Nonlinear Dynamics and Systems Theory, **10** (2) (2010) 175–188

Global Optimization Method for Continuous-Time Sensor Scheduling

S.F. Woon [1,2]*, V. Rehbock [1] and R.C. Loxton [1]

[1] *Department of Mathematics and Statistics, Curtin University of Technology, Bentley, Western Australia 6102, Australia;*
[2] *Physical Science, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia.*

Received: June 24, 2009; Revised: March 26, 2010

Abstract: We consider a situation in which several sensors are used to collect data for signal processing. Since operating multiple sensors simultaneously causes system interference, only one sensor can be active at any one time. The problem of scheduling the operation of the sensors to minimize signal estimation error is formulated as a discrete-valued optimal control problem. This problem cannot be solved using conventional optimization techniques. We instead transform it into an equivalent mixed discrete optimization problem. The transformed problem is then decomposed into a bi-level optimization problem, which is solved using a discrete filled function method in conjunction with a conventional optimal control algorithm. Numerical results show that our algorithm is robust, efficient, and reliable in attaining a near globally optimal solution.

Keywords: *sensor scheduling; time scaling transformation; discrete filled function; optimal control; mixed discrete optimization.*

Mathematics Subject Classification (2000): 37N35, 37N40, 90-04, 90-08, 68W01, 90C06, 90C27, 90C30.

* Corresponding author: mailto:woonsiewfang@yahoo.com

1 Introduction

Sensors are used in various applications, including military surveillance, ground mapping, tracking and recognition of targets, instrumentation, air traffic control, imaging, and robotics [1]. Information collected by the sensors is used to design activities that evolve over time in the underlying system [2]. For example, in a defense system, surveillance sensors are used to detect, identify, and localize targets, assess levels of threat, and deduce enemy intent [3]. In some applications, such as robotics, operating several sensors simultaneously causes interference in the system and thus affects the measurement accuracy [4]. Consequently, it is impossible to operate all of the sensors at once. Instead, we need to schedule the operation of sensors over a given time frame so that the signal estimation error is minimized. We assume in this paper that only one sensor may be active at any one time. The accuracy of the estimation obtained by the sensors increases with a decrease in measurement of noise in a stochastic environment. The work presented here was motivated by [5] and [6]. In [6], the optimal scheduling policy is obtained by solving a quasi-variational inequality. However, the complexity of the model in [6] makes it difficult to compute an optimal solution. On the other hand, [5] considers open-loop policies with switches from one sensor to another. This reference proposes a time scaling transformation, which aims to capture a large variety of possible switching sequences. The sensor scheduling problem, which is formulated as a discrete-valued optimal control problem, is first transformed into an optimal parameter selection problem, and then solved using existing optimal control software. The optimal control for the original problem is determined through a reverse transformation. However, this approach introduces a large number of artificial switches, many of which are not utilized in the optimal solution. As a consequence, the resulting optimization problem has many local minima. A study similar to that considered in [5] is performed in [7], where a combination of a branch and cut technique and a gradient-based method is applied to solve the continuous-time sensor scheduling problem.

We consider a general optimal sensor scheduling problem, which is similar to the one discussed in [5] and [7], and propose a transformation to convert it into an equivalent mixed discrete optimization problem, as discussed in Section 3. Then, we propose a novel global optimization algorithm in Section 4, which incorporates a discrete filled function method and a gradient-based method, to avoid local solutions and speed up the computation. To evaluate the effectiveness of our algorithm, we solve a numerical example from the literature and compare the results with those obtained from the methods in [5] and [7] in Section 5.

2 Problem Formulation

Consider the following system of linear stochastic differential equations on a given probability space $(\Omega, \mathcal{F}, \mathcal{P})$:

$$d\mathbf{x}(t) = A(t)\mathbf{x}(t)dt + B(t)dK(t), \qquad t \in [0, T],$$

with initial condition

$$\mathbf{x}(0) = \mathbf{x}_0.$$

Here, $\{\mathbf{x}(t),\ t \in [0, T]\}$ is a $\mathbb{R}^n$-valued state process representing a signal of interest. It is assumed to be square integrable. The initial state, $\mathbf{x}_0$, is a $\mathbb{R}^n$-valued Gaussian random vector on $(\Omega, \mathcal{F}, \mathcal{P})$ with mean $\bar{\mathbf{x}}_0$ and covariance matrix P_0. Furthermore, $A : [0, T] \to$

$\mathbb{R}^{n\times n}$ and $B : [0,T] \to \mathbb{R}^{n\times p}$ are continuous functions. The process $\{K(t),\ t \in [0,T]\}$ is a standard $\mathbb{R}^p$-valued Brownian motion on $(\Omega, \mathcal{F}, \mathcal{P})$ with mean zero and given covariance matrix $Q \in \mathbb{R}^{p\times p}$, where Q is symmetric and positive semi-definite.

Suppose that there are M sensors for detecting the state process. Only one of these sensors may be operated at any one time. A *sensor schedule* is a function $\phi : [0,T] \to \{1,\dots,M\}$ that returns the active sensor at time t. In other words, $\phi(t) = i$ means sensor i is active at time t. Let Φ be the set of all measurable sensor schedules and let $\mathbf{y}$ be the observation process associated with the scheduling policy ϕ. For any $\phi \in \Phi$, we have the following output equation:

$$d\mathbf{y}(t) = \sum_{i=1}^{M} \chi_{\{t:\phi(t)=i\}}(t)\big\{C_i(t)x(t)dt + D_i(t)dW_i(t)\big\}, \qquad t \in [0,T],$$

and

$$\mathbf{y}(0) = \mathbf{0},$$

where, for each $\mathcal{I} \subset [0,T]$,

$$\chi_{\mathcal{I}}(t) = \begin{cases} 1, & t \in \mathcal{I}, \\ 0, & \text{otherwise,} \end{cases}$$

and $\{W_i(t),\ t \in [0,T]\}$ is a standard $\mathbb{R}^m$-valued Brownian motion with mean zero and covariance matrix $R \in \mathbb{R}^{m\times m}$, where R is symmetric and positive definite, $C_i : [0,T] \to \mathbb{R}^{m\times n}$ and $D_i : [0,T] \to \mathbb{R}^{m\times m}$ are continuous functions.

Each sensor makes an observation of the state process that is contaminated by noise. The history of such observation processes is denoted by $\{\mathbf{y}(s), 0 \le s \le t\}$. The data collected from the M sensors are used to estimate the state $\mathbf{x}$ at time t. The best estimate of $\mathbf{x}(t)$ is known as $\hat{\mathbf{x}}(t)$. Since $\mathbf{y}$ is corrupted by noise, the history observed is uncertain. Let the history of such a process be denoted by the smallest σ-algebra, $\mathcal{F}_t^{\mathbf{y}} = \sigma\{\mathbf{y}(s), 0 \le s \le t\}$. Hence, the optimal mean-square estimate of the state given $\mathcal{F}_t^{\mathbf{y}}$ is $\hat{\mathbf{x}}(t)$, and the associated error covariance is $P(t)$. Then, for a given $\phi \in \Phi$, the optimal $\hat{\mathbf{x}}(t)$ is given by the following theorem. The proof of this theorem may be found in [8].

Theorem 2.1 *For each sensor schedule $\phi \in \Phi$, the optimal mean-square estimate of the state $\hat{\mathbf{x}}(t)$ is the unique solution of the following stochastic differential equation:*

$$\begin{aligned} d\hat{\mathbf{x}}(t) \quad = \quad & \left[A(t) - P(t)\sum_{i=1}^{M} \chi_{\{t:\phi(t)=i\}}(t)C_i^\top(t)\bar{R}_i^{-1}(t)C_i(t)\right]\hat{\mathbf{x}}(t)dt \\ & + \left[P(t)\sum_{i=1}^{M} \chi_{\{t:\phi(t)=i\}}(t)C_i^\top(t)\bar{R}_i^{-1}(t)\right]d\mathbf{y}(t), \quad t \in [0,T], \end{aligned} \tag{1}$$

and

$$\hat{\mathbf{x}}(0) = \bar{\mathbf{x}}_0, \tag{2}$$

where

$$\bar{R}_i^{-1}(t) = \big[D_i(t)R_i(t)D_i^\top(t)\big]^{-1}, \tag{3}$$

and the error covariance matrix $P : [0,T] \to \mathbb{R}^{n\times n}$ is the unique solution of the matrix Riccati differential equation

$$\dot{P}(t) = A(t)P(t)+P(t)A^\top(t)+B(t)QB^\top(t)-P(t)\sum_{i=1}^{M}\chi_{\{t:\phi(t)=i\}}(t)C_i^\top(t)\bar{R}_i^{-1}(t)C_i(t)P(t) \tag{4}$$

with initial condition

$$P(0) = P_0. \tag{5}$$

Clearly, the solution of (4)-(5) depends on the sensor schedule that is chosen. Let $P(\cdot|\phi)$ be the solution corresponding to $\phi \in \Phi$. We formulate the following sensor scheduling problem.

Problem (P). Choose $\phi \in \Phi$ to minimize

$$g_0(\phi) = \alpha\text{trace}\{P(T|\phi)\} + \int_0^T \text{trace}\{P(t|\phi)\}dt, \tag{6}$$

subject to (4) and (5), where α is a non-negative constant.

The objective function (6) is designed to minimize the estimation error during the operation of the system. Note that Problem (P) is a discrete-valued optimal control problem. The main challenge in solving Problem (P) is that the control ϕ is constrained to take values in the discrete set $\{1,\dots,M\}$. Each sensor schedule is completely determined by specifying the values in $\{1,\dots,M\}$ that it assumes and the times when it switches from one value in $\{1,\dots,M\}$ to another. Clearly, only a finite number of switches are able to be implemented in practice, and hence ϕ is a piecewise constant function with a finite number of switches. In other words, to solve Problem (P), we need to determine both the optimal switching sequence and the optimal switching times. Thus, we transform Problem (P) into an equivalent and solvable form in the next section.

3 Problem Transformation

Recall that only one sensor is active at each time and that only a finite number of switches are allowed. Suppose that we allow a sensor schedule ϕ to switch N times during the time horizon. Let $V = \{\mathbf{v} = [v_1,\dots,v_{N+1}]^\top : \ v_i \in \{1,\dots,M\}\}$ be the set of all possible switching sequence vectors. Let $\boldsymbol{\sigma} = [\sigma_1,\dots,\sigma_{N+1}]^\top$, where $\sigma_i \geq 0, \ i = 1,\dots,N+1$, denote the duration for which the corresponding sensor v_i in the sequence is active. Clearly,

$$\sum_{i=1}^{N+1}\sigma_i = T.$$

Let Σ denote the set of all such $\boldsymbol{\sigma}$. Note that under the assumption of a finite number of switches, N, any $\phi \in \Phi$ is completely determined by an element $(\mathbf{v},\boldsymbol{\sigma}) \in V \times \Sigma$, where

$$\phi(t) = v_i, \qquad t \in \left[\sum_{j=1}^{i-1}\sigma_j, \ \sum_{j=1}^{i}\sigma_j\right], \qquad i = 1,\dots,N+1.$$

We introduce a new time variable $\tau \in [0, N+1]$ and consider the fixed partition $\{0, 1, \ldots, N+1\}$. The original time horizon $[0, T]$ is transformed into the new time horizon $[0, N+1]$ as follows:

$$\dot{t}(\tau) = \sigma_i, \qquad \tau \in [i-1, i), \qquad i = 1, \ldots, N+1, \tag{7}$$

with the boundary conditions

$$t(0) = 0 \tag{8}$$

and

$$t(N+1) = T. \tag{9}$$

The original dynamics (4)-(5) are transformed into

$$\begin{aligned} \dot{\tilde{P}}(\tau) \;\;&=\;\; \sigma_i \Big[A(\tau)\tilde{P}(\tau) + \tilde{P}(\tau)A^\top(\tau) + B(\tau)QB^\top(\tau) \\ &\quad - \tilde{P}(\tau)C_{v_i}^\top(\tau)\bar{R}_{v_i}^{-1}(\tau)C_{v_i}(\tau)\tilde{P}(\tau)\Big],\ \tau \in [i-1, i),\ i = 1, \ldots, N+1, \end{aligned} \tag{10}$$

and

$$\tilde{P}(0) = P_0. \tag{11}$$

Hence, the transformed problem is stated formally below. Let $\tilde{P}(\cdot|\mathbf{v}, \boldsymbol{\sigma})$ be the solution of (10)-(11) corresponding to $(\mathbf{v}, \boldsymbol{\sigma}) \in V \times \Sigma$.

Problem (R). Choose $\mathbf{v} \in V$ and $\boldsymbol{\sigma} \in \Sigma$ to minimize

$$g_0(\mathbf{v}, \boldsymbol{\sigma}) = \alpha \text{trace}\{\tilde{P}(N+1|\mathbf{v}, \boldsymbol{\sigma})\} + \sum_{i=1}^{N+1} \int_{i-1}^{i} \text{trace}\{\tilde{P}(\tau|\mathbf{v}, \boldsymbol{\sigma})\}\sigma_i \, d\tau, \tag{12}$$

subject to (7)-(9) and the dynamics (10)-(11), where α is a non-negative constant.

Problem (R), an equivalent problem to Problem (P), is a mixed discrete optimization problem with the discrete variable $\mathbf{v}$ representing the switching sequence and the continuous variable $\boldsymbol{\sigma}$ representing the time length of each mode. We propose to solve Problem (R) by first decomposing it into two levels. Note that for a fixed $\mathbf{v} \in V$, Problem (R) reduces to the following problem.

Problem ($\mathbf{R_1}$). Given $\mathbf{v} \in V$, find a $\boldsymbol{\sigma} \in \Sigma$ to minimize

$$g_0(\boldsymbol{\sigma}|\mathbf{v}) = \alpha \text{trace}\{\tilde{P}(N+1|\boldsymbol{\sigma}, \mathbf{v})\} + \sum_{i=1}^{N+1} \int_{i-1}^{i} \text{trace}\{\tilde{P}(\tau|\boldsymbol{\sigma}, \mathbf{v})\}\sigma_i \, d\tau, \tag{13}$$

subject to (7)-(9) and dynamics (10)-(11), where α is a non-negative constant.
Problem (R_1) is a standard optimal parameter selection problem in a canonical form suitable for the application of a standard algorithm based on the control parameterization concept. For each given $\mathbf{v}$, the optimal value of g_0 in (13) can be determined using an optimal control software, such as MISER3.3 [9], since the switching sequence is fixed. Note that in MISER3.3, the optimal parameter selection problem is solved using a sequential quadratic programming algorithm. The second problem in the proposed

decomposition is defined as follows.

Problem ($\mathbf{R_2}$). Choose $\mathbf{v} \in V$ to minimize the objective function

$$J(\mathbf{v}), \tag{14}$$

where

$$J(\mathbf{v}) = \min_{\boldsymbol{\sigma} \in \Sigma} \quad g_0(\boldsymbol{\sigma}|\mathbf{v}).$$

Note that Problem (R_2) is a purely discrete optimization problem, but computing the value of $J(\mathbf{v})$ requires solving the corresponding Problem (R_1). Hence, Problem (R_1) is a subproblem of Problem (R_2). To obtain a near globally optimal solution for Problem (R), we propose a combined algorithm where Problem (R_2) will be solved using a discrete filled function method and, at each iteration, Problem (R_1) is solved using MISER3.3. For our numerical computations, we have been able to incorporate the discrete filled function method within the MISER3.3 software. The details of the discrete filled function approach are discussed in the next section.

Remark 3.1 Note that the early time scale transformation proposed in [5] introduces a large number of artificial switching instants, typically $N \times M$, most of which are not used in the final optimal solution. As a result, the transformed problem yields many local minima, many of which have high objective values. Our method avoids this difficulty because only N switches are needed.

4 Discrete Filled Function Method

The filled function approach is a global optimization method which was initiated by Ge in the late 1980s [10, 11] to solve continuous global optimization problems. Zhu [12] appears to be the first researcher to adapt the continuous filled function approach in solving discrete optimization problems. However, the filled function proposed by Zhu contains an exponential term, making it difficult to determine an improved point [13] in practice. Since then, various discrete filled functions with improved theoretical properties have been proposed in [13–17] to enhance computational efficiency.

In this paper, we employ a discrete filled function method, which was recently developed in [13], as a part of our proposed algorithm. The basic idea of this method is as follows. We choose an initial sequence and then perform a local search (see Algorithm 4.1 below) to find an initial local minimizer. Then, we construct an auxiliary function, called a filled function, at this local minimizer. By minimizing the filled function, either an improved local minimizer is found or one of the vertices is reached. This process is repeated until no improved local minimizer of the corresponding filled function can be found. The final local minimizer is then taken as an approximation of the global minimizer.

Definition 4.1 For any $\mathbf{v} \in V$, the *neighbourhood* of $\mathbf{v}$ is defined by $N(\mathbf{v}) = \{\mathbf{w} \in V|\ \mathbf{w} = \mathbf{v} \pm \mathbf{e}_i : i = 1, 2, \ldots, N+1\}$. Here, $\mathbf{e}_i$ denotes the i-th standard unit basis vector of $\mathbb{R}^{N+1}$: its i-th component is equal to one and its other components are equal to zero. The set of all feasible directions at $\mathbf{v} \in V$ is defined by $\mathcal{D}(\mathbf{v}) = \{\mathbf{d} \in \mathbb{R}^{N+1} : \mathbf{v} + \mathbf{d} \in N(\mathbf{v})\} \subset \{\pm\mathbf{e}_i,\ i = 1, \ldots, N+1\}$.

Definition 4.2 The sequence $\mathbf{v}^* \in V$ is a *local minimizer* of J if $J(\mathbf{v}^*) \leq J(\mathbf{v})$ for all $\mathbf{v} \in N(\mathbf{v}^*)$. If $J(\mathbf{v}^*) < J(\mathbf{v})$ for all $\mathbf{v} \in N(\mathbf{v}^*) \setminus \{\mathbf{v}^*\}$, then $\mathbf{v}^*$ is a *strict local minimizer* of J. The sequence $\mathbf{v}^*$ is a *global minimizer* of J if $J(\mathbf{v}^*) \leq J(\mathbf{v})$ for all $\mathbf{v} \in V$. If $J(\mathbf{v}^*) < J(\mathbf{v})$ for all $\mathbf{v} \in V \setminus \{\mathbf{v}^*\}$, then $\mathbf{v}^*$ is a *strict global minimizer* of J.

Definition 4.3 $\mathbf{v}$ is a vertex of V if for each $\mathbf{d} \in \mathcal{D}(\mathbf{v})$, $\mathbf{v} + \mathbf{d} \in V$ and $\mathbf{v} - \mathbf{d} \notin V$. Let $\tilde{V}$ denote the set of vertices of V.

Algorithm 4.1 Discrete Steepest Descent Method

1. Choose an initial switching sequence $\mathbf{v} \in V$.
2. If $\mathbf{v}$ is a local minimizer of J, then stop. Otherwise, find a discrete steepest descent direction $\mathbf{d}^* \in \mathcal{D}(\mathbf{v})$ of J.
3. Let $\mathbf{v} = \mathbf{v} + \mathbf{d}^*$. Go to Step 2.

Based on Definitions 4.1-4.3, we call a function $G_{\mathbf{v}^*} : V \mapsto \mathbb{R}$ a *discrete filled function* of J at $\mathbf{v}^*$ if it satisfies the following conditions:
(a) $\mathbf{v}^*$ is a strict local maximizer of $G_{\mathbf{v}^*}$;
(b) Let $\hat{V}(\mathbf{v}^*) = \{\mathbf{v} \in V : \mathbf{v} \neq \mathbf{v}^*, J(\mathbf{v}) \geq J(\mathbf{v}^*)\}$. $G_{\mathbf{v}^*}$ has no local minimizer in the set $\hat{V}(\mathbf{v}^*) \setminus \tilde{V}$;
(c) $\mathbf{v}^{**} \in V \setminus \tilde{V}$ is a local minimizer of J if and only if $\mathbf{v}^{**}$ is a local minimizer of $G_{\mathbf{v}^*}$.

Define

$$G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}) = A_\mu(J(\mathbf{v}) - J(\mathbf{v}^*)) - \rho \parallel \mathbf{v} - \mathbf{v}^* \parallel, \tag{15}$$

where

$$A_\mu(y) = y \cdot \mu \left[(1-c) \left(\frac{1 - c\mu}{\mu - c\mu} \right)^{-y/\omega} + c \right],$$

$\omega > 0$ is a sufficiently small number, and $0 < c \leq 1$ is a constant. The function $G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v})$ is a discrete filled function when certain conditions on the parameters μ and ρ are satisfied. Hence, it has properties (a)-(c) when those conditions on μ and ρ are met. Note that the discrete filled function is constructed based on the following theorems found in [13]. A detailed convergence analysis for this method has also been given in [13].

Definition 4.4 Let $\mathcal{K}$ be a constant satisfying

$$1 \leq \max_{\substack{\mathbf{v}_1, \mathbf{v}_2 \in V \\ \mathbf{v}_1 \neq \mathbf{v}_2}} \parallel \mathbf{v}_1 - \mathbf{v}_2 \parallel \leq \mathcal{K} < \infty,$$

where $\parallel \cdot \parallel$ is the Euclidean norm. Let $0 < \mathcal{L} < \infty$ be the Lipschitz constant such that $|J(\mathbf{v}_1) - J(\mathbf{v}_2)| \leq \mathcal{L} \parallel \mathbf{v}_1 - \mathbf{v}_2 \parallel$, for any distinct $\mathbf{v}_1, \mathbf{v}_2 \in V$.

Theorem 4.1 *If $\rho > 0$ and $0 < \mu < min\{1, \frac{\rho}{\mathcal{L}}\}$, then $\mathbf{v}^*$ is a strict local maximizer of $G_{\mu,\rho,\mathbf{v}^*}$. If $\mathbf{v}^*$ is a global minimizer of J, then $G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}) < 0$ for all $\mathbf{v} \in V \setminus \{\mathbf{v}^*\}$.*

Theorem 4.2 *Let $\mathbf{v}^{**}$ be a strict local minimizer of J with $J(\mathbf{v}^{**}) < J(\mathbf{v}^*)$. If $\rho > 0$ is sufficiently small and $0 < \mu < 1$, then $\mathbf{v}^{**}$ is a strict local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$.*

Theorem 4.3 *Let $\acute{\mathbf{v}}$ be a strict local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$ and let $\bar{\mathbf{d}} \in \mathcal{D}(\acute{\mathbf{v}})$ be a feasible direction at $\acute{\mathbf{v}}$ such that $\parallel \acute{\mathbf{v}} + \bar{\mathbf{d}} - \mathbf{v}^* \parallel > \parallel \acute{\mathbf{v}} - \mathbf{v}^* \parallel$. If $\rho > 0$ is sufficiently small and $0 < \mu < min\{1, \frac{\rho}{2\mathcal{K}^2\mathcal{L}}\}$, then $\acute{\mathbf{v}}$ is a local minimizer of J.*

Corollary 4.1 *Assume that every local minimizer of J is strict. Suppose that $\rho > 0$ is sufficiently small and $0 < \mu < min\{1, \frac{\rho}{2\mathcal{K}^2\mathcal{L}}\}$. Then, $\mathbf{v}^{**} \in V \setminus \tilde{V}$ is a local minimizer of J with $J(\mathbf{v}^{**}) < J(\mathbf{v}^*)$ if and only if $\mathbf{v}^{**}$ is a local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$.*

Interested readers are referred to [13] for proofs of Theorem 4.1-4.3. Clearly, from Corollary 4.1, $G_{\mu,\rho,\mathbf{v}^*}$ must satisfy the condition (c) of the discrete filled function definition if every local minimizer of J is strict under certain conditions on μ and ρ. If the local minimizer of the discrete filled function $G_{\mu,\rho,\mathbf{v}^*}$ found is an improved point, it is also a local minimizer of the original function J. Based on the theoretical framework described above, a discrete filled function algorithm for global optimization can be stated as follows.

Algorithm 4.2 Discrete Filled Function Method

1. Initialize $\mathbf{v}_0 \in V$, ρ_0, μ_0, $\rho_L > 0$, $0 < \hat{\rho} < 1$, and $0 < \hat{\mu} < 1$.
   Let $\rho := \rho_0$ and $\mu := \mu_0$.
   Choose an initial sequence $\mathbf{v}_0 \in V$.

2. Starting from $\mathbf{v}_0$, minimize $J(\mathbf{v})$ using Algorithm 4.1 to obtain a local minimizer $\mathbf{v}^*$ of J.

3. (a) List the neighbouring sequences of $\mathbf{v}^*$ as $N(\mathbf{v}^*) = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_q\}$. Set $\ell := 1$.
   (b) Set the current switching sequence, $\mathbf{v}_c := \mathbf{w}_\ell$.

4. (a) If there exists a direction $\mathbf{d} \in \mathcal{D}(\mathbf{v}_c)$ such that $J(\mathbf{v}_c + \mathbf{d}) < J(\mathbf{v}^*)$, then set $\mathbf{v}_0 := \mathbf{v}_c + \mathbf{d}$ and go to Step 2. Otherwise, go to (b) below.
   (b) Let $\mathcal{D}_1 = \{\mathbf{d} \in \mathcal{D}(\mathbf{v}_c) : \; J(\mathbf{v}_c + \mathbf{d}) < J(\mathbf{v}_c) \text{ and } G_{\mu,\rho,\mathbf{v}}(\mathbf{v}_c + \mathbf{d}) < G_{\mu,\rho,\mathbf{v}}(\mathbf{v}_c)\}$.
   If $\mathcal{D}_1 \neq \emptyset$, set $\mathbf{d}^* := \arg\min_{\mathbf{d} \in \mathcal{D}(\mathbf{v}_c)} \{J(\mathbf{v}_c + \mathbf{d}) + G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}_c + \mathbf{d})\}$.
   Then, set $\mathbf{v}_c := \mathbf{v}_c + \mathbf{d}^*$ and go to Step 4(a). Otherwise, go to (c) below.
   (c) Let $\mathcal{D}_2 = \{\mathbf{d} \in \mathcal{D}(\mathbf{v}_c) : \; G_{\mu,\rho,\mathbf{v}}(\mathbf{v}_c + \mathbf{d}) < G_{\mu,\rho,\mathbf{v}}(\mathbf{v}_c)\}$.
   If $\mathcal{D}_2 \neq \emptyset$, set $\mathbf{d}^* := \arg\min_{\mathbf{d} \in \mathcal{D}(\mathbf{v}_c)} \{G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}_c + \mathbf{d})\}$.
   Then, set $\mathbf{v}_c := \mathbf{v}_c + \mathbf{d}^*$ and go to Step 4(a). Otherwise, go to Step 5.

5. Let $\acute{\mathbf{v}}$ be the obtained local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$.
   (a) If $\acute{\mathbf{v}} \in \tilde{V}$, set $\ell := \ell + 1$. If $\ell > q$, go to Step 6. Otherwise, go to Step 3(b).
   (b) If $\acute{\mathbf{v}} \notin \tilde{V}$, reduce μ by setting $\mu := \hat{\mu}\mu$ and go to Step 4(b).

6. Reduce ρ by setting $\rho := \hat{\rho}\rho$. If $\rho < \rho_L$, terminate the algorithm. The current $\mathbf{v}^*$ is taken as a global minimizer of the problem. Otherwise, set $\ell := 1$ and go to Step 3(b).

The mechanism behind this algorithm can be illustrated as follows. Firstly, the parameters of the discrete filled function $G_{\mu,\rho,\mathbf{v}^*}$ in (15) are initialized to suitable values. These parameters will be reduced gradually in Steps 5 and 6, to ensure that $G_{\mu,\rho,\mathbf{v}^*}$ eventually satisfies properties (a)-(c). The reduction factor of each parameter is also specified at Step 1.

Secondly, we choose an initial sequence $\mathbf{v}_0$ in the feasible region and minimize the original function J. Recall that the value of J is computed using MISER3.3 according to the discussion in the previous section. The objective function value at each sequence in the neighbourhood of $\mathbf{v}_0$ is calculated. The search direction leading to the most improved objective function value in this neighbourhood is chosen according to Algorithm 4.1. The process is repeated until a local minimizer of J, namely $\mathbf{v}^*$, is found. Next, we identify the neighbourhood of $\mathbf{v}^*$ in Step 3. One of the neighbouring points of $\mathbf{v}^*$, denoted by $\mathbf{v}_c$, is set to be an initial point to minimize the discrete filled function $G_{\mu,\rho,\mathbf{v}^*}$ in the following step. Note that $\mathbf{v}^*$ is a local maximizer of $G_{\mu,\rho,\mathbf{v}^*}$ here.

In Step 4, we first check to see if there exists a neighbouring sequence of $\mathbf{v}_c$ that is an improvement over the current minimizer. If such a sequence can be found, then we use it as a starting point to minimize the function J using Algorithm 4.1. Otherwise, if we can find a direction that results in an improvement of both J and G compared with the values at $\mathbf{v}_c$, then we choose the direction which gives the greatest such improvement. If such a direction does not exist, then find a steepest descent direction such that $G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}_c + \mathbf{d}^*) < G_{\mu,\rho,\mathbf{v}^*}(\mathbf{v}_c)$. If none of these directions exists, then $\mathbf{v}_c$ must be a local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$, so we go to Step 5.

If the local minimizer of $G_{\mu,\rho,\mathbf{v}^*}$ is found to be a vertex of the feasible region, choose the next point in $N(\mathbf{v}^*)$ as a starting point to minimize $G_{\mu,\rho,\mathbf{v}^*}$ in Step 5(a). Note that the minimizer of $G_{\mu,\rho,\mathbf{v}^*}$ should be either an improved point or a vertex. Thus, μ is adjusted suitably to satisfy this criteria in Step 5(b).

If no improved sequence is found with the minimization process starting from all neighbouring sequences ending up at the vertices, we reduce ρ, reset $\ell = 1$, and minimize $G_{\mu,\rho,\mathbf{v}^*}$ again with the new value of ρ. The algorithm is repeated until the termination criteria is reached, where ρ reaches its lower bound, ρ_L. In other words, we have minimized the discrete filled function from every search direction from $\mathbf{v}^*$ and failed to find an improved point, even the parameters are small. We repeat Algorithm 4.2 twice, reducing the value of ρ each time to confirm that no better solution can be found. Thus, the final local minimizer $\mathbf{v}^*$ found is taken to be the global solution of J.

To increase the efficiency, we construct a look-up table to store each value of the objective function J computed so far. Thus, we avoid repeated application of the subproblems solution algorithm at the same point. This is essential because computing $J(\mathbf{v})$ involves solving a complex optimal control problem, which takes considerable computational time.

Remark 4.1 Note that a sequential quadratic programming method is employed within MISER3.3 to solve the subproblem (R_1). This is a local search method and thus cannot guarantee the global optimality for the solution of the subproblem. In other words, although we aim to solve the upper level problem globally, the lower level problem may only yield a locally optimal solution. Therefore, we consider our approach to be a heuristic global optimization method with no implied guarantee of finding the overall global optimum. Nevertheless, numerical results demonstrate that good quality solutions can be determined effectively compared with other methods in the literature, such as [5] and [7].

5 Illustrative Example

Consider a sensor scheduling problem with six sensors and seven switches as discussed in [7]. Let $N = 7,\ M = 6,\ n = 2,\ m = 1,\ p = 2,\ T = 8,\ \alpha = 0,\ c = 0.5,\ \mu_0 = 0.1,\ \rho_0 = 0.1,\ \omega = 1,\ \rho_L = 0.001,\ \hat{\rho} = 0.1,\ \hat{\mu} = 0.1$ and consider the following dynamics:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_1(t) \end{bmatrix} = \begin{bmatrix} 0.5 & 1.0 \\ 1.0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} K(t), \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where

$$P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_1(t) = \begin{bmatrix} 1 + 1.2\sin(2t) & 0 \\ 1 + 1.2\sin(2t) & 0 \end{bmatrix}, \qquad D_1(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_1(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_2(t) = \begin{bmatrix} 1 + 0.5\cos(2t) & 1 + 0.5\cos(2t) \\ 0 & 0 \end{bmatrix}, \qquad D_2(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_2(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_3(t) = \begin{bmatrix} 1 + 0.5\sin(2t) & 0 \\ 0 & 1 + 0.5\cos(2t) \end{bmatrix}, \qquad D_3(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_3(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_4(t) = \begin{bmatrix} 0 & 1 + 0.5\cos(2t) \\ 1 + 0.5\sin(2t) & 0 \end{bmatrix}, \qquad D_4(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_4(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_5(t) = \begin{bmatrix} 0 & 0 \\ 1 + 0.5\cos(2t) & 1 + 0.5\sin(2t) \end{bmatrix}, \qquad D_5(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_5(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_6(t) = \begin{bmatrix} 0 & 1 + 1.8\sin(2t) \\ 0 & 1 + 1.8\cos(2t) \end{bmatrix}, \qquad D_6(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_6(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

For the ease of computation, we are able to embed the filled function algorithm into the MISER3.3 program. The algorithm is terminated when $\mu = 1 \times 10^{-41}$ and $\rho = 1 \times 10^{-3}$, at which stage the best local minimizer found cannot be improved. The computation is performed using the modified version of MISER3.3 on a Windows-based PC, with a CPU speed of 2.4GHz and 2GB RAM. We solve Problem (R), which has a total number of 1,679,616 potential switching sequences, using $\mathbf{v_0} = [6,5,2,6,5,2,6,1]^\top$ as the initial sequence and $\boldsymbol{\sigma}_0 = [1,1,1,1,1,1,1,1]^\top$ as the initial guess for $\boldsymbol{\sigma}$. Note that P_0 is initialized as a 2×2 identity matrix. Relevant results obtained are summarized in Table 1. The entries in the $\mathbf{v}^*$ column indicate the optimal solutions for the local searches. From Table 1, $\boldsymbol{\sigma}^* = [0.23501973, 0, 0, 7.7649803, 0, 0, 0, 0]^\top$ for the assumed global minimum indicates that sensors 2, 3, 4, and 5 are not used in the final optimal solution during the tenth iteration. Hence, only two out of six sensors are turned on. The assumed global optimal switching sequence is to turn on sensor 1, followed by sensor 6, with the objective function 14.33176. The number of original function evaluations and filled function evaluations are 5293 and 8517, respectively. This represents 0.32% of the total number of potential sequences. Note that the objective function evaluations do not include those that were obtained from the look-up table.

We tested the problem with five different initial sequences. These are $[1,2,3,4,5,6,1,2]^\top$, $[6,5,4,3,2,1,6,5]^\top$, $[1,6,3,2,4,5,3,1]^\top$, $[1,6,1,6,1,6,1,6]^\top$, and $[6,6,1,2,5,4,2,1]^\top$ using the same $P_\mathbf{0}$ and $\boldsymbol{\sigma}_\mathbf{0}$ as in the first computation. As many as fifty local minima are found during the searches from the various initial sequences. Starting at each initial sequence, the algorithm successfully identified the same assumed discrete global minimum sequence of Problem (R) observed in the first experiment, that is, sensor 1 is followed by sensor 6, with the cost function value $J = 14.33176$. Again, computational results show that only up to 0.32% of the total number of potential sequences are evaluated. The optimal operating schedule for the control and states are depicted in Figure 1. In addition, several different choices of P_0 are tested in our experimentation with various initial switching sequences. The optimal operating schemes for $P_0 = \mathbf{0}$, $P_0 = 6I$, $P_0 = 10I$ are illustrated by Figures 2, 3, and 4, respectively. From

Table 1: Numerical Results for $P_0 = I$

$\mathbf{v}^*$	$\boldsymbol{\sigma}^*$	J
$[1,6,1,1,6,1,6,6]^\top$	$[0.24035917, 0, 0, 0, 0, 0, 7.7525870, 0.0070538593]^\top$	14.649680367412879
$[6,1,6,6,1,6,1,1]^\top$	$[0.17566501, 0.18470974, 0, 7.6396253, 0, 0, 0, 0]^\top$	14.504334985710470
$[1,6,1,6,6,1,1,6]^\top$	$[0.23511799, 0, 0, 7.7648820, 0, 0, 0, 0]^\top$	14.331763146735220
$[1,6,2,6,6,2,2,6]^\top$	$[0.23501894, 0, 0, 7.7649811, 0, 0, 0, 0]^\top$	14.331763102479558
$[1,6,6,6,6,3,3,5]^\top$	$[0.23502083, 0, 0, 7.7649792, 0, 0, 0, 0]^\top$	14.331763102474610
$[1,6,6,6,6,6,5,5]^\top$	$[0.23502039, 0, 0, 7.7649796, 0, 0, 0, 0]^\top$	14.331763102473506
$[1,6,6,6,1,5,6,2]^\top$	$[0.23501994, 0, 0, 7.7649801, 0, 0, 0, 0]^\top$	14.331763102471598
$[1,1,6,6,6,6,5,1]^\top$	$[0.23501894, 0, 0, 7.7649811, 0, 0, 0, 0]^\top$	14.331763102445281
$[1,1,6,6,5,6,6,2]^\top$	$[0.23501979, 0, 0, 7.7649802, 0, 0, 0, 0]^\top$	14.331763102440952
$[1,1,6,6,6,5,2,1]^\top$	$[0.23501973, 0, 0, 7.7649803, 0, 0, 0, 0]^\top$	14.331763102437696

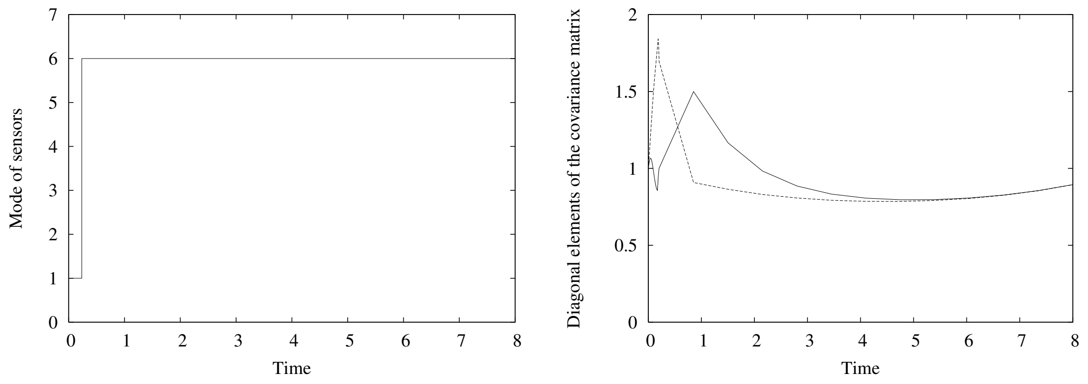

Figure 1: Optimal Sensor Operating Scheme with $P_0 = I$.

these graphs, only the first and sixth sensors are ever used, while the other four are not utilized in any optimal solution.

We also compare the solutions obtained here with those obtained from the other methods proposed in [5] and [7]. These results are summarized in Table 2. Note that the error estimation that we sought is lower than 19.6553, the optimal solution reported in [7], which was obtained using a combination of a branch and bound technique with a gradient-based method. To the best of our knowledge, $P_0 = \mathbf{0}$ is used in [7]. Note that non-zero choices of P_0 lead to even higher objective values when used in conjunction with the solution in [7].

6 Conclusions

A sensor scheduling problem is considered in this paper. It was formulated as a discrete-valued optimal control problem and then transformed into a mixed discrete optimization problem. Then, it was decomposed into a bi-level problem. A new heuristic approach,

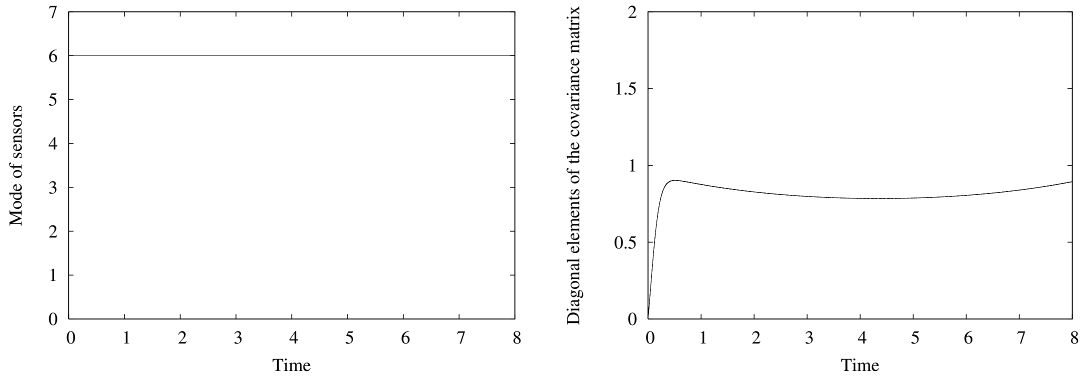

Figure 2: Optimal Sensor Operating Scheme with $P_0 = \mathbf{0}$.

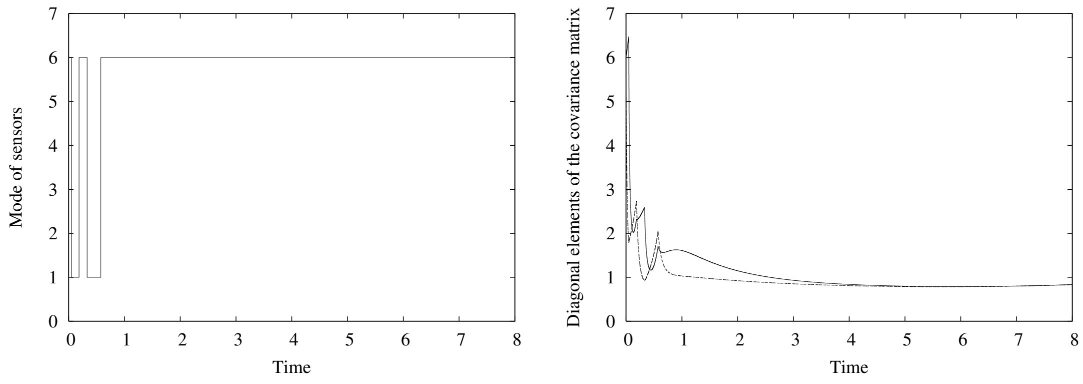

Figure 3: Optimal Sensor Operating Scheme with $P_0 = 6I$.

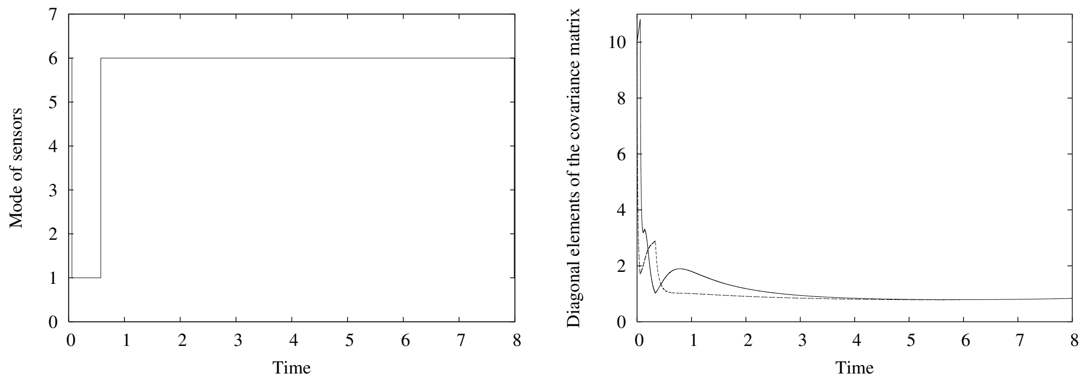

Figure 4: Optimal Sensor Operating Scheme with $P_0 = 10I$.

Table 2: A Comparison of Numerical Results with Other Methods.

Methods	Objective values
Method in [7] with $P_0 = \mathbf{0}$	19.6553
Method in [5] with $P_0 = 10I$	19.2353622
Proposed method with $P_0 = 10I$	16.5697177
Proposed method with $P_0 = 6I$	15.8781106
Proposed method with $P_0 = I$	14.3317631
Proposed method with $P_0 = \mathbf{0}$	12.9949699

which incorporates the discrete filled function algorithm into standard optimal control software, is proposed for finding a global solution of this problem. Numerical results show that the method is efficient, reliable, and robust in solving a complex discrete-valued optimal control problem. The proposed method successfully identified significantly improved solutions compared with other methods available in the literature.

Acknowledgment

The authors would like to thank David Packer from Curtin University of Technology for proof reading the early draft of this manuscript.

References

[1] Hovanessian, S.A. *Introduction to sensor systems.* Artech House, Inc., Norwood, 1988.

[2] Castanon, D. and Carin, L. Stochastic control theory for sensor management. In: *Foundations & Applications of Sensor Management* (A. Hero, D. Castanon, D. Cochran, and K. Kastella, eds.). SpringerLink, Boston, 2008, 7–32.

[3] Musick, S.H. Defence applications. In: *Foundations & Applications of Sensor Management* (A. Hero, D. Castanon, D. Cochran, and K. Kastella, eds.). SpringerLink, Boston, 2008, 257–268.

[4] Chung, T.H., Gupta, V., Hassibi, B., Burdick, J., and Murray, R.M. Scheduling for distributed sensor networks with single sensor measurement per time step. In: *IEEE International Conference on Robotics & Automation*, New Orleans, April 26 – May 1, 2004, 187–192.

[5] Lee, H.W.J., Teo, K.L., and Lim, A.E.B. Sensor scheduling in continuous time. *Automatica* **37**(12) (2001) 2017–2023.

[6] Baras, J.S. and Bensoussan, A. Optimal sensor scheduling in nonlinear filtering of diffusion process. *SIAM Journal of Control and Optimization* **27** (1989) 786–813.

[7] Feng, Z.G., Teo, K.L., and Rehbock, V. Optimal sensor scheduling in continuous time. *Dynamic Systems and Applications* **17** (2008) 331–350.

[8] Ahmed, N.U. *Linear and nonlinear filtering for scientists and engineers*. World Scientific, Singapore, 1998.

[9] Jennings, L.S., Fisher, M.E., Teo, K.L., and Goh, C.J. *MISER3.3–optimal control software: theory and user manual*. http://www.maths.uwa.edu.au/˜les/MISER3.3/ch1.pdf. University of Western Australia, Perth, 2004.

[10] Ge, R. A filled function method for finding a global minimizer of a function of several variables. *Mathematical Programming* **46**(1) (1990) 191–204.

[11] Ge, R. and Huang, C. A continuous approach to nonlinear integer programming. *Applied Mathematics and Computation* **34**(1) (1989) 39–60.

[12] Zhu, W. An approximate algorithm for nonlinear integer programming. *Applied Mathematics and Computations* **93**(2-3) (1998) 183–193.

[13] Ng, C.K., Li, D., and Zhang, L.S. Discrete global descent method for discrete global optimization and nonlinear integer programming. *Journal of Global Optimization* **37**(3) (2007) 357–379.

[14] Ng, C.K., Zhang, L.S., Li, D., and Tian, W.W. Discrete filled function method for discrete global optimization. *Computational Optimization & Applications* **31**(1) (2005) 87–115.

[15] Shang, Y. and Zhang, L. A filled function method for finding a global minimizer on global integer optimization. *Journal of Computational and Applied Mathematics* **181**(1) (2005) 200–210.

[16] Yang, Y. and Liang, Y. A new discrete filled function algorithm for discrete global optimization. *Journal of Computational and Applied Mathematics* **202**(2) (2007) 280–291.

[17] Gu, Y.H. and Wu, Z.Y. A new filled function method for nonlinear integer programming problem. *Applied Mathematics and Computation* **173**(2) (2007) 938–950.