On-line Non-stationary Inventory Control using Champion Competition

Jianfeng Mao
Division of Systems Engineering and Management,
School of Mechanical Aerospace Engineering,
Nanyang Technological University, Singapore 639798,
jfmao@ntu.edu.sg

The commonly adopted assumption of stationary demands cannot actually reflect fluctuating demands and will weaken solution effectiveness in real practice. We consider an On-line Non-stationary Inventory Control Problem (ONICP), in which no specific assumption is imposed on demands and their probability distributions are allowed to vary over periods and correlate with each other. The nature of non-stationary demands disables the optimality of static \((s, S)\) policies and the applicability of its corresponding algorithms. The ONICP becomes computationally intractable by using general Simulation-based Optimization (SO) methods, especially under an on-line decision-making environment with no luxury of time and computing resources to afford the huge computational burden. We develop a new SO method, termed “Champion Competition” (CC), which provides a different framework and bypasses the time-consuming sample average routine adopted in general SO methods. An alternate type of optimal solution, termed “Champion Solution”, is pursued in the CC framework, which coincides the traditional optimality sense under certain conditions and serves as a near-optimal solution for general cases. The CC can reduce the complexity of general SO methods by orders of magnitude in solving a class of SO problems, including the ONICP. A polynomial algorithm, termed “Renewal Cycle Algorithm” (RCA), is further developed to fulfill an important procedure of the CC framework in solving this ONICP. Numerical examples are included to demonstrate the performance of the CC framework with the RCA embedded.

Key words: Simulation-based Optimization, Non-Stationary, Inventory Control, On-line.

1. Introduction

Demands are affected by all kinds of special events in real practice such as weather changes, public holidays, new product promotions, financial crisis and mega conventions, which result in increasing or decreasing or just fluctuating demands. Although the commonly adopted assumption of \(i.i.d.\) (independent and identical distributed) demands works well for the case where products are in a mature stage without disturbing events (Axsäter 2006), it cannot capture the nature of non-stationary demands in general cases. To improve the effectiveness in practice, we should take into account this non-stationary nature in inventory control.
The \((s, S)\) policy is one of the most popular inventory control policies, but the optimality of its static version requires the assumption of \textit{i.i.d.} demands in many inventory systems \cite{scarf1959, iglehart1963, veinott1966, zheng1991} and \cite{beyer1999}. Although the optimality of this two-threshold type policy can be extended to several cases with non-stationary demands in \cite{zipkin2000} and \cite{gallego2001}, the two optimal thresholds \((s, S)\) are no more static, i.e., the optimal policy is \((s_i, S_i)\) that may vary in different period \(i\).

Finding the optimal \((s, S)\) policy is complicated even for \textit{i.i.d.} demands. Some efficient algorithms have been derived in \cite{veinott1966}, \cite{zheng1991} and \cite{fu1994}. Besides the assumption of \textit{i.i.d.} demands, some extra assumptions might also be needed, such as integer-valued demands \cite{zheng1991}.

It becomes much more complicated for cases of non-stationary demands. Some efforts have been made towards the non-stationary inventory control with fixed setup cost \cite{askin1981}, \cite{bookbinder1988}, \cite{bollapragada1999}, \cite{hua2009}. Most of them still require the assumption of mutually independent demands over periods. \cite{askin1981} proposed a heuristics similar to Silver-Meal heuristics \cite{silver1973} and need to explicitly compute the probability distributions of cumulative demands, which is not plausible for demands with complicated patterns. In \cite{bookbinder1988} and \cite{hua2009}, static-dynamic uncertainty approaches were developed for cases of independent demands. \cite{bollapragada1999} approximated non-stationary cases by averaging demands over periods and then computed a stationary policy by utilizing the algorithm in \cite{zheng1991}. Moreover, they aimed at computing the threshold type \((s_i, S_i)\) policy, which may not be computationally promising. Due to non-stationary demands, \((s_i, S_i)\) generally varies over time, which cannot be reused in following periods after one-time computation like the static \((s, S)\) policy.

In this paper, we consider an On-line Non-stationary Inventory Control Problem (\textbf{ONICP}) without imposing any specific assumptions on demands as long as they can be randomly generated through simulations. Since demands can be continuous random variables or discrete random variables with many choices, it is typically impossible to calculate expected values of quantities of interests in closed form and obtain an equivalent deterministic optimization model. Simulation-based Optimization (\textbf{SO}) methods need to be invoked to estimate expected values through sample average approximations to assess solution performances, which are generally time-consuming. As computational efficiency is critical for on-line decision making, general \textbf{SO} methods may not be applicable under a real-time environment. This challenge motivates us to develop a new \textbf{SO} method, called “Champion Competition” (\textbf{CC}), which provides a different framework and bypasses the time-consuming sample average routine in solution assessment. An alternate type of optimal solution is pursued in the \textbf{CC}, termed “Champion Solution” (\textbf{C-Sol}), which coincides the traditional
optimality sense under certain conditions and serves as a near-optimal solution for general cases. The CC can reduce the computational complexity by orders of magnitude in solving a class of SO problems, including this ONICP. To further improve the computational efficiency, a polynomial algorithm, termed “Renewal Cycle Algorithm” (RCA), is developed to fulfill an important procedure of the CC framework in solving the ONICP.

In the rest of paper, we introduce the C-Sol with an alternate optimality sense and develop the CC framework in Section II after reviewing SO methods. We then formulate the ONICP and verify the applicability of CC in Section III. The RCA is further developed based on a structural property identified over single sample-path of the ONICP in Section IV. Numerical results are given in Section V to demonstrate the performance of the CC with the RCA embedded in solving the ONICP. We close with conclusions in Section VI.

2. Champion Competition

2.1. Related Literatures

A general Simulation-based Optimization (SO) problem can be formulated as

$$\min_{u \in \Phi} E[J(u, \omega)]$$  \hspace{1cm} (1)

where $u$ is the decision variable, $\Phi$ is the feasible space of $u$ and $\omega$ represents a sample-path. A closed form of the expected cost function of SO problems is typically impossible to be derived due to infinite or tremendous number of sample-paths. In this ONICP, $\omega$ is a realization of a sequence of demands, which can be continuous random variables or discrete random variables with plenty of choices. It is impossible to express the corresponding expected operating cost in a closed form. Thus, SO methods become necessary in solving this type of problems.

In general, SO methods include two major operations: (i) Assess solutions by averaging evaluations over multiple sample-paths and (ii) Explore new solutions within certain areas based on performance assessments in (i). Evaluations are implemented per solution per sample-path. The total complexity can be measured by the computational efforts for all evaluations, which can be approximated as $M \cdot I \cdot C$, where $M$ is the number of sample-paths generated for assessing a solution, $I$ is the total number of solutions explored and $C$ is the complexity of each evaluation. ($M$ is not necessarily a constant throughout the entire process.) A more accurate assessment requires a bigger $M$ and a better solution needs a greater $I$ for exploring more solutions. Both $M$ and $I$ can be very large in solving a general SO problem. Moreover, an evaluation does not necessarily just generate random numbers and calculate function values. It may involve solving a deterministic optimization problem, such as the ONICP considered in this paper. Then the complexity of each
On-line Non-stationary Inventory Control using Champion Competition

4

evaluation becomes $C(N)$, where $N$ is the dimensionality of the involved deterministic optimization problem and the total complexity of SO methods becomes $M \cdot I \cdot C(N)$. Therefore, solving a SO problem is generally time-consuming.

Many SO methods have been developed over the past few decades. Computational efforts can be saved by either using less $M$ in assessment, such as Ordinal Optimization (OO) (Ho et al. 2008) and Optimal Computing Budget Allocation (OCBA) (Chen and Lee 2011), or by reducing $I$ in search, such as Nested Partition (NP) (Shi and Olafsson 2000) and COMPASS (Hong and Nelson 2006), or by both ways, such as Perturbation Analysis (PA) (Ho and Cao 1991) and Retrospective Optimization (RO) (Chen 1994) (Jin 1998).

OO and OCBA embrace the idea of ranking solutions rather than accurately estimating expected performance values of a solution. An optimal solution or a top $\alpha\%$ solution can be achieved by ranking solutions at a certain confidence level, which consumes less $M$ than accurately estimating expected performance values. OO and OCBA can reduce computational time by smartly allocating necessary $M$ to solutions in assessment.

NP and COMPASS provide efficient search strategies, in which a feasible space can be iteratively divided by certain structures, termed Nested Partition and Most Promising Area respectively, and the majority of search efforts are gradually narrowed down to a small region that likely contains an optimal solution. These methods can reduce computational time by exploring less $I$ solutions. The two groups of aforementioned methods can be surely combined together to get a better computational performance by less $M$ and $I$ at the same time.

PA generally aims at an Infinitesimal Perturbation Analysis (IPA) estimator, that is, an unbiased estimator of gradients of expected performance functions, which can lead to a local optimal solution with less $M$ and $I$. The assumption of stationary process is commonly required in PA to support unbiasedness so that the accuracy of IPA estimators can be improved along a single sample-path over time. IPA estimators may not exist for some cases.

RO is essentially a sample-average approximation method, in which a set of sample-average problems with asymptotically increasing $M$ is sequentially generated and solved within decreasing error tolerance interval. To implement the RO, the five main factors need to be determined: (1) Method for sample-average approximation problems (2) Rule for decreasing error tolerance (3) Rule for increasing number of replications (4) Rule for the $i$th retrospective estimate based on all the retrospective solutions derived before (5) Termination rule. These factors require coordinated fine-tuning for an efficient implementation.

These SO methods generally employ the sample average routine to assess solutions, that is, evaluating quantities of interests over many sample-paths and averaging these evaluations to estimate the expected performance of a solution. Thus, their complexity can still be roughly approximated
as $M \cdot I \cdot C$ with either less $M$ or less $I$ or both, which may not be efficient enough for an on-line environment with on luxury of time and computing resources. It becomes more serious if each evaluation requires solving a deterministic optimization problem with a high dimensionality of $N'$, which implies that $C$ becomes a high complexity of $C(N')$ for each evaluation, such as the ONICP here.

The “Champion Competition” (CC) framework revealed later can bypass the time-consuming sample average routine mentioned above. It can reduce the complexity to $M \cdot C(N'')$, where there is no need to employ any search strategy to explore $I$ solutions and $N''$ is much less than $N'$ required for evaluations in general SO methods. Before that, we will first introduce the “Champion Solution”, an alternate type of optimal solution pursued in the CC framework.

2.2. Champion Solution

Definition 1 The Champion Solution (C-Sol) of (1) is the solution $u^c$ such that

$$\Pr [J(u^c, \omega) \leq J(u, \omega)] \geq 50\%, \quad \text{for any } u \in \Phi.$$ 

2.2.1. Interpretation of C-Sol The C-Sol is essentially the solution that can be better than any other feasible solutions with a higher probability. The NBA Finals can be used as an example to interpret C-Sol, in which the champion (C-Sol) will be determined from two teams (solutions) based on the results in 7 games (sample-paths). The champion (C-Sol) is the team that wins more games (performs better in more sample-paths). If there are infinite number of games (sample-paths), then the C-Sol is the team with 50% more winning ratio (the probability of performing better).

Now what if we have two more solutions? We can adopt the example of president election that was originally used to interpret the Arrow’s Impossibility Theorem in social choice theory [Arrow 1963]. Imagine we have three candidates (solutions) $A$, $B$ and $C$. Each voter (sample-path) will rank the three candidates according to his or her own preference. Now, we randomly pick three voters’ preference lists (sample-paths) as shown in the following table, where $A \succ B$ means $A$ is preferred over $B$, and try to determine the president based on that. We have $\Pr[A \succ B] = 33\%$, $\Pr[A \succ C] = 33\%$ for Candidate $A$, $\Pr[B \succ A] = 67\%$, $\Pr[B \succ C] = 67\%$ for Candidate $B$, and $\Pr[C \succ A] = 67\%$, $\Pr[C \succ B] = 33\%$ for Candidate $C$ based on the opinions of the three voters. Clearly, $B$ should be the president (C-Sol) because $B$ gets a higher preference (performs better) than all the other candidates (solutions) from majority of voters (sample-paths).
2.3. Optimality in Probability

Generally, the C-Sol may not minimize the expected cost in (1). It possesses an alternate optimality sense to the usual “Optimality in Expectation” (which may in fact not be the best choice in some applications), termed “Optimality in Probability”. In the sense of Optimality in Probability, a solution is regarded better than the other if it can perform better than the other with a higher probability no matter how much better over each sample-path. In the example of NBA Finals, the traditional Optimality in Expectation favors the team with a higher average score of multiple games. Imagine the NBA Finals finished in 6 games and the results are shown in the following table. Although Team A is the champion (C-Sol), Team B is more favorable according to the Optimality in Expectation because Team B has a higher average score than Team A.

|       | Game 1 | Game 2 | Game 3 | Game 4 | Game 5 | Game 6 |
|-------|--------|--------|--------|--------|--------|--------|
| Team A | 107    | 103    | 60     | 106    | 66     | 98     |
| Team B | 100    | 97     | 103    | 104    | 101    | 95     |

The C-Sol coincides the optimal solution in the sense of Optimality in Expectation under the following “Non-singularity Condition”.

Non-singularity Condition (NSC):

\[
\Pr [J(u', \omega) \leq J(u'', \omega)] \geq 0.5 \implies E[J(u', \omega)] \leq E[J(u'', \omega)], \quad \forall u', u'' \in \Phi
\]

The interpretation of NSC is that if \( u' \) is more likely better than \( u'' \) (in the sense of resulting in lower cost), then the expected cost under \( u' \) will be lower than the one under \( u'' \). This is consistent with common sense in that any solution \( A \) more likely better than \( B \) should result in \( A \)'s expected performance being better than \( B \)'s. Only “singularities” such as \( J(u', \omega) \gg J(u'', \omega) \) with an unusually low probability for some \((u', u'')\) can affect the corresponding expectations so that this condition may be violated. It is straightforward to verify this NSC for several common cases; for example, consider \( \min_x E(x - Y)^2 \), where \( Y \) is a uniform random variable over \([a, b]\). The optimal solution \((a + b)/2\) satisfies the NSC. For general cases, the C-Sol can serve as a near-optimal solution if the corresponding problem is not that singular.

2.3.1. Existence of C-Sol

Obviously, a C-Sol always exists if there are only two feasible solutions. However, we may not have a C-Sol even for a case with only three feasible solutions. Recall the example of president election. Voter 3 would like to change his or her preference as shown in the following table. We have that \( \Pr[A \succ B] = 67\% \), \( \Pr[A \succ C] = 33\% \) for \( A \), \( \Pr[B \succ A] = 33\% \), \( \Pr[B \succ C] = 67\% \) for \( B \), and \( \Pr[C \succ A] = 67\% \), \( \Pr[C \succ B] = 33\% \) for \( C \). This time no candidate can
be elected as the president (C-Sol) because no one can be preferred over all the other candidates (solutions) from majority of voters (sample-paths).

The case above is also kind-of a singular case, in which solutions’ performances are purely random over different sample-paths. Such a chaotic pattern is not common in real practice. For instance, a student’s performance can be measured by exam scores. Generally, if a student is good, then the student is not supposed to obtain purely random scores in exams and should get above-average scores most of the time.

In the following, we will identify a sufficient condition for the existence of C-Sol by a constructive proof, which specifies a class of SO problems that can be solved by the CC framework revealed later. It should be noted that it is only a sufficient condition and the idea of C-Sol can potentially work for a wider class of SO problems.

Before that, we need to define \( \omega \)-Prob, \( \omega \)-Sol and \( \omega \)-Med for a general SO problem (1), where \( \omega \) stands for a single sample-path.

**Definition 2** An \( \omega \)-Prob of (1) is the deterministic optimization problem below defined over a single sample-path \( \omega \),

\[
\min_{u \in \Phi} J(u, \omega)
\]

**Definition 3** An \( \omega \)-Sol of (1) is the optimal solution of an \( \omega \)-Prob of (1), that is, the solution \( u^\omega \) such that

\[
u^\omega = \arg \min_{u \in \Phi} J(u, \omega).
\]

**Definition 4** The \( \omega \)-Med of (1) is the median of \( \omega \)-Sols of (1), that is, the solution \( u^m \) such that

\[
\Pr[u^\omega \leq u^m] \geq 0.5 \quad \text{and} \quad \Pr[u^\omega \geq u^m] \geq 0.5
\]

where \( u^\omega \) is an \( \omega \)-Sol of (1) for a single sample-path \( \omega \).

\( u^\omega \) is essentially a random variable. The two probabilities above are \( \omega \)-Sol’s cumulative distribution function (cdf) and complementary cumulative distribution function (ccdf) respectively. It should be noted that both probabilities can be strictly more than 0.5 at the same time if \( u^\omega \) is not a continuous random variable.

**Theorem 1** If \( J(u, \omega) \) in (1) is a scalar unimodal function in \( u \) for any \( \omega \), then the \( \omega \)-Med is a C-Sol of (1).
Proof: Since $J(u, \omega)$ is a scalar unimodal function in $u$ for any $\omega$, we have

$$J(u', \omega) \leq J(u'', \omega), \quad \text{for any } u'' < u' < u^\omega,$$

(2)

and

$$J(u', \omega) \leq J(u'', \omega), \quad \text{for any } u^\omega < u' < u''.$$

(3)

Assume $u^m$ is the $\omega$-Med. For any solution $u > u^m$, we have

$$\Pr[J(u^m, \omega) \leq J(u, \omega)] = \Pr[J(u^m, \omega) \leq J(u, \omega)|u^\omega \leq u^m]\Pr[u^\omega \leq u^m]$$

$$+ \Pr[J(u^m, \omega) \leq J(u, \omega)|u^\omega > u^m]\Pr[u^\omega > u^m]$$

(4)

From (3), if $u > u^m$ and $u^m \geq u^\omega$, then $J(u^m, \omega) \leq J(u, \omega)$, which implies that

$$\Pr[J(u^m, \omega) \leq J(u, \omega)|u^\omega \leq u^m] = 1$$

(5)

Since $u^m$ is the $\omega$-Med, we have $\Pr[u^\omega \leq u^m] \geq 0.5$. Combining it with (4) and (5), we have

$$\Pr[J(u^m, \omega) \leq J(u, \omega)] \geq 0.5 + \Pr[J(u^m, \omega) \leq J(u, \omega)|u^\omega > u^m]\Pr[u^\omega > u^m] \geq 0.5$$

The case of $u < u^m$ can be similarly proved. Therefore, $u^m$ satisfies the definition of $C$-Sol

$$\Pr[J(u^m, \omega) \leq J(u, \omega)] \geq 0.5, \quad \text{for any } u \in \Phi.$$

that is, $u^m$ is a $C$-Sol of (1). □

2.4. Convergence Rate of $\omega$-Med

The closed form of the cdf and ccdf of $u^\omega$ cannot be derived in general cases, but they can be approximately constructed through simulations. Assume $M$ sample-paths $\omega_1, \ldots, \omega_M$ are randomly generated and $u^{\omega_1}, \ldots, u^{\omega_M}$ are their corresponding $\omega$-Sols. The cdf and ccdf of $u^\omega$ can be estimated through the two following functions respectively

$$G_M(u) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}(u^{\omega_j} \leq u), \quad \bar{G}_M(u) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}(u^{\omega_j} \geq u),$$

where $\mathbf{1}(\cdot)$ is an indicator function.

Then, the $\omega$-Med $u^m$ can be approximated by finding a solution $\hat{u}^m$ that satisfies $G_M(\hat{u}^m) \geq 0.5$ and $\bar{G}_M(\hat{u}^m) \geq 0.5$, which can be depicted in the example of Figure 1.

For any given $u$, based on the strong law of large number, $G_M(u)$ and $\bar{G}_M(u)$ converge to $\Pr[u^\omega \leq u]$ and $\Pr[u^\omega \geq u]$ respectively with probability 1 as $M$ approaches to infinity. Thus, $\hat{u}^m$ also converges to $u^m$ w.p.1 as $M \to +\infty$. Furthermore, we can show that $\hat{u}^m$ approaches $u^m$ exponentially fast as $M$ increases by the following two theorems for two possible cases respectively. Exponential convergence rate is computationally promising due to less $M$ to approximate the $\omega$-Med.
Theorem 2 considers the case of Pr(\(u^\omega = u^m\)) > 0, which corresponds to the case that \(u^\omega\) is a discrete random variable. Theorem 3 considers the case of Pr(\(u^\omega = u^m\)) = 0, which corresponds to the case that \(u^\omega\) is a continuous random variable. Theorem 2 presents a stronger sense of convergence rate than Theorem 3, which implies that \(\omega\)-Med approximation in discrete cases is more efficient (by less \(M\)) than the one in continuous case.

**Theorem 2** If \(\Pr(u^\omega = u^m) > 0\), then there always exists some constant \(C\) such that

\[
\Pr[\hat{u}^m = u^m] \geq 1 - 2e^{-CM}
\]

*Proof:* Without loss of generality, assume \(\Pr(u^\omega = u^m) = c > 0\), \(\Pr(u^\omega < u^m) = p_1\) and \(\Pr(u^\omega > u^m) = p_2\). From the definition of \(\omega\)-Med, we have \(p_1 + c \geq 0.5\) and \(p_2 + c \geq 0.5\). Combining it with \(p_1 + c + p_2 = 1\) and \(c > 0\), we have

\[
p_1 < 0.5, \quad p_2 < 0.5.
\]

The event \([\hat{u}^m = u^m]\) is equivalent to the event \([\hat{G}_M(u^m) \geq 0.5 and \bar{G}_M(u^m) \geq 0.5]\), which can be further equivalently reduced to \([L_M(\hat{u}^m) < 0.5 and \bar{L}_M(\hat{u}^m) < 0.5]\), where

\[
L_M(u) = \frac{1}{M} \sum_{j=1}^{M} 1(u^\omega_j < u), \quad \bar{L}_M(u) = \frac{1}{M} \sum_{j=1}^{M} 1(u^\omega_j > u).
\]

Therefore, we have

\[
\Pr[\hat{u}^m = u^m] = \Pr[L_M(\hat{u}^m) < 0.5 and \bar{L}_M(\hat{u}^m) < 0.5]
\]

\[
= 1 - \Pr[L_M(u^m) > 0.5 or \bar{L}_M(u^m) > 0.5]
\]

\[
= 1 - (\Pr[L_M(u^m) > 0.5] + \Pr[\bar{L}_M(u^m) > 0.5])
\]

Clearly, \(1(u^\omega_j < u^m), j = 1, ..., M\) are i.i.d. 0-1 random variables and \(E[1(u^\omega_j < u^m)] = p_1\). Then based on Chernoff-Hoeffding Theorem (Hoeffding March 1963), we have for any \(\epsilon > 0\)

\[
\Pr[L_M(u^m) \geq p_1 + \epsilon] \leq e^{-D(p_1 + \epsilon||p_1)M}
\]
where \( D(x||y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y} \). Similarly, we can also have

\[
\Pr[\tilde{L}_M(u^m) \geq p_2 + \epsilon] \leq e^{-D(p_2 + \epsilon||p_2)M}
\]

Combining the two inequalities above with \( p_1 < 0.5 \) and \( p_2 < 0.5 \), we can further have

\[
\Pr[L_M(u^m) > 0.5] \leq e^{-D(0.5||p_1)M}
\]

\[
\Pr[\tilde{L}_M(u^m) > 0.5] \leq e^{-D(0.5||p_2)M}
\]

Combining them with (6), we can finally have

\[
\Pr[u^m = u^m] \geq 1 - e^{-D(0.5||p_1)M} - e^{-D(0.5||p_2)M} \geq 1 - 2e^{-CM}
\]

where \( C = \min(D(0.5||p_1), D(0.5||p_2)) \) \( \square \)

**Theorem 3** If \( \Pr(u^\omega = u^m) = 0 \), then for any \( \epsilon > 0 \), there always exists \( C > 0 \) such that

\[
\Pr[|G_M(u^m) - 0.5| < \epsilon] \geq 1 - 2e^{-CM}.
\]

**Proof:** From \( \Pr(u^\omega = u^m) = 0 \) and the definition of \( u^m \), we have

\[
\Pr[u^\omega \leq u^m] = 1 - \Pr[u^\omega \geq u^m] = 0.5
\]

which implies that

\[
E[G_M(u^m)] = 0.5
\]

Since \( \mathbf{1}(u^\omega_j \leq u^m), j = 1, ..., M \) are i.i.d. 0-1 random variables and \( E[\mathbf{1}(u^\omega_j < u^m)] = 0.5 \), based on Chernoff-Hoeffding Theorem (Hoeffding March 1963), we have for any \( \epsilon > 0 \)

\[
\Pr[G_M(u^m) \geq 0.5 + \epsilon] \leq e^{-D(0.5 + \epsilon||0.5)M} \quad \text{and} \quad \Pr[G_M(u^m) \leq 0.5 - \epsilon] \leq e^{-D(0.5 - \epsilon||0.5)M}
\]

where \( D(x||y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y} \). Therefore, we have

\[
\Pr[|G_M(u^m) - 0.5| < \epsilon] = 1 - \Pr[G_M(u^m) \geq 0.5 + \epsilon] - \Pr[G_M(u^m) \leq 0.5 - \epsilon] \geq 1 - e^{-D(0.5 + \epsilon||0.5)M} - e^{-D(0.5 - \epsilon||0.5)M} \geq 1 - 2e^{-CM}.
\]

where \( C = \min(D(0.5 + \epsilon||0.5), D(0.5 - \epsilon||0.5)) \). \( \square \)
Table 1  The Champion Competition Framework

| Step  | Description                                                                 |
|-------|------------------------------------------------------------------------------|
| Step 1| Verify the unimodality of $J(u, \omega)$;                                  |
| Step 2| Randomly generate $M$ sample-paths $\omega^1, ..., \omega^M$;               |
| Step 3| Solve $M$ $\omega$-Probs $\min_{u \in \Phi} J(u, \omega_1)$, $\ldots$, $\min_{u \in \Phi} J(u, \omega_M)$ to obtain their $\omega$-Sols $u^\omega_1, ..., u^\omega_M$ respectively; |
| Step 4| Sort $u^\omega_1, ..., u^\omega_M$ to construct the cdf $G_M(u)$ and ccdf $\bar{G}_M(u)$; |
| Step 5| A $C$-Sol can be approximated by the solution $\hat{u}_m$ that satisfies $G_M(\hat{u}_m) \geq 0.5$ and $\bar{G}_M(\hat{u}_m) \geq 0.5$. |

2.5. Champion Competition Framework

For a class of SO problems with a scalar unimodal $J(u, \omega)$, a $C$-Sol can be guaranteed and efficiently derived by computing $\omega$-Med based on Theorem 1, 2 and 3. We can finally develop the Champion Competition (CC) framework in Table 1.

As mentioned before, the CC framework above involves no sample average routine to assess solutions and no specific search strategy to explore solutions. Only $M$ $\omega$-Probs are required to solve, each of which is a deterministic optimization problem defined over a single sample-path.

The applicability of the CC framework relies on the unimodality of $J(u, \omega)$ so far, which can be potentially extended if the sufficient condition in Theorem 1 can be relaxed. Once the applicability is verified in Step 1, the CC framework can be easily implemented without using inefficient trial-and-error experiments to tune parameters for a better computational performance. Step 2, 4 and 5 are independent of problems and can be completed in a low complexity. Only Step 3 is problem-specific because different SO problems possess their own specific $\omega$-Probs. As Step 3 normally dominates the complexity of the CC framework, an efficient algorithm for solving $\omega$-Probs will definitely further enhance the overall computational efficiency. Since $\omega$-Prob is a deterministic optimization problem defined over just a single sample-path, more promising structural properties can be explored and exploited to develop an highly efficient solver. (Although sample-average approximation problems defined over multiple sample-paths are also deterministic optimization problems, the structural properties identified over a single sample-path normally becomes invalid and cannot be utilized. Some general deterministic optimization solvers might have to be employed to solve sample-average approximation problems, which compromises the computational efficiency.)

From above, the majority efforts of implementing the CC framework are focused on the applicability verification in Step 1 and the development of $\omega$-Prob solver in Step 3. In the following sections, we will apply the CC framework to the ONICP, in which the applicability is verified in Section 3 and the $\omega$-Prob solver (Renewal Cycle Algorithm) is developed in Section 4.
3. On-line Non-stationary Inventory Control Problem

We will formulate the On-line Non-stationary Inventory Control Problem (ONICP), in which full backlogging and periodic review are adopted. To avoid distraction, the factors of random yield ratio and lead time is temporarily ignored in this section, which will be recovered later.

3.1. Periodic Review in On-line Inventory Control

The following notations are commonly adopted in literatures:

- $x_i$: inventory level in period $i$;
- $d_i$: demand in period $i$;
- $u_i$: order quantity in period $i$;
- $h$: holding cost rate for inventory;
- $p$: penalty cost rate for backlog;
- $K$: setup cost.

A typical inventory control process can be depicted in Figure 2. In each period $i$, we will first order $u_i$ items ($u_i = 0$ means no order is placed in period $i$). Then the demand $d_i$ appears. $x_i$ is the inventory level after that. The cost in period $i$ is calculated based on maintenance cost caused by $x_i$, which can be holding or shortage cost. On-line order decisions will be sequentially made at the beginning of every period, that is, decision points in this ONICP. We look ahead $N$ (or infinite) periods. The on-line order decision is to determine $u_1$, the order quantity for the immediate period that can minimize the expected operating cost within these future $N$ (or infinite) periods.

![Figure 2 On-line Inventory Control Process](image)

The only exact information available at “Now” moment is the initial inventory level $x_0$. The demands $d_1, ..., d_N$ in look-ahead window can be randomly generated through simulations even though they are non-stationary. A sample-path $\omega$ is defined as a realization of demands $\{d_1, ..., d_N\}$ in this ONICP.
3.2. ONICP Formulation

We will aim at an optimal on-line order decision that can minimize the average operating cost within infinite horizons ahead in the **ONICP**. In the following, we will derive the **ONICP** formulation step by step from the case with a finite look-ahead window.

Let $H(x)$ denote the maintenance cost caused by inventory level $x$ (either holding or shortage cost), which is traditionally defined as

$$H(x) = h \cdot \max(x, 0) + p \cdot \max(-x, 0)$$

and $\delta(u)$ denote an indicator function to show whether an order is placed or not,

$$\delta(u) = \begin{cases} 
1 & u > 0 \\
0 & u = 0 
\end{cases}$$

As mentioned before, although the time-dependent $(s_i, S_i)$ policy might be optimal for some non-stationary cases, it will be more efficient to directly optimize the order quantity in $N$ periods ($2N$ variables in total) than to optimize $(s_i, S_i)$ for $i = 1, ..., N$ ($2N$ variables in total). Thus, we formulate the case of looking ahead $N$ periods as

$$\min_{u_1, \ldots, u_N} E \left\{ \sum_{i=1}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right) \right\}$$

s.t. $x_i = x_{i-1} - d_i + u_i$, $i = 1, \ldots, N$.

The cost function is commonly adopted in literatures and represents the expected operating cost including maintenance cost and setup cost within $N$ periods. Since we are only interested in finding the optimal $u_1$ for the immediate period at each decision point, we can explicitly demonstrate the contribution of $u_1$ to the total operating cost in the following equivalent formulation,

$$\min_{u_1} E \left\{ H(x_1) + K \cdot \delta(u_1) + \min_{u_2, \ldots, u_N} E \left\{ \sum_{i=2}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right) \right\} \right\}$$

s.t. $x_i = x_{i-1} - d_i + u_i$, $i = 1, \ldots, N$.

The Hindsight Optimization methods in \cite{Chong2000, Wu2002} can be further utilized to approximate the term of $\min_{u_2, \ldots, u_N} E \left\{ \sum_{i=2}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right) \right\}$ as the expected hindsight-optimal value, that is, $E \left\{ \min_{u_2, \ldots, u_N} \sum_{i=2}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right) \right\}$. Thus, the case of looking ahead $N$ periods can be finally reduced to

$$\min_{u_1} E \left\{ J_N(u_1, \omega) \equiv H(u_1 + x_0 - d_1) + K \cdot \delta(u_1) + \sum_{i=2}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right) \right\}$$

where $L_N(x_1, \omega)$ is an auxiliary optimization problem defined below

$$L_N(x_1, \omega) \equiv \min_{u_2, \ldots, u_N} \sum_{i=2}^{N} \left( H(x_i) + K \cdot \delta(u_i) \right)$$

s.t. $x_i = x_{i-1} - d_i + u_i$, $i = 2, \ldots, N$. 
Based on (8), we can finally formulate the ONICP as
\[
\min_{u_1} E \left\{ J(u_1, \omega) \right\} \equiv \lim_{N \to +\infty} \frac{J_N(u_1, \omega)}{N}
\]
(10)

3.3. CC Applicability Verification

To solve the ONICP, it is equivalent to answer the two questions below in order,

**Question 1:** Whether to order;
(Two options: Yes or No)

**Question 2:** How many to order if “Yes” to Question 1.
(Many options: the order quantity can be continuous or discrete integers)

**Question 1** has only two options. A C-Sol must exist in answering this two-option question. If \( \Pr[u_1^0 = 0] \geq 50\% \), the C-Sol is No; otherwise, it is Yes, where \( u_1^0 \) is an \( \omega \)-Sol of the ONICP.

**Question 2** is conditioned on the “Yes” answer to Question 1, which implies that \( u_1 > 0 \) is assumed in answering Question 2. We will verify the applicability of CC for the ONICP by proving the unimodality of \( J(u_1, \omega) \) for \( u_1 > 0 \) in the following theorem. Before that, we need to first reveal a lemma about \( J_N(u_1, \omega) \).

**Lemma 1** \( J_N(u_1, \omega) \) in (8) is \( K \)-convex in \( u_1 \) for \( u_1 > 0 \).

**Proof:** \( L_N(x_1, \omega) \) is essentially a cost-to-go function in the context of dynamic programming and \( L_N(x_1, \omega) \) can be proved to \( K \)-convex in \( x_1 \) by using a similar way as shown in Section 4.2 in [Bertsekas 2000]. From \( x_1 = u_1 + x_0 - d_1 \), \( L_N(u_1 + x_0 - d_1, \omega) \) is also \( K \)-convex in \( u_1 \).

From the definition of \( H(x) \) in (7), \( H(x_1) \) is convex in \( x_1 \). Similarly, \( H(u_1 + x_0 - d_1) \) is also convex in \( u_1 \).

Recalling the definition of \( J_N(u_1, \omega) \) in (8). From \( u_1 > 0 \), we have
\[
J_N(u_1, \omega) = H(u_1 + x_0 - d_1) + K + L_N(u_1 + x_0 - d_1, \omega)
\]
Combining it with the fact that \( H(u_1 + x_0 - d_1) \) is convex in \( u_1 \) and \( L_N(u_1 + x_0 - d_1, \omega) \) is \( K \)-convex in \( u_1 \), we have \( J_N(u_1, \omega) \) is \( K \)-convex in \( u_1 \) for \( u_1 > 0 \). □

**Theorem 4** \( J(u_1, \omega) \) in (10) is convex in \( u_1 \) for \( u_1 > 0 \).

**Proof:** From Lemma 1 \( J_N(u_1, \omega) \) is \( K \)-convex in \( u_1 \) for \( u_1 > 0 \), that is, it satisfies that for any \( 0 < u_1 < u_1' < u_1'' \)
\[
K + J_N(u_1'', \omega) \geq J_N(u_1', \omega) + \left( \frac{u_1'' - u_1'}{u_1'' - u_1} \right) \left( J_N(u_1', \omega) - J_N(u_1, \omega) \right).
\]
Then we apply limit operator at both sides and can have
\[
\lim_{N \to +\infty} \frac{K + J_N(u''_1, \omega)}{N} \geq \lim_{N \to +\infty} \frac{J_N(u'_1, \omega)}{N} + \left( \frac{u''_1 - u'_1}{u'_1 - u_1} \right) \lim_{N \to +\infty} \frac{(J_N(u'_1, \omega) - J_N(u_1, \omega))}{N}
\]
which implies that for any \(0 < u_1 < u'_1 < u''_1\),
\[
J(u'_1, \omega) \geq J(u''_1, \omega) + \left( \frac{u''_1 - u'_1}{u'_1 - u_1} \right) (J(u'_1, \omega) - J(u_1, \omega)).
\]
The inequality above is equivalent to the definition of function convexity, that is, \(J(u_1, \omega)\) is convex in \(u_1\) for \(u_1 > 0\). \(\Box\)

Theorem 3 implies that \(J(u_1, \omega)\) is unimodal for \(u_1 > 0\), which verifies the applicability of CC for the ONICP.

3.4. Solving \(\omega\)-Prob vs. Estimating Infinite-horizon Average Cost

To assess a solution \(u_1\), the sample average routine is needed in general SO methods to evaluate the value of \(J(u_1, \omega)\), i.e., the Infinite-horizon Average Cost, over \(M\) sample-paths \(\omega_1, ..., \omega_M\), and then calculate the average value of \(\sum_{j=1}^{M} J(u_1, \omega_j) / M\). As it is typically impossible to exactly calculate \(J(u_1, \omega)\), it can only be estimated by

\[
J(u_1, \omega) = \frac{J_N(u_1, \omega)}{N}, \quad \text{for } N \geq N'
\]

This long-term average value of \(\frac{J_N(u_1, \omega)}{N}\) converges slowly as \(N\) increases, especially for cases of non-stationary demands. A very large \(N'\) is needed to accurately estimate the value of \(J(u_1, \omega)\). The function \(J_N(u_1, \omega)\) includes the term \(L_N(x_1, \omega)\), which is actually a \((N'-1)\)-dimensional deterministic optimization problem with complexity of \(C(N')\) if \(N = N'\). Thus, the complexity of estimating the value of \(J(u_1, \omega)\) is also \(C(N')\), where \(N'\) is a very large number.

The CC framework bypasses this sample average routine. It is no need to estimate the value of \(J(u_1, \omega)\) as long as \(\omega\)-Sols can be obtained by solving \(\omega\)-Probs of the ONICP below,

\[
\min_{u_1} \left\{ J(u_1, \omega) = \lim_{N \to +\infty} \frac{J_N(u_1, \omega)}{N} \right\} \tag{12}
\]

Again, as \(J(u_1, \omega)\) cannot be exactly calculated, \(\omega\)-Sols can be derived by solving the problem below instead,

\[
u''_1 = \arg \min_{u_1} \left\{ \frac{J_N(u_1, \omega)}{N} \right\}, \quad \text{for } N \geq N''
\]

For a given \(N\), the problem \(\omega\)-Prob is equivalent to the problem below, which can be regarded as the \(\omega\)-Prob of the ONICP,

\[
\min_{u_1, ..., u_N} \sum_{i=1}^{N} (H(x_i) + K \cdot \delta(u_i)) \tag{14}
\]

\[\text{s.t. } x_i = x_{i-1} - d_i + u_i, \quad i = 1, ..., N.\]
Let \((u^*_1, \ldots, u^*_N)\) denote the optimal solution of the problem (14). Since demands from far future have little influence on order decisions made in early stages, \(u^*_1\) will gradually converge as \(N\) increases and remains unchanged after \(N \geq N''\). The \(\omega\)-Sol of the ONICP \(u^\omega_1\) equals the converged \(u^*_1\), which can be derived by solving the problem (14) with \(N = N''\) in complexity of \(C(N'')\).

As shown in the numerical results later, \(u^*_1\) converges much faster than the long-term average value \(\frac{J_N(u_1, \omega)}{N}\) as \(N\) increases, which implies that \(N'' \ll N'\). Thus, the complexity of solving \(\omega\)-Probs, i.e., \(C(N'')\), is much less than the complexity of estimating the Infinite-horizon Average Cost \(J(u_1, \omega)\), i.e., \(C(N')\).

In the next section, we will develop a polynomial algorithm to solve the \(\omega\)-Prob of the ONICP, that is, Step 3 in the CC framework.

### 4. Renewal Cycle Algorithm for Solving \(\omega\)-Prob of ONICP

The \(\omega\)-Prob of the ONICP (14) has not been well posed yet. Since profits earned from sales are not included in the objective, it would never be optimal to order anything in period \(N\), i.e., the last period, (possibly in last few periods) and mostly ends up with negative inventory levels. The terminal effect of “ordering nothing at last” and “ending with negative inventory” are undesirable, especially when \(N\) is relatively small. It is a common sense that total orders are encouraged to meet total demands for more profits. Since the \(\omega\)-Prob of the ONICP is a deterministic optimization problem as if all the demands within \(N\) periods are known for sure, it is possible to perfectly match total orders and total demands, that is, \(\sum_{i=1}^{N} u_i + x_0 = \sum_{i=1}^{N} d_i\). We can add this equation as a constraint into the problem (14) to avoid the undesirable terminal effect. The \(\omega\)-Prob of the ONICP can be well posed as

\[
\min_{u_1, \ldots, u_N} \sum_{i=1}^{N} \left\{ H(x_i) + K \cdot \delta(u_i) \right\}
\quad \text{s.t. } x_i = x_{i-1} - d_i + u_i, \ i = 1, \ldots, N
\quad \sum_{i=1}^{N} u_i + x_0 = \sum_{i=1}^{N} d_i.
\]  

(15)

Several methods had been proposed to solve problems similar to the \(\omega\)-Prob (15). In [Wagner and Whitin 1958], an efficient algorithm is developed to solve the case without backlogging. In [Zangwill 1966], although backlogging is considered, it is not quite efficient to implement because the dominant set is required to generate and its size grows exponentially with respect to \(N\). We will develop a new algorithm to solve the \(\omega\)-Prob (15) in complexity of \(O(N^2 \log N)\). Before that, we will first identify the Renewal Cycle property.
4.1. Renewal Cycle Property

We will start with the case with zero initial inventory level, that is, \( x_0 = 0 \). The case with non-zero initial inventory can be later equivalently reduced to the case with \( x_0 = 0 \). Let \( u_1^*, ... u_N^* \) and \( x_1^*, ... x_N^* \) denote optimal order quantities and inventory levels of the \( \omega \)-\( \text{Prob} \) (15). A structure of “Renewal Cycle” can be defined below.

**Definition 5** A Renewal Cycle is a continuous set of periods \( \{k, ..., n\} \) such that \( x_k^* = 0, x_n^* = 0 \) and \( x_i^* \neq 0 \) for \( i = k, ..., n - 1 \).

Since \( x_0 = 0 \) and \( x_N = 0 \), there is at least one renewal cycle within the periods \( \{1, ..., N\} \). Before proceeding to the “Renewal Cycle Property”, we need to reveal a lemma below to show that each order should always exactly cover the demands within a number of consecutive periods.

**Lemma 2** If \( u_i^* > 0 \), then there must exist some \( p, q \) such that \( u_i^* = \sum_{j=p+1}^{q} d_j \).

**Proof:** Without loss of generality, assume \( u_i^* = 0 \) for all \( i \) except for \( i_1, ..., i_m \). Let \( U_i = \sum_{j=1}^{i} u_{ij} \) and \( D_i = \sum_{j=1}^{i} d_i \). Then for all \( i_1 \leq i < i_{m+1} \), we have

\[
x_i = \sum_{j=1}^{i} u_j - \sum_{j=1}^{i} d_j = \sum_{j=1}^{i} u_{ij} - \sum_{j=1}^{i} d_i = U_i - D_i
\]

Furthermore, the \( \omega \)-\( \text{Prob} \) (15) with \( x_0 = 0 \) can be equivalently reduced to:

\[
\min_{U_1, ..., U_m} \sum_{i=1}^{m} F_i(U_i)
\]

where

\[
F_i(U_i) = \sum_{i = i_1}^{i_{m+1} - 1} H(U_i - D_i), \quad i_{m+1} = N + 1.
\]

and the optimal solution \( u_{i1}^*, ..., u_{im}^* \) can be recovered from the optimal \( U_1^*, ..., U_m^* \). According to the summation form in the problem (16), the optimal \( U_i^* \) can be obtained by

\[
U_i^* = \arg \min F_i(U_i)
\]

From (17) and the definition of \( H(x) \) in (17), the function \( F_i(U_i) \) is piece-wise linear and convex in \( U_i \) and its optimal solution must be some \( D_i \) for \( i \in \{i_1, i_1 + 1, ..., i_{m+1} - 1\} \). Without loss of generality, assume \( U_i^* = D_q \) and \( U_{i-1}^* = D_p \). Then the optimal solution of \( u_{ij}^* \) can be recovered as

\[
u_{ij}^* = U_i^* - U_{i-1}^* = D_q - D_p = \sum_{i=p+1}^{q} d_i
\]

which completes the proof. □
Based on this lemma, we can address the Renewal Cycle property in the following theorem.

**Theorem 5** *Only one order will be placed within a Renewal Cycle.*

*Proof:* Assume on the contrary that there are multiple orders placed within a renewal cycle. Without loss of generality, assume that the first order is placed in period \( l \). From Lemma 2 there must exist some \( q \) such that

\[
 u^*_l = \sum_{i=1}^{q} d_i
\]

It can be verified that \( q > l \); otherwise we can order \( \sum_{i=1}^{l} d_i \) instead in period \( l \) to get a better solution.

Since there are multiple orders within a renewal cycle, we assume the second order comes in period \( p \). Clearly, the second order must be placed before period \( q \); otherwise the inventory will return to zero in period \( q \) that contradicts to the definition of renewal cycle. Thus we have \( l < p \leq q \).

Another solution can be constructed by replacing \( u^*_l \) and \( u^*_p \) by \( u'_l = \sum_{i=1}^{p} d_i \) and \( u'_p = u^*_p + \sum_{i=p+1}^{q} d_i \), respectively and results in a lower cost, which contradicts to the optimality of \( u^*_i \) and completes the proof. \( \square \)

A typical optimal inventory level trajectory can be illustrated in Figure 3 based on Theorem 5.
4.2. Renewal Cycle Algorithm

Let $J_n$ denote the optimal cost over period $\{1, 2, ..., n\}$ ending with zero inventory level and $G_{k,n}$ denote the optimal cost over a Renewal Cycle over periods $\{k, ..., n\}$. Then we could have

$$J_n = \min_{1 \leq k \leq n-1} \left\{ J_k + G_{k,n} \right\}$$

Based on the Renewal Cycle property in Theorem 5, we could define $G_{k,n}$ as the optimal cost of the following problem:

$$G_{k,n} \equiv \min_{z_k, \ldots, z_n} \sum_{i=k}^{n} H(x_i)$$

s.t. $x_i = - \sum_{j=k}^{i} d_j + u \cdot z_i, \ i = k, \ldots, n$; 

$$\sum_{i=k}^{n} z_i = 1; \quad u = \sum_{i=k}^{n} d_i;$$

$$z_i = \{0, 1\}, \ i = k, \ldots, n.$$  \hspace{1cm} (18)

The constraint of $\sum_{i=k}^{n} z_i = 1$ ensures that only one order is placed within the Renewal Cycle $\{k, ..., n\}$. Based on the definition of Renewal Cycle, the corresponding order quantity $u$ will perfectly cover all the demands within this Renewal Cycle, i.e., $\sum_{i=k}^{n} d_i$. The problem (18) can be easily solved by a binary search algorithm in complexity of $O(\log N)$.

From above, we can finally develop the Renewal Cycle Algorithm (RCA) as shown in Table 2 in which we sequentially solve $J_n$ from $J_1$ to $J_N$. Since $J_N$ is actually the $\omega$-Prob (15), the RCA can solve the $\omega$-Prob of the ONICP in complexity of $O(N^2 \log N)$.

**Table 2 Renewal Cycle Algorithm**

| Step 1: | Start from $n = 1$ and $J_1 = G_{1,1};$ |
| Step 2: | Compute $G_{k,n}$ for $k = 1, ..., n;$ |
| Step 3: | Compute $J_n = \min_{1 \leq k \leq n-1} \{ J_k + G_{k,n} \};$ |
| Step 4: | $n = n + 1; \text{ if } n > N \text{ stop, otherwise goto step 2.}$ |

4.3. Nonzero Initial Inventory Level

We commonly face a nonzero initial inventory level, $x_0 \neq 0$, when solving the ONICP at each decision point. Before looking into the case of $x_0 \neq 0$, we first need to reveal the following lemma.

**Lemma 3** If $x_0 \geq d_1$, then $u^*_1 = 0$. 
Proof: Assume on the contrary that \( u^*_1 > 0 \). We could have
\[
x^*_1 = x_0 - d_1 + u^*_1 > 0
\]
Now we consider another solution by only replacing \( u^*_1 \) and \( u^*_2 \) with \( u'_1 = 0 \) and \( u'_2 = u^*_1 + u^*_2 \) respectively. After this modification, only the inventory level in period 1 is changed to
\[
x'_1 = x_0 - d_1 > 0
\]
From \( x_0 \geq d_1 \) and \( u^*_1 > 0 \), we have \( x^*_1 > x'_1 > 0 \), which implies that the modified solution has a lower holding cost. Moreover, the total setup cost of the modified solution is also no more than the one of the optimal solution. Therefore, the modified solution is better than the optimal solution, which contradicts to the optimality and completes the proof. \( \square \)

There are two cases for non-zero initial inventory: (i) \( x_0 < d_1 \) and (ii) \( x_0 \geq d_1 \). Both of them can be equivalently converted to the \( \omega\text{-Prob} \) with zero initial inventory through the following ways.

**Case (i) \( x_0 < d_1 \):** It can be equivalently reduced to the zero initial inventory case with demands \( \{d'_1, ..., d'_N\} \) as \( d'_1 = d_1 - x_0 \) and \( d'_i = d_i \) for \( i = 2, ..., N \).

**Case (ii) \( x_0 \geq d_1 \):** There must exist \( k \) such that \( x_0 - \sum_{i=1}^{k-1} d_i \geq d_k \) and \( x_0 - \sum_{i=1}^{k} d_i < d_{k+1} \). After iteratively applying Lemma 3 from period 1 to \( k \), we can have \( u^*_i = 0 \) for \( i = 1, ..., k \). Then, the remaining periods \( \{k + 1, ..., N\} \) becomes Case (i).

### 4.4. Lead Time and Yield Ratio

It will also be interesting to consider lead time and yield ratio in solving the \( \omega\text{-Prob} \). Although lead times \( L_i \) and yield ratios \( \alpha_i \) are also random variables like demands \( d_i \), they become deterministic in the \( \omega\text{-Prob} \) over a specific sample-path \( \omega \), which provides us opportunity to efficiently solve the \( \omega\text{-Prob} \) with \( L_i \) and \( \alpha_i \) through the following way. We can first obtain the optimal order quantity \( u^*_i \) by solve the \( \omega\text{-Prob} \) \([15]\) and then modify the order quantity as \( u^*_i / \alpha_i \) and place the order \( L_i \) periods a-head.

### 4.5. Complexity of CC with RCA embedded

We can apply the CC framework with the RCA embedded to efficiently obtain a \( C\text{-Sol} \) of the \( ONICP \) in complexity of \( O(M \cdot N''^2 \log N'') \).

The RCA can also facilitate general SO methods in estimating \( J(u_1, \omega) \), but they still have a complexity of \( O(M \cdot I \cdot N'^2 \log N') \) in solving the \( ONICP \), where \( N' \gg N'' \) as mentioned in the previous section.

The \( C\text{-Sol} \) derived by the CC framework can serve as a near-optimal solution in the sense of “Optimality in Expectation” if the Non-singularity condition is not satisfied. Since the \( C\text{-Sol} \) can
be highly efficiently derived, there might be some extra time and computing resources left under this on-line environment, which can be utilized to improve solution performance by implementing general SO methods with the C-Sol as the initial solution.

5. Numerical Results

We demonstrate the performance of the CC framework with RCA embedded through the following numerical results. All the algorithms here are programmed in Matlab and executed on a computer with Intel i5 CPU 3.2GHz, 4GB RAM and 32-bit Win7. The following parameters for this ONICP are identical to those used in [Zheng 1991],

- Fixed Setup Cost $K = 64$;
- Holding Cost Rate $h = 1$;
- Penalty Cost Rate $p = 9$.

A case of non-stationary demands is considered, in which demand in each period is Poisson distributed and may has a different mean value $\mu_i$. The mean value $\mu_i$ will be randomly picked from a set of numbers from 10 to 75 in increments of 5, that is, \{10, 15, 20, ..., 70, 75\}.

5.1. Estimation of $\omega$-Med

![\omega$-Med Estimation](image)

An example of the estimation of $\omega$-Med is interpreted in Figure 4 in which $M = 200$ sample-paths are generated. $\omega$-Sols are obtained by solving 200 corresponding $\omega$-Probs through the RCA. Then the cdf function of $\omega$-Sol is approximated by the solid line in Figure 4. Finally, the $\omega$-Med can be estimated through the dash line and equals 78 in this case.
5.2. Convergence of $\omega$-Med in $M$

The convergence of $\omega$-Med in the number of sample-paths $M$ is shown in Figure 5 in which $M$ varies from 10 to 1000 in increments of 10. It can be found that the estimation of $\omega$-Med quickly converges within 100 replications, which supports the result in Theorem 2.

5.3. Convergence of $u_1^*$ and Long-term Average Cost in $N$

The convergence results of $u_1^*$ and the long-term average cost $J_N(u_1, \omega)/N$ are shown in Figure 6 and 7 respectively, in which the look-ahead window size $N$ varies from 2 to 100 in increments of 1.
still not quite converged when $N = 100$. Based on this observation, we can choose $N'' = 10$ in (13) to ensure the convergence of $u_1^*$ for this non-stationary case. While we have to pick $N'$ around 100 in (11) for a good approximation of the long-term average cost, which implies that $N' \ll N''$ as mentioned in Section 3.

5.4. Complexity of RCA in $N$

The RCA is a polynomial algorithm to solve the $\omega$-Prob (15). Its computational time is demonstrated in Figure 8 in which $N$ varies from 10 to 200 in increments of 10 and the average CPU time for each $N$ is averaged over 20 sample-paths. It can be found that the CPU time increases
polynomially in $N$ and a 200-dimensional single-sample-path problem can be solved within a second.

5.5. C-Sol Performance

We will demonstrate the C-Sol performance for both stationary and non-stationary demands cases in Table 4 and 5 respectively, in which ten scenarios with 50 periods are randomly generated for both cases and the $(s, S)$ (or $(s_i, S_i)$) policy serves as the benchmark. (The optimal static $(s, S)$ policies had been derived for stationary demands with different $\mu$ in Zheng (1991) and we list them in Table 3 for convenience.)

| $\mu$ | 10  | 15  | 20  | 25  | 30  | 35  | 40  | 45  | 50  | 55  | 60  | 65  | 70  | 75  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $s^*$ | 6   | 10  | 14  | 19  | 23  | 28  | 33  | 37  | 42  | 47  | 52  | 56  | 62  | 67  |
| $S^*$ | 40  | 49  | 62  | 56  | 66  | 77  | 87  | 97  | 108 | 118 | 129 | 75  | 81  | 86  |

The stationary case can be used for the Near-Optimality test for C-Sols because the true optimal policy $(s^*, S^*)$ can be derived. We choose $\mu = 20$ in this stationary case test and its corresponding optimal policy is $(s^* = 14, S^* = 62)$ from Table 3. From Table 4, the threshold policy of $(s^* = 14, S^* = 62)$ achieves a smaller average operating costs than the C-Sols derived by the CC, which confirms that the static $(s, S)$ policy possesses the optimality in expectation. The C-Sols derived by the CC are near-optimal because their average cost is only slightly (2.52%) worse than the one of the true optimal policy. Besides, as the $(s, S)$ policy is optimal in the average sense for finite horizon cases, the C-Sols can perform better in some specific scenarios, such as Scenario 2, 4, 6 and 10 in Table 4.

|   | Cost by $(s, S)$ | Cost by CC | Difference $(s, S) - CC$ | Improvement $(s, S) - CC$/$CC$ |
|---|-----------------|-------------|--------------------------|-------------------------------|
| 1 | 2401            | 2537        | -136                     | -5.66%                        |
| 2 | 2710            | 2517        | 193                      | 7.12%                         |
| 3 | 2525            | 2488        | 37                       | 1.47%                         |
| 4 | 2612            | 2644        | -32                      | -1.23%                        |
| 5 | 2450            | 2730        | -280                     | -11.43%                       |
| 6 | 2390            | 2750        | -360                     | -15.06%                       |
| 7 | 2401            | 2549        | -148                     | -6.16%                        |
| 8 | 2711            | 2538        | 173                      | 6.38%                         |
| 9 | 2410            | 2647        | -237                     | -9.83%                        |
| 10| 2598            | 2442        | 156                      | 6.00%                         |
| Mean| 2520.8        | 2584.2      | -63.4                    | -2.52%                        |
For the non-stationary case, \( \mu_i \) may be different from each other. A common heuristic method is to select \((s_i, S_i)\) according to \( \mu_i \) in each period \( i \) as if demands are stationary with the mean value of \( \mu_i \). For example, if \( \mu_1 = 15, \mu_2 = 30, \mu_3 = 20, \ldots \), then we can look up Table 3 to find their corresponding optimal values, choose \((s_1 = 10, S_1 = 49)\), \((s_2 = 23, S_2 = 66)\), \((s_3 = 14, S_3 = 62)\), ..., to apply in period 1, 2, 3, ..., respectively. Clearly, this heuristic \((s_i, S_i)\) policy is not optimal for the non-stationary case. We compare it with the \( C-Sols \) derived by the \( CC \) in Table 5. It can be found that the \( C-Sols \) performs averagely 15.20% better than the heuristic \((s_i, S_i)\) policy in Table 5.

### Table 5 Non-stationary Demands

|       | Cost by \((s_i, S_i)\) | Cost by \(CC\) | Difference \(C_{ss} - C_{cc}\) | Improvement \((C_{ss} - C_{cc})/C_{ss}\) |
|-------|------------------------|----------------|-------------------------------|----------------------------------|
| 1     | 3506                   | 2861           | 645                           | 18.40%                           |
| 2     | 3642                   | 2980           | 662                           | 18.18%                           |
| 3     | 3467                   | 3046           | 421                           | 12.14%                           |
| 4     | 3611                   | 3086           | 525                           | 14.54%                           |
| 5     | 3540                   | 3033           | 507                           | 14.32%                           |
| 6     | 3519                   | 3068           | 451                           | 12.82%                           |
| 7     | 3516                   | 3048           | 468                           | 13.31%                           |
| 8     | 3782                   | 3128           | 654                           | 17.29%                           |
| 9     | 3440                   | 3024           | 416                           | 12.09%                           |
| 10    | 3567                   | 2907           | 660                           | 18.50%                           |
| Mean  | 3559                   | 3018.1         | 540.9                         | 15.20%                           |

6. **Conclusion**

To capture the nature of non-stationary demands in real practice, we consider the On-line Non-stationary Inventory Control Problem (**ONICP**) in this paper. Massive and time-consuming evaluations are needed in solving this **ONICP** by general Simulation-based Optimization (**SO**) methods, which is not affordable under an on-line environment with no luxury of time and computing resources.

A new **SO** method, termed the “Champion Competition” (**CC**) framework is developed to tackle this computational challenge. The sample average routine commonly adopted in general **SO** methods is to evaluate quantities of interests over multiple sample-paths and average them to estimate the expected performance of a solution, which causes the majority of computational burden. The **CC** provides a different framework that can bypass this time-consuming sample average routine and reduce the complexity of general **SO** methods by orders of magnitude.

The **CC** framework aims at an alternate type of optimal solution, “Champion Solution” (**C-Sol**), which coincides the traditional optimality in expectation under “Non-singularity Condition” and
can serve as a near-optimal solution for general cases. The CC framework can be applied to a class of SO problems that contain a C-Sol and the ONICP is proven to be one of them. A C-Sol of the ONICP can be efficiently derived by applying the CC framework with Renewal Cycle Algorithm (RCA) embedded. The RCA is a polynomial algorithm further developed based on the “Renewal Cycle” property identified over single sample-path of the ONICP, which can efficiently fulfill an important procedure of the CC framework in solving the ONICP.

Future work is aiming at generalizing the C-Sol sufficient condition and extending the idea of CC to a wider class of SO problems.

References
Arrow, Kenneth J. 1963. Social Choice and Individual Values. Yale University Press.

Askin, R. G. 1981. A procedure for production lot sizing with probabilistic dynamic demand. AIIE Transactions 12(2) 132–137.

Axsäter, S. 2006. Inventory Control (Second Edition). Springer.

Bertsekas, D. P. 2000. Dynamic Programming and Optimal Control Vol. 1 (Second Edition). Athena Scientific.

Beyer, D., S. P. Sethi. 1999. The classical average-cost inventory models of iglehart (1963) and veinott and wagner (1965) revisited. Journal of Optimization Theory and Applications 101(3) 523–555.

Bollapragada, Srinivas, Thomas E. Morton. 1999. A simple heuristic for computing nonstationary (s,S) policies. Operations Research 47(4) 576–584.

Bookbinder, James H., Jin-Yan Tan. 1988. Strategies for the probabilistic lot-sizing problem with service-level constraints. Management Science 34(9) 1096–1108.

Chen, C. H., L. H. Lee. 2011. Stochastic Simulation Optimization: An Optimal Computing Budget Allocation. World Scientific Publishing Co.

Chen, H. 1994. Stochastic Root Finding in System Design. Ph.D. Thesis. Purdue University, West Lafayette, Indiana, USA.

Chong, E. K. P., R. L. Givan, H. S. Chang. 2000. A framework for simulation-based network control via hindsight optimization. Proceedings of the 39th IEEE Conference on Decision and Control. 1433–1438.

Fu, M.C. 1994. Sample path derivatives for (s,S) inventory systems. Operations Research 42(2) 351–364.

Gallego, G., Ö. Özer. 2001. Integrating replenishment decisions with advanced demand information. Management Science 47(10) 1344–1360.

Ho, Y. C., X. R. Cao. 1991. Perturbation Analysis of Discrete-Event Dynamic Systems. Kluwer Academic Publisher, Boston.

Ho, Yu-Chi, Qian-Chuan Zhao, Qing-Shan Jia. 2008. Ordinal Optimization: Soft Optimization for Hard Problems. Springer Science & Business Media.
Hoeffding, W. March 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* **58**(301) 13–30.

Hong, L. Jeff, Barry L. Nelson. 2006. Discrete optimization via simulation using compass. *Operations Research* **54**(1) 115–129.

Hua, Z., J. Yang, F. Huang, X. Xu. 2009. A static-dynamic strategy for spare part inventory systems with nonstationary stochastic demand. *Journal of the Operational Research Society* **60** 1254 –1263.

Iglehart, Donald L. 1963. Optimality of (s, S) policies in the infinite-horizon dynamic inventory problem. *Management Science* **9**(2) 259–267.

Jin, J. 1998. *Simulation-Based Retrospective Optimization of Stochastic Systems*. Ph.D. Thesis. Purdue University, West Lafayette, Indiana, USA.

Scarf, Herbert. 1959. The optimality of (S,s) policies in the dynamic inventory problem. *Technical Report, Applied Mathematics and Statistics Laboratory, Stanford University*.

Shi, L., S. Olafsson. 2000. Nested partitions method for global optimization. *Operations Research* **48**(3) 390–407.

Silver, E. A., H. C. Meal. 1973. A heuristic selecting lot-size requirements for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. *Production Inventory Management* **14**(9) 64–74.

Veinott, Arthur F. 1966. On the optimality of (s, S) inventory policies: New conditions and a new proof. *SIAM Journal on Applied Mathematics* **14**(5) 1067–1083.

Veinott, Artur F., Harvey M. Wagner. 1966. Computing optimal (s,S) inventory policies. *Management Science* **11**(5) 525–552.

Wagner, Harvey M., Thomson M. Whitin. 1958. Dynamic version of the economic lot size model. *Management Science* **5**(1) 89–96.

Wu, G., E. K. P. Chong, R. L. Givan. 2002. Burst-level congestion control using hindsight optimization. *IEEE Transactions on Automatic Control, special issue on Systems and Control Methods for Communication Networks* **47**(6) 979–991.

Zangwill, Willard I. 1966. A deterministic multi-period production scheduling model with backlogging. *Management Science* **13**(1) 105–119.

Zheng, Y., A. Federgruen. 1991. Finding optimal (s,S) policies is about as simple as evaluating a single policy. *Operations Research* **39**(4) 654–665.

Zheng, Y. S. 1991. A simple proof for optimality of (s, s) policies in infinite-horizon inventory systems. *Journal of Applied Probability* **28** 802–810.

Zipkin, P. H. 2000. *Foundations of Inventory Management*. McGraw-Hill, Singapore.