DISCRETE OPTIMIZATION VIA SIMULATION WITH STOCHASTIC CONSTRAINTS

A Thesis
Presented to
The Academic Faculty

by

Chuljin Park

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2013

DISCRETE OPTIMIZATION VIA SIMULATION WITH STOCHASTIC CONSTRAINTS

Approved by:

Professor Seong-Hee Kim, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Sigrún Andradóttir
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Mustafa M. Aral
School of Civil and Environment Engineering
Georgia Institute of Technology

Professor David M. Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Valerie Thomas
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 21 June 2013

To my wife and family.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere thanks to my advisor, Dr. Seong-Hee Kim, for her thoughtful guidance, support, understanding, and friendship. I deeply appreciate her for believing in me and guiding me to overcome all the obstacles of this hard study.

I am also sincerely thankful to Dr. Sigrún Andradóttir and Dr. David M. Goldsman who introduced me to the simulation area. Their inspiration and help were extremely valuable for me to complete my doctoral study. In addition I appreciate Dr. Mustafa M. Aral and Dr. Valerie Thomas for guiding me to another research area, environmental management, enlarging my vision, and providing valuable comments as members of my dissertation committee.

Finally, I cannot express enough gratitude for the support of my parents, brother, and friends, who made all of this possible, and most importantly I appreciate my wife, Mi Lim Lee for her encouragement, endless support, and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis, we first develop a new method called penalty function with memory (PFM). PFM consists of a penalty parameter and a measure of constraint violation and it converts a discrete optimization via simulation (DOvS) problem with stochastic constraints into a series of DOvS problems without stochastic constraints. PFM determines a penalty of a visited solution based on past results of feasibility checks on the solution. Specifically, assuming a minimization problem, a penalty parameter of PFM, namely the penalty sequence, diverges to infinity for an infeasible solution but converges to zero almost surely for any strictly feasible solution under certain conditions. For a feasible solution located on the boundary of feasible and infeasible regions, the sequence converges to zero either with high probability or almost surely. As a result, a DOvS algorithm combined with PFM performs well even when optimal solutions are tight or nearly tight. Second, we design an optimal water quality monitoring network for river systems. The problem is to find the optimal location of a finite number of monitoring devices, minimizing the expected detection time of a contaminant spill event while guaranteeing good detection reliability. When uncertainties in spill and rain events are considered, both the expected detection time and detection reliability need to be estimated by stochastic simulation. This problem is formulated as a stochastic DOvS problem with the objective of minimizing expected detection time and with a stochastic constraint on the detection reliability; and it is solved by a DOvS algorithm combined with PFM. Finally, we improve PFM by combining it with an approximate budget allocation procedure. We revise an existing optimal budget allocation procedure so that it can handle active constraints and satisfy necessary conditions for the convergence of PFM.

CHAPTER I

INTRODUCTION

For a complicated industrial or service system with various uncertainties, analytically examining the system performance is usually difficult. Meanwhile, the computing power has been significantly improved in last decades, opening up large opportunities for stochastic simulation as analysis or supporting tools for decision-making.

Researchers often use simulation to compare the performance of a number of simulated systems (or solutions) and have proposed many statistical procedures to efficiently and accurately find the best simulated system. Ranking and Selection (R&S) procedures are useful for comparing a finite number of simulated systems, and either provide statistical validity of correct selection or maximize the probability of correct selection under a finite computing budget. See [9] and [18] for example R&S procedures. R&S procedures are appropriate when the number of systems is no more than several hundred (up to 1,000).

When the search space is too large to simulate all systems at the same time, a different class of procedures, namely optimization via simulation (OvS), is more appropriate. OvS algorithms enable us to find the best or a good system when the objective function needs to be evaluated through simulation. Thorough reviews of OvS research and practice for both discrete and continuous variables can be found in [3] and [11]. In this thesis, we focus on discrete OvS (DOvS). Research directions and guidance for use of DOvS algorithms are provided in [24].

The simulation community has presented a number of DOvS algorithms. [2] presents advantages of using a cumulative sample mean to estimate the value of an optimal solution and provides a structure of adaptive random search algorithms. [5]

provides the balanced explorative and exploitative search with estimation (BEESE) algorithms that simultaneously maintain the global search, the local search, and the improvement of solution estimation. The nested partitions (NP) method due to [33] is a globally convergent DOvS algorithm that works for both continuous and discrete decision variables. [28] proposes DOvS algorithms using the NP framework. Their algorithms use a cumulative sample mean as an estimate of a performance measure and are globally convergent in a bounded discrete solution set. The convergent optimization via most-promising-area stochastic search (COMPASS) of [14] is a framework for discrete optimization that finds a local optimum. Other model-based methods for DOvS problems such as stochastic model reference adaptive search (SMRAS) due to [15] are well established. SMRAS employs a parameterized distribution as a sampling distribution that generates solutions and updates parameters of the sampling distribution using the Kullback-Leibler distance. Although these algorithms are shown to perform well for many stochastic optimization problems, they are not directly applicable when a DOvS problem has stochastic constraints, constraints on secondary performance measures that need to be estimated by stochastic simulation.

Recently, simulation optimization with stochastic constraints on secondary performance measures have been studied actively. For example, one may want to find an inventory policy with the smallest long-run cost per period while the expected number of lost sales is less than or equal to some constant. This problem is more difficult than a pure R&S or stochastically unconstrained OvS problem because of the following reasons:

- Correlation among the performance measures: Primary and secondary performance measures are often correlated in practice. For example, expected queue length and throughput in a queueing model are correlated.

- Solutions near or right on the boundary of feasible and infeasible regions of stochastic constraints: In optimization problems, as discussed in [25], constraints on secondary performance measures almost always represent limits on our ability to minimize or maximize the objective function, implying that optimal solutions are likely to be tight or nearly tight. When the true optimal solution is right on the boundary, the solution can be declared as infeasible (with a high probability even when the number of observations is large) and thus it may not be selected as the optimal.

A constrained R&S problem is formulated in [4], which proposes statistically-valid selection procedures when there is one stochastic constraint and performance measures are correlated. Although their methods are statistically valid, they cannot handle solutions located right on the boundary of the feasible and infeasible regions (i.e. tight solutions with one or more active constraints). Instead, they introduce the concept of error tolerance for stochastic constraints and assume that a decision maker is willing to accept a system whose secondary performance measure is within a constant amount of the constraint threshold as the best if it shows a good primary performance measure. The procedures of [4] are extended to multiple stochastic constraints in [13]. On the other hand, several researchers provide a method of efficiently allocating a finite computing budget in order to maximize the probability of selecting the best feasible solution. For example, see [16], [17], [29], and [36]. Some discuss correlations between primary and secondary performance measures, but none of proposed procedures can handle tight solutions.

In OvS, there have been some efforts to use the concept of the penalty function which is a popular method in deterministic optimization of [26]. With the penalty function, a DOvS algorithm is applied to a series of new DOvS problems without stochastic constraints, rather than applied to the original problem. [21] proposes a penalty function method in which the penalty parameter diverges to infinity as search iteration increases and the measure of violation of a constraint is estimated by a sample average of a secondary performance measure for each constraint. [22] proposes a method of finding saddle points of a DOvS problem with stochastic constraints. These methods can handle correlation among performance measures, but does not perform well when the best feasible solution is tight or nearly tight.

In this thesis, we consider a DOvS problem with stochastic constraints and propose a new method, namely penalty function with memory (PFM). PFM enables us to handle multiple stochastic constraints with general inequalities in DOvS. PFM also uses the idea of the penalty function and reformulates the original problem into a series of new unconstrained problems. However, PFM uses a different penalty parameter whose value converges to 0 for feasible solutions and diverges to infinity for infeasible solutions. More specifically, a penalty value for each solution is determined adaptively based on observed feasibility of the solution.

Chapter II introduces PFM that enables existing DOvS algorithms to handle stochastic constraints. Convergence properties are provided with their proofs. Parameter selection for PFM is discussed and a DOvS algorithm with PFM is tested on three numerical examples. In Chapter III, we formulate a problem designing the optimal water quality monitoring network for river systems as a stochastically constrained DOvS problem. We use NP+PFM, a version of NP combined with PFM, as an optimization algorithm, and address implementation issues specialized for designing the water quality monitoring network. The performance of NP+PFM is tested on the Altamaha River and compared with that of the genetic algorithm (GA) from [39]. Chapter IV provides an approximate budget allocation procedure that further improves the performance of DOvS algorithms combined with PFM. Convergence properties are proven and experimental results on three numerical examples are presented. Finally, Chapter V summarizes main contributions of the thesis.

CHAPTER II

PENALTY FUNCTION WITH MEMORY FOR DISCRETE OPTIMIZATION VIA SIMULATION WITH STOCHASTIC CONSTRAINTS

In this chapter, we propose a new method, namely penalty function with memory (PFM), that can handle multiple stochastic constraints even when performance measures are correlated and tight or nearly tight solutions exist in stochastically constrained DOvS problems. PFM itself is not an optimization algorithm, however, it aids existing DOvS algorithms originally developed for stochastically unconstrained DOvS problems and helps them find an optimal solution in the presence of stochastic constraints. PFM uses the same measure of violation for a stochastic constraint as in [21]. However, PFM differs from their method in that it determines a penalty parameter of a visited solution based on the past results of feasibility checks on the solution. More specifically, PFM guarantees that a sequence of penalty parameters diverges to infinity almost surely (if minimization) for an infeasible solution. For a strictly feasible solution, the sequence is guaranteed to converge to zero almost surely. For a tight solution, the sequence converges to zero either with high probability or almost surely. As a result, a DOvS algorithm combined with PFM works well even when optimal solutions are tight or nearly tight. When the set of feasible solutions is not empty, one version of PFM guarantees that a solution returned by the combined algorithm converges to one of the optimal solutions.

We first present conditions in which a penalty sequence of PFM should satisfy and prove that if the penalty sequence satisfies the conditions, a global convergence property is preserved when an existing globally-convergent algorithm is combined with PFM. Then we present two example penalty sequences: a penalty sequence with constants (PS_c) and a penalty sequence with functions (PS_f). Both sequences guarantee almost sure convergence to infinity and zero for infeasible and strictly feasible solutions, respectively. For a tight solution, PS_c guarantees only that the sequence converges to zero with high probability while PS_f provides almost-sure convergence to zero under some conditions.

This chapter is organized as follows: Section 2.1 defines our problem and notation, provides a framework of existing DOvS algorithms for a stochastically unconstrained DOvS problem, and then presents an example that demonstrates the need for a more sophisticated penalty function. Section 2.2 introduces PFM and two example penalty sequences along with their convergence proofs. Section 2.3 discusses parameter selection for the implementation of PFM. Experimental results of three different numerical examples are presented in Section 2.4, followed by concluding remarks in Section 2.5.

2.1 Background

In this section, we define our problem and notations and review a common structure of existing DOvS algorithms. Then we demonstrate the need for a sophisticated penalty function using a simple example with three systems.

2.1.1 Problem and Notation

Let $\mathbf{x} = (x_1, \ldots, x_d)$ represent a solution and Θ the whole decision variable space, which is a discrete and finite set in $\mathbb{R}^d$, the set of $d-$dimensional real-numbered vectors. We assume that information on deterministic constraints is already reflected in Θ. Let $G(\mathbf{x})$ represent the primary performance measure and $H_\ell(\mathbf{x})$ a secondary performance measure of the ℓth constraint, for any $\ell = 1, 2, \ldots, m$. Then $G_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x})$ represent the ith observation of $G(\mathbf{x})$ and $H_\ell(\mathbf{x})$, respectively, both observed by stochastic simulation. We assume that for any given $\mathbf{x}$, $G_i(\mathbf{x})$, $i = 1, 2, \ldots$, are independent and identically distributed (iid) random variables. For a given $\ell =$

$1, 2, \ldots, m$, $H_{\ell i}(\mathbf{x})$, $i = 1, 2, \ldots$, are also iid. It is possible that $G_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x})$ are correlated. The expectation and variance of $G(\mathbf{x})$ are $\mathbf{E}[G(\mathbf{x})]$ and $\mathbf{Var}[G(\mathbf{x})]$, respectively. Also, for any $\ell = 1, 2, \ldots, m$, the expectation and variance of $H_\ell(\mathbf{x})$ are $\mathbf{E}[H_\ell]$ and $\mathbf{Var}[H_\ell(\mathbf{x})]$, respectively.

Then our DOvS problem with stochastic constraints is defined as follows:

$$\begin{aligned} &\text{argmin}_{\mathbf{x}\in\Theta}\mathbf{E}[G(\mathbf{x})], \\ &\text{subject to } \mathbf{E}[H_\ell(\mathbf{x})] \geq q_\ell, \ \ell = 1, 2, ..., m, \end{aligned} \tag{1}$$

where q_ℓ is a threshold constant of the ℓth stochastic constraint.

To explain our method, we need the following notations:

$\Theta_f \equiv$ the set of all feasible solutions;

$\Theta^* \equiv$ the set of optimal solutions with $\min_{\mathbf{x}\in\Theta_f}\mathbf{E}[G(\mathbf{x})]$;

$\mathbf{x}_o^b \equiv$ an optimal solution in Θ^*;

$\Lambda \equiv$ an index set of all stochastic constraints (i.e., $\Lambda \equiv \{1, 2, ..., m\}$);

$\Lambda_{S(\mathbf{x})} \equiv \{\ell \mid \mathbf{E}[H_\ell(\mathbf{x})] > q_\ell, \ \ell \in \Lambda\}$, an index set of strictly feasible stochastic constraints for $\mathbf{x}$;

$\Lambda_{A(\mathbf{x})} \equiv \{\ell \mid \mathbf{E}[H_\ell(\mathbf{x})] = q_\ell, \ \ell \in \Lambda\}$, an index set of active stochastic constraints for $\mathbf{x}$; and

$\Lambda_{I(\mathbf{x})} \equiv \{\ell \mid \mathbf{E}[H_\ell(\mathbf{x})] < q_\ell, \ \ell \in \Lambda\}$, an index set of infeasible stochastic constraints for $\mathbf{x}$.

We also make the following assumption throughout the chapter.

Assumption 1 *Θ_f is not an empty set and for any $\mathbf{x} \in \Theta$ and $\ell = 1, 2, \ldots, m$, $\mathbf{E}[(G(\mathbf{x}))^2] < \infty$, $\mathbf{Var}[G(\mathbf{x})] < \infty$, $\mathbf{E}[(H_\ell(\mathbf{x}))^2] < \infty$, and $\mathbf{Var}[H_\ell(\mathbf{x})] < \infty$.*

Section 2.2.1 provides short discussion about implementation when Θ_f is empty.

2.1.2 DOvS Algorithms

Let $\mathcal{D}$ represent an existing DOvS algorithm designed for a DOvS problem without any stochastic constraint. Many DOvS algorithms follow a general structure of

Step 1: Set search iteration counter $k = 1$. Choose an initial solution sampling strategy.

Step 2: Sample a fixed number of solutions according to the current sampling strategy and take additional observations from the solutions sampled at iteration k (not from all solutions sampled so far).

Step 3: Update the current optimal solution.

Step 4: If stopping criteria are satisfied, then stop. Otherwise, update the solution sampling strategy, set $k \leftarrow k + 1$, and go to Step 2.

Figure 1: General structure of random search algorithms.

random search algorithms, shown in Figure 1, presented by [3].

When describing $\mathcal{D}$, we need additional notations:

$r \equiv$ counter for the number of visits;

$v_k(\mathbf{x}) \equiv$ the number of visits up to iteration k for $\mathbf{x}$;

$\Delta n_r(\mathbf{x}) \equiv$ the number of new observations obtained at the rth visit for $\mathbf{x}$;

$n_r(\mathbf{x}) \equiv$ the total number of observations obtained up to the rth visit for $\mathbf{x}$;

$n_{v_k}(\mathbf{x}) \equiv n_{v_k(\mathbf{x})}(\mathbf{x})$, the total number of observations obtained up to iteration k for $\mathbf{x}$;

$\overline{G}_k(\mathbf{x}) \equiv \frac{1}{n_{v_k}(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} G_i(\mathbf{x})$, cumulative sample mean of observations $G_i(\mathbf{x})$ for the primary performance measure up to iteration k;

$\overline{H}_{\ell k}(\mathbf{x}) \equiv \frac{1}{n_{v_k}(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} H_{\ell i}(\mathbf{x})$, cumulative sample mean of observations $H_{\ell i}(\mathbf{x})$ for the ℓth secondary performance measure up to iteration k; and

$\hat{\mathbf{x}}_k^* \equiv$ the sample best among all sampled solutions up to iteration k.

Table 1 summarizes all of the notations used in the chapter. Throughout the chapter, we assume that $\mathcal{D}$ satisfies the following assumption:

Assumption 2 *Algorithm $\mathcal{D}$ follows a general structure of random search algorithms in Figure 1 and satisfies the following:*

1. *A solution sampling strategy of $\mathcal{D}$ guarantees that each solution has non-zero probability of being sampled at any iteration k.*

Table 1: Notation.

PROBLEM FORMULATION	Θ	the whole decision variable space which is a discrete and finite set in $\mathbb{R}^d$
	Θ_f	the set of all feasible solutions
	Θ^*	the set of optimal solutions with $\min_{\mathbf{x}\in\Theta_f}\mathbf{E}[G(\mathbf{x})]$
	$\mathbf{x}_o^b$	an optimal solution in Θ^*
	Λ	an index set of all stochastic constraints
	$\Lambda_{S(\mathbf{x})}$	an index set of strictly feasible stochastic constraints for $\mathbf{x}$
	$\Lambda_{A(\mathbf{x})}$	an index set of active stochastic constraints for $\mathbf{x}$
	$\Lambda_{I(\mathbf{x})}$	an index set of infeasible stochastic constraints for $\mathbf{x}$
DOvS	$\mathcal{D}$	an existing DOvS algorithm designed for a stochastically unconstrained problem
	k	search iteration counter $k = 1, 2, \dots$
	r	counter for the number of visits, $r = 1, 2, \dots$
	$\Delta n_r(\mathbf{x})$	the number of new observations obtained at the rth visit for $\mathbf{x}$
	$v_k(\mathbf{x})$	the number of visits up to iteration k for $\mathbf{x}$
	$n_r(\mathbf{x})$	the total number of observations obtained up to the rth visit for $\mathbf{x}$
	$n_{v_k}(\mathbf{x})$	the total number of observations obtained up to iteration k for $\mathbf{x}$
	$\overline{G}_k(\mathbf{x})$	cumulative sample mean of the objective up to iteration k
	$\overline{H}_{\ell k}(\mathbf{x})$	cumulative sample mean of the ℓth secondary performance measure up to iteration k
	$\hat{\mathbf{x}}_k^*$	the sample best among all sampled solutions up to iteration k

2. *An observation allocation strategy of $\mathcal{D}$ guarantees that* $\mathbf{P}\big[\lim_{k\to\infty} n_{v_k}(\mathbf{x}) = \infty\big] = 1$ *for any* $\mathbf{x} \in \Theta$.
3. *To estimate the performance of an objective at each solution, $\mathcal{D}$ uses a strongly consistent estimator.*

Assumption 2 implies that the sampling strategy of $\mathcal{D}$ should guarantee that as iteration k increases both $v_k(\mathbf{x})$ and $n_{v_k}(\mathbf{x})$ go to infinity. In a finite solution set, example algorithms that satisfy Assumption 2 include the NP-based algorithms of [28], random search methods of [2], the balanced explorative and exploitative search with estimation (BEESE) algorithms of [5]. A modified simulated annealing [1] and the stochastic model reference adaptive search [15] also satisfy Assumption 2 and can be combined with PFM. However, the definitions of $\overline{G}_k(\mathbf{x})$ and $\overline{H}_{\ell k}(\mathbf{x})$ need to be changed to sample means based on observations newly obtained at each visit because they do not use cumulative sample means.

2.1.3 Motivating Example

In this subsection, we demonstrate that a naive penalty function method can fail and a sophisticated penalty function is needed in stochastic optimization. Assuming a single stochastic constraint, one may say that Problem (1) can be solved by any existing DOvS algorithm with objective $\arg\min_{x\in\Theta} \overline{G}_k(\mathbf{x}) + M_k \cdot \max\{0, q - \overline{H}_k(\mathbf{x})\}$, where $M_k \to \infty$ as k increases. The reasoning would be that $\max\{0, q - \overline{H}_k(\mathbf{x})\}$ should converge to zero as k increases due to the strong law of large numbers (SLLN) for any feasible solution, which would, in turn, result in a zero penalty value for any feasible solution. Similarly one would argue that the penalty of an infeasible solution should converge to infinity and thus, the algorithm should be able to find a true optimal solution.

However, this reasoning is not true. Consider an example with three systems and a single constraint. We assume that all $G(\mathbf{x})$ and $H_1(\mathbf{x})$, for $\mathbf{x} \in \{1, 2, 3\}$ are normally distributed with variances 1 and mean values as follows:

- System 1: $\mathbf{E}[G(1)] = 1$, $\mathbf{E}[H_1(1)] = 0.3$
- System 2: $\mathbf{E}[G(2)] = 0$, $\mathbf{E}[H_1(2)] = 0$
- System 3: $\mathbf{E}[G(3)] = -1$, $\mathbf{E}[H_1(3)] = -0.3$

We set $M_k = 3k$ and $q_1 = 0$. Thus, system 2 is the optimal solution which is tight. One observation is obtained for each solution at each iteration, and the current sample best $\hat{\mathbf{x}}_k^*$ is a solution with the smallest $\overline{G}_k(\mathbf{x}) + M_k \cdot \max\{0, q - \overline{H}_k(\mathbf{x})\}$. Figure 2 reports the percentage of time that $\hat{\mathbf{x}}_k^*$ is equal to system 2 over 500 macro replications.

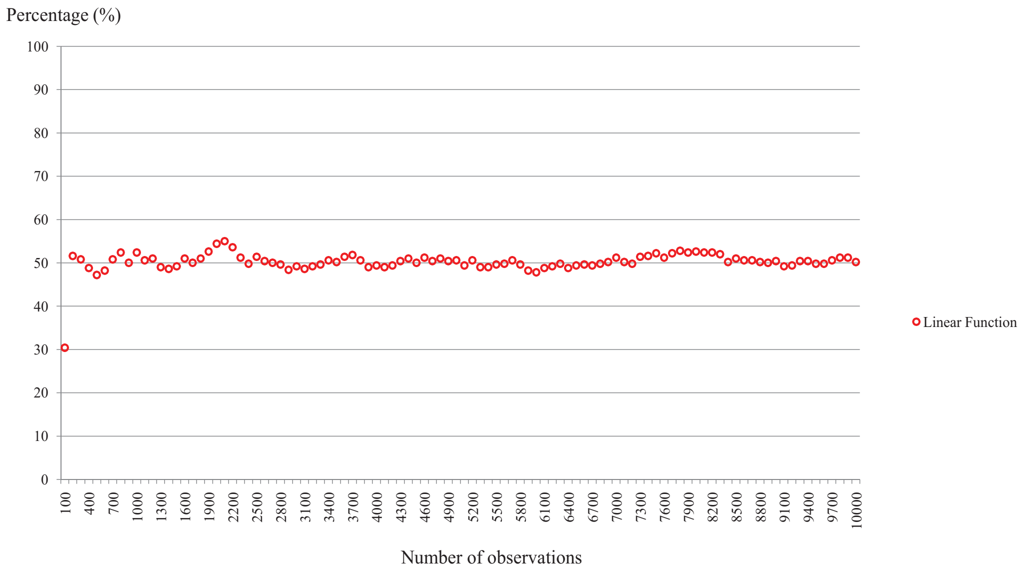


Figure 2: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ of the problem with three systems.

As seen in Figure 2, system 2 is returned as the sample best only around 50% of the time, which clearly demonstrates the failure of a naive penalty function method. [25] provides the following mathematical arguments to explain the need for a penalty function whose value converges to zero for feasible solutions and to infinity for infeasible solutions.

For a constant $\omega > 0$,

$$\begin{aligned}
&\mathbf{P}\{\text{receive a penalty}\}\\
&\quad= \mathbf{P}\{M_k \max\{0, q - \overline{H}_k(\mathbf{x})\} > \omega\}\\
&\quad\geq \mathbf{P}\{M_k(q - \overline{H}_k(\mathbf{x})) > \omega\}\\
&\quad= \mathbf{P}\left\{\frac{\sqrt{n_{v_k}(\mathbf{x})}(\overline{H}_k(\mathbf{x}) - \mathbf{E}[H_1(\mathbf{x})])}{\sqrt{\mathbf{Var}[H_1(\mathbf{x})]}} < \frac{\sqrt{n_{v_k}(\mathbf{x})}}{\sqrt{\mathbf{Var}[H_1(\mathbf{x})]}}(q - \mathbf{E}[H_1(\mathbf{x})] - \omega/M_k)\right\}.
\end{aligned}$$

By the Central Limit Theorem (CLT), the above probability can be calculated from a standard normal distribution as $k \to \infty$ in three cases:

- Infeasible $\mathbf{x}$: M_k should become large quickly, so $M_k \to \infty$ is fine.
- Strictly feasible $\mathbf{x}$: M_k should remain small and ideally go to 0.
- Tight $\mathbf{x}$: M_k must go to 0 to avoid a penalty.

PFM allows its penalty value to assume a complicated form and to depend on observed feasibility/infeasibility of each solution.

2.2 Penalty Function with Memory

In this section, we introduce PFM and prove global convergence of a combined procedure of $\mathcal{D}$ and PFM. Then we propose two example penalty sequences and prove their convergence properties.

2.2.1 Structure of PFM

PFM consists of a penalty sequence and a measure of violation for each constraint. Let $\lambda_\ell^r(\mathbf{x})$ define a penalty sequence of the ℓth constraint at the rth visit for $\mathbf{x}$. To simplify notation, we use $\lambda_\ell^{v_k}(\mathbf{x})$ to denote $\lambda_\ell^{v_k(\mathbf{x})}(\mathbf{x})$ which is a penalty parameter at visit $v_k(\mathbf{x})$ for $\mathbf{x}$. In PFM, the value of $\lambda_\ell^{v_k}(\mathbf{x})$ is determined based on observed (in)feasibility of solution $\mathbf{x}$ and it should go to zero for feasible solutions but infinity for infeasible solutions. A measure of violation for the ℓth constraint is $\max(0, q_\ell - \overline{H}_{\ell k}(\mathbf{x}))$ for $\ell = 1, 2, \ldots, m$. Then PFM is defined as $\sum_{\ell \in \Lambda} \left[\lambda_\ell^{v_k}(\mathbf{x}) \times \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\}\right]$. A new objective function with PFM at search iteration k is $Z_k(\mathbf{x}) = \overline{G}_k(\mathbf{x}) + \sum_{\ell \in \Lambda} \left[\lambda_\ell^{v_k}(\mathbf{x}) \times \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\}\right]$. We show that if a combined algorithm $\mathcal{D}$+PFM returns a solution with the smallest $Z_k(\mathbf{x})$ as $\hat{\mathbf{x}}_k^*$, then $Z_k(\hat{\mathbf{x}}_k^*)$ converges to $\min_{\mathbf{x} \in \Theta_f} \mathbf{E}[G(\mathbf{x})]$ as k increases. We first need the following condition for $\lambda_\ell^{v_k}(\mathbf{x})$:

Condition 1 As k goes to infinity, $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for $\ell \in \Lambda_{S(\mathbf{x})} \cup \Lambda_{A(\mathbf{x})}$ and $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for $\ell \in \Lambda_{I(\mathbf{x})}$ where $\xrightarrow{a.s.}$ denotes almost sure convergence.

Theorem 1 *For all $\mathbf{x} \in \Theta$, if Assumptions 1 and 2 hold and $\lambda_\ell^{v_k}(\mathbf{x})$ satisfies Condition 1, then $Z_k(\mathbf{x}) \xrightarrow{a.s.} \mathbf{E}[G(\mathbf{x})]$ if $\Lambda_{S(\mathbf{x})} \cup \Lambda_{A(\mathbf{x})} = \Lambda$; $Z_k(\mathbf{x}) \xrightarrow{a.s.} \infty$ if $\Lambda_{I(\mathbf{x})} \neq \emptyset$.*

See the Appendix for the proof of Theorem 1. Now, we get a main convergence theorem of $\mathcal{D}$+PFM:

Theorem 2 *If Assumptions 1 and 2 hold and $\lambda_\ell^{v_k}(\mathbf{x})$ satisfies Condition 1, then $Z_k(\hat{\mathbf{x}}_k^*) \xrightarrow{a.s.} \min_{\mathbf{x}\in\Theta_f} \mathbf{E}[G(\mathbf{x})]$ as $k \to \infty$.*

See the Appendix for the proof of Theorem 2.

Remark 1: If $\Theta_f = \emptyset$, then $\min_{\mathbf{x}\in\mathcal{V}_k} Z_k(\mathbf{x})$ diverges as $k \to \infty$. Thus, if a penalty sequence of a sample best does not show convergence to zero but instead shows increasing tendency for large k, a decision maker may declare no feasible solution.

The above theorem implies that the problem of finding an optimal solution in DOvS falls down to the problem of coming up with a penalty sequence that satisfies Condition 1. In the next subsections, we provide two example penalty sequences, PS$_c$ and PS$_f$, and discuss their convergence properties.

2.2.2 Penalty Sequence with Constants (PS$_c$)

We first introduce PS$_c$, an intuitive and easy-to-implement penalty sequence. Let $S_\ell^{v_k}(\mathbf{x}) \equiv \sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}(\mathbf{x})$ where $\zeta_{\ell r}(\mathbf{x}) \equiv \sum_{i=n_{r-1}(\mathbf{x})+1}^{n_{r-1}(\mathbf{x})+\Delta n_r(\mathbf{x})} \frac{H_{\ell i}(\mathbf{x})-q_\ell}{\sqrt{\Delta n_r(\mathbf{x})}}$. Then PS$_c$ is defined as follows:

Penalty Sequence with Constants (PS$_c$)

$$\lambda_\ell^{v_k}(\mathbf{x}) = \begin{cases} \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \theta_a, & \text{if } S_\ell^{v_k}(\mathbf{x}) < 0, \\ \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \theta_d, & \text{if } S_\ell^{v_k}(\mathbf{x}) \geq 0, \end{cases}$$

where $\lambda_\ell^0(\mathbf{x})$ is the initial penalty constant λ_ℓ^0 for any $\mathbf{x} \in \Theta$, θ_a is an appreciation factor, and θ_d is a depreciation factor such that $\theta_a > 1$ and $0 < \theta_d < 1$.

If $\Delta n_r(\mathbf{x})$ is a constant Δn for any $\mathbf{x}$ and r, then $S_\ell^{v_k}(\mathbf{x}) < 0$ and $S_\ell^{v_k}(\mathbf{x}) \geq 0$ are simplified to $\overline{H}_{\ell k}(\mathbf{x}) < q_\ell$ and $\overline{H}_{\ell k}(\mathbf{x}) \geq q_\ell$, respectively.

To prove the convergence properties of PS$_c$, we first need two lemmas.

Lemma 1 *[31] If $\{Z_n, n = 1, 2, \ldots\}$ is a sequence of random variables with $\mathbf{E}[Z_j] = \mu_j < \infty$, $\mathbf{Var}[Z_j] = \sigma_j^2 < \infty$, and $\sum_{j=1}^{\infty} \frac{\sigma_j^2}{j^2} < \infty$, then*

$$\frac{\sum_{j=1}^n Z_j}{n} - \frac{\sum_{j=1}^n \mu_j}{n} \xrightarrow{a.s.} 0, \quad \text{as } n \to \infty.$$

Lemma 1 is known as Kolmogorov's strong law of large numbers. Note that if σ_j^2 is a finite constant for any j, then $\sum_{j=1}^{\infty} \frac{\sigma_j^2}{j^2} < \infty$ is always satisfied because $\sum_{j=1}^{\infty} \frac{1}{j^2} < \infty$.

Lemma 2 *[10] Let $X_1, X_2, \cdots$ be independent random variables each having mean 0 and variance 1 and such that the central limit theorem is applicable. Let $s_k = X_1 + X_2 + \cdots + X_k$ and let N_n denote the numbers of s_k's, $1 \leq k \leq n$, which are positive. Then, for $0 \leq \xi \leq 1$,*

$$\lim_{n\to\infty} \mathbf{P}\left[\frac{N_n}{n} < \xi\right] = \frac{2}{\pi} \arcsin \sqrt{\xi}.$$

Theorem 3 *Under Assumptions 1 and 2, PS$_c$ guarnatees $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for any $\ell \in \Lambda_{S(\mathbf{x})}$ and $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for any $\ell \in \Lambda_{I(\mathbf{x})}$ as $k \to \infty$.*

See the Appendix for the proof of Theorem 3. For $\ell \in \Lambda_{A(\mathbf{x})}$, the following theorem holds:

Theorem 4 *Under Assumptions 1 and 2, PS$_c$ guarantees that $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{D} Y$ as $k \to \infty$ for any $\ell \in \Lambda_{A(\mathbf{x})}$, where*

$$Y = \begin{cases} 0, & \textit{with probility } 1 - \frac{2}{\pi} \textit{arcsin}\sqrt{\frac{-\log\theta_a}{\log\theta_d - \log\theta_a}}, \\ \infty, & \textit{with probility } \frac{2}{\pi} \textit{arcsin}\sqrt{\frac{-\log\theta_a}{\log\theta_d - \log\theta_a}}. \end{cases}$$

See the Appendix for the proof of Theorem 4. Theorem 4 implies that PS$_c$ does not satisfy Condition 1, but a careful choice of θ_a and θ_d makes the penalty sequence converge to zero with high probability. The choice of θ_a and θ_d is discussed in Section 2.3.

2.2.3 Penalty Sequence with Functions (PS$_f$)

While PS$_c$ uses same appreciation and depreciation factors for every solution, PS$_f$ allows each solution to use different appreciation and depreciation factors depending on, so called, the infeasible probability $p_\ell(\mathbf{x})$. The infeasible probability $p_\ell(\mathbf{x})$ represents how likely a solution is declared to be infeasible and is defined as $p_\ell(\mathbf{x}) \equiv \lim_{r\to\infty} \mathbf{P}\big[\zeta_{\ell r}(\mathbf{x}) < 0\big]$. We estimate the infeasible probability by $\hat{p}_\ell^{v_k}(\mathbf{x})$, where

$$\hat{p}_\ell^{v_k}(\mathbf{x}) = \frac{\sum_{r=1}^{v_k(\mathbf{x})} \mathbb{I}\{\zeta_{\ell r}(\mathbf{x}) < 0\}}{v_k(\mathbf{x})}. \tag{2}$$

To implement PS$_f$, a decision maker needs to choose the following parameters:

- two real-valued constants w_a and w_d such that $0 < w_a < 1$ and $0 < w_d < 1$, and two positive integers u and v such that $u < v$;
- $0 < h_1 < \ldots < h_u < 0.5 < h_{u+1} < \ldots < h_v < 1$;
- $1 < a_\nu$ and $0 < d_\nu < 1$, for all $\nu = 1, 2, \ldots, v$.

An estimated error tolerance for the infeasible probability of tight solutions is denoted as γ_ℓ^k and it is calculated as

$$\gamma_\ell^k = \begin{cases} \min\left(h_{u+1} - 0.5, 0.5 - h_u, \min_{\mathbf{x}\in\{\mathbf{x}|\hat{p}_\ell^{v_k}(\mathbf{x})>0.5+\epsilon_{\ell 0},\ \mathbf{x}\in\mathcal{V}_k\}} \frac{\hat{p}_\ell^{v_k}(\mathbf{x})-0.5}{2}\right), \\ \qquad\qquad\qquad\qquad \text{if } \{\mathbf{x}|\hat{p}_\ell^{v_k}(\mathbf{x}) > 0.5 + \epsilon_{\ell 0},\ \mathbf{x} \in \mathcal{V}_k\} \neq \emptyset; \\ \\ \min\left(h_{u+1} - 0.5, 0.5 - h_u\right), \quad \text{otherwise,} \end{cases}$$

where $\epsilon_{\ell 0}$ are very small positive constants close to zero for $\ell = 1, 2, \ldots, m$. The estimated error tolerance γ_ℓ^k defines a range in which a solution $\mathbf{x}$ does not receive any appreciation factor regardless of a feasibility decision at iteration k if $\hat{p}_\ell^{v_k}(\mathbf{x})$ falls in the range. Having such a range is critical when we establish $\lambda_\ell^{v_k}(\mathbf{x})$ with almost sure convergence for a tight solution.

Then PS$_f$ is defined as follows:

Penalty Sequence with Functions (PS$_f$)

$$\lambda_\ell^{v_k}(\mathbf{x}) = \begin{cases} \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \alpha_\ell^{v_k}(\mathbf{x}), & \text{if } S_\ell^{v_k}(\mathbf{x}) < 0, \\ \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \delta_\ell^{v_k}(\mathbf{x}), & \text{if } S_\ell^{v_k}(\mathbf{x}) \geq 0, \end{cases}$$

where $\lambda_\ell^0(\mathbf{x})$ is the initial penalty constant λ_ℓ^0, $\alpha_\ell^{v_k}(\mathbf{x})$ is an appreciation function, and $\delta_\ell^{v_k}(\mathbf{x})$ is a depreciation function whose values are determined by $\hat{p}_\ell^{v_k}(\mathbf{x})$ and the following table:

$\hat{p}_\ell^{v_k}(\mathbf{x})$	$[0, h_1)$	$[h_1, h_2)$	$\ldots$	$[h_u, 0.5 - \gamma_\ell^k)$	$[0.5 - \gamma_\ell^k, 0.5 + \gamma_\ell^k]$	$(0.5 + \gamma_\ell^k, h_{u+1}]$	$\ldots$	$(h_{v-1}, h_v]$	$(h_v, 1]$
$\alpha_\ell^{v_k}(\mathbf{x})$	a_0	a_1	$\ldots$	a_u	w_a	a_{u+1}	$\ldots$	a_{v-1}	a_v
$\delta_\ell^{v_k}(\mathbf{x})$	d_0	d_1	$\ldots$	d_u	w_d	d_{u+1}	$\ldots$	d_{v-1}	d_v

To prove the convergence properties of PS$_f$, we need the following lemma:

Lemma 3 *Let b_n, $n = 1, 2, \ldots$, be a positive real sequence such that $\lim_{n\to\infty} b_n = c < \infty$ for $c \in \mathbb{R}$. Then $\lim_{n\to\infty} \frac{1}{n}\sum_{r=1}^{n} b_r = c$.*

See the Appendix for the proof of Lemma 3.

Theorem 5 *If $\Delta n_r(\mathbf{x}) \to \infty$ as $r \to \infty$ (i.e., $k \to \infty$), then (i) $\hat{p}_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for $\ell \in \Lambda_{S(\mathbf{x})}$; (ii) $\hat{p}_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 1$ for $\ell \in \Lambda_{I(\mathbf{x})}$; and (iii) $\hat{p}_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0.5$ for $\ell \in \Lambda_{A(\mathbf{x})}$.*

See the Appendix for the proof of Theorem 5. Now we present the main theorems of PS$_f$.

Theorem 6 *If Assumptions 1 and 2 hold and $\Delta n_r(\mathbf{x}) \to \infty$ as $r \to \infty$ (i.e., $k \to \infty$), then PS$_f$ guarantees (i) $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for any $\ell \in \Lambda_{S(\mathbf{x})} \cup \Lambda_{A(\mathbf{x})}$; and (ii) $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for any $\ell \in \Lambda_{I(\mathbf{x})}$.*

See the Appendix for the proof of Theorem 6. Although some optimization algorithms such as the simulated annealing method of [1] and SMRAS of [15] allow $\Delta n_r(\mathbf{x})$ to increase as k increases, majority of OvS algorithms assume finite $\Delta n_r(\mathbf{x})$. Next we show the convergence properties of PS$_f$ assuming finite $\Delta n_r(\mathbf{x})$.

Corollary 1 *If Assumptions 1 and 2 hold and $\Delta n_r(\mathbf{x}) < \infty$, then PS$_f$ guarantees $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for any $\ell \in \Lambda_{S(\mathbf{x})}$.*

The proof of Corollary 1 is same as Case (i) in the proof of Theorem 6. For the convergence properties of $\ell \in \Lambda_{A(\mathbf{x})} \cup \Lambda_{I(\mathbf{x})}$, we make the following assumption:

Assumption 3 *For a finite constant $\Delta n(\mathbf{x}) > 0$, $\lim_{r\to\infty} \Delta n_r(\mathbf{x}) = \Delta n(\mathbf{x})$ and $\zeta_{\ell r}(\mathbf{x})$ are independent with a symmetric probability density or mass function.*

Remark 2: In Assumption 3, $\Delta n(\mathbf{x})$ can be different for each solution $\mathbf{x}$ but in the experiment we use equal $\Delta n(\mathbf{x})$ for all solutions $\mathbf{x}$ by setting $\Delta n(\mathbf{x}) = \Delta n$.

Assumption 3 ensures that $p_\ell(\mathbf{x}) = 0.5$ for $\ell \in \Lambda_{A(\mathbf{x})}$ and $p_\ell(\mathbf{x}) > 0.5$ for $\ell \in \Lambda_{I(\mathbf{x})}$. In practice, if $H_{\ell i}(\mathbf{x})$ are either within-replication averages or batch means or if $\Delta n_r(\mathbf{x})$ is large, then $\zeta_{\ell r}(\mathbf{x})$ will be approximately symmetric.

Theorem 7 *If Assumptions 1, 2, and 3 hold, then PS$_f$ guarantees (i) $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for any $\ell \in \Lambda_{I(\mathbf{x})}$ and (ii) $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ for any $\ell \in \Lambda_{A(\mathbf{x})}$ as $k \to \infty$.*

See the Appendix for the proof of Theorem 7.

2.3 Parameter Selection

In this section, we discuss how to choose parameters for implementation of each penalty sequence. We first investigate effects of appreciation and depreciation factors on a convergence or divergence rate of $Z_k(\mathbf{x})$ for a single solution $\mathbf{x}$ and then provide guidelines for parameter selection of PS$_c$ and PS$_f$.

2.3.1 Single Solution

We first study how appreciation and depreciation factors affect the convergence rate of $Z_k(\mathbf{x})$ to $\mathbf{E}[G(\mathbf{x})]$ for a feasible solution and the divergence rate of $Z_k(\mathbf{x})$ to infinity for an infeasible solution. Desirable appreciation and depreciation factors should ensure fast convergence of $Z_k(\mathbf{x})$ to the right value (i.e., either $\mathbf{E}[G(\mathbf{x})]$ or ∞) with high probability.

Let ρ_c represent the probability that $\lambda_\ell^{v_k}(\mathbf{x})$ converges to 0 for any $\ell \in \Lambda_{A(\mathbf{x})}$. That is, ρ_c is the probability of converging to the right value for tight solutions. Then, for fixed ρ_c and θ_a, θ_d is determined by

$$\theta_d = \exp\left(-(\log\theta_a)\left(\sin\frac{\pi(1-\rho_c)}{2}\right)^{-2} + \log\theta_a \right),$$

which is derived from Theorem 4.

To study convergence rates for various combinations of θ_a and ρ_c, we consider a single solution $\mathbf{x}$ with one constraint, $\mathbf{E}[H_1(\mathbf{x})] \geq 0$. In this problem, $H_1(\mathbf{x})$ is normally distributed with variance 1. We consider five different levels of feasibility: (i) $\mathbf{E}[H_1(\mathbf{x})] = 0.5244$ ($p_1(\mathbf{x}) = 0.3$ and clearly feasible); (ii) $\mathbf{E}[H_1(\mathbf{x})] = 0.1257$ ($p_1(\mathbf{x}) = 0.45$ and barely feasible); (iii) $\mathbf{E}[H_1(\mathbf{x})] = 0$ ($p_1(\mathbf{x}) = 0.5$ and tight); (iv) $\mathbf{E}[H_1(\mathbf{x})] = -0.1257$ ($p_1(\mathbf{x}) = 0.55$ and barely infeasible); and (v) $\mathbf{E}[H_1(\mathbf{x})] = -0.5244$ ($p_1(\mathbf{x}) = 0.7$ and clearly infeasible).

We test $\rho_c \in \{0.5, 0.7, 0.8, 0.9\}$ and $\theta_a \in \{\sqrt{1.1}, \sqrt{1.3}, \sqrt{1.5}, \sqrt{1.7}, \sqrt{1.9}\}$. At each search iteration k $\Delta n_r(\mathbf{x})$ is set to a single constant $\Delta n = 1$ for all solutions. Based on 100 macro replications, $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} < 0.1\}$ is estimated if $\mathbf{x}$ is feasible and $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} > 10\}$ is estimated if $\mathbf{x}$ is infeasible. Table 2 reports parameter settings with the largest estimated probability for each level of feasibility. As seen in the table, $\rho_c = 0.9$ and $\theta_a \geq \sqrt{1.3}$ perform equally well for feasible solutions and $\rho_c = 0.5$ and $\theta_a = \sqrt{1.9}$ perform the best for infeasible solutions.

Table 2: Recommended parameter settings for five different levels of (in)feasiblity.

	Clearly Feasible	Barely Feasible	Feasible with a Tight Constraint	Barely Infeasible	Clearly Infeasible
ρ_c	0.9	0.9	0.9	0.5	0.5
θ_a	$\geq \sqrt{1.3}$	$\geq \sqrt{1.3}$	$\sqrt{1.3}$	$\sqrt{1.9}$	$\sqrt{1.9}$

Figure 3 shows estimated $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} < 0.1\}$ for feasible solutions when $\rho_c = 0.9$ and $\theta_a = \sqrt{1.3}$ while Figure 4 shows estimated $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} > 10\}$ for infeasible solutions when $\rho_c = 0.5$ and $\theta_a = \sqrt{1.9}$. For all experiments, we observe that the estimated probability converges to 1 for both strictly feasible and infeasible solutions while it converges to ρ_c for a tight solution. We also note that a desirable combination for a feasible solution is the worst combination for an infeasible solution and vice versa.

2.3.2 Parameter Selection for PS$_c$

Now we discuss how to choose parameters for PS$_c$ which applies the same appreciation and depreciation factors to all solutions. If all solutions are feasible, then the findings

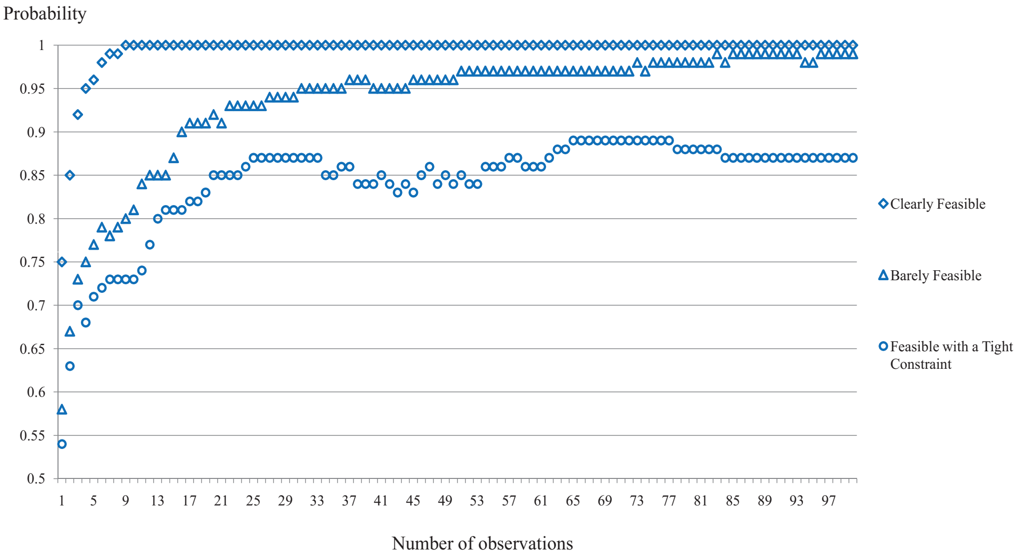


Figure 3: Estimated $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} < 0.1\}$ for feasible solutions when $\rho_c = 0.9$ and $\theta_a = \sqrt{1.3}$.

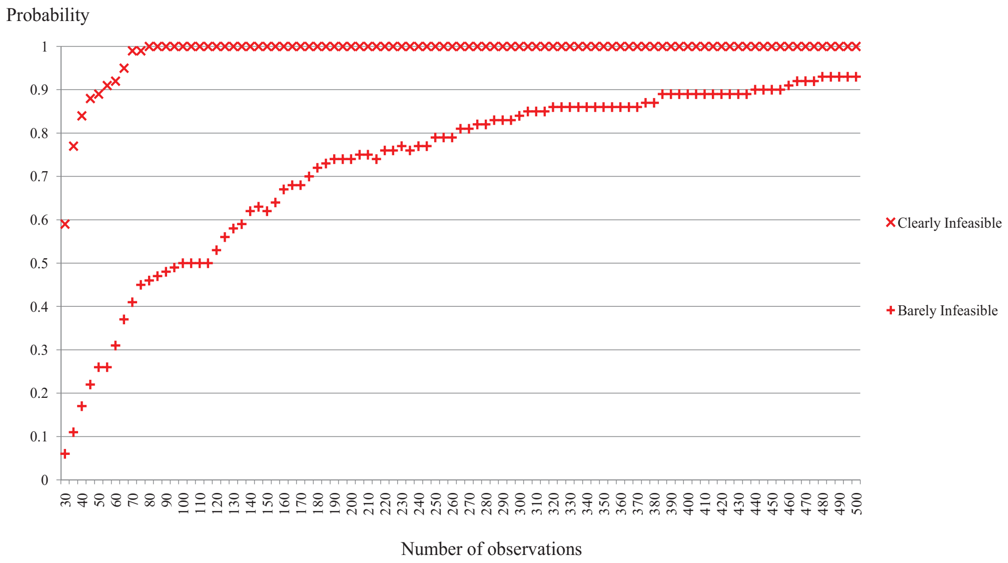


Figure 4: Estimated $\mathbf{P}\{\lambda_\ell^{v_k}(\mathbf{x}) \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\} > 10\}$ for infeasible solutions when $\rho_c = 0.5$ and $\theta_a = \sqrt{1.9}$.

from the previous subsection imply that we should choose $(\rho_c, \theta_a) = (0.9, \sqrt{1.3})$. If all are infeasible, then $(\rho_c, \theta_a) = (0.5, \sqrt{1.9})$ would be desirable. However, most DOvS problems contain both feasible and infeasible solutions with various levels of feasibility/infeasibility. Thus, we need to choose (ρ_c, θ_a) that shows performances robust for both feasible and infeasible solutions. To see the performance of various choices of (ρ_c, θ_a), we revisit the three-system example from Section 2.1.3.

Figure 5 shows the percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ based on 500 macro replications. To focus on small-sample-size behaviors of each combination, Figure 5 reports performances up to 1000 total number of observations although the experiment continued up to 10,000 total number of observations. The figure shows that (i) combinations (ρ_c, θ_a) with the same value of ρ_c exhibit similar behaviors and their probabilities of returning a true optimal solution converge to ρ_c (if observed up to 10,000 observations); (ii) combinations with high ρ_c show slower convergence with a high probability of returning a true optimal solution while combinations with low ρ_c show fast convergence with a low probability of returning a true optimal solution; and (iii) for a given

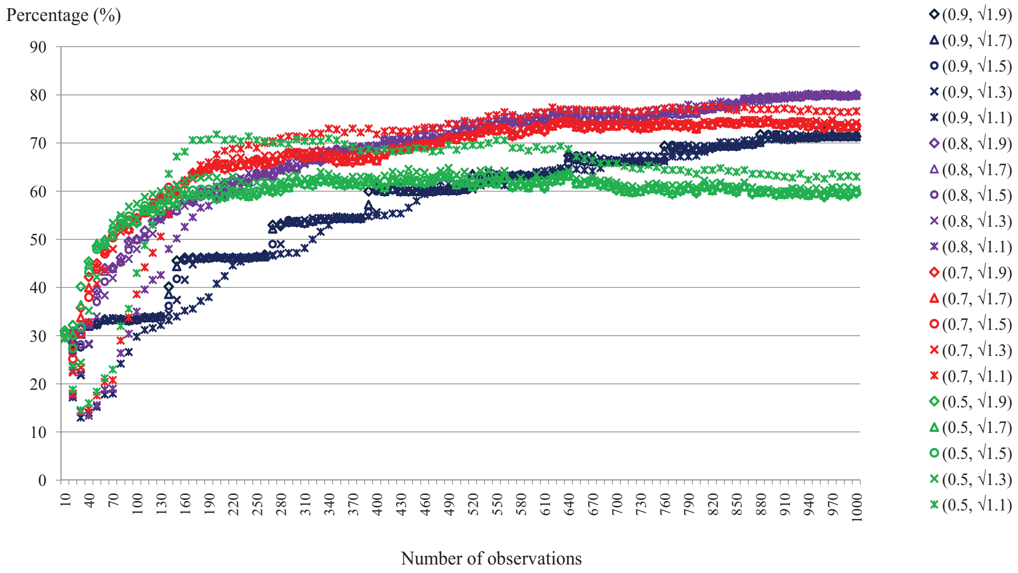


Figure 5: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ using PS$_c$ for the problem with three systems.

ρ_c, $\sqrt{1.3} \leq \theta_a \leq \sqrt{1.9}$ show faster convergence than $\theta_a < \sqrt{1.3}$. To balance between fast convergence and high probability of returning a true optimal solution, $(0.7, \sqrt{1.3})$ is a good compromise. For the largest probability of returning a true optimal solution, $(\rho_c, \theta_a) = (0.9, \sqrt{1.1})$ performs well but has very slow convergence.

2.3.3 Parameter Selection for PS$_f$

Unlike PS$_c$, PS$_f$ adjusts parameters to an effective choice using the infeasible probability $p_\ell(\mathbf{x})$. Intuitively, it makes sense to use $(\rho_c, \alpha_\ell^{v_k}(\mathbf{x})) = (0.5, \sqrt{1.9})$ when $p_\ell(\mathbf{x})$ is close to 1 and $(\rho_c, \alpha_\ell^{v_k}(\mathbf{x})) = (0.9, \sqrt{1.3})$ when $p_\ell(\mathbf{x})$ is close to 0. Table 3 is an example that assigns appreciation and depreciation functions this way. However, our experiments (which we omit due to space) show that the initial performance of PS$_f$ is worse than that of PS$_c$ with $(\rho_c, \theta_a) = (0.7, \sqrt{1.3})$ for the three-system example. It is because inaccurate $\hat{p}_\ell^{v_k}(\mathbf{x})$, when $v_k(\mathbf{x})$ is small, makes feasible or infeasible solutions receive the worst combination $(\rho_c, \alpha_\ell^{v_k}(\mathbf{x}))$ and it takes a while to fix wrong penalty values.

An adaptive PS$_f$ can reduce such initial error: start with parameters robust to

Table 3: An example of PS$_f$.

$\hat{p}_\ell^{v_k}(\mathbf{x})$	$[0, 0.35)$	$[0.35, 0.5 - \gamma_\ell^k)$	$[0.5 - \gamma_\ell^k, 0.5 + \gamma_\ell^k]$	$(0.5 + \gamma_\ell^k, 0.65]$	$(0.65, 1]$
ρ_c	0.9	0.9		0.9	0.5
$\alpha_\ell^{v_k}(\mathbf{x})$	$\sqrt{1.3}$	$\sqrt{1.3}$	0.95	$\sqrt{1.3}$	$\sqrt{1.9}$
$\delta_\ell^{v_k}(\mathbf{x})$	0.0054	0.0054	0.1	0.0054	0.7255

feasibility/infeasibility of solutions as shown in Table 4 and then jump to Table 3 when $v_k(\mathbf{x}) > N_p$ where N_p is the number of visits that makes $\hat{p}_\ell^{v_k}(\mathbf{x})$ accurate enough. For a given level of estimation error of the infeasible probability, one can plan for the value of N_p in advance by using the fact that the standard deviation of probability is maximized at 0.5.

Table 4: Robust PS$_f$.

$\hat{p}_\ell^{v_k}(\mathbf{x})$	$[0.5 - \gamma_\ell^k, 0.5 + \gamma_\ell^k]$	Otherwise
ρ_c		0.7
$\alpha_\ell^{v_k}(\mathbf{x})$	0.95	$\sqrt{1.3}$
$\delta_\ell^{v_k}(\mathbf{x})$	0.5	0.6033

2.4 Numerical Experiments

In this section, we test PFM on three numerical examples: (i) the Goldstein-Price problem, (ii) an (s, S) inventory policy problem, and (iii) the three-system example from Section 2.1.3 to test the performance of PFM when secondary performance measures have asymmetric distributions. The Goldstein-Price problem is introduced in Section 2.4.1 and the (s, S) inventory policy problem is described in Section 2.4.2. In all problems, there is a unique optimal solution $\mathbf{x}_o^b$.

For the Goldstein-Price problem and the (s, S) inventory policy problem, we need a DOvS algorithm because search spaces of the problems are too large to update and compare $Z_k(\mathbf{x})$ for all $\mathbf{x} \in \Theta$ at each iteration. A version of NP [28] is used and combined with PFM. The combined algorithm NP+PFM returns $\hat{\mathbf{x}}_k^* \equiv \text{argmin}_{\mathbf{x} \in \mathcal{V}_k} Z_k(\mathbf{x})$ at each iteration k. Table 5 shows parameters for the implementation of NP where n_0

is the number of observations obtained in the first visit, τ_k is the number of solutions sampled at iteration k, and ω is the number of subregions.

Table 5: Basic parameter settings for NP.

Problem	n_0	τ_k	ω
Goldstein Price Problem	9	16	4
(s, S) Inventory Policy Problem	30	9	2

[21] provide a penalty function for a DOvS problem with one stochastic constraint, which we call the augmented cost function (ACF). We take a straightforward extension of their method to multiple constraints and define ACF as $\sum_{\ell \in \Lambda} \alpha_\ell^k \times \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\}$. We take α_ℓ^k similar to the one used in their online companion:

$$\alpha_\ell^k = \begin{cases} \frac{e^k}{\min_{\mathbf{x} \in \Upsilon_k}(q_\ell - \overline{H}_{\ell k}(\mathbf{x}))}, & \text{if } \Upsilon_k \neq \emptyset; \\ 10^6, & \text{otherwise,} \end{cases}$$

where $\Upsilon_k \equiv \{\mathbf{x} | \overline{H}_{\ell k}(\mathbf{x}) < q_\ell \text{ and } \mathbf{x} \in \mathcal{V}_k\}$. NP+ACF takes $Z_k^{'}(\mathbf{x}) = \overline{G}_k(\mathbf{x}) + \sum_{\ell \in \Lambda} \alpha_\ell^k \times \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\}$ as an estimate of the performance measure in NP.

Let PFM(PS$_c$) and PFM(PS$_f$) represent PFM using PS$_c$ and PS$_f$, respectively, as a penalty sequence. For PS$_c$, we test two different combinations of (ρ_c, θ_a) discussed in Section 2.3.2: $(\rho_c, \theta_a) = (0.7, \sqrt{1.3})$ and $(\rho_c, \theta_a) = (0.9, \sqrt{1.1})$. We denote PS$_c$ with the former combination as PS$_{c1}$ and the latter as PS$_{c2}$. For PS$_c$, we set $\Delta n_r(\mathbf{x}) = \Delta n$ for all solution $\mathbf{x}$, and use $\Delta n = 3$ for the Goldstein-Price problem and $\Delta n = 10$ for the (s, S) inventory policy problem.

For PS$_f$, we employ an adaptive version: Table 4 is used until $v_k(\mathbf{x}) \leq N_p$ and then PS$_f$ uses parameters in Table 3 for $v_k(\mathbf{x}) > N_p$. We set $N_p = 10$ for the three-system example since the problem only includes three systems. For other numerical examples, $N_p = 200$ is used which ensures the standard error in $\hat{p}_\ell^{v_k}(\mathbf{x})$ is no more than 0.035. We denote PS$_f$ with fixed $\Delta n_r(\mathbf{x}) = \Delta n$ as PS$_{f1}$, and PS$_f$ with increasing

$\Delta n_r(\mathbf{x})$ as PS$_{f2}$. For increasing $\Delta n_r(\mathbf{x})$, the following equation is used:

$$\Delta n_r(\mathbf{x}) = \begin{cases} n_0 + \lceil \log r \rceil, & \text{if } r > 1, \\ n_0, & \text{otherwise.} \end{cases} \tag{3}$$

For the three-system example, we set $n_0 = \Delta n$. PS$_c$ uses $\Delta n_r(\mathbf{x}) = \Delta n$ for all three systems where $\Delta n \in \{1, 10, 30\}$. For PS$_f$, $\Delta n_r(\mathbf{x})$ is set to (3). As a competitive method, we apply $M_k = 3k$ as in Section 2.1.3.

We set λ_ℓ^0 for PFM as follows:

$$\lambda_\ell^0 = \begin{cases} \max_{k \le 20} \max_{\mathbf{x} \in \Upsilon_k} \frac{|\overline{G}_k(\mathbf{x}_k^*) - \overline{G}_k(\mathbf{x})|}{|q_\ell - \overline{H}_{\ell k}(\mathbf{x})|}, & \text{if } \Upsilon_k \neq \emptyset \text{ and } k > 20; \\ 10^6, & \text{otherwise,} \end{cases}$$

where $\Upsilon_k \equiv \{\mathbf{x} | \overline{H}_{\ell k}(\mathbf{x}) < q_\ell \text{ and } \mathbf{x} \in \mathcal{V}_k\}$.

2.4.1 Goldstein-Price Problem

The Goldstein-Price problem is one of the famous deterministic and continuous optimization problems with a two-dimensional quadratic function defined as,

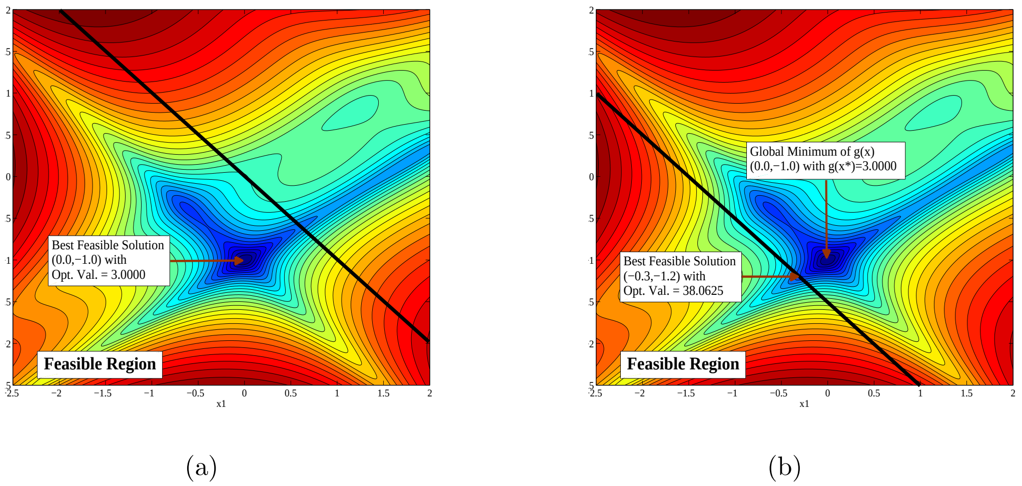
$$\begin{aligned} g(\mathbf{x}) &= \Big\{1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\Big\} \\ &\quad \times \Big\{30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\Big\}. \end{aligned}$$

Let $\phi_i(\mathbf{x})$ and $\psi_{\ell i}(\mathbf{x})$, $\ell = 1, 2, ..., m$ be iid normal random variables with mean zero and standard deviations $0.15\ g(\mathbf{x})$ and $0.15\ (a_\ell x_1 + b_\ell x_2)$ where a_ℓ and b_ℓ are constants. We define $G_i(\mathbf{x}) = g(\mathbf{x}) + \phi_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x}) = a_\ell x_1 + b_\ell x_2 + \psi_{\ell i}(\mathbf{x})$ and want to minimize $\mathbf{E}[G(\mathbf{x})]$ such that $\mathbf{E}[H_\ell(\mathbf{x})] \geq q_\ell$. We set $\Theta = \{-2.50, -2.49, ..., 1.99, 2.00\}^2$ which is a two-dimensional discretized set in $[-2.50,\ 2.00]^2$. The function $g(\mathbf{x})$ has four local minima and the global minimum at $(0, -1)$. In Θ, the largest and smallest values of $g(\mathbf{x})$ are 1,015,685 and 3, respectively.

We first consider a single constraint with one of the following constraints:

$$\mathbf{E}[-x_1 - x_2 + \psi_{1i}] \geq 0.0; \tag{4}$$

$$\mathbf{E}[-x_1 - x_2 + \psi_{1i}] \geq 1.5. \tag{5}$$



(a)

(b)

Figure 6: Contour of the Goldstein-Price function with a single constraint: constraint (4) (left) and constraint (5) (right).

Figure 6(a) shows that the true optimal solution of the problem with constraint (4) is identical to the global minimum of the unconstrained Goldstein-Price function. On the other hand, Figure 6(b) shows that constraint (5) is a difficult stochastic constraint because the optimal solution is located right on the line $-x_1 - x_2 = 1.5$ and there are many superior infeasible solutions near the optimal solution.

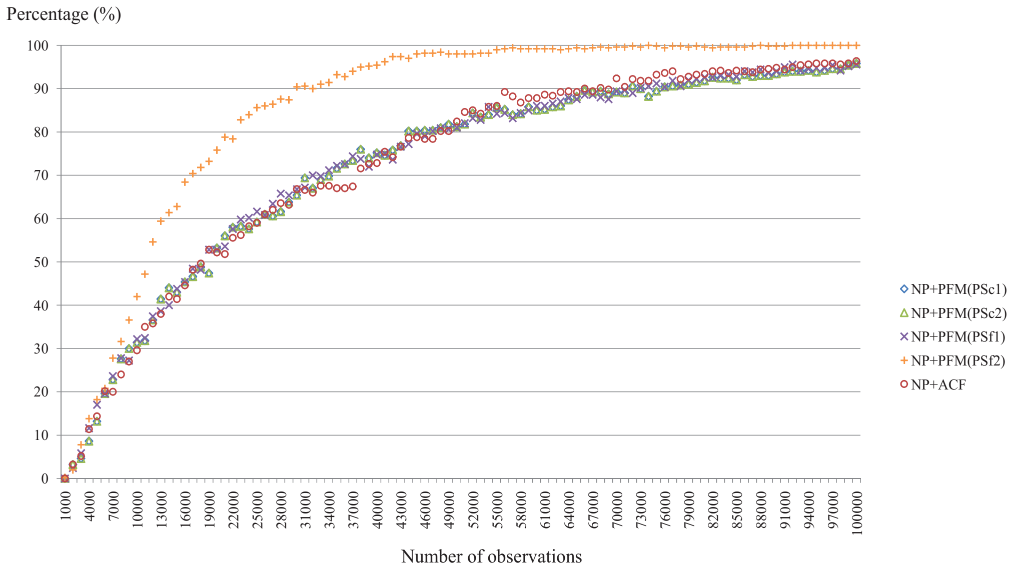


Figure 7: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ with constraint (4).

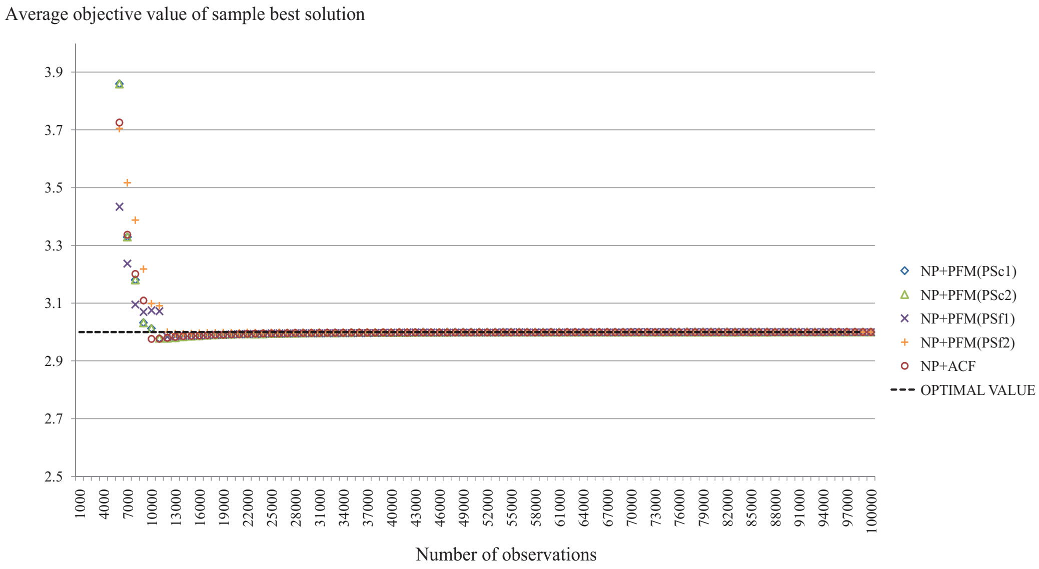


Figure 8: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ with constraint (4).

Figure 7 shows the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ and Figure 8 provides the average objective value of the sample best over 500 macro replications when constraint (4) is considered. Each macro replication is terminated when the total number of observations obtained so far reaches 100,000. As shown in Figure 7, NP+PFM(PS$_{f2}$) accomplishes up to 100% with the fastest rate of convergence and others achieve up to 95% with similar rates of convergence.

Figures 9 and 10 show the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ and the average objective value of the sample best over 500 macro replications when constraint (5) is considered. We arbitrary set $\epsilon_{10} = 0.00617$ for PS$_{f1}$ and PS$_{f2}$. Figure 9 shows that the choice of θ_a and θ_d does matter for a good performance of PS$_c$ in a difficult problem. At the end of the search, NP+PFM(PS$_{f2}$) achieves up to 90% while NP+PFM(PS$_{c2}$) and NP+PFM(PS$_{f1}$) reach around 80% and NP+PFM(PS$_{c1}$) reaches around 70%. NP+ACF finds the true best only around 50% of the time. Figure 10 shows that NP+PFM(PS$_{c1}$), NP+PFM(PS$_{f1}$) and NP+PFM(PS$_{f2}$) obtain better average estimated objective values than NP+PFM(PS$_{c2}$) and NP+ACF.

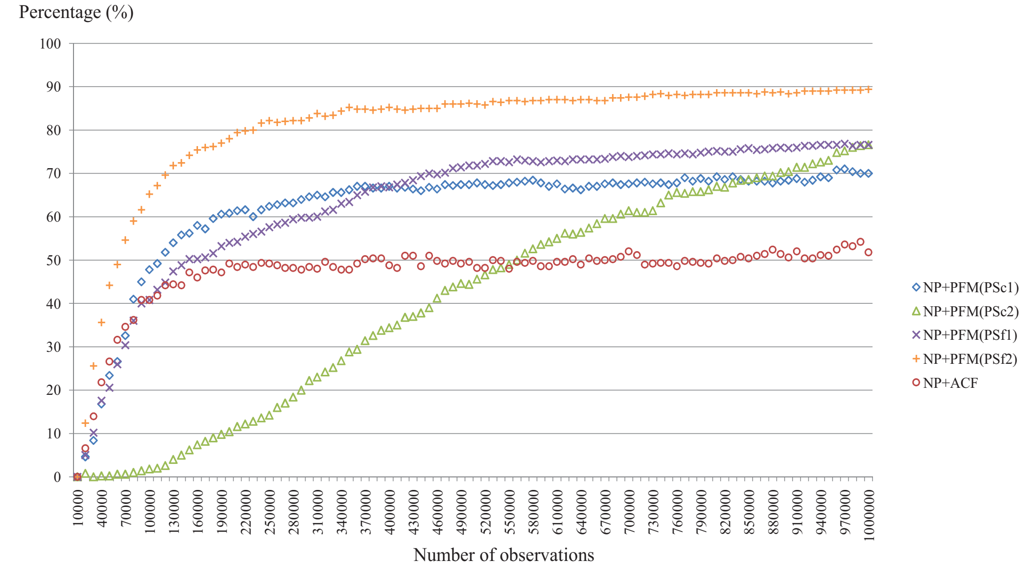


Figure 9: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ with constraint (5).

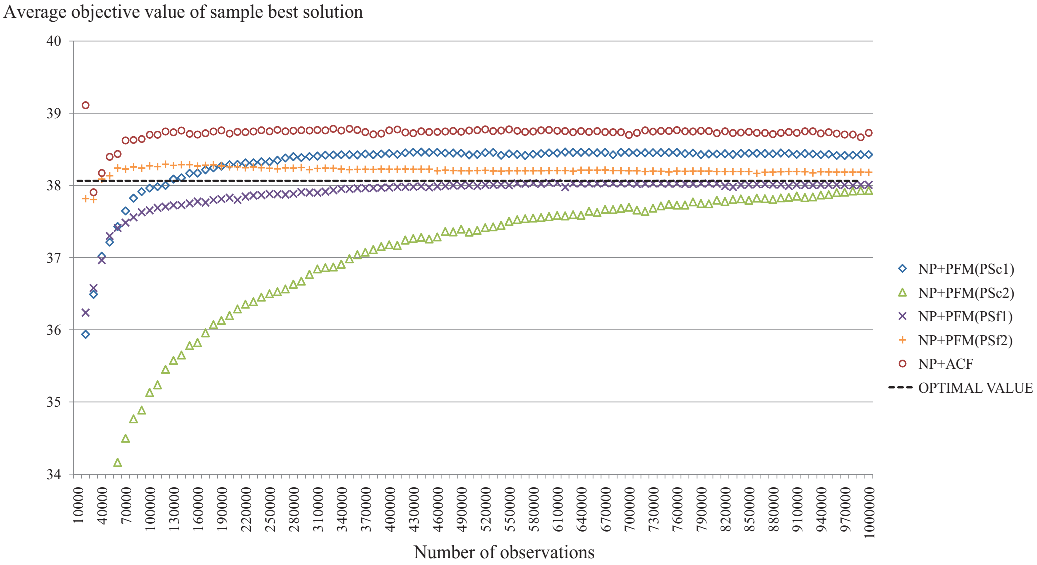


Figure 10: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ with constraint (5).

Note that average objective value of NP+PFM in Figures 10 (and Figure 13 later in the chapter) tends to start below the true optimal value and increases as the total number of observations increases. The true optimal solution $\mathbf{x}_o^b$ is tight and infeasible solutions near the true optimal have better (superior) primary performance measures

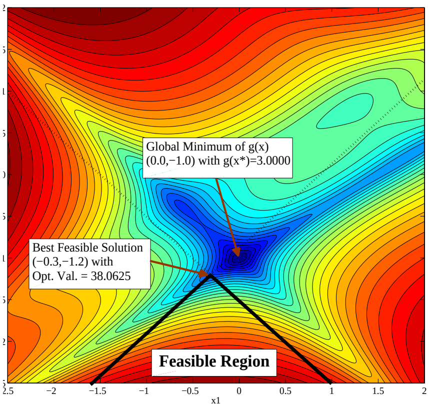


Figure 11: Contour of the Goldstein-Price function with constraints (6).

in the problem setting. Thus, NP+PFM selects one of superior but infeasible solutions as the current sample best at the beginning of search due to a small penalty. As the search goes on, infeasible solutions receive a large penalty value and this helps NP+PFM move away from infeasible solutions and select the true optimal solution or one of feasible solutions near $\mathbf{x}_o^b$.

Now we consider two stochastic constraints:

$$\mathbf{E}[-x_1 - x_2 + \psi_{1i}] \geq 1.5 \quad \text{and} \quad \mathbf{E}[x_1 - x_2 + \psi_{2i}] \geq 0.9. \tag{6}$$

Figure 11 shows that the feasible region becomes smaller than that of the single constraint case and $\mathbf{x}_o^b$ is located on the extreme point of two stochastic constraints with many superior infeasible solutions near $\mathbf{x}_o^b$. We arbitrary set $(\epsilon_{10}, \epsilon_{20}) = (0.00617, 0.0103)$ for PS$_{f1}$ and PS$_{f2}$. As the two constraints are independent and the optimal solution has two tight constraints, the probability that the penalty sequence of PS$_{c1}$ of the optimal solution converges to zero is expected to be $\rho_c \times \rho_c = 0.49$. For PS$_{c2}$, the probability would be 0.81.

Figure 12 shows that NP+PFM(PS$_{f1}$) and NP+PFM(PS$_{f2}$) return the true optimal around 80% of the time as the number of observations increases and NP+PFM(PS$_{c2}$) returns the true optimal around 70% while NP+PFM(PS$_{c1}$) find the true optimal

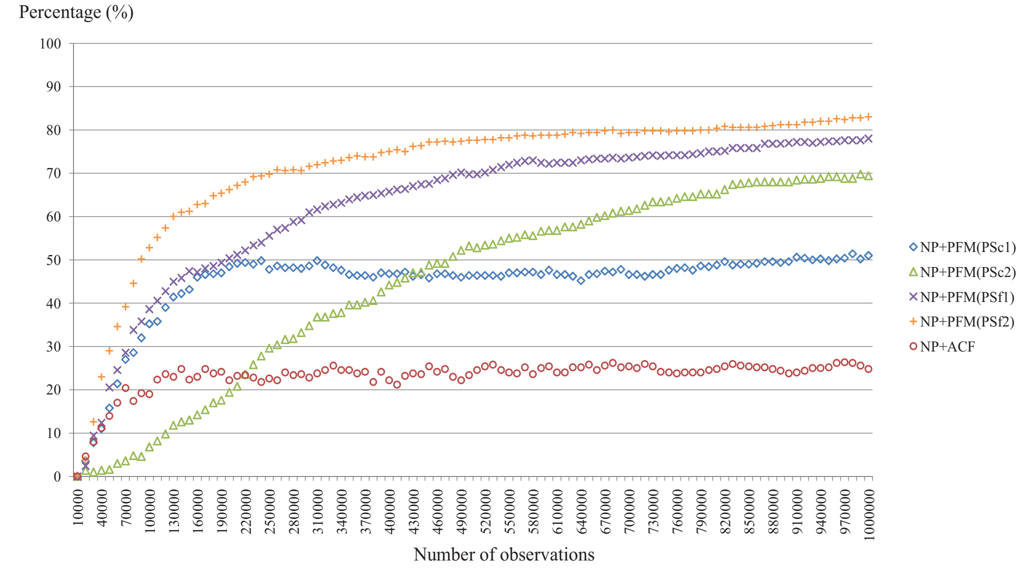


Figure 12: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ with constraints (6).

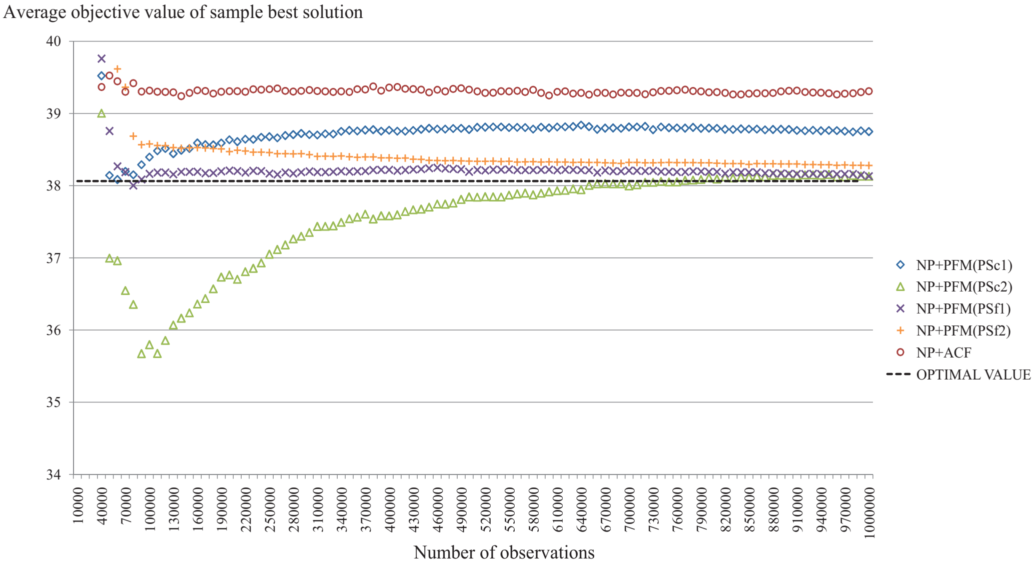


Figure 13: Average objective value of $\hat{\mathbf{x}}_k^*$ with constraints (6).

around 50% and NP+ACF find the true optimal only around 25% of the time. We observe that the probability increases to 81% for NP+PFM(PS$_{c2}$) as we further increase the total number of observations. NP+PFM(PS$_{f1}$) and NP+PFM(PS$_{f2}$) give a better estimated objective value than NP+PFM(PS$_{c1}$), NP+PFM(PS$_{c2}$) and NP+ACF as shown in Figure 13.

2.4.2 (s, S) Inventory Policy Problem

Now we consider an (s, S) inventory policy problem [19] with (i) non-normal observations for the secondary performance measure; and (ii) correlation between the primary and secondary performance measures.

If the inventory level at a review is found to be below s units, then an order is placed to increase the inventory level to S. If not, there is no order. Demand is assumed to be Poisson distributed with mean 25. The solution set is $\Theta = \{(s, S) | 20 \leq s \leq 80,\ 40 \leq S \leq 100,\ s \in \mathbb{Z},\ S \in \mathbb{Z}\}$, where $\mathbb{Z}$ is a set of integers.

We define the failure probability as the probability that a shortage occurs. We want to find values of s and S that minimize the steady-state expected inventory cost per review period while keeping the failure probability less than or equal to 0.01. All analytic results can be obtained by using a Markov chain model, and Figures 14 and 15 show the expected cost and the failure probability of solutions near the optimal solution of the unconstrained problem. The true optimal solution is $\mathbf{x}_o^b = (31, 61)$ and its expected cost and failure probability are 117.3428 and 0.00998, respectively. Many superior infeasible solutions are located near the true optimal solution $\mathbf{x}_o^b$, the stochastic constraint becomes nearly tight at $\mathbf{x}_o^b$, and it is difficult to accurately estimate the secondary performance measure with small samples. Therefore this is a very difficult optimization problem.

We arbitrarily choose $\epsilon_{10} = 0.001$ for PS$_{f1}$ and PS$_{f2}$.

Figure 16 shows the percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ over 500 macro replications when each macro replication is terminated with 2,000,000 total observations. At the end of the searches, NP+PFM(PS$_{f1}$), NP+PFM(PS$_{f2}$), and NP+PFM(PS$_{c2}$) achieve up to 90% and NP+PFM(PS$_{c1}$) shows the percentage around 77% while NP+ACF achieves only up to 40%. NP+PFM shows better average estimate objective values

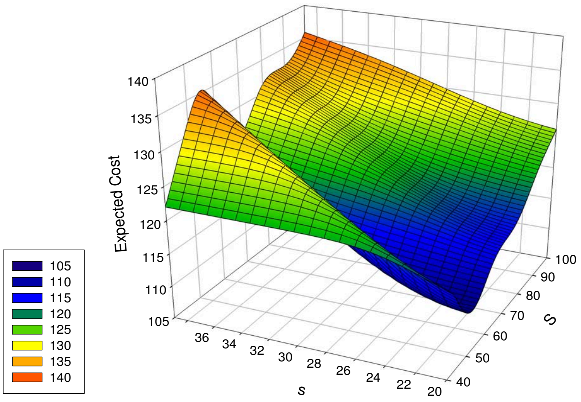


Figure 14: Steady-state expected cost/period of the (s, S) inventory problem.

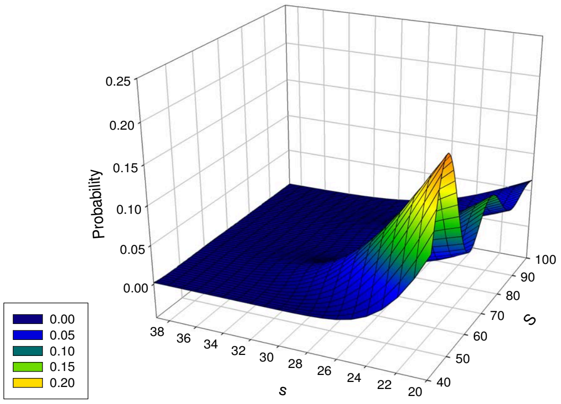


Figure 15: Steady-state failure probability of the (s, S) inventory problem.

than NP+ACF as in Figure 17.

2.4.3 Shortcoming and Recommendation

We use the three-system example to study the performance of PFM when $H_1(\mathbf{x})$ has an asymmetric distribution. More specifically, let Ψ represent an exponential random variable with mean 1. Then three distributions for the secondary performance measure are considered: (i) $H_1(\mathbf{x})$ is normally distributed (symmetric), (ii) $H_1(\mathbf{x}) = \mathbf{E}[H_1(\mathbf{x})]+(\Psi-1)$ (positively skewed), and (iii) $H_1(\mathbf{x}) = \mathbf{E}[H_1(\mathbf{x})]+(1-\Psi)$ (negatively skewed). We arbitrary set$\epsilon_{10} = 0.08$ for PS$_{f1}$ and PS$_{f2}$. Table 6 shows infeasible

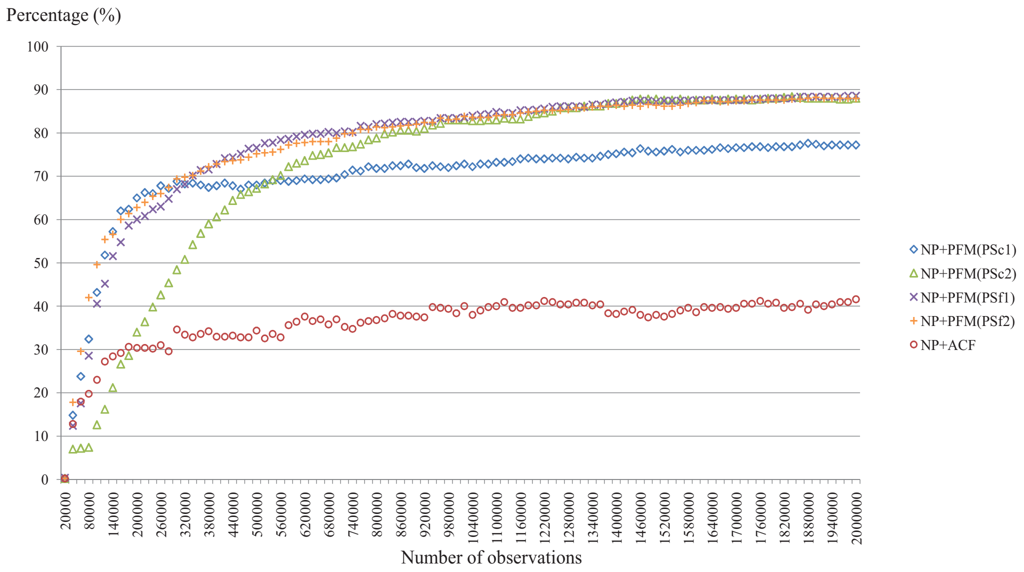


Figure 16: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ in the (s, S) inventory problem.

probabilities of three systems when $\Delta n = 1$ for the three distributions of $H_1(\mathbf{x})$.

Table 6: Infeasible probability with $\Delta n = 1$.

System	$p_1(\mathbf{x})$		
	symmetric $H_1(\mathbf{x})$	positively skewed $H_1(\mathbf{x})$	negatively skewed $H_1(\mathbf{x})$
1	0.3821	0.5034	0.2725
2	0.5000	0.6321	0.3679
3	0.6179	0.7275	0.4966

Figures 18, 19, and 20 show the percentages of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ with $\Delta n = 1$ over 500 macro replications when $H_1(\mathbf{x})$ are symmetric, positively skewed, and negatively skewed, respectively. Each macro replication continues until the total number of observations obtained so far reaches 10,000.

When $H_1(\mathbf{x})$ are symmetric, Figure 18 shows that PFM(PS$_{f1}$) and PFM(PS$_{f2}$) achieve almost 100%, PFM(PS$_{c1}$) and PFM(PS$_{c2}$) achieve close to their ρ_c (0.7 or 0.9), and the linear penalty finds the true optimal solution only 50% of the time. PFM(PS$_{c1}$), PFM(PS$_{f1}$) and PFM(PS$_{f2}$) show similar convergence rates for a small number of observations and this is expected because all three sequences essentially use the same values of ρ_c and θ_a up to $N_p = 10$ visits. However, PFM(PS$_{f1}$) and

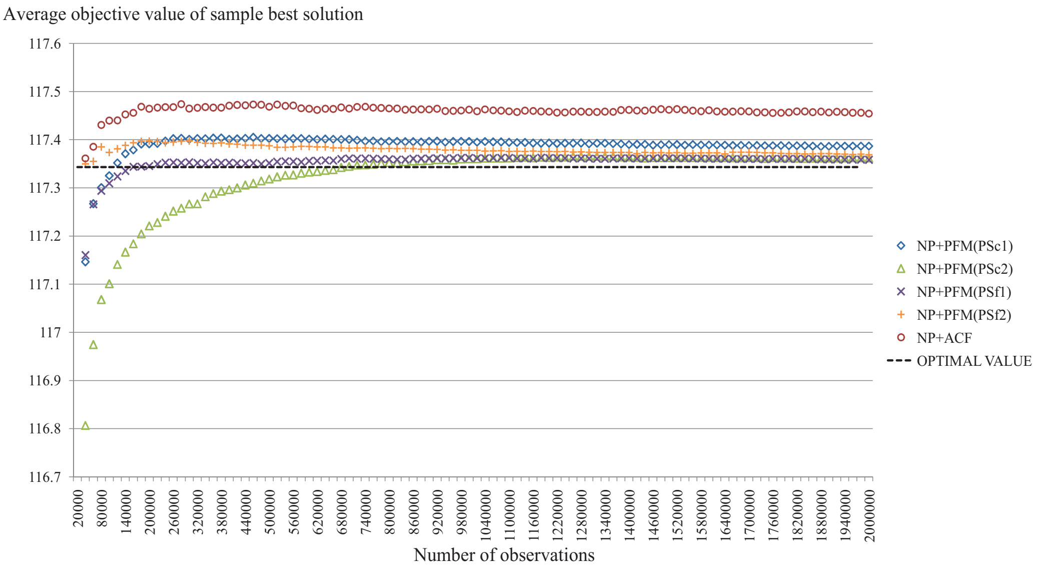


Figure 17: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ in the (s, S) inventory problem.

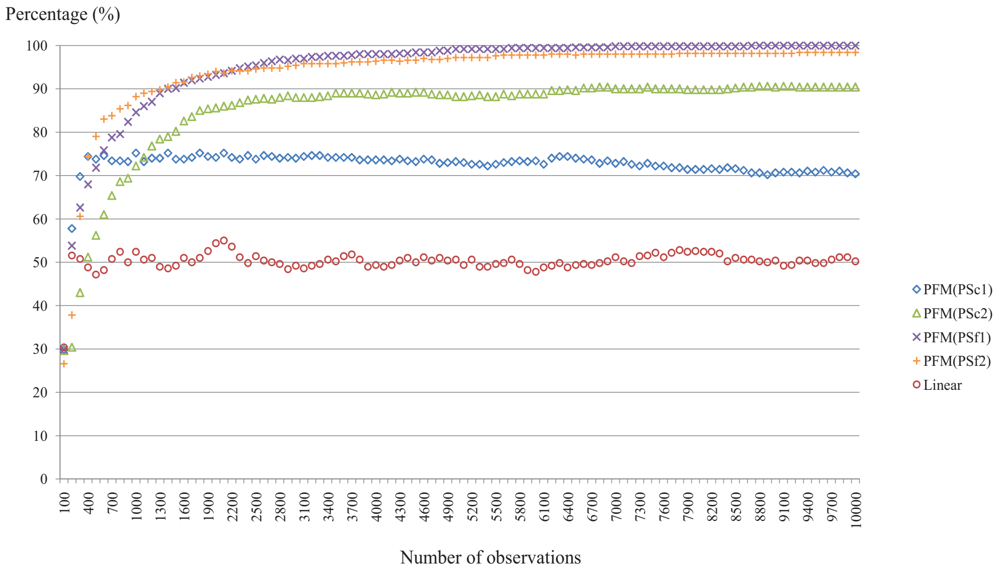


Figure 18: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ for the three-system example with normal $H_1(\mathbf{x})$.

PFM(PS$_{f2}$) achieve a lot better final convergence probability (close to 100%) than PFM(PS$_{c1}$). PFM(PS$_{c2}$) achieves a high convergence probability in the end but shows slow convergence, which is expected from the discussion in Section 2.3.2.

Figures 19 and 20 show that the performance of PFM(PS$_{c1}$) and PFM(PS$_{c2}$) under asymmetric distributions are similar to that under the normal case because their

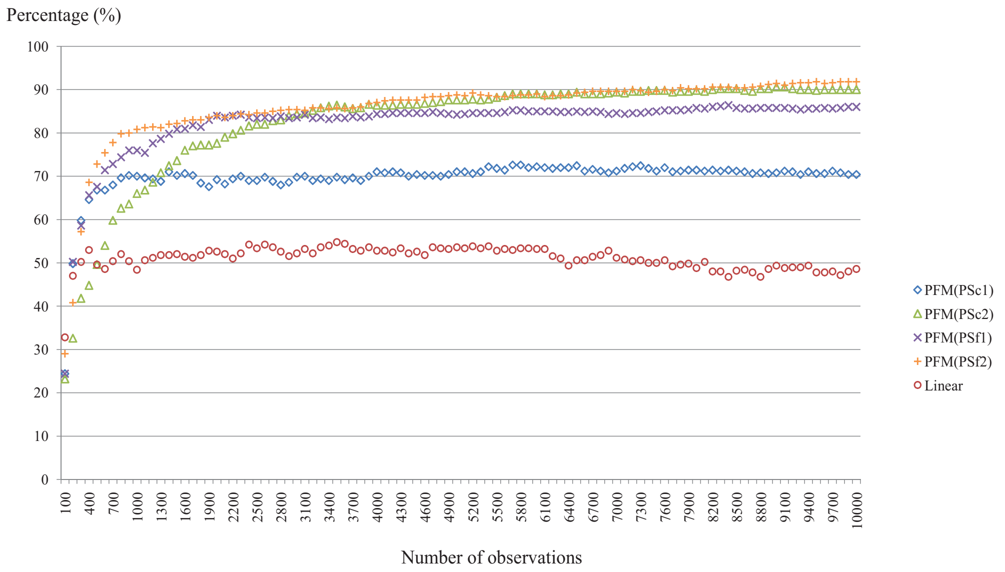


Figure 19: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ for the three-system example with positively skewed $H_1(\mathbf{x})$.

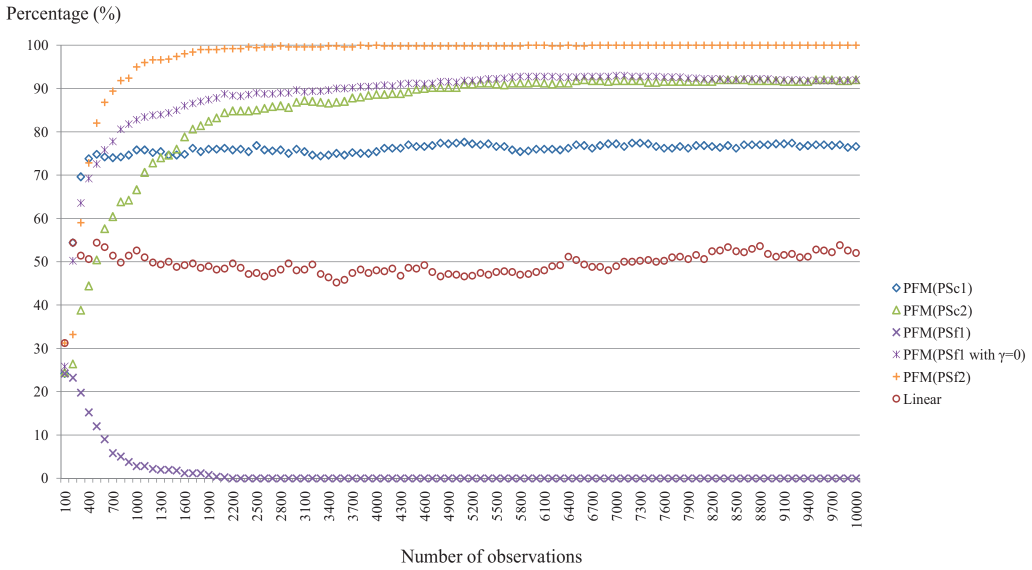
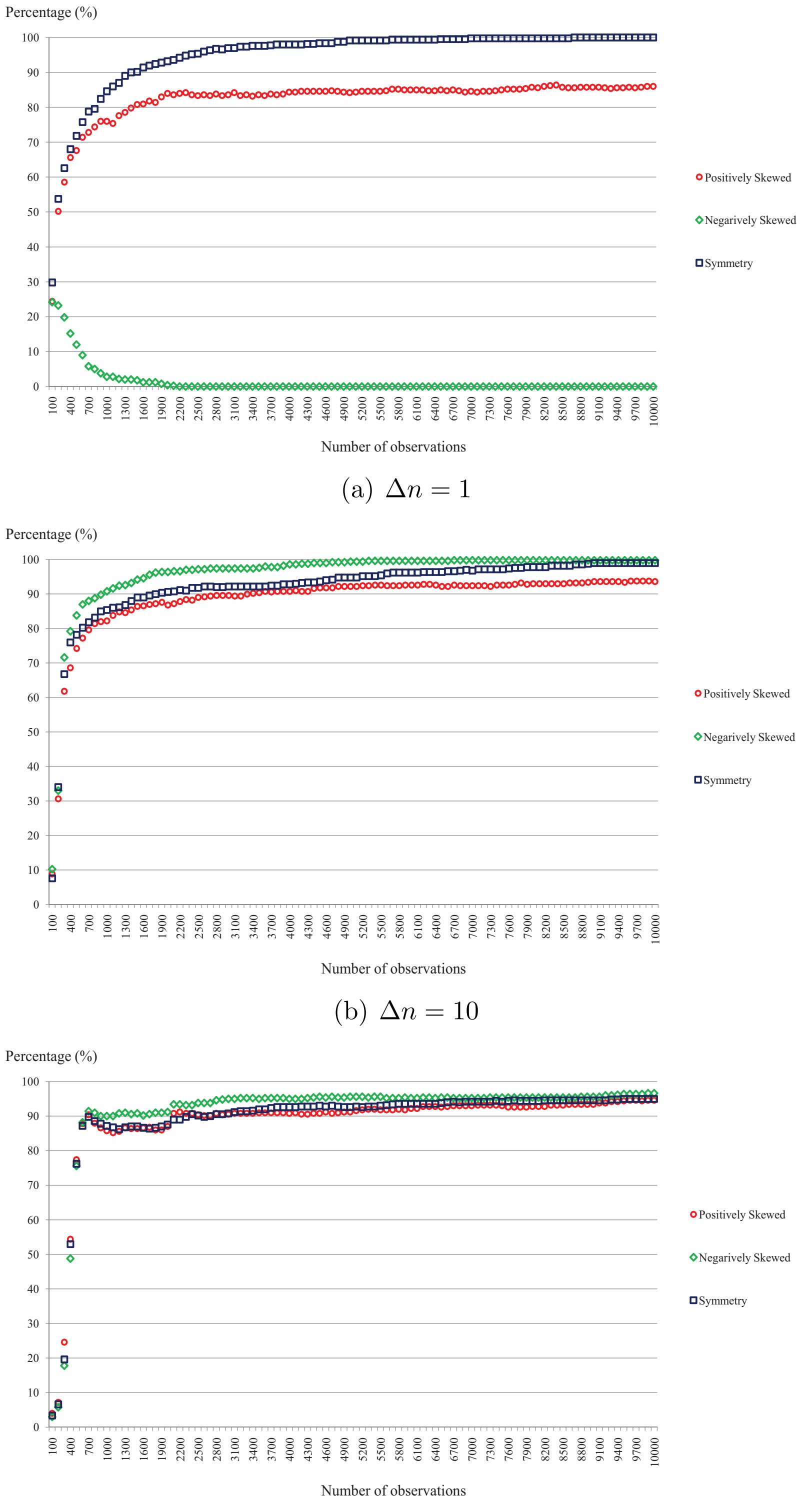


Figure 20: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ for the three-system example with negatively skewed $H_1(\mathbf{x})$.

behaviors do not depend on an underlying distribution of a secondary performance measure. PFM(PS$_{f2}$) also performs well regardless of the underlying distribution but Figure 19 shows slower convergence to 100% under the positively skewed distribution. Although we do not report the results in the chapter due to space, we confirmed that



(a) $\Delta n = 1$

(b) $\Delta n = 10$

(c) $\Delta n = 30$

Figure 21: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ using PS$_{f1}$ with different Δn with PS$_{f1}$.

the probability does approach to 100% as the total number of observations increases. On the other hand, the performance of PFM(PS$_{f1}$) is deteriorated especially when $H_1(\mathbf{x})$ is negatively skewed. In the positively skewed distribution, $p_\ell(\mathbf{x})$ of systems 2 and 3 are greater than 0.5 and the systems are likely to receive $(\rho_c, \alpha_\ell^{v_k}(\mathbf{x})) = (0.7, \sqrt{1.3})$ for $v_k(\mathbf{x}) \leq 10$ and then $(0.9, \sqrt{1.3})$ for $v_k(\mathbf{x}) > 10$. This explains why PFM(PS$_{f1}$) converges to 0.9 for the positively skewed $H_1(\mathbf{x})$. In the negatively skewed distribution of $H_1(\mathbf{x})$, $p_\ell(\mathbf{x})$ of the infeasible but superior system lies in the range of $[0.5 - \gamma_\ell^k, 0.5 + \gamma_\ell^k]$ which makes system 3 receive $0 < w_a < 1$ or $0 < w_d < 1$ quite often. This, in turns, makes $Z_k(\mathbf{x})$ of system 3 converge to $\mathbf{E}[G(3)] = -1$ which is better than the true optimal value, $\mathbf{E}[G(2)] = 0$. As a result, PFM(PS$_{f1}$) tends to choose system 3 as the sample best and performs no better than the naive linear penalty function.

This caveat of PFM(PS$_{f1}$) can be avoided simply by (i) setting $\gamma_\ell^k = 0$ so that PS$_{f1}$ essentially becomes adaptive PS$_c$ whose performance does not depend on the underlying distribution of $\zeta_{\ell r}(\mathbf{x})$; or (ii) using Δn large enough for $\zeta_{\ell r}(\mathbf{x})$ to be approximately symmetric. Figure 20 shows that the performance of PFM(PS$_{f1}$) with $\gamma_\ell^k = 0$ is similar to that of PFM(PS$_{c1}$) at the beginning and then PFM(PS$_{c2}$) later. Figures 21 also shows that PFM(PS$_{f1}$) performs similarly regardless of the underlying distribution of $H_1(\mathbf{x})$ as Δn increases. An experimenter can use a flow chart in Figure 22 to determine which penalty sequence to use.

2.5 Conclusions

In this chapter, we present PFM that converts a DOvS problem with stochastic constraints into a series of unconstrained DOvS problems. PFM determines a penalty value on a solution based on records of fast feasibility checks on the solution and it forces the penalty sequence converge to zero for a strictly feasible solution but diverges to ∞ for an infeasible solution. For even a tight solution, the penalty sequence

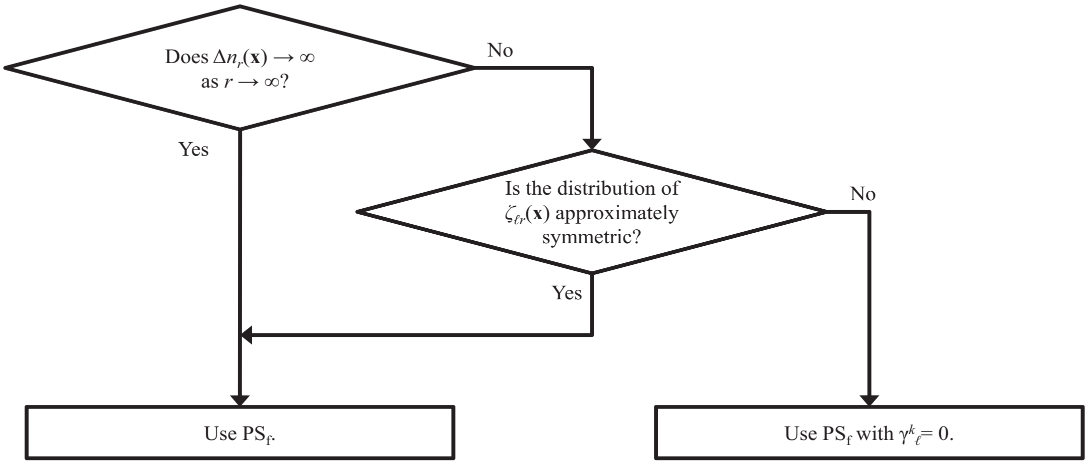

Does $\Delta n_r(\mathbf{x}) \to \infty$ as $r \to \infty$?
No
Yes
Is the distribution of $\zeta_{\ell r}(\mathbf{x})$ approximately symmetric?
No
Yes
Use $\mathrm{PS_f}$.
Use $\mathrm{PS_f}$ with $\gamma^k_\ell = 0$.

Figure 22: A flow chart for the determination of a penalty sequence.

converges to zero with high probability or almost surely under some conditions. We provide two penalty sequences, namely PS_c and PS_f, with proofs of their convergence properties and discuss parameter selections for their implementation. Our findings show that the implementation of PS_c is easier than PS_f and its performance does not depend on the underlying distribution of $H_\ell(\mathbf{x})$. However, its performance depends on the choice of parameters ρ_c and θ_a.

In addition, we show that PS_f with increasing $\Delta n_r(\mathbf{x})$ works well and is robust to the underlying distribution of $H_\ell(\mathbf{x})$. PS_f with fixed $\Delta n_r(\mathbf{x}) = \Delta n$ also works well as long as Assumption 3 is satisfied. If Assumption3 is not satisfied, it could introduces a caveat especially when the distribution is negatively skewed and $\Delta n_r(\mathbf{x})$ is small. Since it is unlikely that means and variances of solutions in the search space and the location of a the optimal solutions are known prior to simulation, either PS_f with increasing $\Delta n_r(\mathbf{x})$ or PS_f with finite Δn and $\gamma^k_\ell = 0$ appears to be a reliable choice.

In this chapter, we make a feasibility decision simply based on comparison between an estimate of secondary performance measures and q_ℓ. However, others (e.g.,

[4], [6], [16], and [29]) have developed more sophisticated statistical methods for feasibility checks. These sophisticated methods can be combined with PFM to further enhance the performance of the combined algorithm $\mathcal{D}$+PFM. This topic is discussed in Chapter IV.

CHAPTER III

DESIGNING OPTIMAL WATER QUALITY MONITORING NETWORK FOR RIVER SYSTEMS USING CONSTRAINED DISCRETE OPTIMIZATION VIA SIMULATION

Maintaining good water quality for river systems is an important problem in environmental engineering. Many researchers have studied effects of contaminants, water purifying techniques, and systemized water quality monitoring. This chapter deals with the problem of designing a water quality monitoring network for river systems where the goal is to find the optimal location of a finite number of monitoring devices to detect a potential contamination event in a river network. Some researchers suggest that a good network design should have small expected detection time and high detection reliability. This problem is difficult because (a) a river system is large and complicated, requiring huge computational time for its process simulation, (b) the two performance measures (detection time and reliability) are observed only through stochastic simulation when uncertainties in spill and rain events are considered, and (c) detection time and reliability criteria are opposing criteria requiring balanced optimal solutions.

Many researchers have studied water quality monitoring for river systems. For example, [35] give a comprehensive review of past approaches to the factors that affect effective water quality monitoring network design. [27] design a monitoring network based on a geometric analysis of river systems and demonstrate a simple application in a hypothetical river system. [38] show that a dynamic analysis of contaminant fate and transport may be necessary to solve this problem by formulating a bi-objective problem (minimizing the detection time and maximizing the reliability) for the hypothetical river; and find the optimal placement of monitoring devices using a genetic algorithm (GA) under relatively simple discrete uniform distributions on spill events. [39] extend their previous work to a more complicated model that accounts for rain events and a larger-sized river, namely, the Altamaha River. In their work, the two objectives are combined into one objective by assigning a large penalty value whenever a spill is not detected. They obtain a pre-determined number of observations to estimate the combined objective prior to optimization and apply the GA, assuming that the number of observations is large enough to ensure that an estimated value for the combined objective function is close to its true expected value. Although their proposed GA is shown to produce a good solution, (i) the GA itself is a heuristic algorithm without any guarantee about convergence to the true best solution; (ii) it is well known that the number of pre-determined observations may be too few, causing high estimation error, or too many, wasting computation time especially when one run takes long. For example, see [24]. Also, due to the use of a large penalty value when they combine two objectives, the GA tends to return a solution with 100% or the highest possible reliability. If a decision maker is interested in finding a solution with the smallest detection time among the solutions whose reliability is at least, say, 95%, the GA may not be the best method because it does not have any control on feasibility of the returned solution.

In this chapter, the water quality monitoring design problem is formulated as an optimization via simulation (OvS) problem with a constraint on reliability, assuming that a decision maker wants to identify the best solution among those whose reliability level is greater than or equal to a constant. In the simulation society, a number of algorithms for an OvS problem have been developed and shown to perform significantly better than heuristic algorithms in a variety of stochastic optimization problems where the objective function needs to be estimated by simulation. An

OvS algorithm sequentially obtains additional observations as needed until stopping criteria are satisfied and finds the best or a near-best solution with global or local convergence.

The previous chapter presents thorough theoretical developments and convergence proofs of PFM. From the convergence properties of PFM, one can choose a good penalty sequence for PFM that is reliable and robust under different mean and variance configurations of solutions in the search space of an optimization problem in consideration. Thus, the contributions of this chapter are on addressing the choices of implementation parameters of NP+PFM specifically tuned for the water quality monitoring design problem; demonstrating that the new optimization algorithm works significantly better than a popularly used method in environmental management; and solving the water quality monitoring design problem for the Altamaha River and studying how the best location changes when more random factors (that have not considered in literature) are considered.

This chapter is organized as follows: Section 3.1 formulates the problem considered in the chapter, describes process simulation for hydrodynamics and contaminant transport in a river, introduces frameworks of DOvS algorithms and PFM, respectively, and gives the NP+PFM algorithm. Section 3.2 discusses detailed implementation issues of the NP+PFM algorithm for designing the optimal water quality monitoring network. Experimental results of applying the NP+PFM algorithm to the Altamaha River are summarized in Section 3.3, followed by concluding remarks in Section 3.4.

3.1 Background

This section includes problem formulation and a description of process simulation with hydrodynamics and contaminant transport in river systems. Then the section introduces general frameworks for DOvS algorithms and PFM, and presents an algorithmic statement of NP+PFM. All notation needed throughout the chapter is summarized in a table of Appendix B.

3.1.1 Problem

A river network system has N nodes and each node can be a possible monitoring station or a possible spill location. Let I be the index set, $I = \{1, 2, ..., N\}$. The number of monitoring devices is M and $M < N$. Each solution $\mathbf{x}$ represents a location of M devices and is denoted as an M dimensional vector, $\mathbf{x} = (x_1, x_2, ..., x_M)$ such that $x_u \in I$ for $u = 1, ..., M$ and $x_u \neq x_v$ for $u \neq v$. It is also assumed that $x_1 < x_2 < ... < x_M$ to prevent the repetition of solutions (e.g., (1, 2, 3), (3, 2, 1), (2, 1, 3) etc. are the same solution). Set Θ is defined as the set of all possible solutions.

Let $t_d(x_u)$ represent detection time at the monitoring location x_u which is the first time when the concentration level at x_u goes over the detection threshold of a monitoring device, C_{th}. Then the elapsed detection time at x_u is defined as,

$$d(x_u) = \begin{cases} t_d(x_u) - S^S, & \text{if a contaminant is detected at } x_u; \\ \infty, & \text{otherwise,} \end{cases}$$

where S^S is a contaminant injection time (i.e., the starting time of a contaminant spill event). The minimum elapsed detection time for $\mathbf{x}$ is defined as

$$t(\mathbf{x}) = \min_{1 \leq u \leq M} d(x_u).$$

An indicator $R(\mathbf{x})$ is

$$R(\mathbf{x}) = \begin{cases} 0, & \text{if none of monitoring devices detects a contaminant (i.e., } t(\mathbf{x}) = \infty); \\ 1, & \text{otherwise.} \end{cases}$$

The two main outputs are $t(\mathbf{x})$ and $R(\mathbf{x})$ and they are only observed by stochastic simulation. Notation $t_i(\mathbf{x})$ and $R_i(\mathbf{x})$ represent observations obtained from the ith run of stochastic simulation.

Let $\mathbf{E}[Y]$ and $\mathbf{Var}[Y]$ represent expectation and variance of a random variable Y, respectively. The expected reliability is defined as $\mathbf{P}(R_i(\mathbf{x}) = 1)$, the probability that a spill is detected. The minimum required reliability level is denoted as q. Then the problem is formulated as follows:

$$\begin{aligned} &\operatorname{argmin}_{\mathbf{x}\in\Theta} && \mathbf{E}[t_i(\mathbf{x}) \mid R_i(\mathbf{x}) = 1] \\ &\text{subject to} && \mathbf{P}(R_i(\mathbf{x}) = 1) \geq q, \\ & && t_i(\mathbf{x}) \in \mathbb{R}^+,\ R_i(\mathbf{x}) \in \{0, 1\},\ 0 < q < 1. \end{aligned} \tag{7}$$

Note that the objective function is the conditional expectation of $t_i(\mathbf{x})$ given the event that a spill is detected. It is assumed that $\mathbf{E}[t_i(\mathbf{x}) \mid R_i(\mathbf{x}) = 1]$, $\mathbf{Var}[t_i(\mathbf{x}) \mid R_i(\mathbf{x}) = 1]$, $\mathbf{E}[R_i(\mathbf{x})]$ and $\mathbf{Var}[R_i(\mathbf{x})]$ exist and are finite real numbers, which implies Assumption 1 satisfied.

The next subsection describes how to generate $t_i(\mathbf{x})$ and $R_i(\mathbf{x})$ using a process simulation with randomness in contaminant spill and rainfall events.

3.1.2 Process Simulation

Process simulation is needed to simulate hydrodynamics and contaminant transport in a river system. A popular software package is the Storm Water Management Model (SWMM) developed by U.S. Environmental Protection Agency (EPA). As in the EPA user manual [30], SWMM is capable of simulating a dynamic flow model with rainfall events and a variety of watershed conditions for an urban area. SWMM takes as inputs (i) geologic and geometric data and basic hydrodynamics data to construct the river, (ii) spill location, spill intensity, and spill time of contaminant, and (iii) rain intensities over a time period for each location. In this chapter, geologic and basic hydrodynamics are fixed but contaminant spill and rainfall events contain randomness.

Three random variables are needed to simulate a spill event for the ith run: the spill location L_i^S, spill intensity I_i^S, and spill starting time S_i^S. Also, we need a random variable which simulates a rainfall event, and there are a number of ways to generate the events. [39] divide the whole river region into M^s number of subregions, called sub-catchments. Each sub-catchment contains neighborly nodes in a subregion and a rain pattern for the sub-catchment is randomly chosen from a number of pre-generated rain patterns. Let P^R_{im} represent a rain pattern for the mth sub-catchment in the ith simulation run. A vector $P^R_i = (P^R_{i1}, \ldots, P^R_{iM^s})$ denotes rain patterns for the entire region in the ith simulation run.

For randomly generated input data (L^S_i, I^S_i, S^S_i, P^R_i) and geologic information (locations, elevations, and shapes of nodes and reaches placed between nodes), one process simulation run returns a large binary output file including concentration levels at each node at every constant inter-reporting time of the simulation clock. Each output file provides one realization of $t_i(\mathbf{x})$ and one realization of $R_i(\mathbf{x})$.

3.1.3 DOvS and PFM

In this section, we explain how we combine a DOvS algorithm with the process simulation and briefly review PFM. As shown in Figure 23, at each search iteration, a DOvS algorithm samples candidate solutions $\mathbf{x}$ and obtains additional observations from each sampled solution. If stopping criteria are satisfied, the algorithm stops and returns the current best solution as the optimal solution. Otherwise, it updates the solution sampling strategy and repeats previous steps. Unfortunately, existing DOvS algorithms assume deterministic constraints and cannot handle stochastic constraints. Thus, they are not directly applicable to the problem considered in this chapter because it has a stochastic constraint on reliability. The previous chapter proposes PFM that enables a DOvS algorithm to solve stochastically constrained DOvS problems. Note that the process simulation requires extensive computational costs but a DOvS algorithm does not. Also, results from the process simulation can be reused if they have been generated and saved. Thus, a simpler version of PFM,

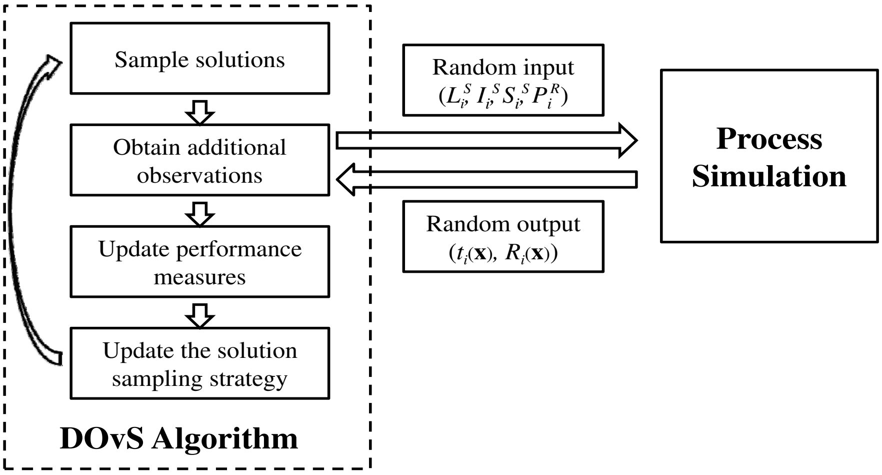

Sample solutions
Obtain additional observations
Update performance measures
Update the solution sampling strategy
DOvS Algorithm
Random input
$(L_i^S I_i^S S_i^S P_i^R)$
Random output
$(t_i(\mathbf{x}), R_i(\mathbf{x}))$
Process Simulation

Figure 23: General structure of DOvS with process simulation.

the penalty sequence with constants (PS$_c$), is used in this chapter. Before presenting the simpler version of PFM, additional notation is needed.

$\overline{T}_k(\mathbf{x}) := \frac{1}{\sum_{i=1}^{n_{v_k}(\mathbf{x})} R_i(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} t_i(\mathbf{x}) R_i(\mathbf{x})$, conditional sample mean of $t_i(\mathbf{x})$ up to iteration k given the event that a spill is detected;
$\overline{R}_k(\mathbf{x}) := \frac{1}{n_{v_k}(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} R_i(\mathbf{x})$, cumulative sample mean of $R_i(\mathbf{x})$ up to iteration k.

PFM consists of a measure of constraint violation and a penalty sequence. For Problem (7), a measure of constraint violation is $\max(0, q - \overline{R}_k(\mathbf{x}))$ and with a constant $\Delta n_r(\mathbf{x}) = \Delta n$, penalty sequence $\lambda^{v_k}(\mathbf{x})$ is defined as

$$\lambda^{v_k}(\mathbf{x}) = \begin{cases} \lambda^{v_{k-1}}(\mathbf{x}) \times \theta_a, & \text{if } \overline{R}_k(\mathbf{x}) < q; \\ \lambda^{v_{k-1}}(\mathbf{x}) \times \theta_d, & \text{if } \overline{R}_k(\mathbf{x}) \geq q, \end{cases} \tag{8}$$

where $\lambda^0(\mathbf{x}) = \lambda^0$ is an initial penalty constant for the constraint, θ_a is an appreciation factor, and θ_d is a depreciation factor such that $\theta_a > 1$ and $0 < \theta_d < 1$. With PFM, $Z_k(\mathbf{x}) = \overline{T}_k(\mathbf{x}) + \lambda^{v_k}(\mathbf{x}) \times \max\{0, q - \overline{R}_k(\mathbf{x})\}$ is calculated. Then, to solve Problem (7), a solution with the smallest $Z_k(\mathbf{x})$ is selected as the current best at search iteration

k.

Section 2.2 shows that a solution with the smallest $Z_k(\mathbf{x})$ at search iteration k converges to the true best feasible solution to Problem (7) as k goes to infinity when (i) PFM is designed in a way that the penalty value of a feasible solution converges to zero but diverges to infinity for an infeasible solution and (ii) a globally convergent DOvS algorithm is applied with the PFM. The version of PFM presented above does satisfy the convergence property when no solution is located right on the boundary of feasible/infeasible regions (i.e., no active constraint). However, if a solution has an active constraint, the penalty value of a solution with an active constraint converges in distribution to a random variable with two possible values: zero and infinity as in Section 2.2.2. Recall that Section 2.3.2 provides a direction for choosing good parameters which balance between fast convergence and high probability of returning a true optimal solution when PS_c is used.

3.1.4 NP+PFM

Among DOvS algorithms, we choose a version of NP presented in [28]. NP focuses on a special region called the most promising region and spends more computational efforts in the most promising region by sampling more solutions from it. More specifically, NP systematically partitions the promising region into a number of subregions. Then it samples and assesses solutions from the subregions. At the same time, it keeps sampling some solutions from the region outside the most promising region called the surrounding region. If the current sample best solution is in one of the subregions, the subregion with the current sample best will be the next most promising region. Otherwise, the most promising region becomes the whole set, Θ. Prior to the description of NP+PFM, some additional notation for NP is defined first:

$\mathcal{R}_k$:= the most promising region at search iteration k;

$\mathcal{R}_k(\ell)$:= the ℓth subregion at search iteration k;

$\Theta \setminus \mathcal{R}_k$:= the surrounding region at search iteration k;

$\mathcal{S}_k$:= the set of solutions sampled at iteration k;

$\mathcal{V}_k$:= the set of all solutions visited up to iteration k;

ω := the number of subregions;

τ_k := the total number of sampled solutions at iteration k; and

$\tau_k(\ell)$:= the number of sampled solutions at iteration k from subregion ℓ.

We specialize NP+PFM for the water quality monitoring design problem and its detailed steps are given in Figure 24.

[34] point out that the performance of NP on a combinatorial type problem highly depends on how to index nodes, partition a search space, and sample solutions. The next section discusses how to efficiently perform each step in NP+PFM including stopping criteria.

3.2 Implementation

This section addresses implementation issues in NP+PFM for the water quality monitoring design problem including indexing, partitioning, sampling, and stopping.

3.2.1 Indexing

Many search methods tend to generate alternative solutions from neighbors of the current best solution. In this chapter, a solution is a M-dimensional vector whose elements are in the increasing order of integers. Changing a few elements in $\mathbf{x}$ up and down generates neighbor solutions. As NP+PFM spends more computing efforts on the most promising region, it would be desirable if neighbors of a solution tend to share the same feasibility with the current solution.

The idea is to index each node based on quality where the quality is defined as the probability that a spill is detected, assuming only one monitoring device is placed at the node and the device does not miss a spill under any circumstances. Then

Algorithm NP + PFM

Step 0. Initialization:
- Perform the indexing algorithm (Section 3.2.1.).
- Set $k = 1$, $\mathcal{R}_k = \Theta$, and $\mathcal{V}_k = \emptyset$.
- Sample an initial solution $\hat{\mathbf{x}}_0^*$ randomly from Θ.
- Select constants ω, τ_k, Δn, $\lambda^0(\mathbf{x})$, θ_a and θ_d.

Step 1. Partitioning: (Section 3.2.2.)
- Partition $\mathcal{R}_k$ into ω disjoint subregions, $\mathcal{R}_k(1), \mathcal{R}_k(2), ..., \mathcal{R}_k(\omega)$. If $\mathcal{R}_k$ is a singleton, set $\mathcal{R}_k(1) = \mathcal{R}_k$ and $\mathcal{R}_k(2) = \ldots = \mathcal{R}_k(\omega) = \emptyset$.
- Set $\mathcal{R}_k(\omega + 1) = \Theta \setminus \mathcal{R}_k$ which denotes the surrounding region.

Step 2. Sampling Solutions: (Section 3.2.3.)
- From each region $\mathcal{R}_k(j), j = 1, 2, ..., \omega + 1$, sample $\tau_k(j)$ solutions. Always sample $\hat{\mathbf{x}}_{k-1}^*$ so that $\hat{\mathbf{x}}_{k-1}^* \in \mathcal{S}_k$.
- Include all sampled solutions $\mathbf{x}$ into $\mathcal{S}_k$.
- If $\mathbf{x} \notin \mathcal{V}_k$ for any $\mathbf{x} \in \mathcal{S}_k$, then $\mathcal{V}_k = \mathcal{V}_k \cup \{\mathbf{x}\}$.

Step 3. Estimating the Promising Index:
- For each $\mathbf{x} \in \mathcal{S}_k$, take Δn observations, set $n_{v_k}(\mathbf{x}) = n_{v_{k-1}}(\mathbf{x}) + \Delta n$ where $n_0(\mathbf{x}) = 0$, for any $\mathbf{x} \in \Theta$, and update $Z_k(\mathbf{x})$.
- Select $\hat{\mathbf{x}}_k^*$ such that $\hat{\mathbf{x}}_k^* \equiv \text{argmin}_{\mathbf{x} \in \mathcal{V}_k} Z_k(\mathbf{x})$.

Step 4. Selecting the Most Promising Region and Backtracking:
- Determine j^* such that $\hat{\mathbf{x}}_k^* \in \mathcal{R}_k(j^*)$.
- If $\mathcal{R}_k(j^*) \subset \mathcal{R}_k$, then $\mathcal{R}_{k+1} = \mathcal{R}_k(j^*)$. Otherwise, $\mathcal{R}_{k+1} = \Theta$.
- Set $k = k + 1$.

Step 5. Stopping Rule (Section 3.2.4.)
- If the stopping rule is satisfied, then stop and return $\hat{\mathbf{x}}_k^*$ as the best solution. Otherwise go to Step 1.

Figure 24: Algorithm NP + PFM for the water quality monitoring design problem.

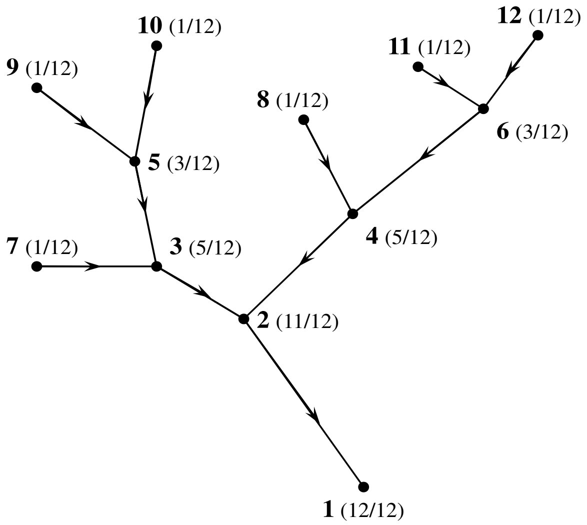


Figure 25: The hypothetical river with 12 nodes indexed based on quality.

nodes are indexed in the decreasing order of the quality. If there are nodes with the same quality, they are indexed by the increasing order of the distance from the node with the highest quality. Figure 25 shows quality levels in parenthesis for all nodes in the hypothetical river [27], assuming equal spill probability at each node (i.e., 1/12), and the indices of nodes based on the quality levels. This indexing method helps neighbors of a solution show similar levels of reliability.

3.2.2 Partitioning

The current most promising region $\mathcal{R}_k$ needs to be partitioned into two subregions, $\mathcal{R}_k(1)$ and $\mathcal{R}_k(2)$, if $\omega = 2$. The partitioning scheme in this chapter is based on the bisection method. More specifically, the algorithm selects an element of $\mathbf{x}$ whose range of possible values contains two or more integers. Then the partitioning is done by dividing the range into approximately two equal ranges.

Figure 26 illustrates the partitioning scheme for the monitoring system with three monitoring devices in the hypothetical river, assuming no backtracking occurs by $k = 5$. Although the partitioning is presented only for $\omega = 2$ in this chapter, it can

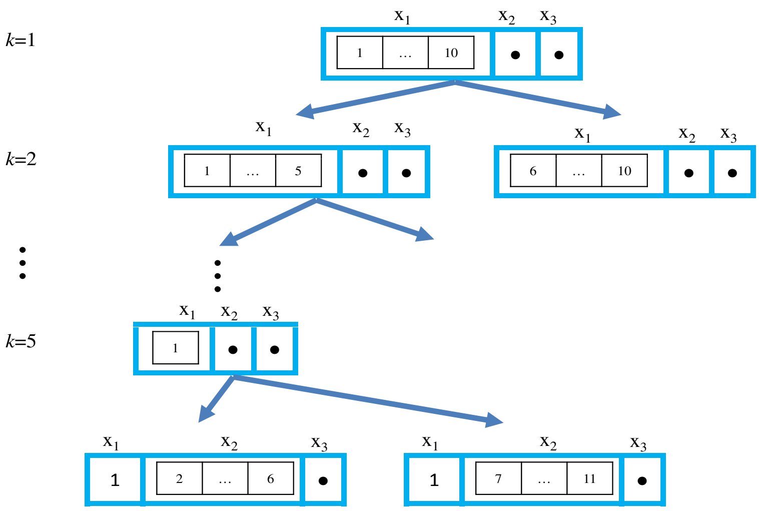


Figure 26: Partitioning for the hypothetical river.

be easily extended to $\omega \geq 3$, in which case the range can be divided into ω number of equal ranges.

3.2.3 Sampling

Solutions are sampled from three regions: two subregions, $\mathcal{R}_k(1)$ and $\mathcal{R}_k(2)$, and the surrounding region, $\Theta \setminus \mathcal{R}_k$. Since the size of two subregions can be different, $\tau_k(1)$ and $\tau_k(2)$ are set to proportional to the sizes of subregions 1 and 2, respectively. Although the size of the surrounding region $\Theta \setminus \mathcal{R}_k$ tends to be much bigger than $\mathcal{R}_k$, the $\lfloor \tau_k/2 \rfloor$ or $|\mathcal{R}_k|$ number, whichever is smaller, of solutions are sampled from $\mathcal{R}_k$ and then the rest $\tau_k - \tau_k(1) - \tau_k(2)$ number of solutions are sampled from the surrounding region. This ensures that NP+PFM visits more solutions in the most promising region. Moreover, $\tau_k(1)$ and $\tau_k(2)$ are proportional to the sizes of $\mathcal{R}_k(1)$ and $\mathcal{R}_k(2)$, respectively. Detailed steps of the sampling scheme is given in Figure 27.

In general, it is important that a search algorithm visits a few number of neighbor solutions to the current best solution because good solutions tend to be herded together. Visiting neighbor solutions is more important for OvS problems with stochastic constraints. Finding the best feasible solution becomes difficult when the best

Sampling scheme

1. Set $\begin{cases} \tau_k(1) = \min\Big(|\mathcal{R}_k(1)|, \max\Big(1, \lfloor \frac{|\mathcal{R}_k(1)|}{|\mathcal{R}_k|} \cdot \frac{\tau_k}{2} \rfloor\Big)\Big); \\ \tau_k(2) = \min\Big(|\mathcal{R}_k(2)|, \lfloor \frac{\tau_k}{2} \rfloor - \tau_k(1)\Big); \text{ and} \\ \tau_k(3) = \tau_k - \tau_k(1) - \tau_k(2). \end{cases}$

2. Sample $\tau_k(1)$ and $\tau_k(2)$ number of different solutions from $\mathcal{R}_k(1)$ and $\mathcal{R}_k(2)$ using the uniform sampling, respectively.

3. If $\mathcal{R}_k$ is not a singleton (i.e., $|\mathcal{R}_k| \geq 2$), sample $\tau_k(3)$ number of different solutions by the uniform sampling from $\Theta \setminus \mathcal{R}_k$. Otherwise, sample $\lfloor \frac{\tau_k}{2} \rfloor$ number of different solutions by the local search sampling and the rest number of different solutions by the uniform sampling from $\Theta \setminus \mathcal{R}_k$.

Figure 27: Sampling scheme for NP+PFM.

feasible solution is the one whose reliability is close to the minimum reliability level, q. An extreme case occurs when the reliability of the best feasible solution is exactly equal to q, in which case there always exists positive probability that the solution is declared as infeasible at each visit to the solution. This, in turns, increases the chance of labeling the region that contains the best feasible as the surrounding region. PFM is designed to ensure that NP+PFM eventually selects the best feasible as long as there is nonzero probability of visiting the solution. When the most promising region is not a singleton, the sampling scheme ensures visiting some neighbor solutions to the current best. To ensure that this happens also when $\mathcal{R}_k$ is a singleton, a local search sampling is adopted. In the local search sampling, two methods are used. The first method randomly selects an element x_{u^*} from the current best solution in the singleton set $\mathcal{R}_k$, and then change the value of x_{u^*} by ±1 randomly. The second method varies the value of x_{u^*} to any integer between x_{u^*-1} and x_{u^*+1}. For example, if the solution in a singleton set $\mathcal{R}_k$ is $(1, 4, 7)$ and $x_{u^*} = 4$ (or $u^* = 2$), then the first method randomly selects either $(1, 3, 7)$ or $(1, 5, 7)$ with equal probability and the second method generates one solution among $(1, 2, 7), (1, 3, 7), (1, 5, 7)$, and $(1, 6, 7)$ with equal probability. About half of $\lfloor \tau_k/2 \rfloor$ number of solutions are generated using the first method and the other half number of solutions are generated using the second method.

3.2.4 Stopping Criteria

The global convergence is achieved when k goes to infinity but, in practice, the algorithm should terminate with finite search iterations. [14] give some popular stopping criteria. In this chapter, the following stopping criterion is used: the algorithm stops when event E_1 occurs consecutively n_E times, where

$$E_1 := \{\hat{\mathbf{x}}^*_k = \hat{\mathbf{x}}^*_{k-1} \ , \ |Z_k(\hat{\mathbf{x}}^*_k) - Z_{k-1}(\hat{\mathbf{x}}^*_{k-1})| < \epsilon \ , \ \mathcal{R}(k) \text{ is a singleton}\} \tag{9}$$

for a small positive constant ϵ. The decision maker needs to choose ϵ and n_E.

3.3 Case Study

This section presents the performance of NP + PFM compared to the GA algorithm in [39] on the water quality monitoring design problem for the Altamaha River. The Altamaha River is located in Georgia, U.S.A. and known to have the largest watershed in the State. Figure 28 shows the Altamaha River with one hundred nodes which are located on the most upstream points, confluences, and points evenly distributed along each river reach. Each node can be a possible monitoring location or a spill location. The Altamaha River is composed of 60 river reaches and total 62 river junctions. To construct river system, U.S. Geological Survey (USGS) in the National Elevation Dataset is used as in [39]. Each process simulation continues until the simulation clock reaches 40 days and uses every 15 minutes of the simulation clock as the inter-reporting time.

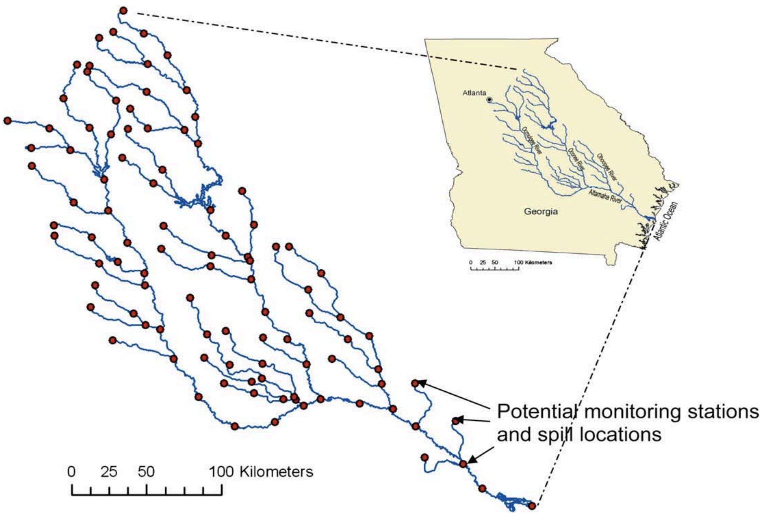


Figure 28: Shape of the Altamaha River and possible monitoring locations [39].

[39] define

$$d'(x_u) = \begin{cases} \text{detected time} \ - \ \text{spill time}, & \text{if detected at } x_u; \\ P_v, & \text{otherwise,} \end{cases}$$

where P_v is a constant penalty value and $t'(\mathbf{x}) = \min_{1 \le u \le M} d'(x_u)$. Then they solve $\operatorname{argmin}_{\mathbf{x} \in \Theta} \mathbf{E}[t'(\mathbf{x})]$ by the GA algorithm.

As discussed in the beginning of this chapter, the GA tends to return a solution with 100% or the highest possible reliability. For implementation of the GA, three parameters are needed: the number of observations (i.e., SWMM runs), a generation size, and a population size. Large values of these parameters help the GA return a solution close to the true best solution but at the cost of computational efforts. In general, it is hard to pick these parameters that balance between computational efficiency and accuracy in a returned solution. In addition, when a decision maker is more interested in finding a solution with the shortest detection time at the slight cost of reliability (e.g., 99% or 95%), it is hard to use the GA because there is no available information about what values of P_v would be appropriate to find the best feasible solution.

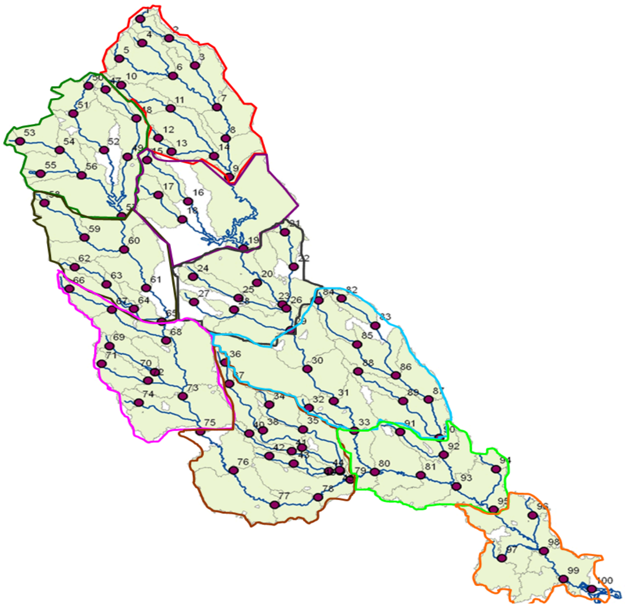


Figure 29: Ten sub-catchments of the Altamaha River [37].

3.3.1 Experimental Setup

Spill location L_i^S is assumed as a discrete random variable from sample space $\{1, 2, \ldots, 100\}$ and it is modeled as either a uniform discrete random variable or a non-uniform discrete random variable. The non-uniformly distributed L_i^S case is assumed to detect a specific type of chemicals produced by paper mill factories. Two paper mill factories exist close to nodes 30 and 68 in Figure 29 around the Altamaha River [23] thus the probability of occurring a spill at the two nodes is assumed to be ten times higher than the probability at the other nodes. Threshold C_{th} for the monitoring devices is set to $C_{th} \in \{0.0001, 0.05\}$. Then, four different cases are examined: $C_{th} = 0.0001\ mg/\ell$ and uniformly distributed L_i^S; $C_{th} = 0.0001\ mg/\ell$ and non-uniformly distributed L_i^S; $C_{th} = 0.05\ mg/\ell$, and uniformly distributed L_i^S; and $C_{th} = 0.05\ mg/\ell$ and non-uniformly distributed L_i^S. Under each case, S_i^S, I_i^S, P_i^R, M, and q are varied.

For a spill event, only one single instantaneous spill is considered. Spill starting time S_i^S is uniformly distributed between 0 and 240 hours in the simulation clock and intensity of a spill I_i^S is uniformly distributed between 10 and 1000 g/ℓ. The hydrodynamics of the river system is adopted from [37] where the steady-state hydraulic system is calibrated for the flow pattern in the river based on the data obtained from annual average flow rates measured in 2006 at twenty USGS gauging stations that are distributed throughout the river network. In this application, all lakes and impoundments were approximated as river reaches to simplify the network with an adjustment to the length of the reach.

The rain events for the case study are also generated in the same way as in [37]. The Altamaha River watershed is divided into ten sub-catchments as in Figure 29. The rainfall measurements are obtained from different USGS observation stations close to these 10 sub-catchments in the year 2006. Then, using the results of the statistical analysis of these observations, five rain patterns are generated for each sub-catchment. Note that these five rain patterns are different for each sub-catchment and thus there are total fifty rain patterns for the entire watershed. Also, each rain pattern describes time-dependent rainfall events and keeps changing hydrologic conditions in each sub-catchment during process simulation. For each SWMM run, one out of five rain patterns is randomly selected for each sub-catchment, which defines P_i^R. All nodes in the same sub-catchment have the same rain pattern during process simulation. The number of monitoring devices M is set to $M \in \{5, 6, 7, 10\}$. Minimum reliability requirement q is $q \in \{0.9, 1.0\}$, if $C_{th} = 0.0001$ mg/ℓ. Otherwise $q \in \{0.9, 0.95\}$ are used since there is no solution with 100% detection reliability.

In a SWMM run, simulating hydrodynamics under a random rain event P_i^R takes long but simulating contaminant transports under the hydrodynamics can be done relatively fast. Thus, [37] design one SWMM run to generate 100 observations of $t'(\mathbf{x})$ by (i) simulating hydrodynamics with P_i^R and (ii) performing 100 different spill events, each with randomly generated I_i^S and S_i^S, under the hydrodynamics. Moreover, [37] generate 1000 SWMM runs (thus, 100,000 observations of $t'(\mathbf{x})$) and apply the GA to find a solution with the smallest sample mean of the 100,000 observations. Following their experiments, when the spill location is uniformly distributed, one SWMM run performs 100 spill events (one spill event at each node with randomly generated I_i^S and S_i^S) under shared hydrodynamics and obtain 100 observations of $t(\mathbf{x})$ and $R(\mathbf{x})$. When the spill location is non-uniformly distributed, 10 spill events are simulated at both nodes 30 and 68 and one spill event is simulated at each of the other nodes in one SWMM run. Thus, one SWMM run performs 118 spill events resulting in 118 observations of $t_i(\mathbf{x})$ and $R_i(\mathbf{x})$, respectively. While the GA completes 1000 SWMM runs prior to optimization search, NP+PFM performs additional SWMM runs as the optimization search continues and NP+PFM stops either when the stopping criterion in Section 3.2.4 is satisfied with $n_E = 10$ and $\epsilon = 0.5$ or when the total number of SWMM runs reaches 100.

For the implementation of the GA, it always takes 1000 SWMM runs for each solution visited. As parameters of the GA, a population size of 100 and a generation size of 400 are initially used. Unless the GA with the initial population and generation sizes finds a comparable solution to that of NP+PFM, the population and generation sizes are increased up to 200 and 800, respectively, until a comparable solution is found by the GA. This chapter reports only the best results obtained by the GA with this parameter adjustment.

It is clear that monitoring devices should be located toward downstream if one is to increase reliability, but this tends to increase the expected detection time. Thus there exists positive dependence between reliability and the expected detection time. It implies that the best feasible solution is likely (but not always) to have reliability exactly equal to or close to q. Thus θ_a and θ_d need to be selected carefully to ensure both convergence and efficiency as discussed in section 2.3.2. Recall ρ_c represent the probability that $\lambda_\ell^{v_k}(\mathbf{x})$ converges to 0 for any active constraint. It is known that $\sqrt{1.3} \leq \theta_a \leq \sqrt{1.9}$ and $0.7 \leq \rho_c \leq 0.9$ can generally be a good compromise regarding efficiency and accuracy for PS_c. Thus we select $\theta_a = \sqrt{1.5}$ and $\theta_d = \frac{\sqrt{1.5}}{10}$ which ensures $\rho_c \approx 0.8082$ by Theorem 4. For the implementation of NP+PFM, $\tau_k = 200$ is used. Also, $\Delta n = 100$ is selected for uniform spill probability case and $\Delta n = 118$ is selected for non-uniform spill probability cases. This ensures that each iteration of NP+PFM makes one SWMM run.

Three performance measures are reported to compare the GA and NP+PFM: NUM, EMD and ER. NUM represents the number of SWMM runs required by each optimization algorithm until it stops. It takes about 2 hours to complete one SWMM run on a PC with 2.6 GHz Intel Core 3 Quad Q8300 CPU while one iteration of NP+PFM takes less than two seconds. This implies that, with a single CPU, it would take about 2000 hours (83 days) to perform 1000 SWMM runs. In this chapter, multiple CPUs were used to obtain 1000 SWMM runs in two weeks. Since computation cost mainly depends on the number of SWMM runs needed, NUM is used as a measure for computation cost. Note that NUM is always 1000 for the GA because it requires the 1000 SWMM runs to be completed prior to optimization. On the other hand, NUM is usually smaller than 1000 for NP+PFM because it only continues until stopping criteria are satisfied.

EMD and ER are quality measures for solutions returned by the two competing methods. EMD denotes estimated conditional minimum detection time in minutes and is calculated as conditional sample average of $t_i(\mathbf{x})$ given $R_i(\mathbf{x}) = 1$ over 1000 SWMM runs. EMD can be interpreted as estimated time interval between the time when a spill occurs and the time when the spill is detected. ER represents estimated reliability. ER is calculated as sample average of $R_i(\mathbf{x})$ over 1000 SWMM runs. Note that EMD and ER values reported in the tables are calculated based on all observations from 1000 SWMM runs, even when NP+PFM stops with a fewer number of iterations than 1000. EMD and ER are meaningful up to the tenths digit and the ten-thousandths digit, respectively.

3.3.2 Results

The main results are summarized in this section. For all figures in this section, the node indices are generated by rules in Section 3.1.

3.3.2.1 Uniform spill probability and small threshold monitoring system

Figure 30 shows solutions returned by NP+PFM in red circles and GA in black triangles when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S, and $q = 1.0$ for $M = 5, 6, 7,$ and 10. When $M = 5$ and 7, both NP+PFM and GA return exactly the same solution and when $M = 6$ and 10, only one placement from NP+PFM is different of that from GA.

Table 7 shows that NP+PFM uses a smaller number of SWMM runs (smaller NUM) and yet returns equal or better solutions than GA. More specifically, when $M = 6$, NP+PFM stopped only after 107 SWMM runs and the returned solution's EMD is 16 minutes shorter than that of the solution returned by the GA with 1000 SWMM runs. That is, the GA and NP+PFM find a similar solution with respect to EMD but NP+PFM saves approximately 1786 hours of CPU computation time when using a single PC with 2.6 GHz Intel Core 3 Quad Q8300 CPU.

Figure 31 and Table 8 show results when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.9$. As GA returns the same solutions when $q = 1.0$, only the results of NP+PFM are reported as discussed earlier. The optimal solutions when $q = 0.9$ tend to be located in the more upstream than those when $q = 1.0$. Table 8 shows NUM, EMD, and ER when $q = 0.9$. EMD are smaller than those when $q = 1.0$. Note that $C_{th} = 0.0001\ mg/\ell$ is so small that it rarely misses any spill. As a result, the estimated reliability is simply the maximum probability that the monitoring devices can achieve to detect a spill assuming no miss.

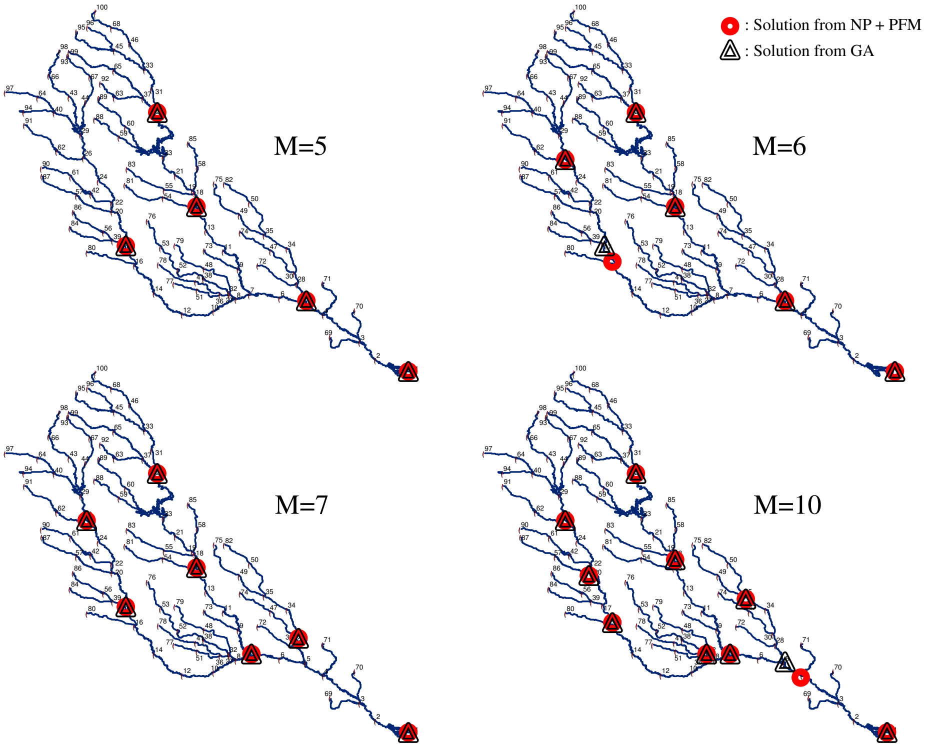


Figure 30: Optimal solutions when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S, and $q = 1.0$.

Table 7: Results of NP+PFM and GA when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S and $q = 1.0$.

M	NP+PFM			GA		
	NUM	EMD	ER	NUM	EMD	ER
5	91	3156.1	1.0000	1000	3156.1	1.0000
6	107	2718.8	1.0000	1000	2734.8	1.0000
7	277	2329.0	1.0000	1000	2329.0	1.0000
10	757	1917.3	1.0000	1000	1923.3	1.0000

Table 8: Results of NP+PFM when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.9$.

M	NUM	EMD	ER
5	154	2782.7	0.9300
6	321	2407.1	0.9300
7	205	2222.7	0.9300
10	459	1738.1	0.9100

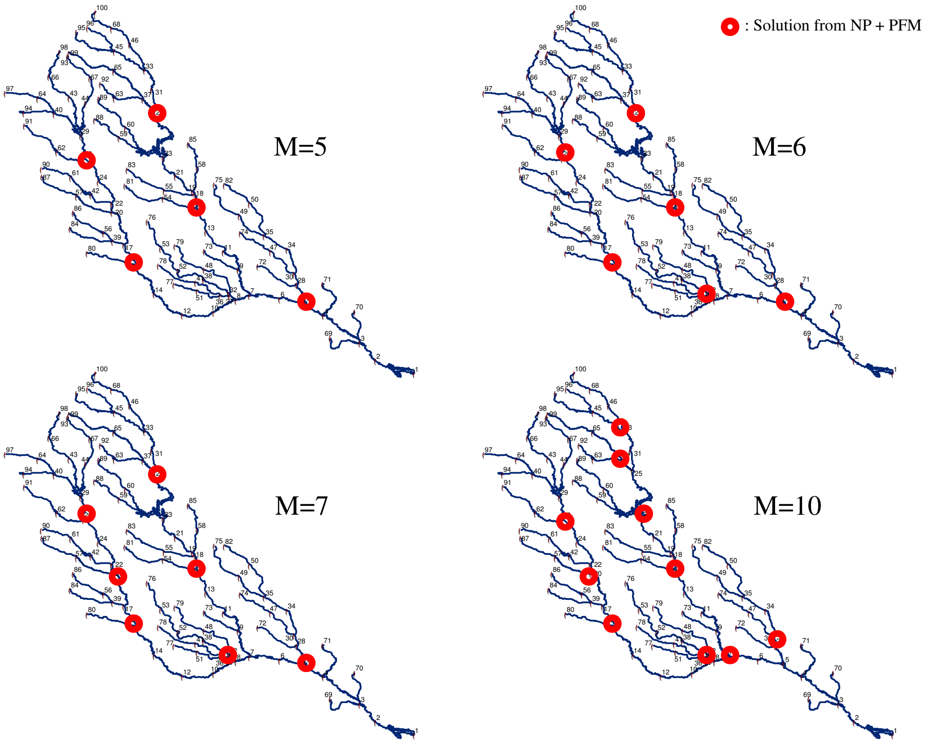


Figure 31: Optimal solutions when $C_{th} = 0.0001\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.9$.

Table 9: Results of NP+PFM and GA when $C_{th} = 0.0001\ mg/\ell$, non-uniformly distributed L_i^S, and $q = 1.0$.

M	NP+PFM			GA		
	NUM	EMD	ER	NUM	EMD	ER
5	79	2778.2	1.0000	1000	2778.2	1.0000
6	126	2407.1	1.0000	1000	2407.1	1.0000
7	241	2122.8	1.0000	1000	2122.8	1.0000
10	827	1677.9	1.0000	1000	1762.2	1.0000

Table 10: Results of NP+PFM when $C_{th} = 0.0001\ mg/\ell$, non-uniformly distributed L_i^S and $q = 0.9$.

M	NUM	EMD	ER
5	120	2450.4	0.9407
6	305	2144.6	0.9407
7	156	1930.0	0.9407
10	602	1488.9	0.9068

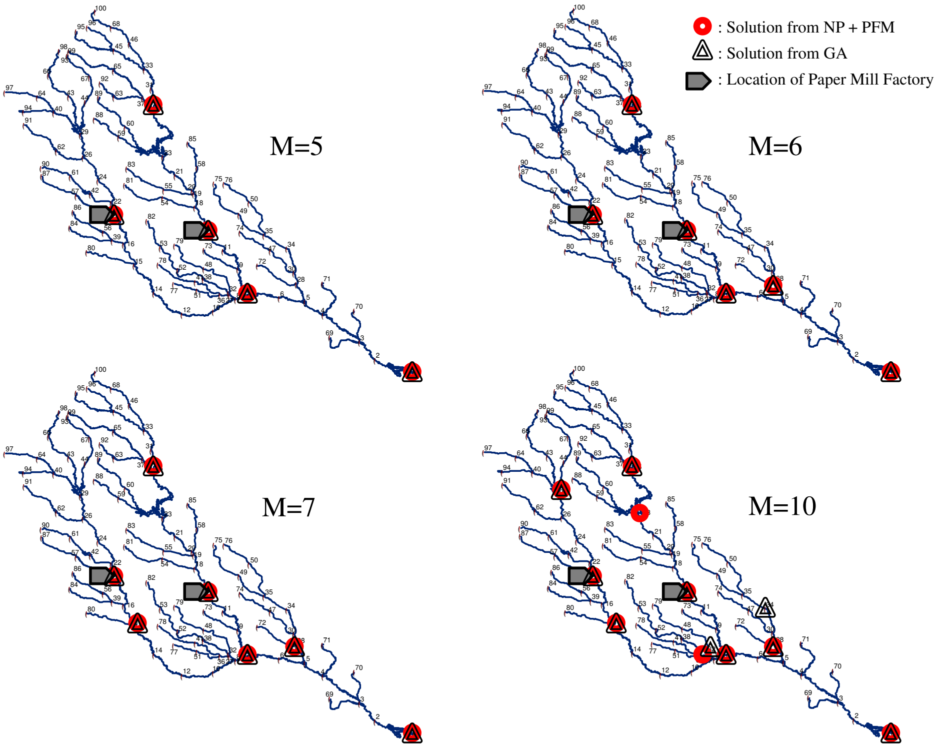


Figure 32: Optimal solutions when $C_{th} = 0.0001 mg/\ell$, non-uniformly distributed L_i^S, and $q = 1.0$.

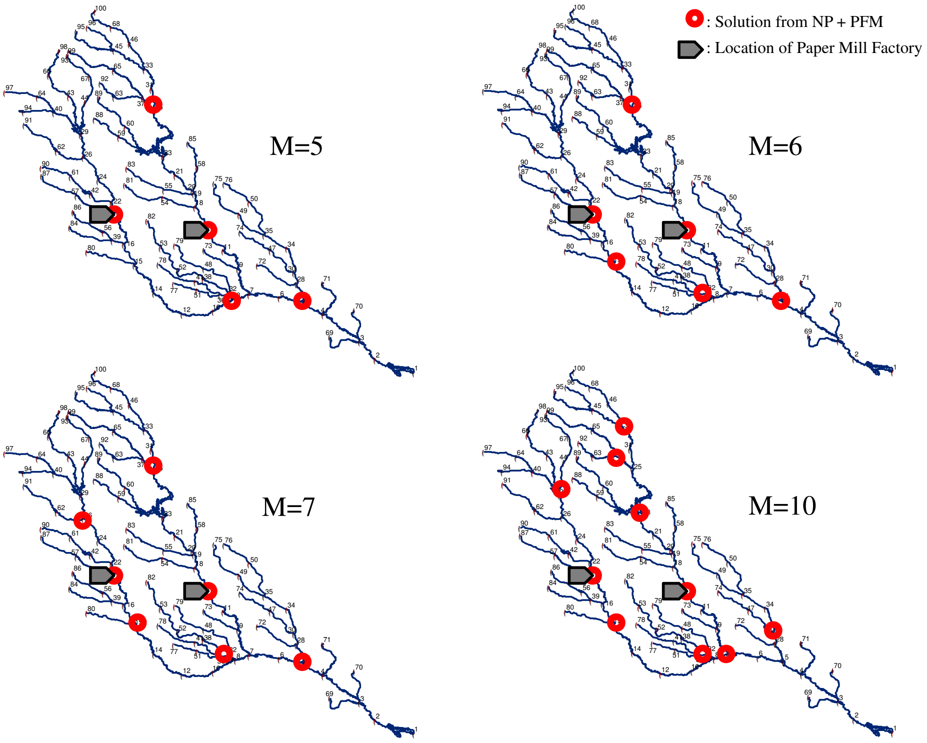


Figure 33: Optimal solutions when $C_{th} = 0.0001\ mg/\ell$, non-uniformly distributed L_i^S, and $q = 0.9$.

3.3.2.2 Non-uniform spill probability and small threshold monitoring system

Now this section discusses the results with $C_{th} = 0.0001\ mg/\ell$ and non-uniformly distributed L_i^S. Figures 32 and 33 show optimal locations of monitoring devices when $q = 1.0$ and $q = 0.9$, respectively. In Figure 32, solutions from NP+PFM and GA are exactly same when $M = 5, 6$, and 7, but for $M = 10$ case, results from NP+PFM and GA are different. The monitoring devices are located exactly on the nodes close to the paper mill factories for any value of M and for both NP+PFM and GA. When $q = 0.9$, monitors tend to be placed upward compared to the $q = 1.0$ case as shown in Figure 32 and Figure 33, but two monitoring devices are still located exactly on the nodes closed to the paper mill factories.

Tables 9 and 10 show three main performance measures. Table 9 shows that NP+PFM returns the same or a better solution with significantly smaller number of SWMM runs than the GA does.

3.3.2.3 Uniform spill probability and large threshold monitoring system

Next this section considers the case with uniformly distributed L_i^S but larger threshold C_{th} (i.e., $C_{th} = 0.05\ mg/\ell$). The larger threshold C_{th} increases the chance that a monitoring device misses a spill due to flow speed and dilution by dynamic flow and rain. Figure 34 shows solutions returned by NP+PFM and GA and Table 11 shows NUM, EMD, and ER of the returned solutions. NP+PFM returns solutions with smaller EMD and smaller ER. Especially, when $M = 10$, a solution from NP+PFM saves about 400 minutes according to EMD but its detection reliability is also dropped by 1.34 % compared to a solution from GA.

By comparing Figures 31 and 35, one can notice that the optimal locations of monitoring devices when $C_{th} = 0.05\ mg/\ell$ tend to be placed in more downstream than those when $C_{th} = 0.0001\ mg/\ell$. Also, Table 12 shows that NP+PFM needs more SWMM runs (thus, larger NUM) until it stops and the estimated detection

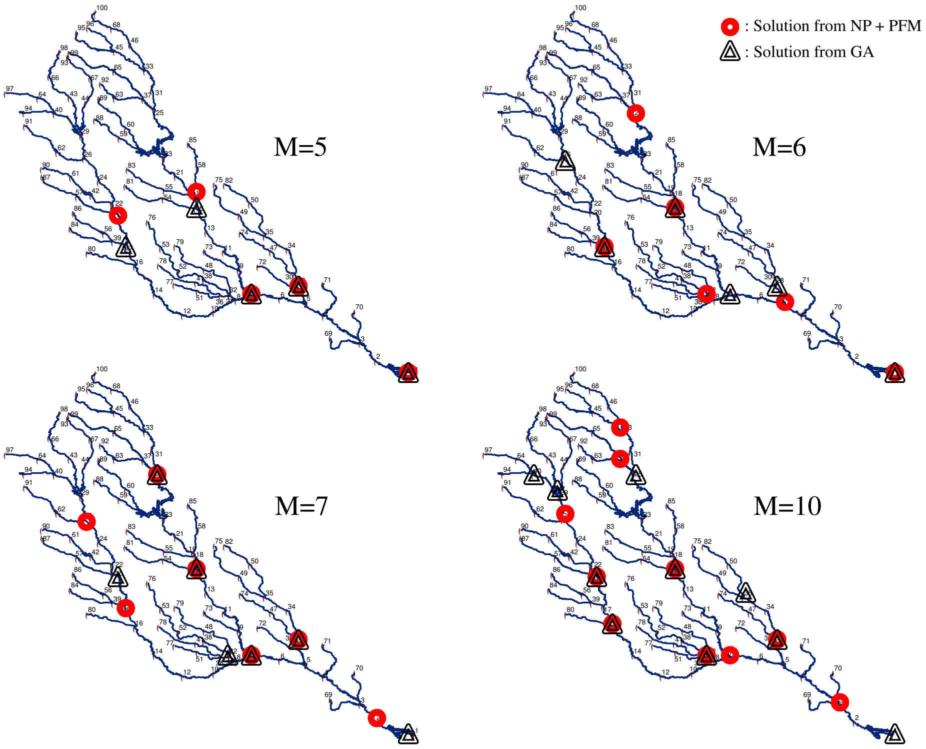


Figure 34: Optimal solutions with $C_{th} = 0.05\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.95$.

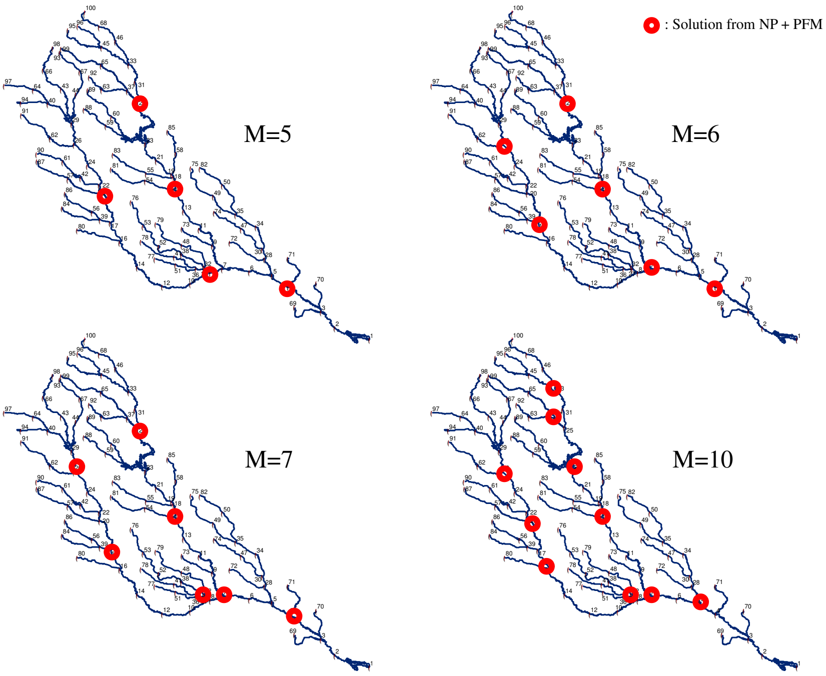


Figure 35: Optimal solutions with $C_{th} = 0.05\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.9$.

Table 11: Results of NP+PFM and GA when $C_{th} = 0.05\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.95$.

M	NP+PFM			GA		
	NUM	EMD	ER	NUM	EMD	ER
5	893	3750.7	0.9534	1000	3779.9	0.9559
6	656	3230.0	0.9570	1000	3322.1	0.9600
7	598	2764.7	0.9553	1000	3043.0	0.9728
10	809	2272.6	0.9626	1000	2670.3	0.9760

Table 12: Results of NP+PFM when $C_{th} = 0.05\ mg/\ell$, uniformly distributed L_i^S, and $q = 0.9$.

M	NUM	EMD	ER
5	614	3349.7	0.9041
6	702	2851.2	0.9018
7	800	2668.4	0.9162
10	586	2130.3	0.9021

times (EMD) are much larger when $C_{th} = 0.05\ mg/\ell$ than when $C_{th} = 0.0001\ mg/\ell$. It is well expected because this case is more difficult than the first case with the small threshold. The estimated reliability levels (ER) of the returned solutions are greater than $q = 0.9$, providing an evidence that the solutions are feasible, but are smaller than the reliability levels obtained when $C_{th} = 0.0001\ mg/\ell$.

3.3.2.4 Non-uniform spill probability and large threshold monitoring system

Finally, non-uniformly distributed L_i^S are considered with $C_{th} = 0.05\ mg/\ell$. Figure 36 and Table 13 show the optimal solutions returned by NP+PFM and GA and their performance measures, respectively. Similar to Table 11, NP+PFM provides solutions whose EMD is smaller and ER is also smaller than a solution from GA.

Solutions returned by NP+PFM are shown in Figure 37 and Table 14 when $q = 0.9$ is considered. Compared to the second case where $C_{th} = 0.0001\ mg/\ell$ and non-uniformly distributed L_S^i, this is a more difficult case because a spill can be missed. NP+PFM needs more SWMM runs until it stops and EMD are larger than those

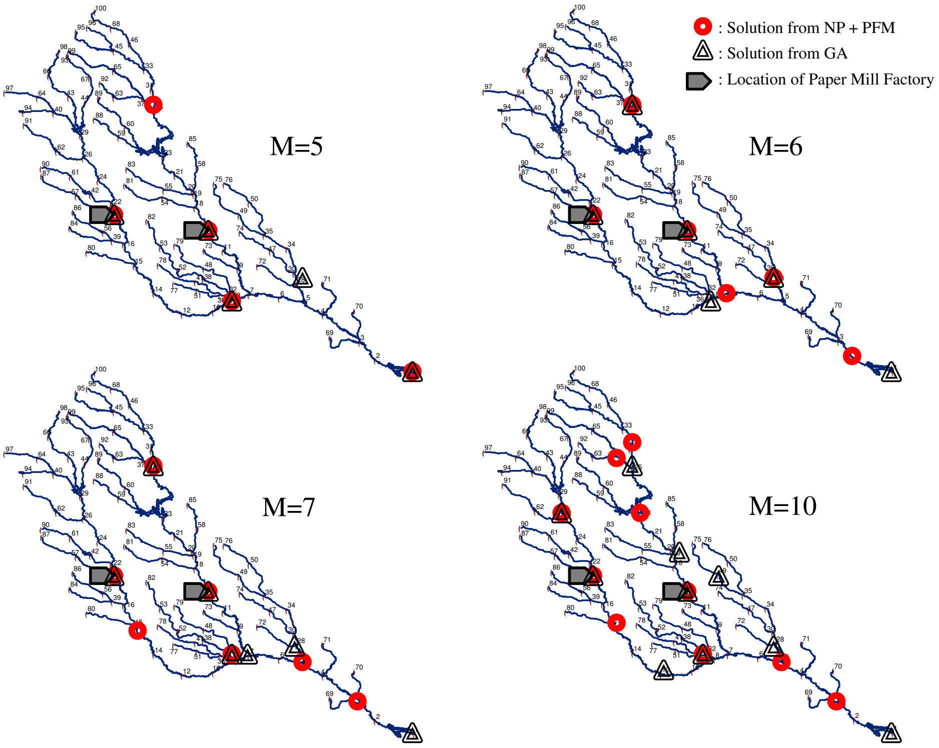


Figure 36: Optimal solutions with $C_{th} = 0.05\ mg/\ell$, non-uniform L_i^S, and $q = 0.95$.

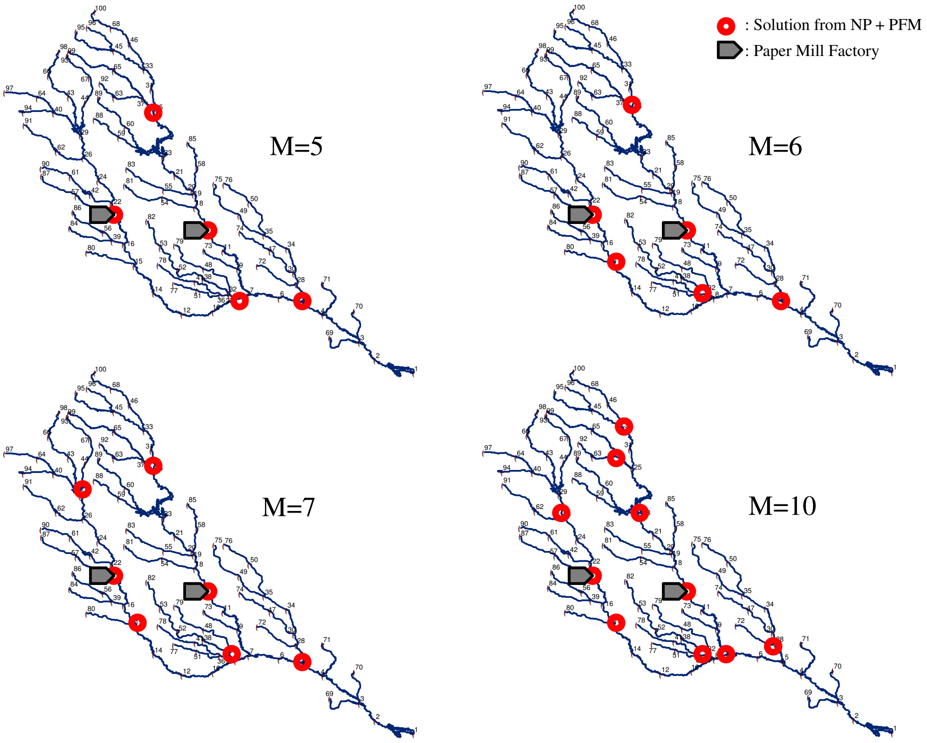


Figure 37: Optimal solutions with $C_{th} = 0.05\ mg/\ell$, non-uniform L_i^S, and $q = 0.9$.

Table 13: Results of NP+PFM and GA when $C_{th} = 0.05\ mg/\ell$, non-uniformly distributed L_i^S, and $q = 0.95$.

M	NP+PFM			GA		
	NUM	EMD	ER	NUM	EMD	ER
5	421	3198.3	0.9527	1000	3446.4	0.9674
6	954	2795.4	0.9571	1000	2895.5	0.9706
7	544	2472.8	0.9507	1000	2660.3	0.9763
10	1000	1961.5	0.9545	1000	2269.9	0.9808

Table 14: Results of NP+PFM when $C_{th} = 0.05\ mg/\ell$, non-uniformly distributed L_i^S, and $q = 0.9$.

M	NUM	EMD	ER
5	320	2834.9	0.9035
6	350	2498.5	0.9109
7	543	2261.1	0.9132
10	675	1835.3	0.9135

from the second case. The returned solutions for $M = 5, 6, 7$, and 10 seem all feasible based on estimated reliability ER.

3.4 Conclusions

This chapter considers the problem of designing a water quality monitoring network for river systems where the goal is to find the optimal location of a finite number of monitoring devices that minimizes the expected detection time of a contaminant spill event while guaranteeing detection reliability greater or equal to some constant. The problem is formulated as a DOvS problem with a stochastic constraint and implementation issues in a DOvS algorithm, namely NP+PFM, are discussed.

The advantage of NP+PFM is that it obtains additional observations as needed while the GA requires the decision maker to determine how many observations to take prior to optimization. The chosen number of observations for the GA may be too small, introducing large estimation error in performance measures which in turns increases a chance to return a solution quite different from the true optimal, or too large, wasting time on taking unnecessary observations. In addition, NP+PFM with the penalty sequence used in the chapter has a guarantee on global convergence when there is no active constraint while the GA is a heuristic algorithm that does not provide such guarantee.

The experimental results on the Altamaha River show that NP+PFM handles feasibility of solutions on a stochastic constraint effectively and returns a good feasible solution. Also, NP+PFM finds the same or better solution than the GA algorithm with significant savings in computational efforts. The proposed algorithm can be applicable to other river systems and extended to more realistic and complex settings such as those that account for multiple spills, complicated rain patterns, or detection error of a monitoring device.

CHAPTER IV

IMPROVING PERFORMANCE OF PENALTY FUNCTION WITH MEMORY BY APPROXIMATE SIMULATION BUDGET ALLOCATION

In discrete optimization via simulation (DOvS), decision variables are discrete and the number of potential solutions is large, requiring a search method to determine which solutions to simulate in the next iteration. A number of efficient DOvS algorithms are presented. For example, see [2], [5], [14], [15], [28], [33], and [40].

Recently stochastically constrained DOvS problems have received attentions from the simulation community. For the problems, we propose penalty function with memory (PFM) to handle stochastic constraints on secondary performance measures in Chapter II. PFM determines a penalty sequence of each solution based on past results of feasibility checks and a DOvS algorithm combined with PFM, denoted as $\mathcal{D}$+PFM, can solve a stochastically constrained DOvS problem. To ensure a good performance of $\mathcal{D}$+PFM, two things need to happen: (i) PFM should make a correct decision on feasibility with high probability when a solution is visited; and (ii) a DOvS algorithm should find the most promising region (or solution) correctly at each search iteration. The version proposed in Chapter II currently takes the equal number of observations on each visited solution (equal allocation) and uses cumulative sample means to make a feasibility decision or find a feasible promising region. If we can efficiently and accurately select the most promising region (or solution) with accurate feasibility checks, the performance of $\mathcal{D}$+PFM can be further improved.

For stochastically unconstrained DOvS problems, a number of ranking and selection (R&S) procedures are used to further improve efficiency or accuracy of a DOvS algorithm. For example, [8] and [28] propose statistical comparison procedures for either cleaning up solutions at the end of the search (in order to find the best among all solutions visited during the search) or assisting a DOvS algorithm with finding a correct promising region. Although the statistical comparison procedures are shown to be very useful for clean-up, it is known that they tend to 'overdo' when they are applied within a DOvS algorithm at each search iteration because they try to provide a guarantee on the probability of correctly selecting the true best solution (PCS). On the other hand, there are statistical procedures called optimal computing budget allocation (OCBA). These procedures allocate a finite computing budget (either count or time) among a finite number of solutions to maximize PCS. OCBA often suffers due to low PCS or inability telling how many observations would be needed for a correct selection in R&S. However, OCBA finds a good application in DOvS because DOvS algorithms need to allocate a finite budget among visited solutions at each search iteration. For example, [12] presents that OCBA improves the performance of optimization algorithms significantly compared to equal allocation.

To improve the performance of $\mathcal{D}$+PFM for stochastically constrained DOvS problems, constrained R&S procedures can be considered. [4] and [13] propose statistically valid procedures that select the best feasible solution with PCS guarantee in the presence of multiple stochastic constraints. They are good candidate procedures for clean-up at the end of the optimization search. [16] and [20] provide optimal budget allocation methods that allocate finite sampling budget to maximize the probability of correctly selecting the best feasible solution. [16] allows observations to have general marginal distributions but it is hard to implement their procedure when the number of solutions is large. On the other hand, [20] provides a procedure, called OCBA-CO, that is easy to implement even with a large number of solutions but the procedure requires the normal assumption on observations.

In this chapter, we focus on improving the performance of $\mathcal{D}$+PFM by combining it with a budget allocation method. More specifically, we consider OCBA-CO from [20] due to its easiness in implementation and applicability to a large number of solutions, modify it so that convergence properties of $\mathcal{D}$+PFM can be preserved when the modified budget allocation method is combined with $\mathcal{D}$+PFM, and test how much improvement is achieved over equal allocation. We call the modified budget allocation method the approximate budget allocation (ABA). ABA is different from OCBA-CO in two aspects:

1. OCBA-CO always considers a fixed set of simulated solutions and re-allocates cumulative total number of simulation budget among all solutions in the set at each iteration. On the other hand, the size of a set of sampled solutions and identities of elements in the set can be different in $\mathcal{D}$+PFM at each iteration. Thus ABA allocates only new additional simulation budget among sampled solutions at each iteration.

2. Assuming a minimization problem, OCBA-CO chooses a solution with the smallest sample mean of the primary performance measure as the sample best among solutions whose estimates of the secondary performance measures satisfy stochastic constraints. However, ABA combined with $\mathcal{D}$+PFM (which we denote as $\mathcal{D}$+PFM+ABA) define the sample best as a solution with the smallest sum between sample mean of the primary performance measure and penalty value.

This chapter is organized as follows: Section 4.1 defines our problem and notation, and then briefly reviews PFM and OCBA-CO. Section 4.2 introduces $\mathcal{D}$+PFM+ABA and provides its asymptotic convergence properties along with proofs. Experimental results of $\mathcal{D}$+PFM+ABA on the three numerical examples are presented in Section 4.3, followed by concluding remarks in Section 4.4.

4.1 Background

In this section, we define our problem and notation, and review PFM and OCBA-CO.

4.1.1 Problem

From Section 2.1.1, recall that Θ represents the whole decision variable space which is a discrete and finite set in $\mathbb{R}^d$ and $\mathbf{x} = (x_1, \ldots, x_d)$ represents a solution (or decision variable). Let $G(\mathbf{x})$ and $H_\ell(\mathbf{x})$ represent the primary performance measure and the secondary performance measure of the ℓth constraint, respectively and $G_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x})$ represent their ith observations from stochastic simulation. Then our DOvS problem with stochastic constraints is defined as follows:

$$\begin{aligned} &\text{argmin}_{\mathbf{x}\in\Theta}\mathbf{E}[G(\mathbf{x})], \\ &\text{subject to } \mathbf{E}[H_\ell(\mathbf{x})] \geq q_\ell, \;\; \ell \in \Lambda, \end{aligned} \tag{10}$$

where m is the number of stochastic constraints and Λ is a set of all indices for stochastic constraints (i.e., $\Lambda = \{1, 2, \ldots, m\}$). We also make the following assumption throughout the chapter.

Assumption 4 *The original problem (10) satisfies the following:*

1. $G_i(\mathbf{x})$ *are normally distributed and independent for all* $i = 1, 2, \ldots,$ *given any* $\mathbf{x} \in \Theta$*,*
2. $H_{\ell i}(\mathbf{x})$ *are normally distributed and independent for all* $i = 1, 2, \ldots,$ *given any* $\ell \in \{1, 2, \ldots, m\}$ *and any* $\mathbf{x} \in \Theta$*.*
3. *There exists a unique best feasible solution.*

The first two assumptions in Assumption 4 are often used in R&S. In practice, $G_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x})$ have approximate normal distributions, if $G_i(\mathbf{x})$ and $H_{\ell i}(\mathbf{x})$ are either within-replication averages or batch means. We test the performance of $\mathcal{D}$+PFM+ABA when the normal assumption is violated in Section 4.3.3.

4.1.2 PFM

We review PFM. In this chapter, "solution $\mathbf{x}$ is visited" means that solution $\mathbf{x}$ is simulated to obtain additional observations. Thus, it is possible that a solution is sampled under a DOvS algorithm $\mathcal{D}$, but not visited if no additional observation is obtained from the solution based on a simulation budget allocation method. We list notation for PFM below:

$k \equiv$ iteration counter;

$r \equiv$ counter for the number of visits;

$v_k(\mathbf{x}) \equiv$ the number of visits up to iteration k for $\mathbf{x}$;

$n_r(\mathbf{x}) \equiv$ the total number of observations obtained up to the rth visit for $\mathbf{x}$;

$n_{v_k}(\mathbf{x}) \equiv n_{v_k(\mathbf{x})}(\mathbf{x})$, the total number of observations obtained up to iteration k for $\mathbf{x}$;

$\Delta n_r(\mathbf{x}) \equiv$ the number of new observations obtained at the rth visit for $\mathbf{x}$;

$\overline{G}_k(\mathbf{x}) \equiv \frac{1}{n_{v_k}(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} G_i(\mathbf{x})$, cumulative sample mean of observations $G_i(\mathbf{x})$ for the primary performance measure up to iteration k;

$\overline{H}_{\ell k}(\mathbf{x}) \equiv \frac{1}{n_{v_k}(\mathbf{x})} \sum_{i=1}^{n_{v_k}(\mathbf{x})} H_{\ell i}(\mathbf{x})$, cumulative sample mean of observations $H_{\ell i}(\mathbf{x})$ for the ℓth secondary performance measure up to iteration k;

$\hat{\mathbf{x}}_k^* \equiv$ the sample best among all sampled solutions up to iteration k; and

$\mathbf{x}_o^b \equiv$ the true best feasible solution up to iteration k.

As discussed in Section 2.2, a new objective function with PFM at search iteration k is

$$Z_k(\mathbf{x}) = \overline{G}_k(\mathbf{x}) + \sum_{\ell \in \Lambda} \left[\lambda_\ell^{v_k}(\mathbf{x}) \times \max\{0, q_\ell - \overline{H}_{\ell k}(\mathbf{x})\}\right], \tag{11}$$

where $\lambda_\ell^{v_k}(\mathbf{x})$ is a penalty sequence of the ℓth constraint at visit $v_k(\mathbf{x})$ for $\mathbf{x}$. Whenever a solution $\mathbf{x}$ is visited, a feasibility check is performed. If $\mathbf{x}$ is declared as feasible, a positive constant smaller than 1 (depreciation factor) is multiplied to the previous penalty sequence value, $\lambda_\ell^{v_k-1}(\mathbf{x})$. Similarly, if $\mathbf{x}$ is declared as feasible, a positive constant larger than 1 (appreciation factor) is multiplied to $\lambda_\ell^{v_k-1}(\mathbf{x})$. In Section 2.2, two example penalty sequences, PS$_c$ and PS$_f$ are presented. PS$_f$ generally performs better than PS$_c$ as shown in Sections 2.3 and 2.4 because PS$_c$ uses same appreciation and depreciation factors for every solution while PS$_f$ adaptively adjusts appreciation and depreciation factors according to the level of feasibility of each solution.

When PS$_f$ with finite $\Delta n_r(\mathbf{x})$ is used, we should guarantee Assumption 3: both $\lim_{r\to\infty} \Delta n_r(\mathbf{x}) = \Delta n(\mathbf{x})$ and symmetric marginal distributions of all secondary performance measures. However, a simulation budget allocation method tends to change $\Delta n_r(\mathbf{x})$ at every iteration and we may have a problem with either $\lim_{r\to\infty} \Delta n_r(\mathbf{x}) = \Delta n(\mathbf{x})$ or symmetry of the underlying distribution of $H_\ell(\mathbf{x})$. Therefore, throughout this chapter, we consider only two penalty sequences that do not require Assumption 3: (i) PS$_f$ with increasing $\Delta n_r(\mathbf{x})$ and (ii) an adaptive penalty sequence with constants (APS$_c$).

More specifically, APS$_c$ is constructed as follows: Recall $S_\ell^{v_k}(\mathbf{x}) \equiv \sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}(\mathbf{x})$ where $\zeta_{\ell r}(\mathbf{x}) \equiv \sum_{i=n_{r-1}(\mathbf{x})+1}^{n_{r-1}(\mathbf{x})+\Delta n_r(\mathbf{x})} \frac{H_{\ell i}(\mathbf{x})-q_\ell}{\sqrt{\Delta n_r(\mathbf{x})}}$ and the infeasible probability,

$$\hat{p}_\ell^{v_k}(\mathbf{x}) = \frac{\sum_{r=1}^{v_k(\mathbf{x})} \mathbb{I}\{\zeta_{\ell r}(\mathbf{x}) < 0\}}{v_k(\mathbf{x})}. \tag{12}$$

Prior to running APS$_c$, the following parameters should be chosen:

- $0 < h_1 < \ldots < h_u < 0.5 < h_{u+1} < \ldots < h_v < 1$;
- $1 < a_\nu$ and $0 < d_\nu < 1$, for all $\nu = 1, 2, \ldots, v$.

Then APS$_c$ is defined as follows:

Adaptive Penalty Sequence with Constants (APS$_c$)

$$\lambda_\ell^{v_k}(\mathbf{x}) = \begin{cases} \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \alpha_\ell^{v_k}(\mathbf{x}), & \text{if } S_\ell^{v_k}(\mathbf{x}) < 0, \\ \lambda_\ell^{v_{k-1}}(\mathbf{x}) \times \delta_\ell^{v_k}(\mathbf{x}), & \text{if } S_\ell^{v_k}(\mathbf{x}) \geq 0, \end{cases}$$

where $\lambda_\ell^0(\mathbf{x})$ is the initial penalty constant λ_ℓ^0, $\alpha_\ell^{v_k}(\mathbf{x})$ is an appreciation function, and $\delta_\ell^{v_k}(\mathbf{x})$ is a depreciation function whose values are determined by $\hat{p}_\ell^{v_k}(\mathbf{x})$ and the following table:

$\hat{p}_\ell^{v_k}(\mathbf{x})$	$[0, h_1)$	$[h_1, h_2)$	$\ldots$	$(h_{v-1}, h_v]$	$(h_v, 1]$
$\alpha_\ell^{v_k}(\mathbf{x})$	a_0	a_1	$\ldots$	a_{v-1}	a_v
$\delta_\ell^{v_k}(\mathbf{x})$	d_0	d_1	$\ldots$	d_{v-1}	d_v

Note that APS$_c$ is PS$_f$ with finite $\Delta n_r(\mathbf{x})$ and $\gamma_\ell^k = 0$ but its convergence proofs follow convergence proofs of PS$_c$ in Chapter II. In this chapter, if simulation budget allocated at each iteration increases, then PS$_f$ is used. If the simulation budget is finite but can be different at each iteration, then APS$_c$ is used.

4.1.3 OCBA-CO

In this section, we review OCBA-CO in [20]. OCBA-CO considers a fixed set, say Ω. Let T define a total computing budget we want to allocate and $N(\mathbf{x})$ define the number of simulation observations for solution $\mathbf{x}$ (i.e., $T = \sum_{\mathbf{x}\in\Omega} N(\mathbf{x})$). The goal of the optimal budget allocation is to intelligently control each $N(\mathbf{x})$ in a way that PCS is maximized. PCS is defined as follows:

$$\begin{aligned} \text{PCS} &= \mathbf{P}\left\{\cap_{\ell\in\Lambda}(\bar{H}_{\ell N}(\mathbf{x}_o^b) \geq q_\ell) \right. \\ &\qquad \left. \cap_{\mathbf{x}\in\Omega, \mathbf{x}\neq\mathbf{x}_o^b}\left\{\Big[\ \cap_{\ell\in\Lambda}\,(\bar{H}_{\ell N}(\mathbf{x}) \geq q_\ell) \cap (\bar{G}_N(\mathbf{x}_o^b) > \bar{G}_N(\mathbf{x}))\Big]^c\right\}\right\}, \end{aligned}$$

where $\bar{G}_N(\mathbf{x}) = \frac{\sum_{i=1}^{N(\mathbf{x})} G_i(\mathbf{x})}{N(\mathbf{x})}$ and $\bar{H}_{\ell N}(\mathbf{x}) = \frac{\sum_{i=1}^{N(\mathbf{x})} H_{\ell i}(\mathbf{x})}{N(\mathbf{x})}$. The OCBA-CO problem is defined as follows:

$$\max_{N(\mathbf{x}), \forall \mathbf{x}\in\Omega} \text{PCS} \quad \text{subject to} \quad \sum_{\mathbf{x}\in\Omega} N(\mathbf{x}) = T. \tag{13}$$

However, it is known that there is no closed-form for PCS. Bonferroni inequality provides that PCS is bounded below by APCS (i.e., PCS $\geq$ APCS) which is defined as

$$\begin{aligned}\text{APCS} &= \sum_{\ell\in\Lambda}\mathbf{P}\big(\bar{H}_{\ell N}(\mathbf{x}_o^b) \geq q_\ell\big) + (1-m) \\ &\quad - \sum_{\mathbf{x}\in\Omega,\mathbf{x}\neq\mathbf{x}_o^b}\Big[\min\Big[\min_{\ell\in\Lambda}\mathbf{P}\big(\bar{H}_{\ell N}(\mathbf{x}) \geq q_\ell\big), \mathbf{P}\big(\bar{G}_N(\mathbf{x}_o^b) > \bar{G}_N(\mathbf{x})\big)\Big]\Big].\end{aligned} \tag{14}$$

Let $\ell^*(\mathbf{x})$ define the most critical constraint (i.e., $\ell^*(\mathbf{x}) \equiv \arg\min_{\ell\in\Lambda}\mathbf{P}\{\bar{H}_{\ell N}(\mathbf{x}) \geq q_\ell\}$), then, we can define two sets Ω_O and Ω_F as follows.

$$\begin{aligned}\Omega_O &\equiv \{\mathbf{x}|\mathbf{x}\neq\mathbf{x}_o^b, \mathbf{x}\in\Omega, \mathbf{P}(\bar{H}_{\ell^* N}(\mathbf{x}) \geq q_{\ell^*}(\mathbf{x})) \geq \mathbf{P}(\bar{G}_N(\mathbf{x}_o^b) > \bar{G}_N(\mathbf{x}))\}, \text{ and} \\ \Omega_F &\equiv \{\mathbf{x}|\mathbf{x}\neq\mathbf{x}_o^b, \mathbf{x}\in\Omega, \mathbf{P}(\bar{H}_{\ell^* N}(\mathbf{x}) \geq q_{\ell^*}(\mathbf{x})) < \mathbf{P}(\bar{G}_N(\mathbf{x}_o^b) > \bar{G}_N(\mathbf{x}))\}.\end{aligned}$$

Note that Ω_O represents the set of solutions where optimality is a more dominant issue and Ω_F represents the set of solutions where feasibility is a more dominant issue. Then APCS of (14) can be rewritten as

$$\begin{aligned}\text{APCS} &= \sum_{\ell\in\Lambda}\mathbf{P}\big(\bar{H}_{\ell N}(\mathbf{x}_o^b) \geq q_\ell\big) + (1-m) \\ &\quad - \sum_{\mathbf{x}\in\Omega_F}\mathbf{P}\big(\bar{H}_{\ell^* N}(\mathbf{x}) \geq q_\ell\big) - \sum_{\mathbf{x}\in\Omega_O}\mathbf{P}\big(\bar{G}_N(\mathbf{x}_o^b) > \bar{G}_N(\mathbf{x})\big).\end{aligned} \tag{15}$$

Let $\beta(\mathbf{x})$ be the proportion of the total budget allocated to solution (i.e., $N(\mathbf{x}) = \beta(\mathbf{x})\cdot T$). Instead of Problem (13) we can solve the following problem to optimally allocate T:

$$\max_{N(\mathbf{x}),\forall\mathbf{x}\in\Omega} \text{APCS} \quad \text{subject to} \quad \sum_{\mathbf{x}\in\Omega}\beta(\mathbf{x}) = 1. \tag{16}$$

Recall $\sigma_0^2(\mathbf{x}) = \mathbf{Var}(G(\mathbf{x}))$ and $\sigma_\ell^2(\mathbf{x}) = \mathbf{Var}(H_\ell(\mathbf{x}))$, for any $\ell \in \Lambda$. In [20], the following theorem is provided and proved:

Lemma 4 *Define the noise-to-signal ratio, $\eta(\mathbf{x})$, as follows:*

$$\eta(\mathbf{x}) = \begin{cases} \frac{\sigma_0(\mathbf{x})}{\mathbf{E}[G(\mathbf{x})]-\mathbf{E}[G(\mathbf{x}_o^b)]}\sqrt{1+\frac{\sigma_0^2(\mathbf{x}_o^b)/\beta(\mathbf{x}_o^b)}{\sigma_0^2(\mathbf{x})/\beta(\mathbf{x})}}, & \textit{if } \mathbf{x}\in\Omega_O, \\ \frac{\sigma_{\ell^*}(\mathbf{x})}{q_{\ell^*}-\mathbf{E}[H_{\ell^*}(\mathbf{x})]}, & \textit{if } \mathbf{x}\notin\Omega_O. \end{cases}$$

Then, APCS *is asymptotically (as $T\to\infty$) maximized if*

$$\begin{aligned}\frac{\beta(\mathbf{x})}{\beta(\mathbf{y})} &= \left(\frac{\eta(\mathbf{x})}{\eta(\mathbf{y})}\right)^2, \forall \mathbf{x} \neq \mathbf{y} \neq \mathbf{x}_o^b, \ \textit{and} \\ \beta(\mathbf{x}_o^b) &= \max(\beta_O(\mathbf{x}_o^b), \beta_F(\mathbf{x}_o^b)),\end{aligned}$$

where $\beta_O(\mathbf{x}_o^b) = \sigma_0(\mathbf{x}_o^b)\sqrt{\sum_{\mathbf{x}\in\Omega_O} \frac{\beta^2(\mathbf{x})}{\sigma_0^2(\mathbf{x})}}$ *and* $\frac{\beta_F(\mathbf{x}_o^b)}{\beta(\mathbf{x})} = \left(\frac{\eta(\mathbf{x}_o^b)}{\eta(\mathbf{x})}\right)^2, \forall \mathbf{x} \neq \mathbf{x}_o^b$.

As in [20], we assume $\beta(\mathbf{x}_o^b) >> \beta(\mathbf{x})$ for all $\mathbf{x} \in \Omega_O$ in this chapter. This assumption produces $\eta(\mathbf{x}) \approx \frac{\sigma_0(\mathbf{x})}{\mathbf{E}[G(\mathbf{x})]-\mathbf{E}[G(\mathbf{x}_o^b)]}$ and it makes implementation of OCBA-CO easier. Based on Lemma 4, a heuristic sequential allocation procedure, OCBA-CO, is constructed to solve (16) and detailed steps of OCBA-CO are given in [20].

4.2 Methodology

In this section, we provide $\mathcal{D}$+PFM+ABA and prove its convergence properties.

4.2.1 $\mathcal{D}$+PFM+ABA

$\mathcal{D}$+PFM+ABA represents a framework combining $\mathcal{D}$+PFM of Chapter II with a revised OCBA-CO procedure, ABA. To explain $\mathcal{D}$+PFM+ABA, we first need additional notation.

$\mathcal{Q}_k \equiv$ a set of solutions to which we apply ABA at iteration k;

$\mathcal{V}_k \equiv$ a set of all visited solutions up to iteration k;

$\Delta N_k(\mathbf{x}) \equiv$ the number of new observations obtained at iteration k for $\mathbf{x}$

$\Delta T_k \equiv$ total number of new observations obtained at iteration k;

$s_{0k}^2(\mathbf{x}) \equiv$ sample variance of all $G_i(\mathbf{x})$ obtained up to iteration k;

$s_{\ell k}^2(\mathbf{x}) \equiv$ sample variance of all $H_{\ell i}(\mathbf{x})$ obtained up to iteration k, for $\ell \in \Lambda$; and

$\tilde{\mathbf{x}}_k \equiv$ the sample best among all solutions in $\mathcal{Q}_k$.

OCBA-CO always considers the same set of solutions, say Ω, but in ABA, $\mathcal{Q}_k$ can be different at each iteration. The budget allocation ABA in $\mathcal{D}$+PFM+ABA, is designed to solve the following problem:

$$\max_{\Delta N_k(\mathbf{x}), \forall \mathbf{x} \in \mathcal{Q}_k} \text{PCS} \quad \text{subject to} \quad \sum_{\mathbf{x}\in\mathcal{Q}_k} \Delta N_k(\mathbf{x}) = \Delta T_k. \tag{17}$$

Let $\hat{\eta}_k(\mathbf{x})$ represent a strongly consistent estimator of $\eta(\mathbf{x})$ and $\hat{\beta}_k(\mathbf{x})$ represent an estimator $\beta(\mathbf{x})$ at iteration k. $\tilde{\mathcal{Q}}_{kO}$ and $\tilde{\mathcal{Q}}_{kF}$ represent estimates of $\mathcal{Q}_{kO}$ and $\mathcal{Q}_{kF}$ in implementation where $\mathcal{Q}_{kF}$ ($\mathcal{Q}_{kO}$) represents the set of solutions in $\mathcal{Q}_k$ in which optimality (feasibility) is a more dominant issue. An implementation summary of $\mathcal{D}$+PFM+ABA is provided in Figure 38.

Remark 3: In practice, an active constraint often produces an extremely small $\bar{H}_{\ell^* k}(\mathbf{x}) - q_{\ell^*}(\mathbf{x})$, which, in turn, results in an extremely large $\hat{\eta}_k(\mathbf{x})$. To prevent such cases, when calculating $\hat{\eta}_k(\mathbf{x})$, we use $\max\{\epsilon_{c\ell^*}, |\bar{H}_{\ell^* k}(\mathbf{x}) - q_{\ell^*}(\mathbf{x})|\}$ instead of $\bar{H}_{\ell^* k}(\mathbf{x}) - q_{\ell^*}(\mathbf{x})$, where $\epsilon_{c\ell^*}$ is an error tolerance for the constraint ℓ^*, the amount of error a decision maker is willing to take.

Remark 4: Similarly, if $\bar{G}_k(\mathbf{x})$ is very close to $\bar{G}_k(\tilde{\mathbf{x}}_k)$, then $\hat{\eta}_k(\mathbf{x})$ becomes extremely large. To avoid such cases, when calculating $\hat{\eta}_k(\mathbf{x})$, we use $\max\{\epsilon_{ob}, |\bar{G}_k(\mathbf{x}) - \bar{G}_k(\tilde{\mathbf{x}}_k)|\}$ instead of $\bar{G}_k(\mathbf{x}) - \bar{G}_k(\tilde{\mathbf{x}}_k)$, where ϵ_{ob} is an indifference zone parameter, practically meaningful difference in the primary performance measure worth detecting.

When a problem includes a small search space Θ, then we can simply sample all solutions at every k (i.e., $\mathcal{Q}_k = \Theta$). However, if a search space Θ is large, then sampling all solutions at every k is not desirable (or impossible). Instead, existing DOvS algorithms use a solution sampling strategy that forms $\mathcal{Q}_k$ as a subset of Θ. Depending on DOvS algorithms, $\mathcal{Q}_k$ converges to a fixed set or keeps changing:

- Converging $\mathcal{Q}_k$: As k increases, $\mathcal{Q}_k$ converges to a fixed set, $\mathcal{Q}_\infty$. This happens if SMRAS of [15], NP of [32], and CE of [12] are used as $\mathcal{D}$ because these algorithms take $\mathcal{Q}_k = \mathcal{V}_k$ and $\mathcal{V}_k$ eventually converges to Θ in the algorithms.

- Changing $\mathcal{Q}_k$: As k increases, $\mathcal{Q}_k$ keeps changing but its cardinality is fixed. This happens if NP of [28], a random search of [2], and BEESE of [5] are used as $\mathcal{D}$ because these algorithms take $\mathcal{Q}_k$ as a set of sampled solutions at iteration k.

Algorithm : $\mathcal{D}$+PFM+ABA

Step 0. Initialization:

- Set $k = 1$. Select n_0, ϵ_{ob}, and $\epsilon_{c\ell}$ for all $\ell \in \Lambda$.
- Choose a DOvS algorithm as $\mathcal{D}$, set $\mathcal{V}_k = \emptyset$, and $v_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Theta$.

Step 1. Sampling:

- Sample solutions using the solution sampling strategy of $\mathcal{D}$ to form $\mathcal{Q}_k$.
- Set $n_1(\mathbf{x}) = n_0$ and $v_k(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{Q}_k \setminus \mathcal{V}_{k-1}$ (i.e., solutions newly visited).
- Calculate $\bar{G}_k(\mathbf{x})$, $s^2_{0k}(\mathbf{x})$, $\bar{H}_{\ell k}(\mathbf{x})$, $S^1_\ell(\mathbf{x})$, and $s^2_{\ell k}(\mathbf{x})$, for all $\ell \in \Lambda$ and all $\mathbf{x} \in \mathcal{Q}_k \backslash \mathcal{V}_{k-1}$.
- Set $\mathcal{V}_k = \mathcal{V}_{k-1} \cup \mathcal{Q}_k$ and find $\tilde{\mathbf{x}}_k = \arg\min_{\mathbf{x} \in \mathcal{Q}_k} Z_k(\mathbf{x})$ where $Z_k(\mathbf{x})$ is defined in (11).

Step 2. Updating:

- Set

$$\begin{aligned}
\tilde{\mathcal{Q}}_{kO} &= \left\{\mathbf{x} \mid \mathbf{x} \neq \tilde{\mathbf{x}}_k,\ \mathbf{x} \in \mathcal{Q}_k,\ \frac{q_{\ell^*}(\mathbf{x}) - \bar{H}_{\ell^* k}(\mathbf{x})}{s_{\ell^* k}(\mathbf{x})} \leq \frac{\bar{G}_k(\mathbf{x}) - \bar{G}_k(\tilde{\mathbf{x}}_k)}{s_{0k}(\mathbf{x})}\right\} \\
\tilde{\mathcal{Q}}_{kF} &= \left\{\mathbf{x} \mid \mathbf{x} \neq \tilde{\mathbf{x}}_k,\ \mathbf{x} \in \mathcal{Q}_k,\ \frac{q_{\ell^*}(\mathbf{x}) - \bar{H}_{\ell^* k}(\mathbf{x})}{s_{\ell^* k}(\mathbf{x})} > \frac{\bar{G}_k(\mathbf{x}) - \bar{G}_k(\tilde{\mathbf{x}}_k)}{s_{0k}(\mathbf{x})}\right\}
\end{aligned}$$

where $\ell^*(\mathbf{x}) \equiv \arg\min_{\ell \in \Lambda} q_\ell - \bar{H}_{\ell k}(\mathbf{x})$, $s^2_{\ell^* k}(\mathbf{x}) = s^2_{\ell^*(\mathbf{x})k}(\mathbf{x})$, and $q_{\ell^*}(\mathbf{x}) = q_{\ell^*(\mathbf{x})}$.

- If $\mathbf{x} \in \tilde{\mathcal{Q}}_{kO}$, $\hat{\eta}_k(\mathbf{x}) = \frac{s_{0k}(\mathbf{x})}{\max\{\epsilon_{ob}, |\bar{G}_k(\mathbf{x}) - \bar{G}_k(\tilde{\mathbf{x}}_k)|\}}$. Otherwise, $\hat{\eta}_k(\mathbf{x}) = \frac{s_{\ell^* k}(\mathbf{x})}{\max\{\epsilon_{c\ell^*}, |\bar{H}_{\ell^* k}(\mathbf{x}) - q_{\ell^*}(\mathbf{x})|\}}$.

Step 3. Allocating:

- If $\mathbf{x} \notin \mathcal{Q}_k$, set $\hat{\beta}_k(\mathbf{x}) = 0$. Otherwise, find $\hat{\beta}_k(\mathbf{x})$ such that $\sum_{\mathbf{x} \in \mathcal{Q}_k} \hat{\beta}_k(\mathbf{x}) = 1$, $\frac{\hat{\beta}_k(\mathbf{x})}{\hat{\beta}_k(\mathbf{y})} = \left(\frac{\hat{\eta}_k(\mathbf{x})}{\hat{\eta}_k(\mathbf{y})}\right)^2$ for all $\mathbf{x} \neq \mathbf{y} \neq \tilde{\mathbf{x}}_k$, and $\hat{\beta}_k(\tilde{\mathbf{x}}_k) = \max(\beta^{\tilde{O}}_{k+1}, \beta^{\tilde{F}}_{k+1})$, where $\beta^{\tilde{O}}_{k+1} = s_{0k}(\tilde{\mathbf{x}}_k)\sqrt{\sum_{\mathbf{x} \in \tilde{\mathcal{Q}}_{kO}} \left(\frac{\hat{\beta}_k(\mathbf{x})}{s_{0k}(\mathbf{x})}\right)^2}$ and $\beta^{\tilde{F}}_{k+1} = \hat{\beta}_k(\mathbf{x}) \left(\frac{\hat{\eta}_k(\tilde{\mathbf{x}}_k)}{\hat{\eta}_k(\mathbf{x})}\right)^2$ for all $\mathbf{x} \neq \tilde{\mathbf{x}}_k$.
- Determine the number of additional simulation observations for each $\mathbf{x} \in \mathcal{Q}_k$:

$$\Delta N_k(\mathbf{x}) = \lfloor \Delta T_k \hat{\beta}_k(\mathbf{x}) \rfloor + \mathbb{I}\big\{U < \Delta T_k \hat{\beta}_k(\mathbf{x}) - \lfloor \Delta T_k \hat{\beta}_k(\mathbf{x}) \rfloor\big\}, \tag{18}$$

where $\mathbb{I}\{\cdot\}$ represent an indicator function and U is a uniform random variable between 0 and 1. If $\Delta N_k(\mathbf{x}) > 0$, then set $v_k(\mathbf{x}) = v_{k-1}(\mathbf{x}) + 1$ and $\Delta n_{v_k}(\mathbf{x}) = \Delta N_k(\mathbf{x})$. Otherwise, set $v_k(\mathbf{x}) = v_{k-1}(\mathbf{x})$ and $\Delta n_{v_k}(\mathbf{x}) = 0$.

- Adjust the allocation so that $\sum_{\mathbf{x} \in \mathcal{Q}_k} \Delta n_{v_k}(\mathbf{x}) = \Delta T_k$.

Step 4. Simulating:

- Obtain $\Delta n_{v_k}(\mathbf{x})$ additional observations and update $n_{v_k}(\mathbf{x}) = n_{v_{k-1}}(\mathbf{x}) + \Delta n_{v_k}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{Q}_k$.
- Update $\bar{G}_{k+1}(\mathbf{x})$, $s^2_{0,k+1}(\mathbf{x})$, $\bar{H}_{\ell,k+1}(\mathbf{x})$, $S^{v_{k+1}}_\ell(\mathbf{x})$, and $s^2_{\ell,k+1}(\mathbf{x})$, for all $\ell \in \Lambda$ and for all $\mathbf{x} \in \mathcal{Q}_k$.

Step 5. Stopping Rule:

- Update the sample best $\hat{x}^*_k = \arg\min_{\mathbf{x} \in \mathcal{V}_k} Z_k(\mathbf{x})$.
- If a stopping rule is satisfied, then stop and return $\hat{\mathbf{x}}^*_k$ as the best feasible solution. Otherwise, update the solution sampling strategy, set $k \leftarrow k + 1$, and go to Step 1.

Figure 38: Algorithmic statements of an $\mathcal{D}$+PFM+ABA.

Locally convergent algorithms, such as COMAPSS of [14] and [40], can also be used as $\mathcal{D}$. In this case, Q_k converges to a set of neighbor solutions of one of a local optimal solution. However, stochastic constraints and a definition of a local optimum in [14] and [40] can unexpectedly create too many local optima near the infeasible and feasible boundary. This will be further discussed in Section 4.3.2.

4.2.2 Convergence Properties

We first provide a lemma needed to guarantee convergence of $\mathcal{D}$+PFM+ABA.

Lemma 5 *Under Assumptions 1 and 2, if $\Delta T_k > 0$ for any k, then $\mathcal{D}$+PFM+ABA guarantees*

$$\mathbf{P}\big[\lim_{k\to\infty} v_k(\mathbf{x}) = \infty\big] = 1 \quad \textit{and} \quad \mathbf{P}\big[\lim_{k\to\infty} n_{v_k}(\mathbf{x}) = \infty\big] = 1 \textit{ for any } \mathbf{x} \in \Theta.$$

Proof of Lemma 5. If $\Delta T_k \to \infty$ as $k \to \infty$, then $\Delta N_k(\mathbf{x}) \to \infty$ because $\beta_k(\mathbf{x}) > 0$ for any $\mathbf{x} \in \mathcal{Q}_k$ at any k. On the other hand, if ΔT_k is finite, then $\Delta N_k(\mathbf{x}) = \lfloor \Delta T_k \hat{\beta}_k(\mathbf{x}) \rfloor + \mathbb{I}\big\{U < \Delta T_k \hat{\beta}_k(\mathbf{x}) - \lfloor \Delta T_k \hat{\beta}_k(\mathbf{x}) \rfloor\big\}$ can be 0 and a "sampled" solution is not always "visited". However, for any $\mathbf{x}$ and k, since $\beta_k(\mathbf{x}) > 0$, $\mathbf{P}\big\{U < \Delta T_k \hat{\beta}_k(\mathbf{x}) - \lfloor \Delta T_k \hat{\beta}_k(\mathbf{x}) \rfloor\big\} > 0$ (i.e., the probability of visiting $\mathbf{x}$ is always positive when $\mathbf{x}$ is sampled at iteration k, which guarantees $v_k(\mathbf{x}) \to \infty$). Therefore, with Assumption 2, the result directly follows. □

Remind that in this chapter, APS$_c$ is used with finite ΔT_k and PS$_f$ is used with $\Delta T_k \to \infty$. Then, the following corollary shows the convergence properties of $\mathcal{D}$+PFM+ABA.

Corollary 2 *Under Assumptions 1, 2, and 4, $\mathcal{D}$+PFM+ABA satisfies Theorems 3 and 4 when APS$_c$ (with finite ΔT_k) is used. Similarly, $\mathcal{D}$+PFM+ABA satisfies Theorem 6 when PS$_f$ (with $\Delta T_k \to \infty$) is used.*

Proof of Corollary 2. The results follow directly from Lemma 5.□

Most OCBA methods calculate computing budget based on the total number of observations obtained up to the current iteration over Θ as in OCBA-CO procedure of [20]. However, ABA uses a different allocation scheme (18) that calculates computing budget only based on the number of new observations over $\mathcal{Q}_k$. The next theorem shows that the allocation scheme (18) still guarantees that the number of observations obtained for solution $\mathbf{x}$ up to iteration k, $n_{v_k}(\mathbf{x})$, is proportional to $\beta(\mathbf{x})$, if $\hat{\beta}_k(\mathbf{x})$ is a strongly consistent estimator of $\beta(\mathbf{x})$.

Theorem 8 *Assume $\hat{\beta}_k(\mathbf{x})$ is a strongly consistent estimator of $\beta(\mathbf{x})$. For a finite constant ΔT, if either $\lim_{k\to\infty}\Delta T_k = \Delta T$ or $\lim_{k\to\infty}\Delta T_k = \infty$ holds, then (18) of $\mathcal{D}$+PFM+OCBA with converging $\mathcal{Q}_k$ (i.e., $\mathcal{Q}_k = \mathcal{V}_k$) guarantees that $\lim_{k\to\infty}\frac{n_{v_k}(\mathbf{x})}{n_{v_k}(\mathbf{y})} = \frac{\beta(\mathbf{x})}{\beta(\mathbf{y})}$ for any $\mathbf{x}\neq\mathbf{y}$, $\mathbf{x}\in\Theta$, and $\mathbf{y}\in\Theta$.*

Proof of Theorem 8. Assumption 2 guarantees $\mathcal{V}_k \to \Theta$. We first provide the proof with $\lim_{k\to\infty}\Delta T_k = \Delta T$. For any $\mathbf{x}\in\Theta$, by CMT, $\Delta T_k\hat{\beta}_k(\mathbf{x}) \xrightarrow{a.s.} \Delta T\beta(\mathbf{x})$, as $k\to\infty$. For $\mathbf{x}\in\Theta$,

$$\begin{aligned}
\lim_{k\to\infty}\frac{n_{v_k}(\mathbf{x})}{k} &= \lim_{k\to\infty}\frac{n_0+\sum_{j=1}^{k}\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor + \mathbb{I}\big\{U<\Delta T_j\hat{\beta}_j(\mathbf{x})-\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor\big\}}{k}\\
&= \lim_{k\to\infty}\frac{n_0}{k}+\frac{\sum_{j=1}^{k}\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor}{k}+\frac{\sum_{j=1}^{k}\mathbb{I}\big\{U<\Delta T_j\hat{\beta}_j(\mathbf{x})-\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor\big\}}{k}\\
&= \lfloor\Delta T\beta(\mathbf{x})\rfloor + \mathbf{P}\big\{U<\Delta T\beta(\mathbf{x})-\lfloor\Delta T\beta(\mathbf{x})\rfloor\big\} \quad \text{(By Lemmas 1 and 3)}\\
&= \Delta T\beta(\mathbf{x}). \qquad (19)
\end{aligned}$$

For any $\mathbf{x}\neq\mathbf{y}$, $\mathbf{x}\in\Theta$, and $\mathbf{y}\in\Theta$, (19) implies

$$\lim_{k\to\infty}\frac{n_{v_k}(\mathbf{x})}{n_{v_k}(\mathbf{y})} = \lim_{k\to\infty}\frac{n_{v_k}(\mathbf{x})}{k}\frac{k}{n_{v_k}(\mathbf{y})} = \frac{\beta(\mathbf{x})}{\beta(\mathbf{y})}.$$

Next, we provide the proof with $\lim_{k\to\infty}\Delta T_k = \infty$. For a fixed k and $1\le j\le k$, define a sequence $\chi_{j,k}(\mathbf{x}) = \frac{\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor+\mathbb{I}\big\{U<\Delta T_j\hat{\beta}_j(\mathbf{x})-\lfloor\Delta T_j\hat{\beta}_j(\mathbf{x})\rfloor\big\}}{\Delta T_k}$. Then,

$$\frac{\Delta T_k\hat{\beta}_k(\mathbf{x})-1}{\Delta T_k}\le\frac{\lfloor\Delta T_k\hat{\beta}_k(\mathbf{x})\rfloor}{\Delta T_k}\le\chi_{k,k}(\mathbf{x})\le\frac{\lceil\Delta T_k\hat{\beta}_k(\mathbf{x})\rceil}{\Delta T_k}\le\frac{\Delta T_k\hat{\beta}_k(\mathbf{x})+1}{\Delta T_k}$$

We have $\lim_{k\to\infty} \frac{\Delta T_k \hat{\beta}_k(\mathbf{x})-1}{\Delta T_k} = \lim_{k\to\infty} \hat{\beta}_k(\mathbf{x}) - \frac{1}{\Delta T_k} = \beta(\mathbf{x})$ and similarly, we also have $\lim_{k\to\infty} \frac{\Delta T_k \hat{\beta}_k(\mathbf{x})+1}{\Delta T_k} = \beta(\mathbf{x})$. Therefore, we have $\lim_{j\to\infty} \chi_{j,\infty}(\mathbf{x}) = \beta(\mathbf{x})$.

$$\begin{aligned}
\lim_{k\to\infty} \frac{n_{v_k}(\mathbf{x})}{k\Delta T_k} &= \lim_{k\to\infty} \frac{n_0 + \sum_{j=1}^{k} \lfloor \Delta T_j \hat{\beta}_j(\mathbf{x}) \rfloor + \mathbb{I}\{U < \Delta T_j \hat{\beta}_j(\mathbf{x}) - \lfloor \Delta T_j \hat{\beta}_j(\mathbf{x}) \rfloor\}}{k\Delta T_k} \\
&= \lim_{k\to\infty} \frac{n_0}{k\Delta T_k} + \frac{\sum_{j=1}^{k} \chi_{j,k}(\mathbf{x})}{k} \\
&= \beta(\mathbf{x}) \quad \text{(By Lemma 3).} \qquad (20)
\end{aligned}$$

For any $\mathbf{x} \neq \mathbf{y}$, $\mathbf{x} \in \Theta$, and $\mathbf{y} \in \Theta$, (20) implies

$$\begin{aligned}
\lim_{k\to\infty} \frac{n_{v_k}(\mathbf{x})}{n_{v_k}(\mathbf{y})} &= \lim_{k\to\infty} \frac{n_{v_k}(\mathbf{x})}{k\Delta T_k} \frac{k\Delta T_k}{n_{v_k}(\mathbf{y})} \\
&= \frac{\beta(\mathbf{x})}{\beta(\mathbf{y})}.\square
\end{aligned}$$

Remark 5: By Lemma 5 and the strong law of large numbers (SLLN), all estimates converge to true values (i.e., as $k \to \infty$, $\bar{G}_k(\mathbf{x}) \xrightarrow{a.s.} \mathbf{E}[G(\mathbf{x})]$, $s_{0k}^2(\mathbf{x}) \xrightarrow{a.s.} \mathbf{Var}[G(\mathbf{x})]$, $\bar{H}_{\ell k}(\mathbf{x}) \xrightarrow{a.s.} \mathbf{E}[H_\ell(\mathbf{x})]$, and $s_{\ell k}^2(\mathbf{x}) \xrightarrow{a.s.} \mathbf{Var}[H_\ell(\mathbf{x})]$ for any ℓ and any $\mathbf{x} \in \Theta$). As a result, the sets, $\mathcal{Q}_{kO}$ and $\mathcal{Q}_{kF}$, can be determined asymptotically correct and the sample allocation $\hat{\beta}_k(\mathbf{x})$ is an asymptotically consistent estimator $\beta(\mathbf{x})$ under assumptions as $k \to \infty$.

If we take $\mathcal{Q}_k$ as a set of sampled solutions at iteration k (i.e., changing $\mathcal{Q}_k$) in Theorem 8 while keeping the other conditions same, then the main results in Theorem 8 do not hold any more. In fact, we cannot derive any asymptotic results on allocating the total number of observations over Θ, $\sum_{\mathbf{x}\in\Theta} n_{v_k}(\mathbf{x})$, with changing $\mathcal{Q}_k$.

4.3 Numerical Result

In this section, we revisit the three numerical examples from Section 2.4: (i) the three-system example, (ii) the Goldstein-Price problem, and (iii) the (s, S) inventory policy problem. We test $\mathcal{D}$+PFM+ABA on these three examples with both tight optimal and strictly feasible optimal solutions.

$\mathcal{D}$+PFM(APS$_c$)+ABA and $\mathcal{D}$+PFM(PS$_f$)+ABA represent $\mathcal{D}$+PFM+ABA with APS$_c$ and PS_f, respectively. APS$_c$ uses same parameters from Table 3, but γ_ℓ^k is set to $\gamma_\ell^k = 0$.

For the three-system example, we do not need a DOvS algorithm $\mathcal{D}$ because there are only three solutions. We sample all three solutions at each iteration (i.e., $\mathcal{Q}_k = \Theta$); apply PFM(APS$_c$)+ABA; and compare their performances with those of PFM(APS$_c$), PFM(PS$_f$), and OCBA-CO. We set $\Delta T_k = 9$ when APS$_c$ is used and $\Delta T_k = 3(n_0 + \lceil \log k \rceil)$ when PS$_f$ is used.

For the Goldstein-Price problem and the (s, S) inventory policy problem, the search space is large and a DOvS algorithm is needed. We take NP as $\mathcal{D}$ and combine it with PFM+ABA. The performance of NP+PFM+ABA is compared with those of NP+PFM. When we combine NP+PFM with ABA, we take $\mathcal{Q}_k$ as a set of sampled solutions at iteration k and set $\Delta T_k = \Delta n \cdot |\mathcal{Q}_k \cap \mathcal{V}_{k-1}|$ for APS$_c$ and $\Delta T_k = \sum_{\mathbf{x} \in \mathcal{Q}_k \cap \mathcal{V}_{k-1}} (n_0 + \lceil \log v_k(\mathbf{x}) \rceil)$ for PS$_f$, where $| \cdot |$ represents the cardinality of a set. The rest of parameter settings are the same as in Section 2.4. NP+PFM without ABA uses equal allocation in a sense that the same number of additional observations across all sampled solutions except newly visited solutions are obtained at each iteration for APS$_c$ or the same number of additional observations are obtained for solutions with the same number of visits for PS$_f$.

4.3.1 Three-System Example

Mean and variance configuration of three systems are given in Section 2.4.3. In this problem, we arbitrary select $\epsilon_{c1} = \epsilon_{ob} = 0.01$ for ABA and $\epsilon_{10} = 0.04$ for PFM.

Figure 39 shows the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ over 500 macro replications for the three-system example. Each run terminates when the total number of observations obtained from all visited solutions so far reaches 3,000. As shown in Figure

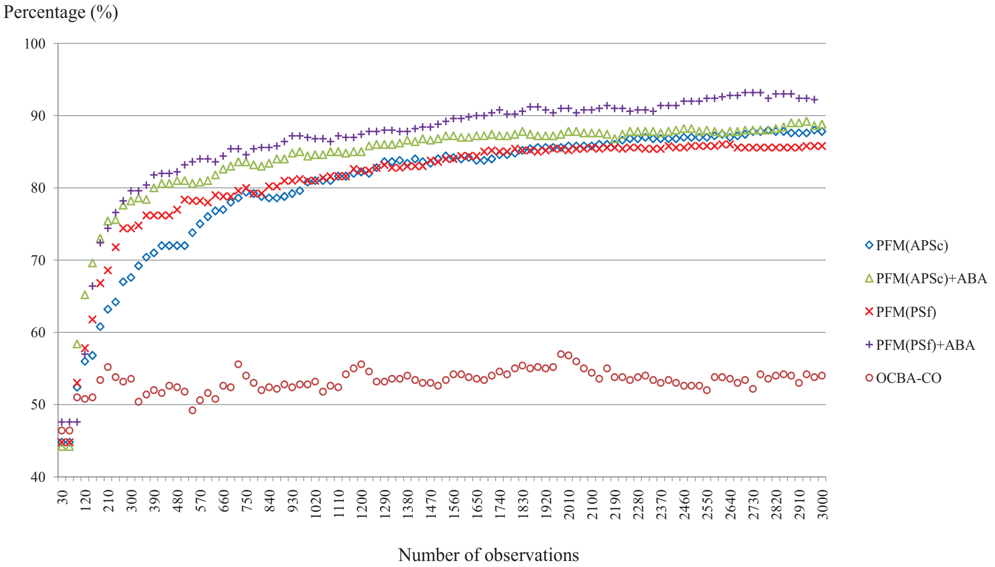


Figure 39: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ for the three-system example with a tight solution.

39, PFM(PS$_f$)+ABA achieves the highest percentage of time over 90% while OCBA-CO achieves the lowest percentage around 50%. It is not surprising that OCBA-CO does not perform well because OCBA-CO cannot handle tight solutions. Also, comparing PFM(PS$_f$)+ABA with PFM(PS$_f$) (or PFM(APS$_c$)+ABA with PFM(APS$_c$)), one can observe improvement due to ABA over equal allocation. The percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ is about 5% $\sim$ 8% higher in both small and large observation sizes when ABA is combined with PFM(PS$_f$). However, for APS$_c$, ABA shows about 10% increase in the percentage compared to equal allocation only when the total number of observations is small but the advantage disappears as more observations are taken.

In order to examine the small-sample behaviors more closely, we change $\mathbf{E}[H_1(2)] = 0$ to $\mathbf{E}[H_1(2)] = 0.03$ so that there is no tight solution while system 2 is still the true best feasible solution. Each run terminates when the total number of observations obtained from all visited solution reaches 1,000. Figure 40 shows the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ over 500 macro replications for the three-system example. In this case, OCBA-CO achieves a higher percentage up to 80% but does not outperform

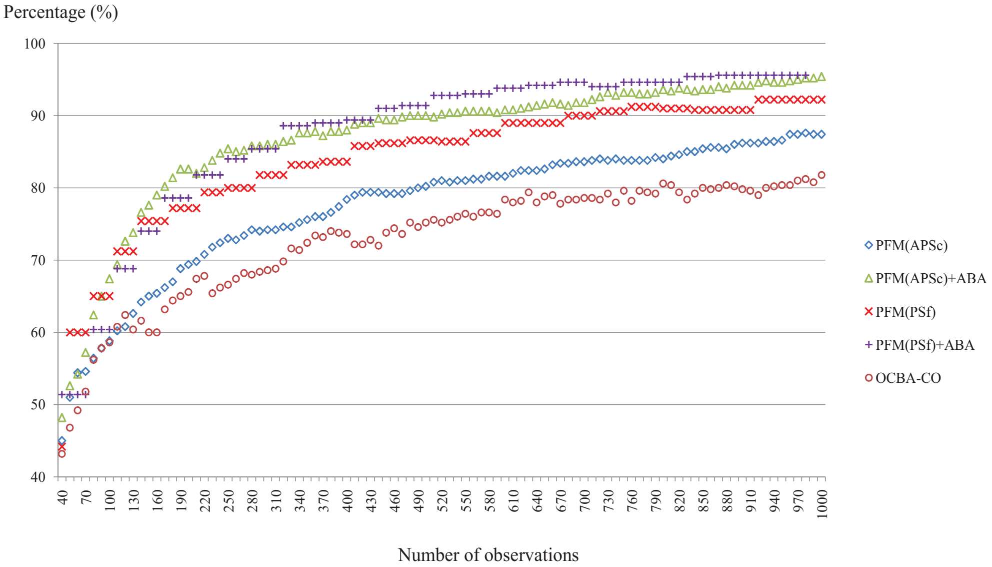


Figure 40: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ for the three-system example without any tight solution.

any variant procedure of PFM. ABA improves the performance of PFM compared to equal allocation, especially when APS_c is used.

4.3.2 Goldstein-Price Problem

We consider the Goldstein-Price problem with a difficult constraint (5) from Section 2.4.1 to examine the performance of NP+PFM+ABA. We arbitrary select an error tolerance, $\epsilon_{c1} = 0.005$, and an indifference zone paramenter $\epsilon_{ob} = 0.1$ for ABA. For PFM with PS_f, we use $\epsilon_{10} = 0.04$.

Figure 41 represents the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ over 500 macro replications for the Goldstein-Price problem with Constraint (5) and each replication is terminated with one million total number of observations. When PS_f is used as a penalty sequence, ABA slightly improves the performance of the combined procedure compared to equal allocation at the beginning, but as the total number of observations increases, both NP+PFM(PS_f)+ABA and NP+PFM(PS_f) show similar performance, achieving 90% of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$. On the other hand, when APS_c is

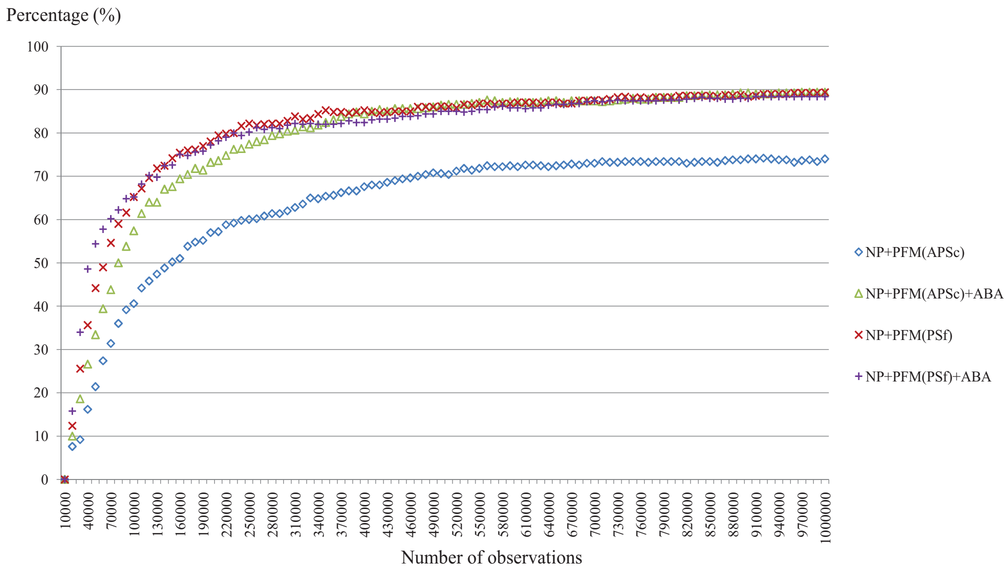


Figure 41: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ with Constraint (5).

used as a penalty sequence, NP+PFM(APS$_c$)+ABA shows a significant improvement over NP+PFM(APS$_c$). More specifically, NP+PFM(APS$_c$)+ABA achieves 90% and performs almost similar to procedures with PS$_f$ after the total number of observations reaches 310,000. However, the percentage of NP+PFM(PS$_f$) is significantly smaller than that of NP+PFM(PS$_f$)+ABA, showing up to 20% difference. As we observe in the three-system example, ABA brings more significant improvements when it is used with APS$_c$. Figure 42 shows average objective values at the sample best. The figure implies that NP+PFM(APS$_c$) tends to select superior infeasible solutions as the sample best at the beginning of the search while the other three procedures do not.

If a locally convergent DOvS algorithm, such as COMPASS of [14], is considered as $\mathcal{D}$, many local optima can be created unexpectedly by the definition of the neighbor solutions used in [14] when stochastic constraints exist. [14] and [40] define neighbor solutions as solutions that differ by ±1 in only one coordinate. For the Goldstein-Price problem with Constraint (5), Figure 43 shows that any solution on the constraint line

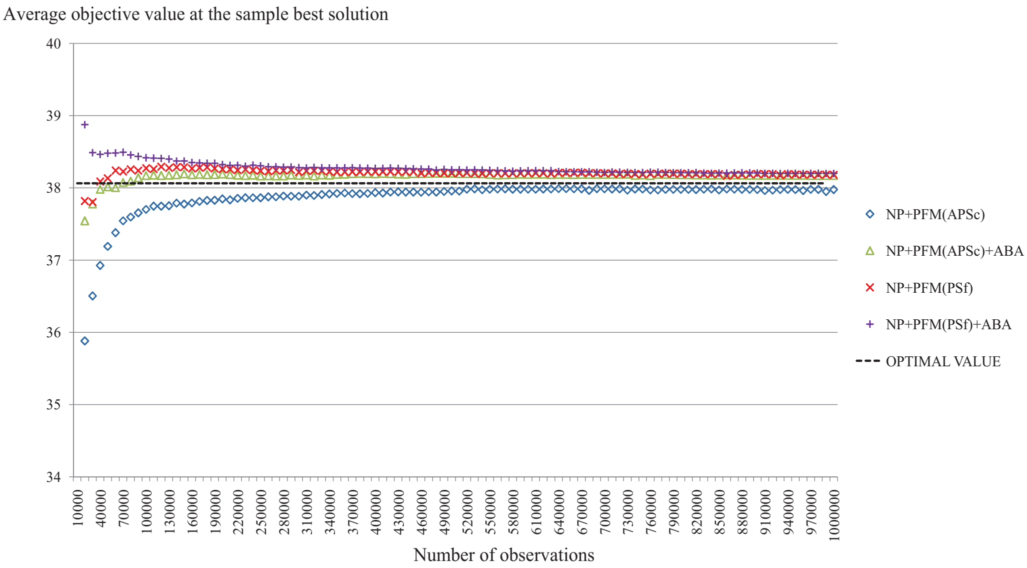


Figure 42: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ with Constraint (5).

becomes a local optimal solution. More specifically, solution $\mathbf{x}_0$ has four solutions in the neighborhood, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$. Note that$\mathbf{x}_1$ and $\mathbf{x}_2$ are infeasible but have better $\mathbf{E}[G(\mathbf{x})]$ than $\mathbf{x}_0$ while $\mathbf{x}_3$ and $\mathbf{x}_4$ are feasible and have worse $\mathbf{E}[G(\mathbf{x})]$ than $\mathbf{x}_0$. In this case, $\mathbf{x}_0$ becomes a local optimal solution because $\mathbf{x}_0$ is the best feasible solution among all solutions in the neighborhood. As a result, any solution on the constraint

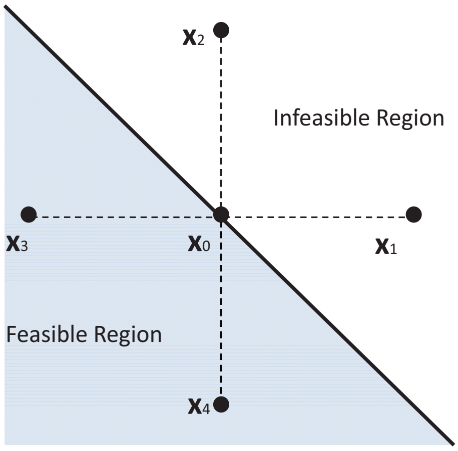


Figure 43: Local optimal solution in the Goldstein-Price problem with Constraint (5).

line becomes a local optimal solution and COMPASS is likely to stop when it reaches a solution on the constraint line. Thus, when there exist stochastic constraints, using a locally convergent algorithm as $\mathcal{D}$ needs some cautions.

Now we consider an easier constraint:

$$\mathbf{E}[-x_1 - x_2 + \psi_{1i}] \geq 1.499. \tag{21}$$

With Constraint (21), the true optimal solution is still the same but now it is strictly feasible.

Figure 44 shows the percentage of time that $\mathbf{x}_k^* = \mathbf{x}_o^b$ and Figure 45 shows the average objective value of the sample best over 500 macro replications. To focus on behaviors of procedures when the total number of observations is small, we terminate each macro replication when the total number of observations reaches 200,000. NP+PFM(PS$_f$)+ABA performs the best. ABA improves the performance of combined procedures compared to equal allocation but greater improvement is achieved when APS$_c$ is used as a penalty sequence for PFM.

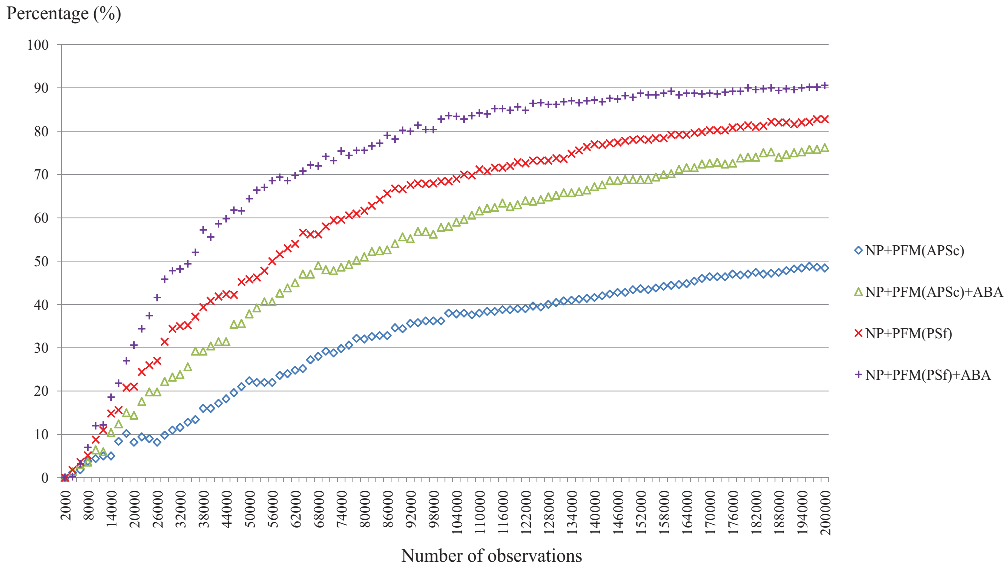


Figure 44: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ with constraint (21).

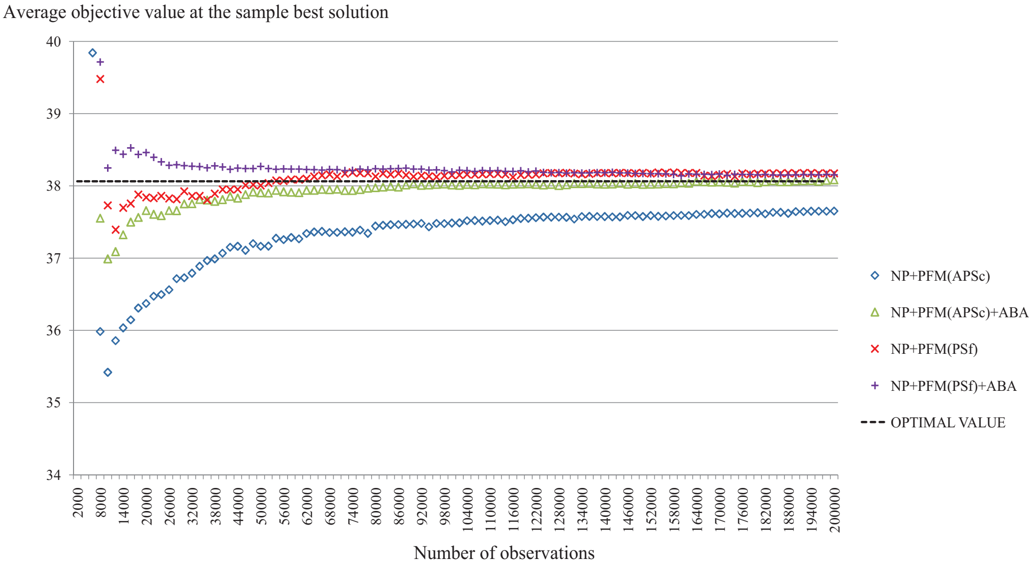


Figure 45: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ with constraint (21).

4.3.3 (s, S) Inventory Policy Problem

We revisit the (s, S) inventory policy problem in Section 2.4 to test the performance of NP+PFM+ABA when a problem includes non-normal observations for the primary and secondary performance measures and there exists correlation across performance measures. We arbitrary select an error tolerance, $\epsilon_{c1} = 0.0001$, and an indifference zone paramenter $\epsilon_{ob} = 0.1$ for ABA. For PFM with PS$_f$, we select $\epsilon_{10} = 0.0001$.

We first consider the same problem in Section 2.4 where the true optimal solution is strictly feasible but nearly tight. Figure 46 shows the percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ over 500 macro replications when each macro replication is terminated with 2,000,000 total observations. At the end of the search, NP+PFM(APS$_c$)+ABA and NP+PFM(PS$_f$)+ABA achieve over 90% while NP+PFM(APS$_c$) and NP+PFM(PS$_f$) achieve under 90%. Figure 47 shows average objective value of the sample best. Average objective values of NP+PFM(APS$_c$)+ABA and NP+PFM(PS$_f$)+ABA start above the true optimal value while NP+PFM(APS$_c$) and NP+PFM(PS$_f$) start under the true optimal value. This supports that ABA helps accurate feasibility checks at

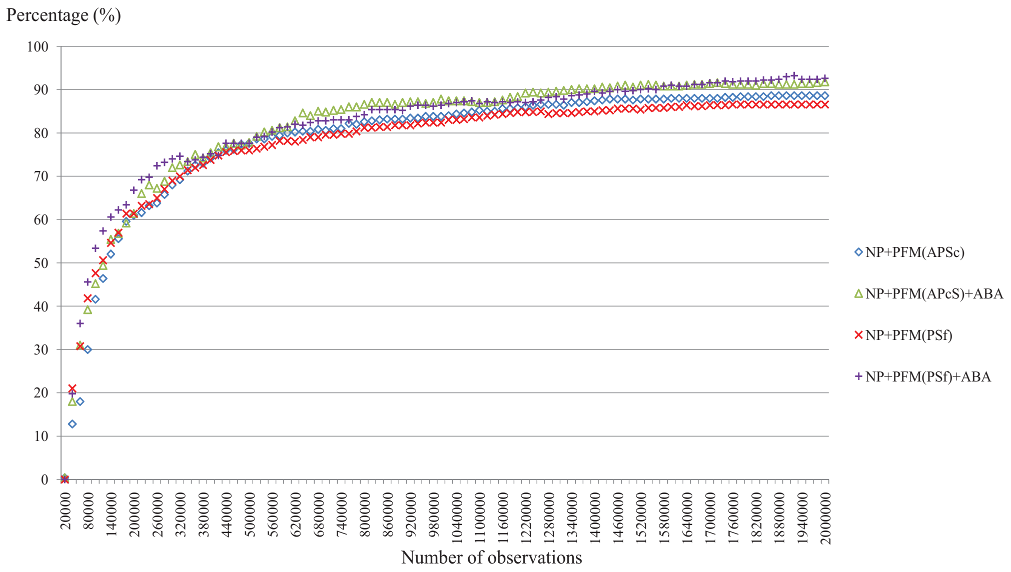


Figure 46: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ in the (s, S) inventory problem with a difficult constraint.

the beginning of the search.

Now we consider an easier constraint where failure probability should be less than equal to 0.05. The true optimal solution changes to $\mathbf{x}_o^b = (24, 58)$ and its expected cost and failure probability are 113.0864 and 0.04878, respectively.

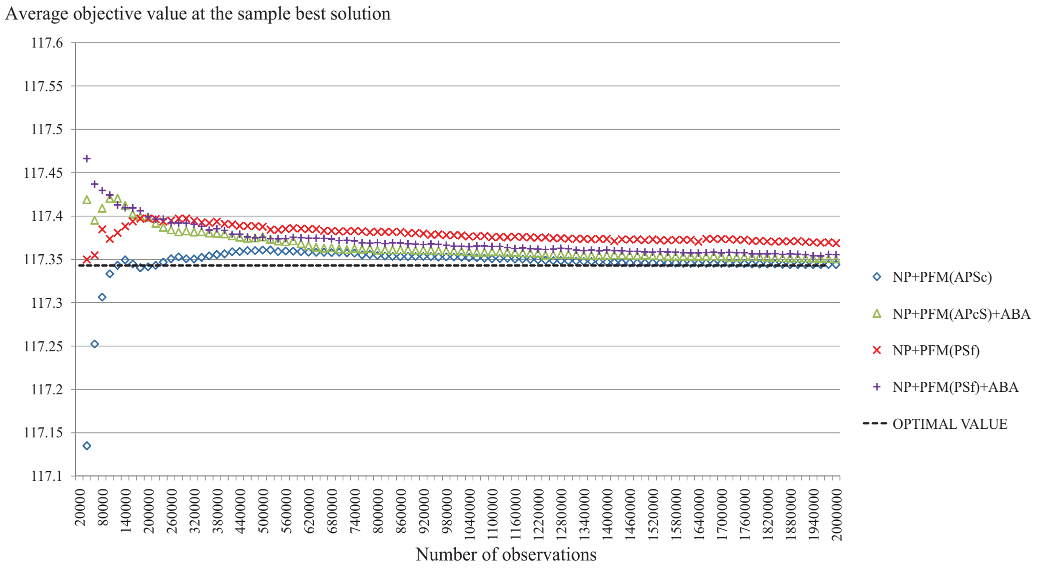


Figure 47: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ in the (s, S) inventory problem with a difficult constraint..

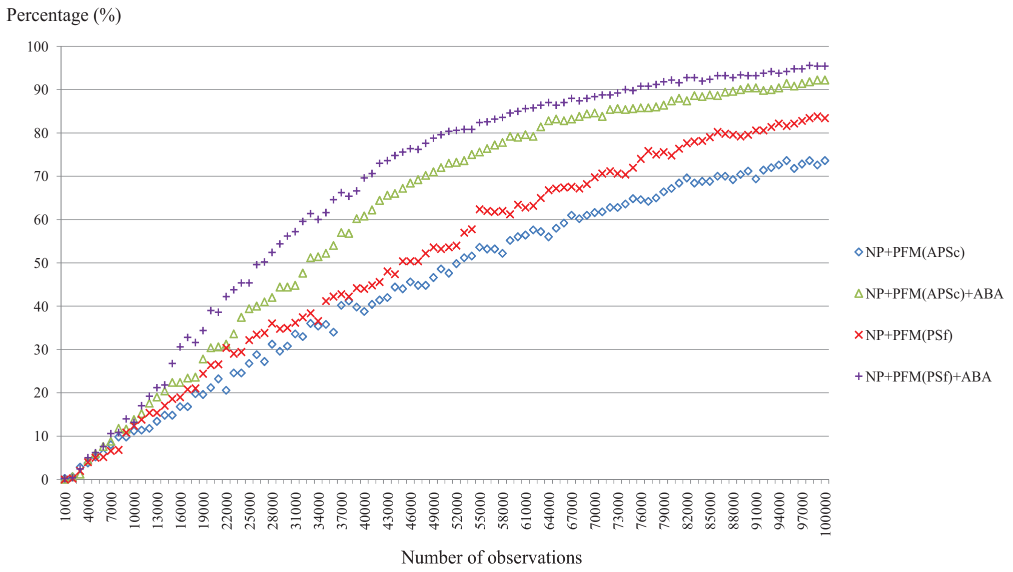


Figure 48: Percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ in the (s, S) inventory problem with a relaxed constraint.

Figures 48 and 49 show the percentage of time that $\hat{\mathbf{x}}_k^* = \mathbf{x}_o^b$ and the average objective value at the sample best over 500 macro replications. To focus on the small sample behaviors, each macro replication is terminated with 100,000 total observations. Both figures show that ABA brings significant savings compared to equal allocation. When

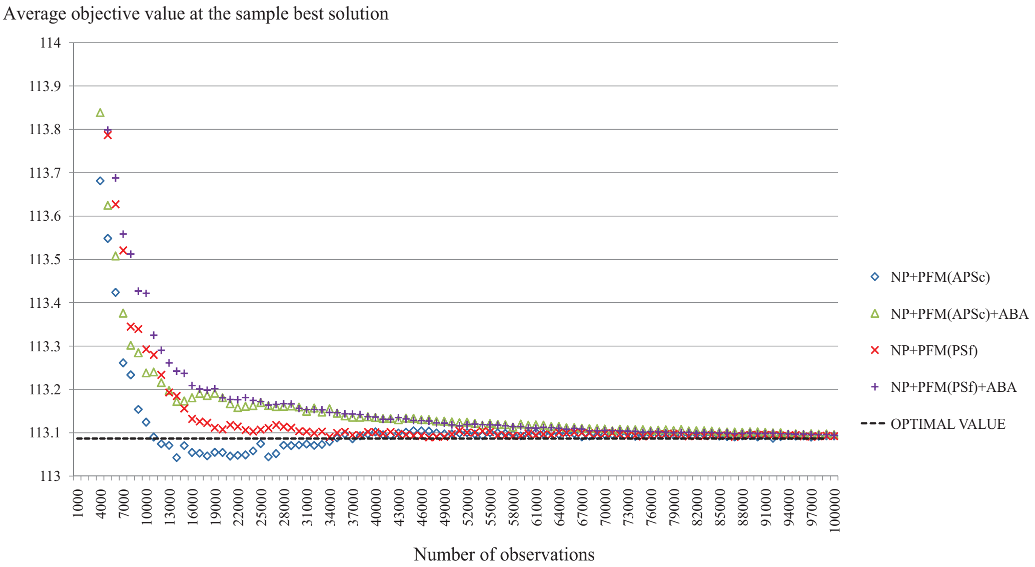


Figure 49: Average estimated objective value of $\hat{\mathbf{x}}_k^*$ in the (s, S) inventory problem with a relaxed constraint.

APS$_c$ is used as a penalty sequence in PFM, NP+PFM(APS$_c$)+ABA achieves about 20% higher percentage than NP+PFM(APS$_c$). On the other hand, when PS$_f$ is used as a penalty sequence in PFM, the combined procedure with ABA achieves about 15% higher percentage than the procedure with equal allocation.

4.4 Conclusions

In this chapter, we improve the performance of $\mathcal{D}$+PFM by combining it with ABA. ABA is a modified version of OCBA-CO of [20]. ABA can handle active constraints and satisfies necessary conditions for convergence properties of PFM. Convergence properties of the combined procedure is provided with proofs. Our experimental results on the three numerical examples show that ABA improves the performance of $\mathcal{D}$+PFM. Greater improvement is observed when the total number of observations is small/medium or APS$_c$ is used as a penalty sequence in PFM.

CHAPTER V

CONTRIBUTIONS

We present a novel method, PFM, which converts a stochastically constrained DOvS problem into a series of stochastically unconstrained DOvS problems. To the best of our knowledge, PFM is the first work that (i) adaptively determines penalty values based on observed feasibility of solutions and (ii) can handle tight solutions among simulation-based optimization algorithms. The proposed method is general enough to handle problems that occur in a wide range of applications including IE/OR, health care, environmental and energy management.

We apply PFM to an important problem in environmental management, namely the water quality monitoring network design problem for river systems. The purpose of the problem is to find the optimal location of a finite number of monitoring devices that minimizes the expected detection time of a contaminant spill event while guaranteeing good detection reliability. We formulate this problem as a DOvS problem with a stochastic constraint and present $\mathcal{D}$+PFM. Experimental results on the Altamaha River shows that our algorithm performs better than an existing popular method in environmental management in terms of both efficiency and accuracy.

Finally, we present ABA, that allocates simulation budget among sampled solutions at each iteration instead of taking equal number of observations from each sampled solution. Our experimental results on three examples shows that ABA improves the performance of $\mathcal{D}$+PFM significantly, especially when the total number of observations is small or a small budget needs to be allocated at each iteration as in APS$_c$.

APPENDIX A

Proof of Theorem 1. Assumption 2 and SLLN ensure that $\overline{G}_k(\mathbf{x})$ converges to $\mathbf{E}[G(\mathbf{x})]$ and $\overline{H}_{\ell k}(\mathbf{x})$ converges to $\mathbf{E}[H_\ell(\mathbf{x})]$ almost surely. By the continuous mapping theorem (CMT, Theorem 5.5 of [7]) and Condition 1, the results follow. □

Proof of Theorem 2. Recall that $\mathcal{V}_k$ is a set of all solutions visited by $\mathcal{D}$+PFM up to iteration k and Θ_f is the set of all feasible solutions. At iteration k, $\mathcal{D}$+PFM selects $\hat{\mathbf{x}}_k^*$ such that $\hat{\mathbf{x}}_k^* \equiv \text{argmin}_{\mathbf{x}\in\mathcal{V}_k} Z_k(\mathbf{x})$.

As $k \to \infty$,

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{V}_k} Z_k(\mathbf{x}) &\overset{a.s.}{\longrightarrow} \min_{\mathbf{x}\in\Theta} Z_k(\mathbf{x}) \quad \text{(by Assumption 2)} \\ &\overset{a.s.}{\longrightarrow} \min_{\mathbf{x}\in\Theta_f} \mathbf{E}[G(\mathbf{x})]. \quad \text{(by Theorem 1)} \end{aligned}$$

This implies that $\mathbf{P}\{\lim_{k\to\infty} Z_k(\hat{\mathbf{x}}_k^*) = \min_{\mathbf{x}\in\Theta_f} \mathbf{E}[G(\mathbf{x})]\} = 1$.

□

Proof of Theorem 3. We start with the case for $\ell \in \Lambda_{S(\mathbf{x})}$. Let $\zeta_{\ell r}^s(\mathbf{x})$ represent a random variable $\zeta_{\ell r}^s(\mathbf{x}) = \sum_{i=n_{r-1}(\mathbf{x})+1}^{n_{r-1}(\mathbf{x})+\Delta n_r(\mathbf{x})} \frac{H_{\ell i}(\mathbf{x})-\mathbf{E}[H_\ell(\mathbf{x})]}{\sqrt{\Delta n_r(\mathbf{x})}}$. Note that $\zeta_{\ell r}^s(\mathbf{x})$, $r = 1, 2, \ldots$, are independent random variables with mean zero and a finite variance $\mathbf{Var}(H_{\ell i}(\mathbf{x}))$. Let $\mathbb{I}(\cdot)$ represent an indicator function. Then,

$$\begin{aligned} &\mathbb{I}\left\{S_\ell^{v_k}(\mathbf{x}) \geq 0\right\} \\ &= \mathbb{I}\left\{\frac{\sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}^s(\mathbf{x})}{v_k(\mathbf{x})} + \frac{\sum_{r=1}^{v_k(\mathbf{x})} \sum_{i=n_{r-1}(\mathbf{x})+1}^{i=n_{r-1}(\mathbf{x})+\Delta n_r(\mathbf{x})} \frac{\mathbf{E}[H_\ell(\mathbf{x})]-q_\ell}{\sqrt{\Delta n_r(\mathbf{x})}}}{v_k(\mathbf{x})} \geq 0\right\} \\ &= \mathbb{I}\left\{\frac{\sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}^s(\mathbf{x})}{v_k(\mathbf{x})} \geq -\frac{\sum_{r=1}^{v_k(\mathbf{x})} \sqrt{\Delta n_r(\mathbf{x})} \cdot (\mathbf{E}[H_\ell(\mathbf{x})] - q_\ell)}{v_k(\mathbf{x})}\right\} \\ &\geq \mathbb{I}\left\{\frac{\sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}^s(\mathbf{x})}{v_k(\mathbf{x})} \geq q_\ell - \mathbf{E}[H_\ell(\mathbf{x})]\right\} \qquad (\because\ \Delta n_r(\mathbf{x}) \geq 1 \text{ and } \mathbf{E}[H_\ell(\mathbf{x})] - q_\ell > 0) \end{aligned}$$

By Lemma 1, there exists $N_\ell(\mathbf{x}) \in \mathbb{Z}^+$ such that $\frac{\sum_{r=1}^{v_k(\mathbf{x})} \zeta_{\ell r}^S(\mathbf{x})}{v_k(\mathbf{x})} \geq q_\ell - \mathbf{E}[H_\ell(\mathbf{x})]$ for any $k \geq N_\ell(\mathbf{x})$. This implies that solution $\mathbf{x}$ is declared as feasible after $k \geq N_\ell(\mathbf{x})$ and

θ_d is kept multiplied to $\lambda_\ell^{v_k}(\mathbf{x})$, resulting the sequence to converge to 0 almost surely.

Using similar arguments, it can be shown that $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for any $\ell \in \Lambda_{I(\mathbf{x})}$.

□

Proof of Lemma 3. For any $\epsilon_b > 0$, there exists a positive integer N_b such that $|b_n - c| < \frac{\epsilon_b}{2}$ for all $n \geq N_b$. Since b_n converges to a finite value, there exists a finite constant M_b such that $|b_n - c| < M_b$ for all $n < N_b$. Therefore, we can select $N \geq N_b$ such that $\frac{(N_b-1)M_b}{N} < \frac{\epsilon_b}{2}$. By the triangular inequality, for $n \geq N$,

$$\begin{aligned}\left|\frac{1}{n}\sum_{r=1}^{n} b_r - c\right| &\leq \frac{1}{n}\sum_{r=1}^{n}|b_r - c| \\ &< \frac{(N_b-1)M}{n} + \frac{(n-N_b+1)\epsilon_b}{2n} \\ &< \frac{\epsilon_b}{2} + \frac{\epsilon_b}{2} = \epsilon_b.\end{aligned}$$

Thus, $\lim_{n\to\infty} \frac{1}{n}\sum_{r=1}^{n} b_n = c$. □

Proof of Theorem 4. By Assumption 1, $\sigma_\ell^2(\mathbf{x}) = \mathbf{Var}(H_{\ell i}(\mathbf{x})) < \infty$. Let $T_\ell^{v_k}(\mathbf{x}) = \sum_{r=1}^{v_k(\mathbf{x})} \mathbb{I}\{S_\ell^r(\mathbf{x}) \geq 0\}$ (the number of feasible decisions up to the $v_k(\mathbf{x})$th visit). Then for any $\epsilon > 0$,

$$\begin{aligned}&\lim_{k\to\infty} \mathbf{P}\left[\lambda_\ell^{v_k}(\mathbf{x}) < \epsilon\right] \\ &= \lim_{k\to\infty} \mathbf{P}\left[\theta_d^{T_\ell^{v_k}(\mathbf{x})} \cdot \theta_a^{v_k(\mathbf{x})-T_\ell^{v_k}(\mathbf{x})} < \epsilon\right] \\ &= \lim_{k\to\infty} \mathbf{P}\left[\theta_a^{v_k(\mathbf{x})} \cdot \left(\frac{\theta_d}{\theta_a}\right)^{T_\ell^{v_k}(\mathbf{x})} < \epsilon\right] \\ &= \lim_{k\to\infty} \mathbf{P}\left[T_\ell^{v_k}(\mathbf{x})(\log\theta_d - \log\theta_a) < \log\epsilon - v_k(\mathbf{x})\log\theta_a\right] \\ &= \lim_{k\to\infty} \mathbf{P}\left[\frac{T_\ell^{v_k}(\mathbf{x})}{v_k} > \frac{\frac{\log\epsilon}{v_k} - \log\theta_a}{\log\theta_d - \log\theta_a}\right] \\ &= 1 - \frac{2}{\pi}\arcsin\sqrt{\frac{-\log\theta_a}{\log\theta_d - \log\theta_a}} \quad \text{(by Lemma 2).}\end{aligned}$$

Therefore, $\lambda_\ell^{v_k}(\mathbf{x})$ converges to 0 with probability $1 - \frac{2}{\pi}\arcsin\sqrt{\frac{-\log\theta_a}{\log\theta_d - \log\theta_a}}$ and diverges to ∞ with probability $\frac{2}{\pi}\arcsin\sqrt{\frac{-\log\theta_a}{\log\theta_d-\log\theta_a}}$ as $k \to \infty$. □

Proof of Theorem 5. Let $J_r \equiv \mathbb{I}\{\zeta_{\ell r}(\mathbf{x}) < 0\}$. Then, J_r, $r = 1, 2, \ldots$ are independent Bernoulli random variables with success probability $\mathbf{E}[J_r]$.

$$\begin{aligned}
&\lim_{r\to\infty} \mathbf{E}[J_r] \\
&= \lim_{\Delta n_r(\mathbf{x})\to\infty} \mathbf{P}\left\{ \frac{1}{\Delta n_r(\mathbf{x})} \sum_{i=1}^{\Delta n_r(\mathbf{x})} H_{\ell i}(\mathbf{x}) < q_\ell \right\} \\
&= \lim_{\Delta n_r(\mathbf{x})\to\infty} \mathbf{P}\left\{ \frac{\sum_{i=1}^{\Delta n_r(\mathbf{x})} (H_{\ell i}(\mathbf{x}) - \mathbf{E}[H_{\ell i}(\mathbf{x})])}{\sigma_\ell(\mathbf{x})\sqrt{\Delta n_r(\mathbf{x})}} < \frac{\sqrt{\Delta n_r(\mathbf{x})}(q_\ell - \mathbf{E}[H_{\ell i}(\mathbf{x})])}{\sigma_\ell(\mathbf{x})} \right\}
\end{aligned}$$

By Lemma 3 and the CLT, we have

$$\lim_{r\to\infty} \frac{\sum_{j=1}^{r} \mathbf{E}[J_j]}{r} = \lim_{r\to\infty} \mathbf{E}[J_r] = \begin{cases} \Phi(-\infty) = 0, & \text{if } \ell \in \Lambda_{S(\mathbf{x})}; \\ \Phi(\infty) = 1, & \text{if } \ell \in \Lambda_{I(\mathbf{x})}; \\ \Phi(0) = 0.5, & \text{if } \ell \in \Lambda_{A(\mathbf{x})}, \end{cases} \tag{22}$$

where $\Phi(\cdot)$ represents the cumulative distribution of the standard normal random variable.

For $\ell \in \Lambda_{A(\mathbf{x})}$, by Lemma 1 and (22) we have

$$\mathbf{P}\left\{ \lim_{v_k(\mathbf{x})\to\infty} \left| \hat{p}_\ell^{v_k}(\mathbf{x}) - \frac{\sum_{r=1}^{v_k(\mathbf{x})} \mathbf{E}[J_r]}{v_k(\mathbf{x})} \right| + \left| \frac{\sum_{r=1}^{v_k(\mathbf{x})} \mathbf{E}[J_r]}{v_k(\mathbf{x})} - 0.5 \right| = 0 \right\} = 1,$$

and the triangular inequality implies that

$$\mathbf{P}\left\{ \lim_{v_k(\mathbf{x})\to\infty} |\hat{p}_\ell^{v_k}(\mathbf{x}) - 0.5| = 0 \right\} = 1.$$

Similar arguments apply to $\ell \in \Lambda_{I(\mathbf{x})} \cup \Lambda_{S(\mathbf{x})}$, which completes the proof. □

Proof of Theorem 6. (i) When $\ell \in \Lambda_{S(\mathbf{x})}$: By Theorem 3, there exists $N_\ell(\mathbf{x}) \in \mathbb{Z}^+$ such that $\mathbf{x}$ is declared as feasible for any $k \geq N_\ell(\mathbf{x})$. Thus, the penalty sequence receives $\delta_\ell^{v_k}(\mathbf{x})$ which is a number smaller than 1. As a result, the sequence converges to 0 almost surely.

When $\ell \in \Lambda_{A(\mathbf{x})}$: By Theorem 5, for any $k \geq N_\ell^{'}(\mathbf{x})$,

$$0.5 - \epsilon_{\ell 0} \leq \hat{p}_\ell^{v_k}(\mathbf{x}) \leq 0.5 + \epsilon_{\ell 0}, \tag{23}$$

for $0 < \epsilon_{\ell 0} < \gamma_\ell^k$. Thus, the penalty sequence receives either $\alpha_\ell^{v_k}(\mathbf{x})$ or $\delta_\ell^{v_k}(\mathbf{x})$ for any $k \geq N_\ell^{'}(\mathbf{x})$. As both $\alpha_\ell^{v_k}(\mathbf{x})$ and $\delta_\ell^{v_k}(\mathbf{x})$ are less than 1, the sequence converges to 0 almost surely.

$\underline{\text{(ii) When } \ell \in \Lambda_{I(\mathbf{x})}}$: By Theorem 3, there exists $N_\ell(\mathbf{x}) \in \mathbb{Z}^+$ such that the penalty sequence for the constraint ℓ of an infeasible solution $\mathbf{x}$ receives only appreciation function $\alpha_\ell^{v_k}(\mathbf{x})$ after $N_\ell(\mathbf{x})$. Also, by Theorem 5, there exists $N_\ell^{'}(\mathbf{x})$ such that for any $k \geq N_\ell^{'}(\mathbf{x})$

$$\hat{p}_\ell^{v_k}(\mathbf{x}) > h_{u+1}.$$

Then, for any $k \geq \max(N_\ell(\mathbf{x}), N_\ell^{'}(\mathbf{x}))$, $\mathbf{x}$ is declared infeasible and a penalty sequence keeps receiving an appreciating factor which is greater than 1. This results in $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} \infty$ for any $\ell \in \Lambda_{I(\mathbf{x})}$. □

Proof of Theorem 7. $\underline{\text{(i) When } \ell \in \Lambda_{I(\mathbf{x})}}$: By Theorem 3, there exists $N_\ell(\mathbf{x}) \in \mathbb{Z}^+$ such that the penalty sequence for the infeasible constraint ℓ of $\mathbf{x}$ receives only $\alpha_\ell^{v_k}(\mathbf{x})$ after $N_\ell(\mathbf{x})$.

Now, let p_ℓ^I represent the infeasible probability for constraint ℓ that is greater than but the closest to 0.5.

By Lemma 3 and similar argument used in the proof of Theorem 5, there exists $N_\ell^{'}(\mathbf{x}) \in \mathbb{Z}^+$ such that

$$p_\ell(\mathbf{x}) - \epsilon_{\ell 0} \leq \hat{p}_\ell^{v_k}(\mathbf{x}) \leq p_\ell(\mathbf{x}) + \epsilon_{\ell 0}. \tag{24}$$

Set a constant $0 < \epsilon_{\ell 0} < \frac{p_\ell^I - 0.5}{4}$ and $\epsilon_{\ell 1} = \frac{p_\ell^I - 0.5}{4}$. Note that Assumption 3 implies $p_\ell(\mathbf{x}) \geq p_\ell^I > 0.5$ for any infeasible solution $\mathbf{x}$. Then the left hand side of (24) satisfies

$$p_\ell(\mathbf{x}) - \epsilon_{\ell 0} \geq p_\ell^I - \epsilon_{\ell 0} > p_\ell^I - \left(\frac{p_\ell^I - 0.5}{4}\right) = \frac{3}{4}p_\ell^I + \frac{0.5}{4}.$$

Thus, we get

$$\frac{3}{4}p_\ell^I + \frac{0.5}{4} < \hat{p}_\ell^{v_k}(\mathbf{x}) \leq p_\ell(\mathbf{x}) + \epsilon_{\ell 0}. \tag{25}$$

Recall

$$\gamma_\ell^k = \begin{cases} \min\left(h_{u+1} - 0.5, 0.5 - h_u, \min_{\mathbf{x}\in\{\mathbf{x}|\hat{p}_\ell^{v_k}(\mathbf{x})>0.5+\epsilon_{\ell 0},\ \mathbf{x}\in\mathcal{V}_k\}} \frac{\hat{p}_\ell^{v_k}(\mathbf{x})-0.5}{2}\right), \\ \qquad\qquad\qquad\qquad \text{if } \{\mathbf{x}|\hat{p}_\ell^{v_k}(\mathbf{x}) > 0.5 + \epsilon_{\ell 0},\ \mathbf{x} \in \mathcal{V}_k\} \neq \emptyset; \\ \\ \min\left(h_{u+1} - 0.5, 0.5 - h_u\right), \quad \text{otherwise.} \end{cases}$$

Consequently, when the set of infeasible solutions is not empty, as $k \to \infty$,

$$\begin{aligned} \gamma_\ell^k &\xrightarrow{a.s.} \min\left(h_{u+1} - 0.5, 0.5 - h_u, \min_{\mathbf{x}\in\{\mathbf{x}|\hat{p}_\ell^{v_k}(\mathbf{x})>0.5+\epsilon_{\ell 0},\ \mathbf{x}\in\Theta\}} \frac{\hat{p}_\ell^{v_k}(\mathbf{x}) - 0.5}{2}\right) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Assumption 2)}, \\ &\xrightarrow{a.s.} \min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right) \qquad \text{(by Lemma 3 and CMT)}. \end{aligned}$$

Also, there exists $N_\ell^{''}(\mathbf{x}) \in \mathbb{Z}^+$ such that for any $k \geq \max\left(N_\ell^{'}(\mathbf{x}), N_\ell^{''}(\mathbf{x})\right)$,

$$\gamma_\ell^k \leq \min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right) + \epsilon_{\ell 1} \quad \text{and} \quad \gamma_\ell^k \leq \frac{p_\ell^I - 0.5}{2} + \epsilon_{\ell 1}.$$

This implies that for any $k \geq \max\left(N_\ell^{'}(\mathbf{x}), N_\ell^{''}(\mathbf{x})\right)$

$$0.5 + \gamma_\ell^k \leq 0.5 + \tfrac{3(p_\ell^I - 0.5)}{4} = \tfrac{3}{4} p_\ell^I + \tfrac{0.5}{4}. \tag{26}$$

For $k \geq \max(N_\ell(\mathbf{x}), N_\ell^{'}(\mathbf{x}), N_\ell^{''}(\mathbf{x}))$, $\mathbf{x}$ is declared infeasible, a penalty sequence of $\mathbf{x}$ keeps receiving an appreciation factor, and (25) and (26) ensure that the appreciation factor is greater than 1. Thus, the penalty sequence diverges to infinity almost surely.

<u>(ii) When $\ell \in \Lambda_{A(\mathbf{x})}$:</u> For any $\ell \in \Lambda_{A(\mathbf{x})}$, $p_\ell = 0.5$ by Assumption 3. We need to consider two cases: when there exists solution $\mathbf{x}$ that violates constraint ℓ and thus p_ℓ^I exists and when all solutions are feasible with respect to constraint ℓ and thus p_ℓ^I does not exist.

If p_ℓ^I exists, there exists $N_\ell^{'''}(\mathbf{x}) \in \mathbb{Z}^+$ such that for $k \geq N_\ell^{'''}(\mathbf{x})$,

$$\frac{\min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right)}{2} \leq \gamma_\ell^k. \tag{27}$$

Also, there exists $N_\ell^{''''}(\mathbf{x}) \in \mathbb{Z}^+$ such that for any $k \geq N_\ell^{''''}(\mathbf{x})$,

$$0.5 - \frac{\epsilon_{\ell 2}}{2} \leq \hat{p}_\ell^{v_k}(\mathbf{x}) \leq 0.5 + \frac{\epsilon_{\ell 2}}{2}, \tag{28}$$

where $\epsilon_{\ell 2} = \frac{\min\left(h_{u+1}-0.5, 0.5-h_u, \frac{p_\ell^I - 0.5}{2}\right)}{2}$. From (27), for $k \geq N_\ell^{'''}$,

$$\begin{aligned} 0.5 - \gamma_\ell^k &\leq 0.5 - \frac{\min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right)}{2}, \quad \text{and} \\ 0.5 + \gamma_\ell^k &\geq 0.5 + \frac{\min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right)}{2}. \end{aligned}$$

Then from (27) and (28), for $k \geq \max(N_\ell^{'''}(\mathbf{x}), N_\ell^{''''}(\mathbf{x}))$,

$$0.5 - \gamma_\ell^k \leq 0.5 - \frac{\epsilon_{\ell 2}}{2} \leq \hat{p}_\ell^{v_k}(\mathbf{x}) \leq 0.5 + \frac{\epsilon_{\ell 2}}{2} \leq 0.5 + \gamma_\ell^k,$$

and $\alpha_\ell^{v_k}(\mathbf{x}) = w_a < 1$ which proves $\lambda_\ell^{v_k}(\mathbf{x}) \xrightarrow{a.s.} 0$ as $k \to \infty$ for any $\ell \in \Lambda_{A(\mathbf{x})}$.

If p_ℓ^I does not exist, $\min\left(h_{u+1} - 0.5, 0.5 - h_u, \frac{p_\ell^I - 0.5}{2}\right)$ needs to be replaced with $\min(0.5 - h_u, h_{u+1} - 0.5)$ and the results follow by similar argument. □

APPENDIX B

PROBLEM FORMULATION	N	the number of nodes in a river system
	I	the index set, $I = \{1, 2, ..., N\}$
	M	the number of monitoring devices ($M < N$)
	$\mathbf{x}$	a solution vector representing locations of M monitoring devices.
	$t_d(x_u)$	spill detection time at the monitoring location x_u
	S^S	spill starting time
	$d(x_u)$	if detected at x_u, $d(x_u) = t_d(x_u) - S^S$; otherwise, $d(x_u) = \infty$
	$t(\mathbf{x})$	the minimum elapsed detection time for $\mathbf{x}$ (i.e., $t(\mathbf{x}) = \min_{1 \le u \le M} d(x_u)$)
	$R(\mathbf{x})$	if a spill is detected, $R(\mathbf{x}) = 1$; otherwise, $R(\mathbf{x}) = 0$
	$t_i(\mathbf{x})$	an observed value of $t(\mathbf{x})$ from the ith simulation run
	$R_i(\mathbf{x})$	an observed value of $R(\mathbf{x})$ from the ith simulation run
PROCESS SIMULATION	C_{th}	detection threshold of a monitoring device
	L_i^S	spill location for the ith simulation run
	I_i^S	spill intensity for the ith simulation run
	S_i^S	spill starting time for the ith simulation run
	P_R^{im}	a rain pattern for the mth sub-catchment in the ith simulation run
	P_R^i	$(P_R^{i1}, \ldots, P_R^{iM^s})$ where M^S is the number of sub-catchment
DOvS & PFM	$v_k(\mathbf{x})$	the number of visits to $\mathbf{x}$ up to iteration k
	$n_{v_k}(\mathbf{x})$	the total number of observations obtained up to iteration k for $\mathbf{x}$
	$\overline{T}_k(\mathbf{x})$	conditional sample mean of $t_i(\mathbf{x})$ up to iteration k given that a spill is detected
	$\overline{R}_k(\mathbf{x})$	cumulative sample mean of $R_i(\mathbf{x})$ up to iteration k
NP+PFM	$\mathcal{R}_k$	the most promising region at search iteration k
	$\mathcal{R}_k(\ell)$	the ℓth subregion at search iteration k
	$\Theta \setminus \mathcal{R}_k$	the surrounding region at search iteration k
	$\mathcal{S}_k$	the set of solutions sampled at iteration k
	$\mathcal{V}_k$	the set of all solutions visited up to iteration k
	ω	the number of subregions
	τ_k	the total number of sampled solutions at iteration k
	$\tau_k(\ell)$	the number of sampled solutions at iteration k from subregion ℓ
	$\hat{\mathbf{x}}_k^*$	the current best solution obtained by NP+PFM (i.e., a solution with the smallest $Z_k(\mathbf{x})$ at iteration k)

REFERENCES

[1] Alrefaei, M. H. and Andradóttir, S., "A simulated annealing algorithm with constant temperature for discrete stochastic optimization," *Management Science*, vol. 45, no. 5, pp. 748–764, 1999.

[2] Andradóttir, S., "Accelerating the convergence of random search methods for discrete stochastic optimization," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 9, no. 4, pp. 349–380, 1999.

[3] Andradóttir, S., "An overview of simulation optimization via random search," *The Handbook of OR & MS: Simulation, S. G. Henderson and B. L. Nelson, eds.*, vol. 13, pp. 617–631, 2006.

[4] Andradóttir, S. and Kim, S.-H., "Fully sequential procedures for comparing constrained systems via simulation," *Naval Research Logistics*, vol. 59, pp. 403–421, 2010.

[5] Andradóttir, S. and Prudius, A. A., "Balanced explorative and exploitative search with estimation for simulation optimization," *INFORMS Journal on Computing*, vol. 21, pp. 193–208, 20090.

[6] Batur, D. and Kim, S.-H., "Finding feasible systems in the presence of constraints on multiple performance measures," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 20, no. 3, pp. 1–26, 2010.

[7] Billingsley, P., *Convergence of Probability Measures.* John Wiley & Sons, New York, 1978.

[8] Boesel, J., Nelson, B. L., and Kim, S.-H., "Using ranking and selection to "clean up after simulation optimization," *Operations Research*, vol. 51, no. 5, pp. 814–825, 2003.

[9] Chick, S. E., "Subjective probability and bayesian methodology," *Handbooks in operations research and management science, S. G. Henderson and B. L. Nelson, eds.*, vol. 13, pp. 225–257, 2006.

[10] Erdös, P. and M., K., "On the number of positive sums of independent random variables," *Bull. Amer. Math. Soc*, vol. 53, pp. 1011–1020, 1947.

[11] Fu, M. C., "Optimization for simulation: Theory vs. practice, journal = INFORMS Journal on Computing, year = 2002, volume = 14, number = 3, pages = 192–215,,"

[12] He, D., Lee, L. H. L., Chen, C. H., Fu, M. C. F., and Wasserkrug, S., "Simulation optimization using the cross-entropy method with optimal computing budget allocation," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 20, no. 4, pp. 1–22, 2010.

[13] Healey, M. C., *Advances in ranking and selection: variance estimation and constraints.* PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2010.

[14] Hong, L. J. and Nelson, B. L., "Discrete optimization via simulation using compass," *Operations Research*, vol. 54, no. 1, pp. 115–129, 2006.

[15] Hu, J., Fu, M. C., and Marcus, S. I., "A model reference adaptive search method for stochastic global optimization, journal = Communications in Information & Systems, year = 2008, volume = 8, number = 3, pages = 245–276,,"

[16] Hunter, S. R. and Pasupathy, R., "Optimal sampling laws for stochastically constrained simulation optimization," *INFORMS Journal on Computing*, p. DOI: 10.1287/ijoc.1120.0519, 2012.

[17] Kabirian, A. and Ólafsson, S., "Selection of the best with stochastic constraints," *Proceedings of the 2009 Winter Simulation Conference, M. D. Rossetti, R. R. Hill, B. Johansson and R. G. Ingalls, eds.*, IEEE, Piscataway, New Jersey. 2009.

[18] Kim, S.-H. and Nelson, B. L., "Selecting the best system," *Handbooks in operations research and management science, S. G. Henderson and B. L. Nelson, eds.*, vol. 13, pp. 501–534, 2006.

[19] Koenig, L. W. and Law, A. M., "A procedure for selecting a subset of size m containing the ℓ best of k independent normal populations, with applications to simulation," *Communications in Statistics-Simulation and Computation*, vol. 14, no. 3, pp. 719–734, 1985.

[20] Lee, L. H., Pujowidianto, N. A., Li, L. W., Chen, C. H., and Yap, C. M., "Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2940–2945, 2012.

[21] Li, J., Sava, A., and Xie, X., "Simulation-based discrete optimization of stochastic discrete event systems subject to non closed-form constraints," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2900–2904, 2009.

[22] Luo, Y. and Lim, E., "Simulation-based optimization over discrete sets with noisy constraints," *Proceedings of the 2011 Winter Simulation Conference, S. Jain, R. R. Creasey, H. Himmelspach, K. P. White, and M. Fu, eds*, pp. 4008 – 4020, IEEE, Piscataway, New Jersey. 2011.

[23] McCarthy, P., *Pulp mills, pulp and paper mills, paper mills in Georgia.* [online]. Atlanta, GA: Georgia Institute of Technology. Available from: http://www.cpbis.gatech.edu/data/mills-online?state=Georgia. [Accessed 28 January 2012].

[24] Nelson, B. L., "Optimization via simulation over discrete decision variables," *TutORials in Operations Research, J. J. Hasenbein, ed.*, pp. 193–207, 2010.

[25] Nelson, B. L., *Foundations and methods of stochastic simulation: A first course.* Springer-Verlag, forthcoming, 2013.

[26] Nocedal, J. and Wright, S. J., *Numerical optimization.* Springer Verlag, 1999.

[27] Ouyang, H. T., Yu, H., Lu, C. H., and Luo, Y. H., "Design optimization of river sampling network using genetic algorithms," *Journal of Water Resources Planning and Management*, vol. 134, pp. 83–87, 2008.

[28] Pichitlamken, J. and Nelson, B. L., "A combined procedure for optimization via simulation," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 13, no. 2, pp. 155–179, 2003.

[29] Pujowidianto, N. A., Lee, L. H., Chen, C. H., and Yap, C. M., "Optimal computing budget allocation for constrained optimization," *Proceedings of the 2009 Winter Simulation Conference, M. D. Rossetti, R. R. Hill, B. Johansson and R. G. Ingalls, eds.*, IEEE, Piscataway, New Jersey. 2009.

[30] Rossman, L. A., *Storm water management model user's manual, version 5.0.* National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.

[31] Sen, P. K. and M., S. J., *Large sample methods in statistics : an introduction with applications.* hapman & Hall, New York, 1993.

[32] Shi, L. and Chen, C. H., "A new algorithm for stochastic discrete resource allocation optimization," *Journal of Discrete Event Dynamic Systems: Theory and Applications*, vol. 10, pp. 271–294, 2010.

[33] Shi, L. and Ólafsson, S., "Nested partitions method for stochastic optimization," *Methodology and Computing in Applied Probability*, vol. 2, no. 3, pp. 271–291, 2000.

[34] Shi, L. and Ólafsson, S., *Nested partitions method, theory and applications*, vol. 109. Springer Verlag, 2009.

[35] Strobl, R. O. and Robillard, P. D., "Network design for water quality monitoring of surface freshwaters: A review," *Journal of Environmental Management*, vol. 87, no. 4, pp. 639–648, 2008.

[36] Szechtman, R. and Yücesan, E., "A new perspective on feasibility determination," *Proceedings of the 2008 Winter Simulation Conference, S. J. Mason, R. R. Holl, L. Mönch, O. Rose, T. Jefferson and J. W. Fowler, eds.*, IEEE, Piscataway, New Jersey. 2008.

[37] Telci, I. T. and Aral, M. M., "Contaminant source location identification in river networks using water quality monitoring systems for exposure analysis," *Water Quality, Exposure and Health*, vol. 2, no. 3, pp. 205–218, 2011.

[38] Telci, I. T., Nam, K., Guan, J., and Aral, M. M., "Real time optimal monitoring network design in river networks," *World Environmental and Water Resources Congress 2008, R. Babcock and R. Walton, eds.*, pp. 1–10, 2008.

[39] Telci, I. T., Nam, K., Guan, J., and Aral, M. M., "Optimal water quality monitoring network design for river systems," *Journal of Environmental Management*, vol. 90, no. 3–4, pp. 2987–2998, 2009.

[40] Xu, J., Nelson, B. L., and Hong, J. L., "An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems," *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 133–146, 2013.