \( g_i \) increases \( p_i \), but this must decrease the other criticalities \( p_k \) (k=1,...,n+1; k≠i), so for k≤n, \( g_k \) will have to be increased iteratively. During the process, \( p_{n+1} \) converges (from above) to \( C/s \). In the constrained case, the major difference is that some inputs will not require adjustment at all. So we start at \( g_i^L = \text{Max} \{0, \text{arg}\{F_i(G-g_i)=1-c_i/s}\} \), and adjust the inputs iteratively to satisfy \( p_i = c_i/s \). We only adjust \( g_i \), however, if \( c_i/s \) strictly exceeds the current criticality.

As [2] briefly noted, for large \( n \) this approach can be implemented by simulation. We adopted the sample-path optimization by simulation approach [4], where we simulate \( \{Y_i\}_{i=1,...,n} \) \( r \) times and optimize for the resulting sample. In the limit, as \( r \rightarrow \infty \), the search result converges to the optimal solution almost surely (because the model is convex). In the next section we discuss the empirical quality of the results as a function of \( r \). Here, from a theoretical point of view, we note that if \( r < s/c_i \) then it is not sufficient to measure the criticality of input \( i \) appropriately. Similarly, adjustments that miss the ideal criticality by less than \( 1/r \) should be considered exact with a given \( r \). To resolve this issue we can first round the desired \( p_i^R \) values to appropriate \( p_i^R \) values, such that \( p_i^R r \) is integer. (Good \( p_i^R \) values may be obtained by minimizing \( \Sigma c_i |p_i^R - p_i^*| \).) If \( p_i < p_i^R \), we increase \( g_i \) until \( p_i = p_i^R \) is obtained. As a rule of thumb, to control both the expected rounding error and the sampling error, we suggest setting \( r \geq 3 \max\{s/c_i\} \). Since \( \max\{s/c_i\} \geq n \), the required sample size is at least \( o(n) \) and we may expect computational complexity that is polynomial in \( n \). If this complexity becomes excessive, the use of blocks may be beneficial. Denote the criticality of block \( k \) by \( P_k \), calculated as the sum of all \( p_i \) within the block, and similarly
\( P^*_k \) is the sum of the relevant \( p^*_i \), and we can also obtain \( P^*_i \) for the block similarly to the single input case. Unconstrained inputs with small optimal criticalities can then be combined to blocks. Using such blocks, we need fewer repetitions. However, small inputs with \( v_i^U = c_i < v_i^L \) and \( p_i^R = 0 \) cannot be resolved this way. For safety, we may have to specify \( g_i = 0 \) for them even if their criticality (in the sample) drops to zero. Similar blocks were used by [13] for a slightly different purpose and the numerical experience there showed a very small detrimental impact on the objective function. One reason for this robustness is that, for unconstrained inputs (but not for constrained ones), the objective function is flat near optimum. In addition, small \( c_i \) inputs are inherently less important.

Finally, note that blocks allow a hierarchical approach to the solution: First we adjust few major blocks correctly, then we adjust inputs within the blocks. This can help devise effective heuristics for very large systems.

6. Observations Based on Numerical Evidence

It is well known that the use of \( SL^U_i \) (i.e., \( g^L_i \)) as a heuristic is very effective for the unconstrained case with high \( SL^* \). For example, [2] reported results within 0.45% of optimum. [9] reports good results with a similar heuristic where the \( SL^U_i \) values are decreased by a constant factor to satisfy \( \Pi SL_i = SL^* = C/s \) (as per Eq. 4). But the performance of such heuristics deteriorates for low service level and the latter does not even apply for the constrained case. In such instances it is easy to construct examples where the optimal solution involves significant postponement of unconstrained activities leading to high savings. Thus, our main practical challenge is to solve low \( SL^* \) cases with many inputs (where \( SL^* \) may be low due to tight constraints or low tardiness penalty). Our current approach seems to provide an adequate answer to these needs.
To verify this, we constructed a basic spreadsheet application capable of handling up to 20 inputs with up to 8000 repetitions in the sample. The application demonstrated that problems of this size are easy to solve, and with dedicated software much larger instances should be easy too. We also used the application for a preliminary study of the effect of $r$ on the quality of the solution. Suppose we consider two independent samples, 1 and 2. Let $Z_j(g_k)$ denote the objective function value of sample $j$ ($j=1,2$) with the optimal vector of ordering times based on analyzing sample $k$ ($k=1,2$). Similarly, $Z_j(g^*)$ is the respective objective function value when the globally optimal (and unknown) $g^*$ vector is used. Because $g_k$ minimizes $Z$ for sample $k$, $Z_k(g_k) \leq Z_k(g^*)$ and $Z_k(g_k) \leq Z_k(g_{3-k})$. It follows that $Z_k(g_{3-k})-Z_k(g_k) \geq Z_k(g_{3-k})-Z_k(g^*)$. The left hand side of this inequality is observable, so we obtain an upper bound for the cost of using $g_{3-k}$ (the "wrong" ordering time vector) for one other sample with the same $r$. Table 1 (where $\Delta_k$ represents $Z_k(g_{3-k})-Z_k(g_k)$) reports the results of one such experiment with various $r$ values. We selected all the parameters randomly, without any attempt to obtain a particularly interesting result, but we did adjust SL to a target: $SL^T=75\%$. As a result, eight of the twenty inputs had $g_{1_i}^L=0$; of these, six were clearly constrained (with $v_{1_i}^L>c_i$), one remained constrained, and one required adjustment during the search. Because the example is constrained, $SL^*<75\%$ was obtained: 59\% in this case, with $SL^L=57\%$. In this example we obtained $\max_i(s/c_i)=320$, so using our recommended minimal sample size--$r \geq 3\max_i(s/c_i)$--would require about 1000 repetitions (we tested for $r=250, 500, 1000, 2000,$ and 4000). The difference can also be compared to the standard deviation of $Z$, $\sigma$--which is readily estimable by the spreadsheet results. In the table we normalized these comparisons by using the standard error, $\sigma_e$ (i.e., $\sigma/\sqrt{r}$), instead of $\sigma$. However, when compared to $\sigma$ (about 1020 for this example when estimated by 8000 repetitions) the differences were invariably negligible even for as few
as 250 repetitions--so our rule of thumb is probably conservative. (In the table, $\sigma_e$ is based on $\hat{\sigma}=1020$: the figures might be up to about 10% different--and in this case usually lower--if based on the $\hat{\sigma}$ estimates achieved in each pair of runs.) Of course, the difference between $g_1$ and $g_2$ is stochastic; e.g., in the table, the $r=1000$ case seems to involve similar gates. Nonetheless, the experiment as a whole suggests that the approach provides a good approximation. Notably, by increasing $SL^T$ we would obtain more constrained inputs, leading to nearly optimal results even with very low $r$. If $F_1(x)<1$ for any $x<\infty$, then in the limit, as $SL^T \to 1$, $g \to 0$. But we don’t need a large sample to adjust $g_1$ to 0.

<table>
<thead>
<tr>
<th>$r$</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_1(g_2)$</td>
<td>21092.11</td>
<td>21175.68</td>
<td>21179.25</td>
<td>21155.98</td>
<td>21158.47</td>
</tr>
<tr>
<td>$Z_1(g_1)$</td>
<td>21064.74</td>
<td>21149.71</td>
<td>21177.27</td>
<td>21145.87</td>
<td>21153.92</td>
</tr>
<tr>
<td>$\Delta_1/\sigma_e$</td>
<td>0.42</td>
<td>0.57</td>
<td>0.06</td>
<td>0.44</td>
<td>0.28</td>
</tr>
<tr>
<td>$Z_2(g_1)$</td>
<td>21254.25</td>
<td>21210.29</td>
<td>21115.87</td>
<td>21165.70</td>
<td>21152.68</td>
</tr>
<tr>
<td>$Z_2(g_2)$</td>
<td>21218.26</td>
<td>21198.15</td>
<td>21112.69</td>
<td>21157.70</td>
<td>21148.98</td>
</tr>
<tr>
<td>$\Delta_2/\sigma_e$</td>
<td>0.56</td>
<td>0.27</td>
<td>0.10</td>
<td>0.35</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Table 1: Evaluating the consequences of various $r$ choices

Finally, spreadsheet applications such as ours are not geared to measure computation time well. However, using Excel® Goal Seek functionality to satisfy the optimality condition for any gate was practically instantaneous. In contrast, using the Solver functionality to seek the local minimum took several seconds per gate. As expected, both approaches yielded the same final result (subject to rounding error).
7. Conclusion

We consolidated and enhanced the assembly coordination model, which was originally introduced independently by three sets of authors. One of our theoretical contribution is just a re-interpretation of the previous optimality condition as an exact generalization of the newsvendor model for n inputs: the criticality of each input should be proportional to its marginal economic value. Another contribution is Lagrange analysis of the constrained version of the ACM, showing that the newsvendor generalization still applies. We presented new bounds and provided initial numerical results by a simulation-based approach. This numerical experience indicates that large constrained instances can be solved effectively.

Acknowledgements: The first author is grateful to Candace A. Yano for substantial help during the initial preparation of this paper. The revision of this paper was supported by the Engineering Research Center at American University of Armenia, Yerevan, while the first author was visiting there in 2005.

References


