DEVELOPMENT OF A SPECTRAL UNMIXING PROCEDURE USING A GENETIC ALGORITHM AND SPECTRAL SHAPE

SUBIR CHOWDHURY
Bachelor of Science, University of Lethbridge, 2010

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfilment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Geography
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

ABSTRACT

Spectral unmixing produces spatial abundance maps of endmembers or 'pure' materials using sub-pixel scale decomposition. It is particularly well suited to extracting a greater portion of the rich information content in hyperspectral data in support of real-world issues such as mineral exploration, resource management, agriculture and food security, pollution detection, and climate change. However, illumination or shading effects, signature variability, and the noise are problematic. The Least Square (LS) based spectral unmixing technique such as Non-Negative Sum Less or Equal to One (NNSLO) depends on "shade" endmembers to deal with the amplitude errors. Furthermore, the LS-based method does not consider amplitude errors in abundance constraint calculations, thus, often leads to abundance errors. The Spectral Angle Constraint (SAC) reduces the amplitude errors, but the abundance errors remain because of using fully constrained condition. In this study, a Genetic Algorithm (GA) was adapted to resolve these issues using a series of iterative computations based on the Darwinian strategy of 'survival of the fittest' to improve the accuracy of abundance estimates. The developed GA uses a Spectral Angle Mapper (SAM) based fitness function to calculate abundances by satisfying a SAC-based weakly constrained condition. This was validated using two hyperspectral data sets: (i) a simulated hyperspectral dataset with embedded noise and illumination effects and (ii) AVIRIS data acquired over Cuprite, Nevada, USA. Results showed that the new GA-based unmixing method improved the abundance estimation accuracies and was less sensitive to illumination effects and noise compared to existing spectral unmixing methods, such as the SAC and NNSLO. In case of synthetic data, the GA increased the average index of agreement between true and estimated abundances by

19.83% and 30.10% compared to the SAC and the NNSLO, respectively. Furthermore, in case of real data, GA improved the overall accuracy by 43.1% and 9.4% compared to the SAC and NNSLO, respectively.

ACKNOWLEDGEMENTS

> "I should premise that I use the term Struggle for Existence in a large and Metaphorical Sense, including dependence of one being on another, and including (which is more important) success in leaving progeny."
> - Charles Robert Darwin, Origin of Species (1859).

This study was carried out at the Alberta Terrestrial Imaging Centre (ATIC) of the University of Lethbridge and was funded primarily by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Lethbridge. In acknowledgement, the following thanks are deserved to those who have helped and supported my thesis. I wish to thank my co-supervisors, Dr. Karl Staenz and Dr. Derek R. Peddle, committee members, Dr. Jinkai Zhang and Dr. Adriana Predoi-Cross, and Dr. Craig Coburn and Dr. Phil Teillet, graduate students, Ilia Parshakov, Logan Pryor, and Shiyong Xu and ATIC employees, David Rolfson and Dr. Nadia Rochdi for their help, expertise, and encouragement. Furthermore, I am grateful to my father, Subodh Chandra Chowdhury, my mother, Rupasree Chowdhury, my wife, Sathi Rani Roy, and my brother, Sudipto Chowdhury for their support and encouragement through everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

p	total number of endmember
q	total number of spectral bands
$\boldsymbol{e}_i$	$(q \times 1)$ vector (reflectance spectrum) of the i^{th} endmember
α_i	abundance of the i^{th} endmember
$\boldsymbol{m}$	$(q \times 1)$ vector (reflectance spectrum) of the pixel
$\boldsymbol{n}$	$(q \times 1)$ noise vector
$\boldsymbol{a}$	$(p \times 1)$ vector of abundances
$\boldsymbol{E}$	$(q \times p)$ matrix of endmembers
$D(\boldsymbol{m}, \boldsymbol{Ea})$	distance function
$\boldsymbol{Q}$	$(g \times h)$ orthonormal matrix, where $g \geq h$
$\boldsymbol{I}$	identity matrix with g columns
$\boldsymbol{c}$	column vector of $\boldsymbol{E}$
$\boldsymbol{R}$	invertible $(g \times g)$ upper triangular matrix
ζ_{target}	$(q \times 1)$ vector of the target spectrum
$\zeta_{reference}$	$(q \times 1)$ vector of the reference spectrum

∂	spectral angle between ζ_{target} and $\zeta_{reference}$
$\boldsymbol{E}'$	$(q \times p)$ matrix of $\boldsymbol{e}_i' = \boldsymbol{e}_i / \|\boldsymbol{e}_i\|$
$\boldsymbol{a}'$	$(p \times 1)$ vector of $\alpha_i' = \alpha_i / \|\boldsymbol{e}_i\| / \|\boldsymbol{m}\|$
$\boldsymbol{m}'$	$(q \times 1)$ vector of $\boldsymbol{m}' = \boldsymbol{m} / \|\boldsymbol{m}\|$
θ_{ij}	spectral angle between $\boldsymbol{e}_i$ and j^{th} endmember $\boldsymbol{e}_j$
θ_z	solar zenith angle
θ_p	slope angle
γ_i	incident angle
ϕ_o	aspect angle
ϕ_a	solar azimuth angle
φ_i	spectral angle between $\boldsymbol{e}_i$ and $\boldsymbol{m}$
ω_i	coefficient of the i^{th} endmember
$\boldsymbol{\Omega}$	coefficient vector, where $\boldsymbol{\Omega} = \left[\omega_1, \omega_2, \omega_3, ..., \omega_p\right]$
$\boldsymbol{z}_i$	zero-mean random noise vector of the i^{th} endmember
η_i	scale factor to model the signature variability
$\boldsymbol{\Psi}$	vector that models additive noise, where $\boldsymbol{\Psi} = \left[\psi_1, \psi_2, \psi_3, ..., \psi_p\right]$

τ	models the illumination fluctuation
v	percentage of signature variability
x	the pixel number
t	total number of pixels
b	band number
k	total number of bands
w_{ih}	true abundances for the i^{th} endmember of the h^{th} pixel
α_{ih}	estimated abundances for the i^{th} endmember of the h^{th} pixel
u_{ih}	difference between w_{ih} and the mean of the true abundance over all pixels
d_{ih}	difference between α_{ih} and mean of the estimated abundance over all pixels

CHAPTER 1: INTRODUCTION

Hyperspectral imaging plays an important role in surface material discrimination due to its high-spectral resolution that covers many narrow contiguous bands usually from the visible to short-wave infrared wavelength region (i.e., from 400 nm to 2500 nm; Jensen, 2004). Information extraction from hyperspectral image data can be a very meticulous process. In order to achieve the best result from information extraction, it is often mandatory to perform data pre-processing, i.e., calibration, assessment and removal of sensor artifacts, and removal of atmospheric effects. Some of the most popular information extraction methods are spectral matching, feature fitting, matched filtering, and spectral unmixing (Jensen, 2004). Due to the mixture of spectral signatures of objects in a pixel recorded by the Instantaneous Field-Of-View (IFOV) of a sensor, spectral unmixing is evident (Miao *et al.*, 2007).

The spectral signature of an object refers to its reflectance spectrum within a specific wavelength region (e.g., VNIR and SWIR). The linear unmixing is a standard unmixing technique that decomposes a measured spectrum into area abundances of a collection of constituent endmembers, where the set of corresponding fractions or abundances refer to the proportion of each endmember present in the pixel (Adams *et al.*, 1993). An endmember represents a 'pure' material present in a scene, for instance, a particular type of mineral (e.g., kaolinite, muscovite, or alunite) or vegetation (e.g., wheat, canola, or barley).

Decomposing mixed pixels using linear unmixing is usually a three-step process. These general steps are data dimensionality reduction, endmember selection and identification, and abundance estimation (Jensen, 2004).

Typically, in order to obtain endmembers, reducing the number of bands or constructing new components through different data transformation techniques can eliminate the redundant information among the hyperspectral bands. Some of the dimension reduction techniques are: Principal Component Analysis (PCA; Lee *et al.*, 1991; Jensen, 2004), Minimum Noise Fraction (MNF; Green *et al.*, 1988), Singular Value Decomposition (SVD; Ball *et al.*, 2004), and Optical Real-time Adaptive Spectral Identification System (ORASIS; Bowles *et al.*, 1997).

Endmembers can be obtained from field/laboratory measurements (library endmembers), directly from the image data (image endmembers; Zhang *et al*., 2008), or by modelling (Peddle *et al*., 1999). Image endmembers can be manually identified based on their absorption features after the dimensionality reduction step (Keshava *et al*., 2002). Spectral signature of a target material might not match with its library endmember because of atmospheric illumination, viewing effects, or noise (Parra *et al*., 2000). In case of vegetation, it is due to the time-dependent phenological stage if the spectra are not acquired under the same illumination and viewing geometry as the image data (Hostert *et al*., 2003). Image endmember extraction can be performed using either a statistical or a geometrical approach. Geometric approaches usually assume the presence of pure pixels that represent endmembers; however, pure pixels are not mandatory in statistical approaches (Nascimento *et al.*, 2005; Bioucas-Dias, 2009). Several endmember selection methods can be used to find pure pixels such as the N-FINDER (Winter, 1999), Iterative

Error Analysis (IEA; Neville *et al.*, 1999), Automated Morphological Endmember Extraction (AMEE; Plaza *et al.*, 2002), ORASIS, Vertex Component Analysis (VCA; Nascimento *et al.*, 2005), and Simplex Growing Algorithm (SGA; Chang *et al.*, 2006). In the geometric approach, the vertices of spectral vectors belong to a simplex set corresponding to the endmembers. A geometric approach, fitting a simplex of minimum volume to the dataset, is used where the pure pixel assumption is not fulfilled. Algorithms employ this geometric approach are the Successive Projection Algorithm (SPA; Zhang *et al.*, 2008), Minimum Volume Enclosing Simplex (MVES; Chan *et al.*, 2009), Minimum Volume Simplex Analysis (MVSA; Li *et al.*, 2008), and Non-negative Matrix Factorization Minimum Volume Transform (NMF-MVT; Tao *et al.*, 2007), Simplex Identification via Split Augmented Lagrangian (SISAL; Bioucas-Dias, 2009) algorithm. A statistical approach, e.g., the Independent Component Analysis (ICA; Pearlmutter *et al.*, 1997; Parra *et al.*, 2000) considers that the abundances are statistically independent and can be modeled as mixtures of Dirichlet densities (Nascimento *et al.*, 2005).

The outcome of linear unmixing is abundance maps as illustrated in Figure 1. Linear unmixing with *a priori* information of the endmember spectra is known as an abundance estimation problem (Rosario-Torres, 2004). The research conducted for this thesis was focused on achieving optimal solution for an abundance estimation problem using a Genetic Algorithm (GA; Holland 1975; Goldberg, 1989; Mathworks, 2011). The GA, a series of iterative computations that efficiently explores large search spaces without being trapped into the local minimum, is a promising alternative to conventional heuristic methods (Gen *et al.*, 2000; Yao *et al.*, 2003; Huang *et al.*, 2006; Farzam *et al.*, 2008). It is a general adaptive optimization search methodology that generates optimal

solutions for search and optimization problems in various sectors, such as in engineering design (Zhao *et al.*, 2012), traffic and telecommunication routing (Vidal *et al.*, 2011), chemical kinetic analysis (Omata *et al.*, 2012), and finance and investment strategies (Mehta *et al.*, 1999). In addition, it was employed in previous linear unmixing studies and achieved better performance for abundance estimation (Farzam *et al.*, 2008).

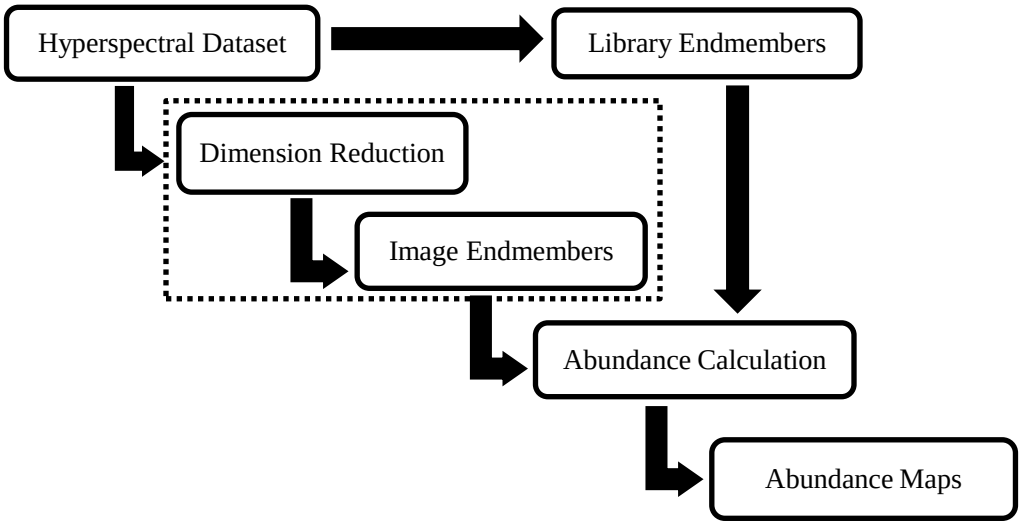


Figure 1: Linear unmixing process using either image endmembers or library endmembers.

1.1 Linear Mixture Model

In the literature, the linear mixture model is widely used for unmixing as it is easier to implement and more flexible in different applications than the non-linear model (Rogge *et al.*, 2006; Sanchez *et al.*, 2010). It assumes no interaction between endmembers and treats a pixel as a linear combination of endmembers weighted by their corresponding fractions or abundances plus an error term. It can be written as follows (Adams *et al.*, 1993; Boardman et al., 1993; Staenz *et al.*, 1999; Rosario-Torres, 2004):

$$\boldsymbol{m} = \sum_{i=1}^{p} \boldsymbol{e}_i \alpha_i + \boldsymbol{n} = \boldsymbol{E}\boldsymbol{a} + \boldsymbol{n}\,, \tag{1}$$

where $\boldsymbol{E}$ is a $(q \times p)$ matrix of endmembers, p is the total number of endmember, q is total number of spectral bands, $\boldsymbol{e}_i$ is the spectral signature of the i^{th} endmember, $\boldsymbol{a}$ is a $(p \times 1)$ vector of abundances, α_i is the abundance of the i^{th} endmember, $\boldsymbol{m}$ is a $(q \times 1)$ vector of the pixel spectrum, and $\boldsymbol{n}$ is a $(q \times 1)$ noise vector or an error term. The noise vector is considered negligible if it is an additive white Gaussian noise (Adams *et al.*, 1986). Therefore, Eq. 1 can be written as follows:

$$\boldsymbol{m} = \sum_{i=1}^{p} \boldsymbol{e}_i \alpha_i = \boldsymbol{E}\boldsymbol{a}\,. \tag{2}$$

The abundance vector needs to satisfy one of the following constraints.

Let $i = 1, \ldots, p$,

$$-\infty < \alpha_i < \infty\,, \text{ or} \tag{3}$$

$$0 \le \alpha_i \le 1 \text{ and } \sum_{i=1}^{p} \alpha_i = 1.0\,, \text{ or} \tag{4}$$

$$0 \le \alpha_i \le 1 \text{ and } \sum_{i=1}^{p} \alpha_i \le 1.0\,. \tag{5}$$

An unconstrained condition in Eq. 3 considers the range of abundance values from minus infinity to plus infinity, which could produce unrealistic abundances (Keshava *et al.*, 2002). The fully constrained case in Eq. 4 considers all abundances as non-negative and the sum of all abundances is equal to one (Staenz *et al.*, 1999; Rosario-Torres *et al.*, 2007; Farzam *et al.*, 2008). It requires a complete set of endmembers and overestimates or underestimates abundances in case of missing endmembers (Shang *et al.*, 2008).

Missing endmembers are often obvious in real world scenario as it is difficult to detect all endmembers present in a scene. The weakly constrained unmixing in Eq. 5 constrains abundances as non-negative and the sum of all abundances must be less than or equal to one (Staenz *et al.*, 1999; Rosario-Torres, 2004; Shang *et al.*, 2008). It minimizes the problem with overestimation or underestimation of abundances in case of missing endmember (Shang *et al.*, 2008; Rosario-Torres, 2004).

1.2 Applications of Linear Unmixing

The linear unmixing is widely used to solve real-world problems associated with mineral exploration, vegetation mapping, environmental monitoring, pollution detection, and climate change (Keshava *et al.*, 2002). For instance, the mineral exploration tasks include identifying minerals under tundra vegetation cover in the Canadian Arctic (Staenz *et al.*, 2000) and analyzing minerals in the bright salty soils of Gusev crater on Mars (Parente *et al.*, 2009). In addition, the vegetation mapping approaches comprise addressing the spectral reflectance problem of crops affected by the spectral contribution of background soils and shadows (Peddle *et al.*, 2005) and estimating crop fraction along with vegetation indices (Liu *et al.*, 2008). Moreover, the environmental monitoring applications include mapping water turbidity distribution in a floodplain lake together with autocorrelation modeling (Alcântara *et al.*, 2009), estimating fractional abundances of land-cover components and delineating potential erosion areas in tropical watersheds (De-Asis *et al.*, 2008). Additionally, one of the pollution detection tasks contains estimating acid mine drainage (AMD) in mine tailings (Richter *et al.*, 2008). Furthermore, snow cover estimation (Foppa *et al.*, 2002) using spectral unmixing is an

example of the climate change detection. Recently, medical science is also using abundance maps for cancer detection, blood testing, and identification of bacterial infections (Sirkeci, 2001). Therefore, realizing the importance of above-mentioned applications, it is obvious to ensure the accuracy of linear unmixing results, i.e., abundance maps.

1.3 Factors Influencing the Accuracy of Unmixing Results

There are many factors contributing to the radiant energy received by an imaging spectrometer. These factors often influence linear unmixing by producing overestimated or underestimated abundances. The accuracy of estimated abundances is influenced by several factors, such as illumination fluctuation, signature variability, and noise (Healey *et al.*, 1999; Tyo *et al.*, 2000; Shaw *et al.*, 2003; Nascimento *et al.*, 2005; Rabah *et al.*, 2011). The illumination perturbations occur due to surface topography and equally affect all bands (Nascimento *et al.*, 2005). The spectral mixture of materials interacting with incident sunlight contributes to the total reflected signal. The terrain slope and shadow affect the reflected signal and can have a serious effect on estimating abundances (Rouse *et al.*, 2005; Rabah *et al.*, 2009, 2010, and 2011). In addition, atmospheric and weathering effects, age-induced colour fading due to oxidization, and contamination due to chemical composition causes spectral signature variability of the surface material (Keshava *et al.*, 2002; Nascimento *et al.*, 2005).

1.4 Constraint Optimization Problem

An abundance estimation method calculates abundances by solving a constraint optimization problem. It searches for a solution to the constraint condition that the variables must satisfy to minimize a distance function (Deep *et al.*, 2008; Tsoulos, 2009). The constraint optimization problem derived from the linear mixture model in Eq.2 by imposing Eq. 5 can be written as follows (Rosario-Torres *et al.*, 2004, 2005, and 2007; Shang *et al.*, 2008):

$$\textbf{\textit{Minimize:}}\ D(\boldsymbol{m}, \boldsymbol{Ea}), \tag{6}$$

$$\textbf{\textit{Subject to:}}\ 0 \le \alpha_i \le 1 \text{ and } \sum_{i=1}^{p} \alpha_i \le 1.0 \text{ or } \sum_{i=1}^{p} \alpha_i = 1.0, \tag{7}$$

where $D(\boldsymbol{m}, \boldsymbol{Ea})$ is a distance function that leads to the solution of abundances.

1.5 How does Genetic Algorithm work?

Holland (1975) and his colleagues, inspired by the Darwinian natural selection, first introduced the concept of the Genetic Algorithm (GA) in the 1960s and 1970s. It is based on the theory of evolution in biology that plays a vital role in the way humankind survives in nature. In order to understand the GA, certain biological expressions need to be clarified. The building block of life, a cell, contains the blueprint of species, i.e., a set of chromosome, also called the strings of DNA (Mitchell, 1996). Each chromosome comprises a number of functional blocks of DNA called genes. Each gene is located in a particular position of the chromosome and encodes a certain type of protein representing various possibilities of an individual property, such as different blood types, different eye colours, and level of intelligence. In nature, exchanging genes between pairs of parent

chromosomes during sexual reproduction or crossover produces new pairs of child chromosomes. The result of a DNA replication error from parent to offspring causes a permanent change in the DNA sequence (heredity elements) of a gene. A mutation occurs when a protein encoded by a gene is altered due to certain incidents, such as radiation, viruses, transposons, and mutagenic chemicals (Mitchell, 1996; Bertram, 2000). Accumulation of mutations through a large number of generations can lead to the evolution of new species. The core mechanism of evolution, the fitness of species or populations, is determined by their survival and reproduction, i.e., contribution of their genes to the next generation (Orr, 2009).

In order to solve a constraint optimization problem (e.g., Eq. 6), the GA randomly generates an initial population of candidate solutions or chromosomes (Goldberg, 1989). Each chromosome contains a number of genes and each gene contains an encoded value within the bounds of a constrained condition (e.g., Eq. 7; Chang, 2007). The quality assessment of each chromosome, the fitness value calculation, is carried out employing its encoded values to a fitness function, which is a distance function to be minimized (Huang *et al.*, 2006). Chromosomes are ranked based on their fitness value, where the top ranking fitter chromosomes are copied multiple times, because crossover requires a pair of chromosome to produce a child (Mathworks, 2011). Chromosomes are selected randomly as parents to participate in crossover, mutation, and elite operation. A crossover operator randomly exchanges genes between a pair of parents and produces a child, and a mutation operator alters a random gene of a parent and produces a child (Figure 2). In addition, an elite operator copies the fittest parent as a child without any gene alternation. The number of total children in a new population is the same as the

number of total parents in an old population. Children from a new generation replace the parents from older generation, and this process goes through fitness evaluation and termination condition (e.g., number of generations) checking until an optimal solution is obtained as illustrated in Figure 3 (Ja'fari *et al.*, 2012; Mathworks, 2011).

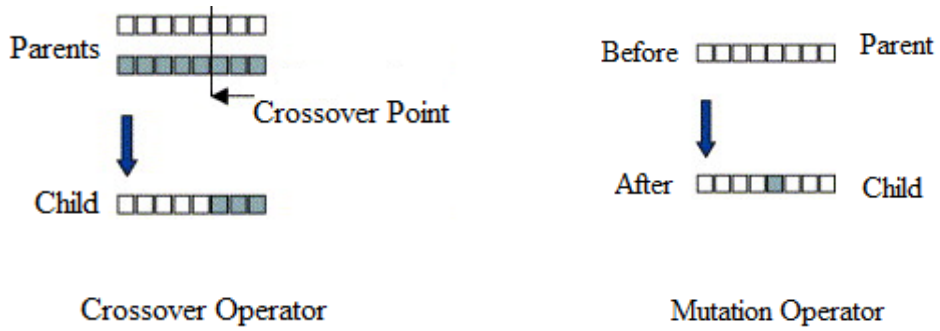


Figure 2: Crossover and mutation operation (modified from Huang et al., 2006).

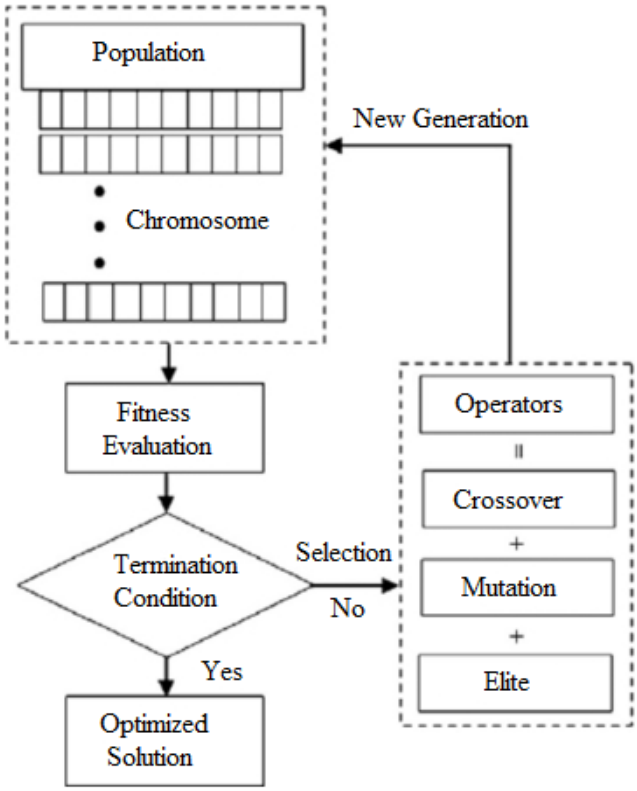


Figure 3: The GA flowchart (modified from Ja'fari et al., 2012).

1.6 Existing GA-based Constraint Optimization Approaches

Some of the most popular GA-based constraint optimization approaches such as the Simple Genetic Algorithm (SGA; Goldberg, 1989; Sastry, 2007), Self-organizing Migrating Genetic Algorithm (C-SOMGA; Deep *et al.*, 2008), and Novel Genetic Algorithm (NGA; Tsoulos, 2009) were studied prior to the development of the proposed GA. The SGA, C-SOMGA, and NGA are discussed in the Appendix section.

1.7 Objectives

The main objective of this study is to implement algorithms that will improve the accuracy of abundance maps. More specifically, the objectives were as follows.

a) Perform constraint optimization using a GA to improve the accuracy of abundance maps as well as to make it less sensitive to illumination effects, signature variability, and noise; and
b) Compare and validate the accuracy of the proposed GA-based abundance quantification approach against existing approaches using synthetic and real hyperspectral datasets.

1.8 Hypotheses

Three main hypotheses were tested in this thesis:

a) A weakly constrained condition can be constructed to check the illumination effect in each pixel;

b) The GA can achieve an optimal solution of a constraint optimization problem by minimizing a properly designed fitness function; and

c) The GA-based unmixing results can be less sensitive to illumination effects, signature variability, and noise compared to advanced abundance estimation methods.

CHAPTER 2: BACKGROUND

In many cases, the endmembers of a scene are known where the only remaining task is abundance calculations. This chapter is mainly focused on the constraint optimization where the advantages and disadvantages of advanced abundance estimation algorithms are presented.

2.1 Multiple Endmember Spectral Mixture Analysis Algorithm (MESMA)

Roberts *et al.* (1998 and 2007) implemented the MESMA-based abundance calculation in the VIPER Tools software, which considers the number and type of endmembers, which vary from pixel to pixel in a scene. It deals with the illumination effect using a photometric shade (i.e., a shade is set to zero reflectance) or non-photometric shade (i.e., a shade spectrum is used from a spectral library). The approach can create linear mixing models from one to three types of non-shade endmembers, where each non-shade endmember contains the combination of its sub-type endmembers (e.g., a water endmember with 3 sub-types: water with sediments, water without sediments, and water with chlorophyll). MESMA selects the models for each pixel based on the four parameters (Roberts *et al.*, 1998 and 2007):

a) The range of endmember fractions, e.g., the minimum endmember fraction varies from -0.5 to 1;
b) The shade fraction (i.e., range from 0 to 1) to ensure that deep shadow and water remain distinct;

c) The root mean square error (RMSE) threshold (e.g., default threshold is 0.025), which is calculated as

$$RMSE = \frac{\|\boldsymbol{n}\|_2}{q-1} \ , \tag{8}$$

d) where $\| \bullet \|_2$ is an Euclidean norm, e.g., $\|\boldsymbol{n}\|_2 = (n_1^2 + n_2^2 + ... + n_p^2)^{\frac{1}{2}}$ where a higher or a lower threshold, respectively increases or decreases the likelihood for a candidate model to be selected; and

e) A residual condition consist of a threshold and a number of consecutive bands to apply the threshold, where all models that produce residual spectra exceeding the threshold for the consecutive bands are excluded.

After selecting the models for each pixel, the MESMA produces a number of results such as an abundance map for each endmember (maximum 3 abundance maps), a shade abundance, an RMSE map with residuals, and a classification map with the dominating abundances.

The disadvantages of this method are as follows:

a) It required extensive sub-type spectra for each endmember type;

b) The non-shade endmember fraction calculation is limited to maximum three endmember types only; and

c) It produces error-prone results if the four parameters discussed earlier are not set properly; in addition, the adjustment of these parameters is done on a trial and error basis.

2.2 Non-Negative Sum-To-One Algorithm (NNSTO)

Rosario-Torres and Velez-Reyes (2004, 2005, and 2007) investigated several Least Square (LS) based abundance estimation methods: NNSLO (Lawson *et al.*, 1974; Björck, 1996), the Non-Negative Sum-To-One (NNSTO; Lawson *et al.*, 1974; Björck, 1996), Non-Negative Least Square (NNLS; Lawson *et al.*, 1974), Unconstrained Least Square (ULS; Keshava *et al.*, 2002), Chang Least Square Sum-to-One (CLSSTO; Chang *et al.*, 2003), Expectation Maximization Maximum Likelihood (EMML; Shepp *et al.*, 1982; Velez-Reyes *et al.*, 2003), and Image Space Reconstruction Algorithm (ISRA; Velez-Reyes *et al.*, 2003). Among these algorithms, the NNSLO estimated abundances had a better match with their physical meanings as well as was more robust to noise compared to other algorithms. The reason is it imposed the weakly constrained condition and considers the existence of the low reflectance pixels for abundance estimation. The NNSLO algorithm defined the constraint optimization problem as follows:

$$\textbf{\textit{Minimize:}}\ \left\|\boldsymbol{E}\boldsymbol{a}-\boldsymbol{m}\right\|_2^2 \text{ and} \tag{9}$$

$$\textbf{\textit{Subject to:}}\ 0\le\alpha_i\le 1 \text{ and } \sum_{i=1}^{p}\alpha_i \le 1.0\,. \tag{10}$$

Rosario-Torres and Velez-Reyes (2004, 2005, and 2007) established the NNSLO algorithm using the following steps to solve Eq. 9 based on the LS solution from Lawson *et al.* (1974).

a) Compute QR-decomposition (Trefethen *et al.*, 1997; Press *et al.*, 2007) for the matrix $\boldsymbol{E}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ 0\end{bmatrix}$, where $\boldsymbol{Q}$ is an ($g\times h$) orthonormal matrix (i.e., $g \ge h$ and $\boldsymbol{Q}^T\boldsymbol{Q}=\boldsymbol{I}$, where $\boldsymbol{I}$ is an identity matrix with g columns) and $\boldsymbol{R}$ is an invertible ($g\times g$) upper

triangular matrix (i.e., all entries below the main diagonal are zero). In addition, compute $\boldsymbol{c} = \boldsymbol{Q}^T \boldsymbol{m}$, where $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are the columns of $\boldsymbol{E}$. Therefore, Eq. 9 can be transformed as follows:

$$\textbf{\textit{Minimize:}} \left\| \boldsymbol{Ra} - \boldsymbol{c}_1 \right\|_2^2 \text{ and} \tag{11}$$

$$\textbf{\textit{Subject to:}} \ 0 \le \alpha_i \le 1 \text{ and } \sum_{i=1}^{p} \alpha_i \le 1.0 . \tag{12}$$

Compute unconstraint abundance vector, $\boldsymbol{X}_{ULS} = \boldsymbol{R}^{-1} \boldsymbol{c}_1$.

b) Set $\boldsymbol{G} = \left[\boldsymbol{I} \ \ 1 \right]^T$.

c) Set $\boldsymbol{f} = \left[0 \ \ 1 \right]^T$.

d) Set $\boldsymbol{H} = \left[-\boldsymbol{X}^T{}_{ULS} \ \ 1^T \boldsymbol{X}_{ULS} \ \ -1 \right]^T$.

e) Set $\boldsymbol{L} = \left[\boldsymbol{G}^T \ \ \boldsymbol{H}^T \right]^T$.

f) Set $\boldsymbol{r} = \boldsymbol{Lu} - \boldsymbol{f}$, where $\boldsymbol{u}$ is the solution for $\boldsymbol{L}$ and $\boldsymbol{f}$.

g) Set $\xi_i = -\boldsymbol{r}_i / \boldsymbol{r}_{p+1}, \ 0 \le i \le p$, where $\boldsymbol{Z}$ is a $(1 \times p)$ vector of ξ_i of i^{th} endmember.

h) The solution is $\boldsymbol{a} = \boldsymbol{X}_{ULS} + \boldsymbol{R}^{-1} \boldsymbol{Z}$.

In the literature, existing LS-based methods including the NNSLO depend on the shade endmembers to deal with the illumination effects and do not use the constraint condition (e.g., Eqs. 2, 3, or 4) to check if a pixel is assigned to an illumination or amplitude error (Rosario-Torres *et al.*, 2007; Farzam *et al.*, 2008; Rabah *et al.*, 2011).

2.3 The Spectral Angle Constraint Algorithm (SAC)

A recent abundance estimation method, the Spectral Angle Constraint (SAC; Rabah *et al.*, 2009, 2010, and 2011), was established from the concept of the Spectral Angle Mapper (SAM; Kruse *et al.*, 1993; Staenz *et al.*, 1999; Park *et al.*, 2007). The SAM is widely used as a spectral similarity measure, because of its insensitivity to the pixel illumination. It calculates the angle between a target material and a reference material, where a smaller angle indicates a closer match. For instance, the spectral angle, ∂, between a target spectrum, ζ_{target} and a reference spectrum, $\zeta_{reference}$ can be mathematically written as follows:

$$\partial = \cos^{-1}\left(\frac{\zeta_{target}\ \zeta_{reference}}{\left\|\zeta_{target}\right\|_2\ \left\|\zeta_{reference}\right\|_2} \right) . \tag{13}$$

The SAM assumes that the lines connecting the origin and each spectrum point are the vectors of the direction of the spectra corresponding to the range of possible illuminations with all possible positions of the materials (Kruse *et al.*, 1993). A pixel with poor illumination will fall closer to the origin than a pixel with greater illumination, although both pixels might have the same reflectance characteristics. Accordingly, the SAM is insensitive to changes in pixel illumination as increasing or decreasing illumination only changes the magnitude, not the direction. SAM considers that the reflected signal amplitude variation occurs due to the sunlight only, where the effects of diffusion from the skylight, cloud light, and transmitted light through the material is ignored (Adler-Golden *et al.*, 2001; Adams *et al*., 2006).

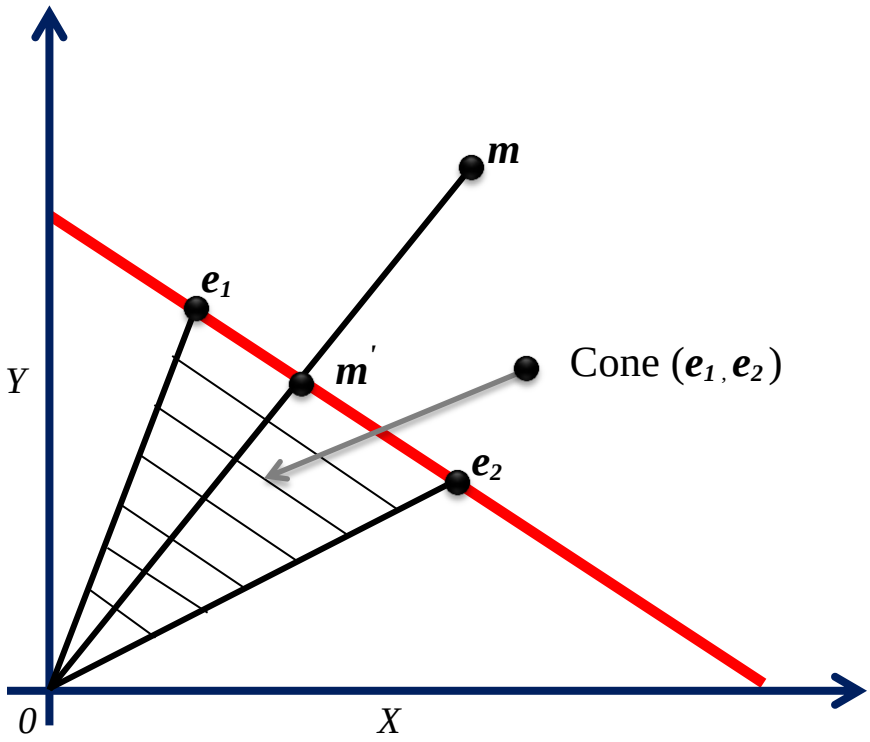


Figure 4: SAC illumination minimization. For example, the pixel $\boldsymbol{m}$ *crosses the convex cone formed by endmembers* $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ *in a two-dimensional space with X and Y. After correcting* $\boldsymbol{m}$ *to* $\boldsymbol{m}'$*, the pixel belongs to the convex cone (modified from Rabah et al., 2011).*

The SAC minimizes the illumination effects across the convex cone formed by endmembers such as shown in Figure 4 using the spectral angle as a constraint condition without relying on shade endmembers (Rabah *et al.*, 2009, 2010, and 2011). It corrects a pixel $\boldsymbol{m}$ contaminated by changing illumination effects, which crosses the convex cone $(\boldsymbol{e}_1, \boldsymbol{e}_2)$. After correcting its amplitude error, the pixel $\boldsymbol{m}'$ belongs to the cone $(\boldsymbol{e}_1, \boldsymbol{e}_2)$, where the direction remains constant. Rabah *et al.* (2009 and 2011) transformed Eq. 5 as follows:

$$\boldsymbol{m}' = \sum_{i=1}^{p} \boldsymbol{e}_i' \alpha_i' = \boldsymbol{E}' \boldsymbol{a}' , \tag{14}$$

where $\boldsymbol{m}' = \boldsymbol{m} / \|\boldsymbol{m}\|_2$, $\boldsymbol{e}_i' = \boldsymbol{e}_i / \|\boldsymbol{e}_i\|_2$, $\alpha_i' = \alpha_i / \|\boldsymbol{e}_i\|_2 / \|\boldsymbol{m}\|_2$, $\boldsymbol{E}'$ is a $(q \times p)$ matrix of $\boldsymbol{e}_i'$, $\boldsymbol{a}'$ is a $(p \times 1)$ vector of α_i', and $\boldsymbol{m}'$ is a $(q \times 1)$ vector of the corrected pixel spectrum. Rabah *et al.* (2011) defined the minimization problem as follows:

$$\textbf{\textit{Minimize:}} \quad \|\boldsymbol{E}'\boldsymbol{a}' - \boldsymbol{m}'\|_2^2 \quad \text{and} \tag{15}$$

$$\textbf{\textit{Subject to:}} \quad 0 \le \alpha_i \le 1 \ \text{and} \sum_{i=1}^{p} \alpha_i = 1.0, \tag{16}$$

where $\boldsymbol{E}'$ is a full rank matrix (i.e., all vectors in this matrix are linearly independent) and the solution without constraints is given by the Moore-Penrose pseudo-inverse of $\boldsymbol{E}'$, i.e., $(\boldsymbol{E}'^{T}\boldsymbol{E}')^{-1}\boldsymbol{E}'^{T}$. In this case, $\boldsymbol{E}'^{T}\boldsymbol{E}' = \boldsymbol{\Gamma}$, where $\Gamma_{ij} = cos(\theta_{ij})$, θ_{ij} measures the spectral angle between the i^{th} endmember $\boldsymbol{e}_i$ and j^{th} endmember $\boldsymbol{e}_j$ where $i = 1,...,p$, and $j = 1,...,p$ (Rabah *et al.*, 2009 and 2011). $\boldsymbol{a}'$ is calculated as follows:

$$\boldsymbol{a}' = \boldsymbol{\Gamma}^{-1}\boldsymbol{E}'^{T}\boldsymbol{m}', \tag{17}$$

where $\boldsymbol{E}'^{T}\boldsymbol{m}' = \boldsymbol{\Lambda}$ with $\Lambda_i = cos(\varphi_i)$ and φ_i is the spectral angle between $\boldsymbol{e}_i$ and $\boldsymbol{m}'$. As there are infinity solutions in Eq. 17, a coefficient vector, $\boldsymbol{\Omega}$, is selected to verify a fully constraint condition in Eq. 16 so that

$$\boldsymbol{a} = \boldsymbol{\Omega}.*\boldsymbol{a}', \tag{18}$$

where $\boldsymbol{\Omega} = \left[\omega_1, \omega_2, \omega_3, ..., \omega_p\right]$, ω_i is a coefficient of the i^{th} endmember and .* is an array multiplication operator ($\boldsymbol{\Omega}.*\boldsymbol{a}'$ is the element-by-element product of the arrays $\boldsymbol{\Omega}$ and $\boldsymbol{a}'$, where both arrays have same size). Rabah *et al.*, (2009 and 2011) calculated abundances using Eq. 18 as follows:

$$\alpha_i = \alpha_i' / \sum_{i=1}^{p} (\alpha_i'), \tag{19}$$

where $\omega_i = 1/\sum_{i=1}^{p} (\alpha_i')$ satisfy the fully constrained condition in Eq. 16.

Though the SAC method is less sensitive to illumination variability of the same material, it may overestimate or underestimate abundances in case of missing endmembers because of imposing the fully constraint condition (Shang *et al.*, 2008).

2.4 The Least Square based Genetic Algorithm (LS-GA)

Farzam *et al.* (2008) developed the LS-GA to estimate abundances because most of the traditional LS-based methods (e.g., the NNSLO algorithm) were mathematically complex, expensive to implement and cannot be solved analytically. The LS-GA method achieved the solution of the following constrained optimization problem as follows:

$$\textbf{\textit{Minimize:}} \left| (\boldsymbol{Ea} - \boldsymbol{m})^T (\boldsymbol{Ea} - \boldsymbol{m}) \right| \text{ and} \tag{20}$$

$$\textbf{\textit{Subject to:}}\ 0 \le \alpha_i \le 1 \text{ and} \sum_{i=1}^{p} \alpha_i = 1.0, \tag{21}$$

where $\left| \bullet \right|$ means the absolute value. Farzam et al. (2008) described the steps of LS-GA as follows:

a) **Encoding** – Abundances are encoded as real values. Initially, a random uniform population of 40 chromosomes is generated.

b) **Fitness** – The LS-based fitness function in Eq. 20 is used with a fully constraint condition in Eq. 21. It computes the fitness of each chromosome.

c) **New Population** – The following steps are repeated to create new populations, where the number of chromosome is the same as the initial population.

d) **Selection** – Fitter chromosomes are copied multiple times. Chromosomes are then selected as parents via stochastic uniform selection.

e) **Crossover** – 60% of total parents are selected for crossover.

f) **Mutation** – 2% of total parents are selected for uniform mutation.

g) **Accepting** – Children from steps 5 and 6 are assigned to a new population.

h) **Replace** – a new population replaces the old population.

i) **Test** – the algorithm is stopped if the termination conditions (error tolerance and maximum number of generation) are satisfied.

j) **Loop** – If conditions are not satisfied, go to step 3.

Although the LS-GA developed by Farzam *et al.* (2008) is robust to noise and improved abundance estimation, the problem with illumination effect remains the same as those with the NNSLO algorithm (Rabah *et al.* 2011). Imposing fully constraint condition is also another disadvantage of this algorithm (Rosario-Torres *et al.*, 2007; Shang *et al.*, 2008).

CHAPTER 3: DATASETS

Preliminary experiments were conducted using synthetic data and a real dataset, acquired over Cuprite, Nevada (USA) with the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS).

3.1 Synthetic Data Creation

The synthetic mixtures were generated using varying parameters of illumination fluctuation, spectral signature variability, and Signal-to-Noise Ratio (SNR) with a set of mineral endmembers to mimic a realistic scenario. Nine mineral endmembers (Kaolinite KGA1, Dumortierite, Nontronite, Alunite, Sphene, Pyrophilite, Halloysite, Muscovite and Kaolinite CM9) with 221 bands between 400 nm - 2500 nm as shown in Figure 5 were used to generate the mixtures. These mineral endmembers were acquired from the United States Geological Survey (USGS) spectral library (Clark *et al.*, 1993; Plaza *et al.*, 2011). Changing the aforementioned parameters generated 12 synthetic data cubes. The size of each synthetic data cube is 100 pixels × 100 pixels × 221 bands. True abundances of the endmembers were acquired from Plaza *et al.* (2011). These abundances mimic a real-world scenario, where pure pixels are rarely found in the scene and the mixtures are extreme in the boundaries of clusters as illustrated in Figure 6. The synthetic data were created using the linear mixture model as follows (Nascimento *et al.*, 2005):

$$\boldsymbol{m} = \tau \sum_{i=1}^{p} \alpha_i \boldsymbol{e}_i \eta_i + \boldsymbol{n} \text{ and } \boldsymbol{n} = \tau \sum_{i=1}^{p} \alpha_i \boldsymbol{z}_i + \boldsymbol{\Psi}, \tag{22}$$

where η_i is the scale factor to model the signature variability characterized by spectral shape invariance, $\mathbf{z}_i$ is the zero-mean random noise vector of the i^{th} endmember that models the spectral signature variability not achieved by η_i, τ models the illumination fluctuation, and $\boldsymbol{\Psi} = \left[\psi_1, \psi_2, \psi_3, ..., \psi_p\right]$ is the vector that models additive sensor noise for various SNR in decibel (dB).

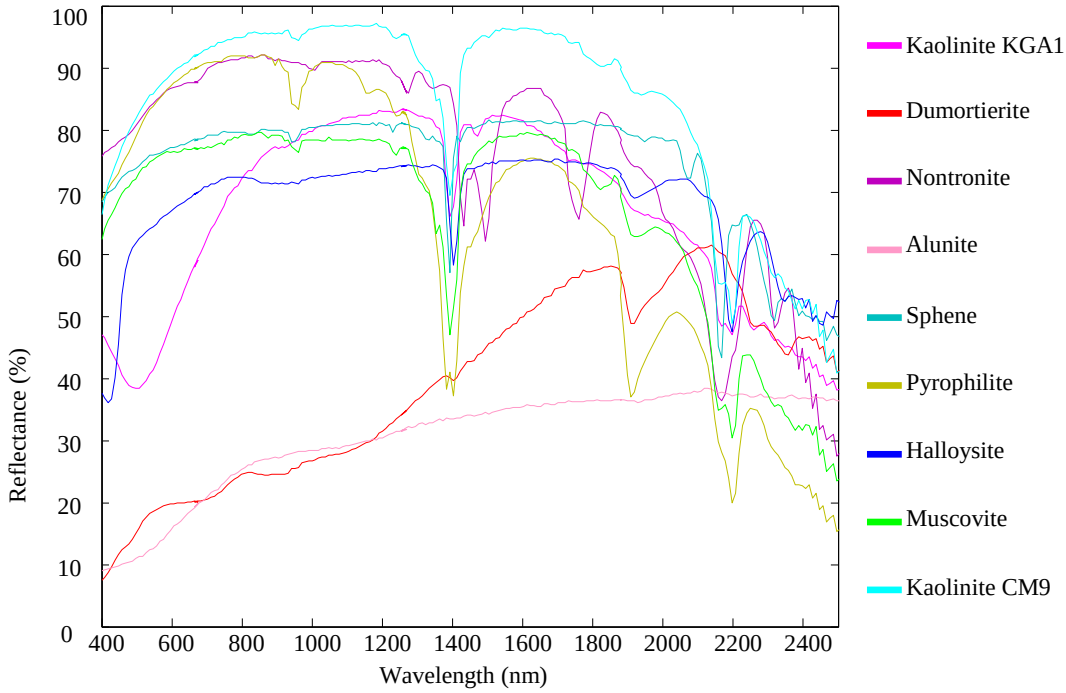


Figure 5: Mineral endmembers used to generate and unmix the synthetic data.

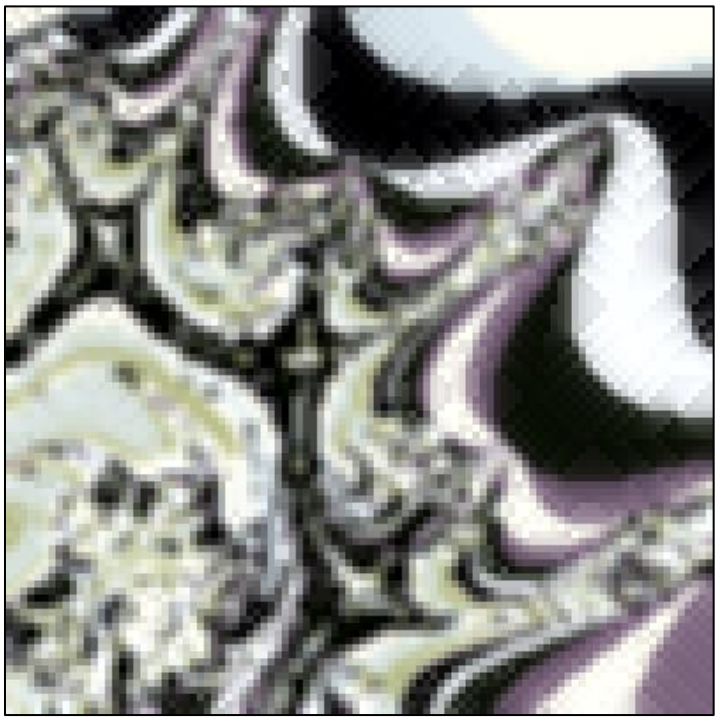


Figure 6: Clusters of a synthetic dataset with 100 pixels × 100 pixels × 221 bands shown in red (606.43 nm), green (517.60 nm), and blue (448.61 nm), without applying noise, spectral signature variability, and illumination fluctuation.

The reflectance of an inclined surface was modeled by multiplying the reflectance of the flat surface, $\sum_{i=1}^{p} \alpha_i \boldsymbol{e}_i$ by the illumination fluctuation, τ in Eq. 22. The illumination fluctuation was calculated as follows (Teillet *et al.*, 1982; Riaño *et al.*, 2003):

$$\tau = \cos\gamma_i \big/ \cos\theta_z \quad \text{and} \tag{23}$$

$$\cos\gamma_i = \cos\theta_p \cos\theta_z + \sin\theta_p \sin\theta_z \cos(\phi_a - \phi_o), \tag{24}$$

where γ_i is the incident angle, which is the angle between the normal to the ground and the sun rays, θ_p is the slope angle, θ_z is the solar zenith angle, ϕ_a is the solar azimuth angle, and ϕ_o is the aspect angle. The values of the τ matrix were generated using a

random uniform distribution with a range between 0 and 1.28. This range was calculated using Eqs. 23 and 24, where θ_z = 38.5° and ϕ_a = 162.8° from an AVIRIS[1] scene acquired over rugged terrain, θ_p = 0° to 60°, and ϕ_o = 0° to 360° were used. These angles are illustrated in Figure 7a and the topographic effect on inclined surface is shown in Figure 7b.

The signature variability was modeled as follows (Nascimento *et al.*, 2005):

$$\eta_i = (1+v) \tag{25}$$

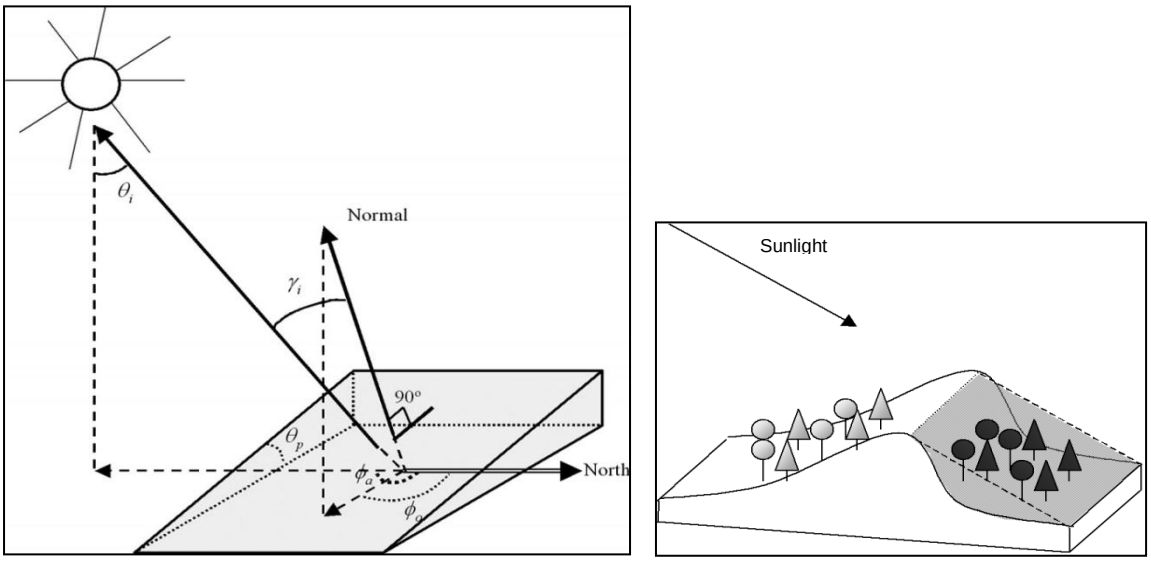
where v represents the percentage of signature variability. Eq. 25 with v = 0%, 5%, and 10% were applied to Eq. 22.

The noise vector $\boldsymbol{n}$ in Eq. 22 was generated by applying SNR = 15, 30, 60, and 90 dB as follows (Nascimento *et al.*, 2005):

$$SNR_{dB} = 10\log_{10}\left(\sum_{x=1}^{t} \left(\tau \sum_{i=1}^{p} \alpha_i \boldsymbol{e}_i \eta_i \right)_x^2 \Big/ \sum_{x=1}^{t} \left(\tau \sum_{i=1}^{p} \alpha_i \boldsymbol{z}_i + \psi_i \right)_x^2 \right), \tag{26}$$

where x is the pixel number and t is the total number of pixels.

[1] "3.1 AVIRIS Imagery, Rugged Terrain", German Remote Sensing Data Center. Accessed March 31, 2012: http://atcor.dlr.de/results4_en.html.



(a) *(b)*

Figure 7: (a) Contribution of angles to model illumination effects and (b) topographic effect on the reflectance of an inclined surface (Figure modified from Riano et al., 2003).

3.2 Real Data

ATmospheric REMoval (ATREM) and Empirical Flat Field Optimized Reflectance Transform (EFFORT) corrected AVIRIS reflectance data of Cuprite, Nevada, USA of year 1995 obtained from the ENVI[2] tutorial package was used for testing the GA-based unmixing (Gao *et al.*, 1990; Boardman *et al.*, 1998; Zhang *et al.*, 2008). This hyperspectral data cube has 400 × 350 pixels, 50 bands of short-wave infrared data between 2000 nm to 2500 nm with ~10 nm spectral resolution and 20 m spatial resolution. The study area Cuprite (Figure 8), a mining district located in west-central Nevada, is considered as an ideal location for unmixing experiments because it is used extensively by the hyperspectral community due to the presence of large outcrops of spectrally distinct alteration minerals and low vegetation cover (Abrams *et al.*, 1977;

[2] ENVI Software - ITT Visual Information Solutions. Accessed Jan 07, 2011: http://www.ittvis.com/ProductServices/ENVI.aspx.

Kruse *et al.*, 1990; Hook *et al.*, 1990; Clark *et al.*, 1993b; Rowan *et al.*, 2003; Zhang *et al.*, 2008). The study site consists of Tertiary volcanic rocks and Country rocks, where Tertiary rocks have been hydrothermally altered primarily to bleach silicified rocks and opalized rocks with limonitic argillized rocks, and Country rocks are primarily Cambrian phyllitic siltstone and limestones (Rowan *et al.*, 2003). Nine mineral endmembers (Muscovite-medlow-Al, Muscovite-medhi-Al, Montmorillonite-Na, Kaolinite KGa-2 (pxl), Kaolinite KGa-1 (wxl), Hematite .02+Quartz .98, Chlorite+Muscovite, Chalcedony, and Alunite GDS82), acquired from the USGS[3] digital spectral library, are used for unmixing Figure 9).

Figure 8: AVIRIS reflectance data of Cuprite shown in red (2100 nm), green (2200 nm), and blue (2340 nm).

[3] "USGS Spectroscopy Lab - Clark and others, 2007 Spectral Library splib06a, Data Series 231", USGS Digital Spectral Library splib06a. Accessed March 31, 2012: http://speclab.cr.usgs.gov/spectral.lib06/ds231/datatable.html.

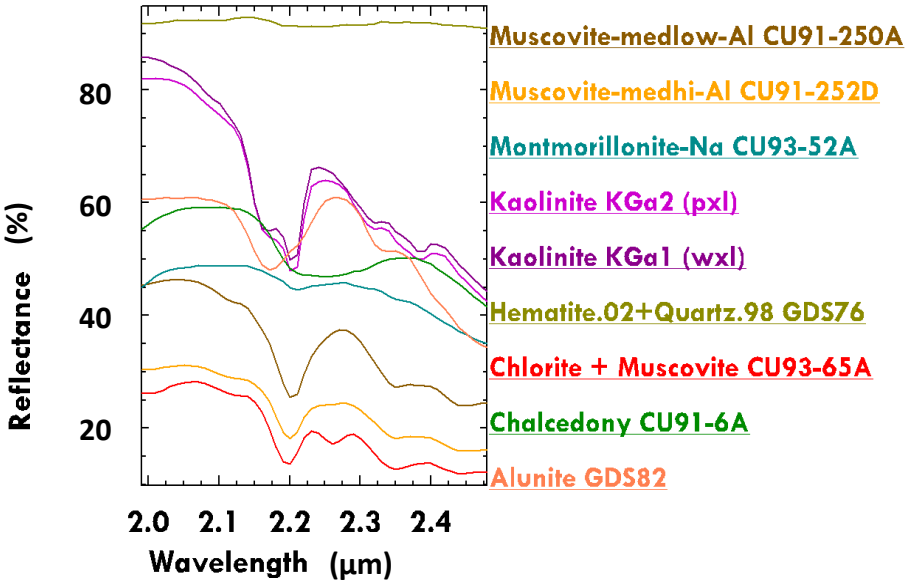


Figure 9: Nine mineral endmembers used to unmix the AVIRIS Cuprite data.

CHAPTER 4: METHODS

In this chapter, a new constraint optimization problem is formulated as well as the development of a properly designed GA to solve this optimization problem is described.

4.1 Formulation of a New Constraint Optimization Problem

A new constraint optimization problem is developed based on the concept of the spectral angle and weakly constrained condition. The purpose is to take the benefits of both the SAC (Rabah *et al.* 2011) and NNSLO (Rosario-Torres *et al.*, 2007) algorithm. The proposed constraint optimization problem can then be written as follows:

$$\textbf{\textit{Minimize:}}\ \left|\cos^{-1}\left(\frac{(\boldsymbol{E}'\boldsymbol{a}')\boldsymbol{m}'}{\left\|\boldsymbol{E}'\boldsymbol{a}'\right\|_2\ \left\|\boldsymbol{m}'\right\|_2}\right)\right|, \tag{27}$$

$$\textbf{\textit{Subject to:}}\ 0 \le \omega_i\alpha_i' \le 1, \text{ and } \sum_{i=1}^{p}\omega_i\alpha_i' \le 1.0, \tag{28}$$

where the SAM (Kruse *et al.*, 1993) function in Eq. 27 measures the spectral angle (δ) between $\boldsymbol{m}'$ and $\boldsymbol{E}'\boldsymbol{a}'$ (Eq. 14). An example of the spectral angle (δ) minimization is shown in Figure 10. The weakly constrained condition in Eq. 28 is constructed using $a_i = \omega_i a_i'$ from Eq. 18, while the $\boldsymbol{a}'$ vector is calculated using Eq. 17. In this study, the goal of the proposed GA is to calculate the optimal results of the $\boldsymbol{\Omega}$ vector (i.e., $\boldsymbol{\Omega} = \left[\omega_1, \omega_2, \omega_3, ..., \omega_p\right]$) by minimizing Eq. 27 to satisfy Eq. 28.

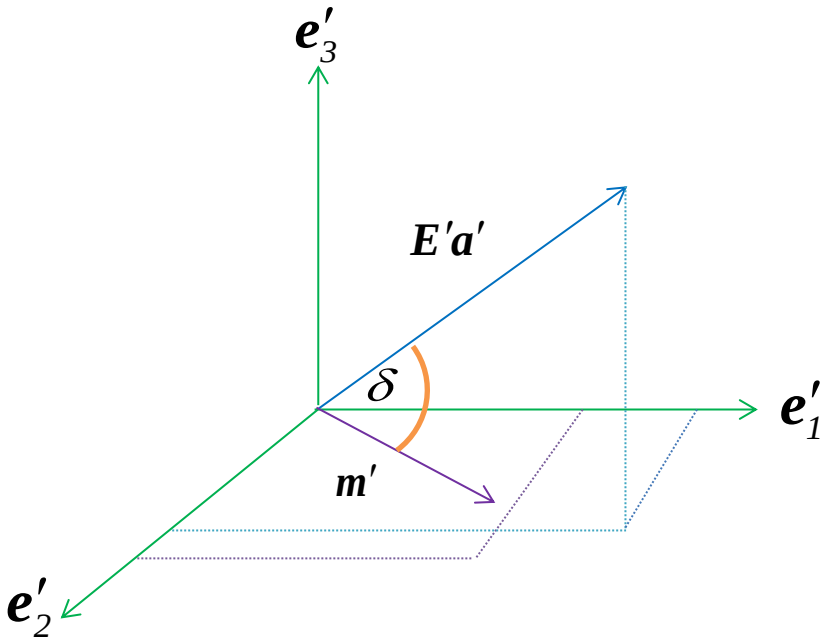


Figure 10: An example of spectral angle, δ, between a target spectrum, $\boldsymbol{m}'$, and a reference spectrum, $\boldsymbol{E}'\boldsymbol{a}'$, in a three-dimensional space with endmember $\boldsymbol{e}'_1, \boldsymbol{e}'_2$, and $\boldsymbol{e}'_3$.

Figure 11 illustrates a four-phase process of unmixing. The hyperspectral data cube and the endmember spectra in phase 1 are the inputs to phase 2, where the SAC calculates $\boldsymbol{a}'$. In phase 3, the proposed GA calculates $\boldsymbol{\Omega}$ by taking the inputs of Eqs. 27 and 28 from phase 2. Finally, abundance vectors are calculated in phase 4.

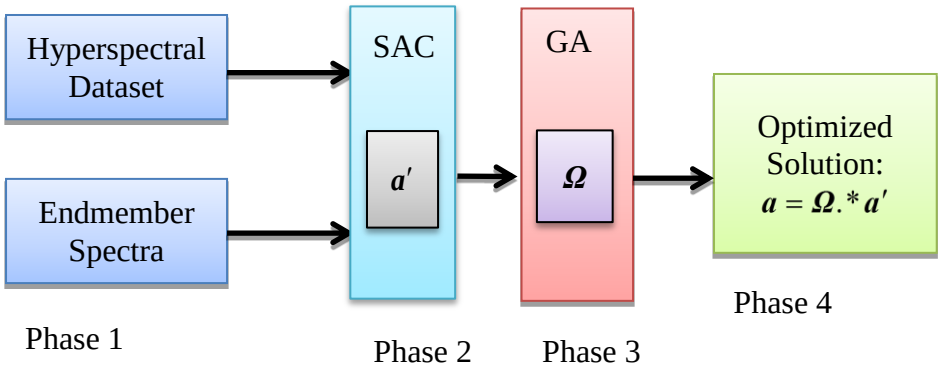


Figure 11: Four phases of the GA-based unmixing with the computation process of $\boldsymbol{a}$ (real abundances) using the element-by-element product of the arrays $\boldsymbol{\Omega}$ (coefficients) and $\boldsymbol{a}'$ (normalized abundances).

4.2 The GA Platform Selection

There are many public domain GA packages such as a C++ GA toolbox developed by the Illinois Genetic Algorithms Laboratory[4] at the University of Illinois, a GA developed in C by the Kanpur Genetic Algorithms Laboratory[5] at the Indian Institute of Technology, the JGAP[6] developed in JAVA, and RMD_GA[7] and IDLGA[8] developed in IDL. However, none of them has a rich GA library like the Matlab (Mathworks, 2011) to solve linear or nonlinear constrained optimization problems. Therefore, the integrated GA functionalities within the flexible parameter adjustment options (e.g., population size, crossover probability, number of generations, etc.) in Matlab were chosen as an ideal development platform. Farzam *et al.* (2008) also successfully utilized the GA functionalities in the Matlab for abundance calculation. More specifically, Matlab provides a set of standard library functions for the GA with preparation of the problem to be solved, visualization of the optimization process, and reporting and saving of results (Farzam *et al.*, 2008; Bioucas-Dias, 2009; Mathworks, 2011). In addition, its integrated mathematical functions assist solving the optimization problems.

---

[4] "Single and Multiobjective Genetic Algorithm Toolbox in C++". Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois, Urbana-Champaign, USA. Accessed Jan 07, 2011: http://illigal.org/2007/06/05/single-and-multiobjective-genetic-algorithm-toolbox-in-c.

[5] "Single-objective GA code in C (for Windows and Linux)". Kanpur Genetic Algorithms Laboratory (KANGAL), Indian Institute of Technology, Kanpur, India. Accessed Jan 07, 2011: http://www.iitk.ac.in/kangal/codes.shtml.

[6] JGAP: Java Genetic Algorithms Package. Accessed Jan 07, 2011: http://jgap.sourceforge.net.

[7] Rob Dimeo's IDL Programs. Accessed Jan 07, 2011: http://www.ncnr.nist.gov/staff/dimeo/idl_programs.html.

[8] The IDL Genetic Algorithm page. Accessed Jan 07, 2011: http://www.xs4all.nl/~ajwwag.

4.3 Proposed GA

The concepts from the SGA, C-SOMGA, and NGA were taken into account to develop the proposed algorithm. The GA approach does not achieve global minimum unless its parameters, e.g., population size, number of generations, crossover probability, and mutation probability are set properly (Srinivas *et al.*, 1994; Chang, 2007). The adjustment of these parameters determines whether the algorithm will find a near-optimal solution efficiently (Zhang *et al.*, 2007). The GA was designed in the Matlab (Mathworks, 2011) platform as follows, which is illustrated in Figure 12.

1) Encode each chromosome as a double data type (real value).
2) Set the values of the following parameters (Tsoulos, 2009):
    a) Population size,
    b) Crossover rate,
    c) Elite count (i.e., the number of top ranking chromosomes that advances to the next generation without crossover and mutation),
    d) Number of generations, and
    e) Number of stall generation (i.e., the number of generation after which there is no change in the fitness value).
    f) Use the following steps to set the values of each parameter:
    g) Select a random pixel m_o from the scene,
    h) Start the iteration loop for a parameter, *S*, with its upper and lower limit using the GA (e.g., population size from 1 to 100, crossover rate from 0.01 to 1.0, number of generation and stall generation from 1 to 100, and elite count from 1 to 5),
    i) Record the SAM fitness score for each value of *S*,

j) End iteration loop, and

k) Use the value of *S* that achieves the best fitness score to unmix the entire scene.

3) Use the population size from the previous step and create a random initial population satisfying Eq. 28.

4) Start the iteration loop by minimizing the fitness function in Eq. 27 subject to Eq. 28.

5) Compute the fitness value of each chromosome using Eq. 27.

6) If one of the following termination criteria is satisfied, end iteration loop and provide the fittest chromosome as an optimal result, otherwise, go to the next step:

   a) the maximum generations specified by the user (i.e., 100) is reached,

   b) the fitness value of a chromosome is less than or equal to the fitness limit (i.e., 0), or

   c) the weighted average change of the fitness value of a chromosome over 80 stall generations is less than the function tolerance of 1e-6.

7) Rank the fitness values of each chromosome and select parent chromosomes for reproduction using stochastic uniform selection based on their fitness ranking.

8) Produce crossover children applying a scattered crossover rate of 50% to the selected parents (Mathworks, 2011).

9) Produce mutation children applying adaptive feasible mutation satisfying Eq. 28 for the remaining selected parents (Mathworks, 2011).

10) Replace the parent population by the children population and continue iteration loop to step 6 towards next generation.

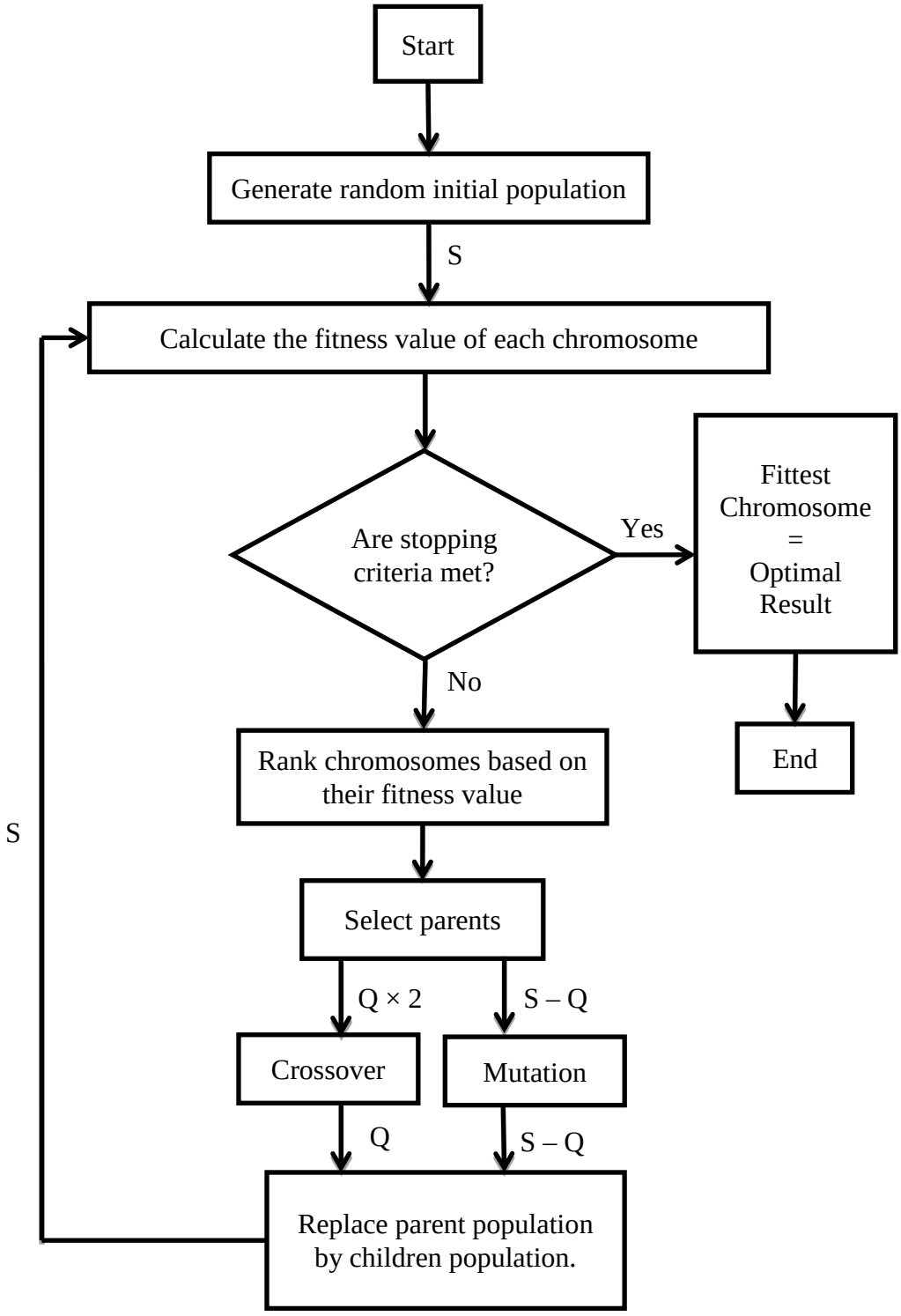


Figure 12: The GA scheme with population size S, number of crossover parents Q × 2, number of crossover children Q, and number of mutation parents or children S - Q.

4.4 Description of the GA Parameters

The parameters of the proposed GA and their working mechanisms are described in the following sections and illustrated in Figure 13.

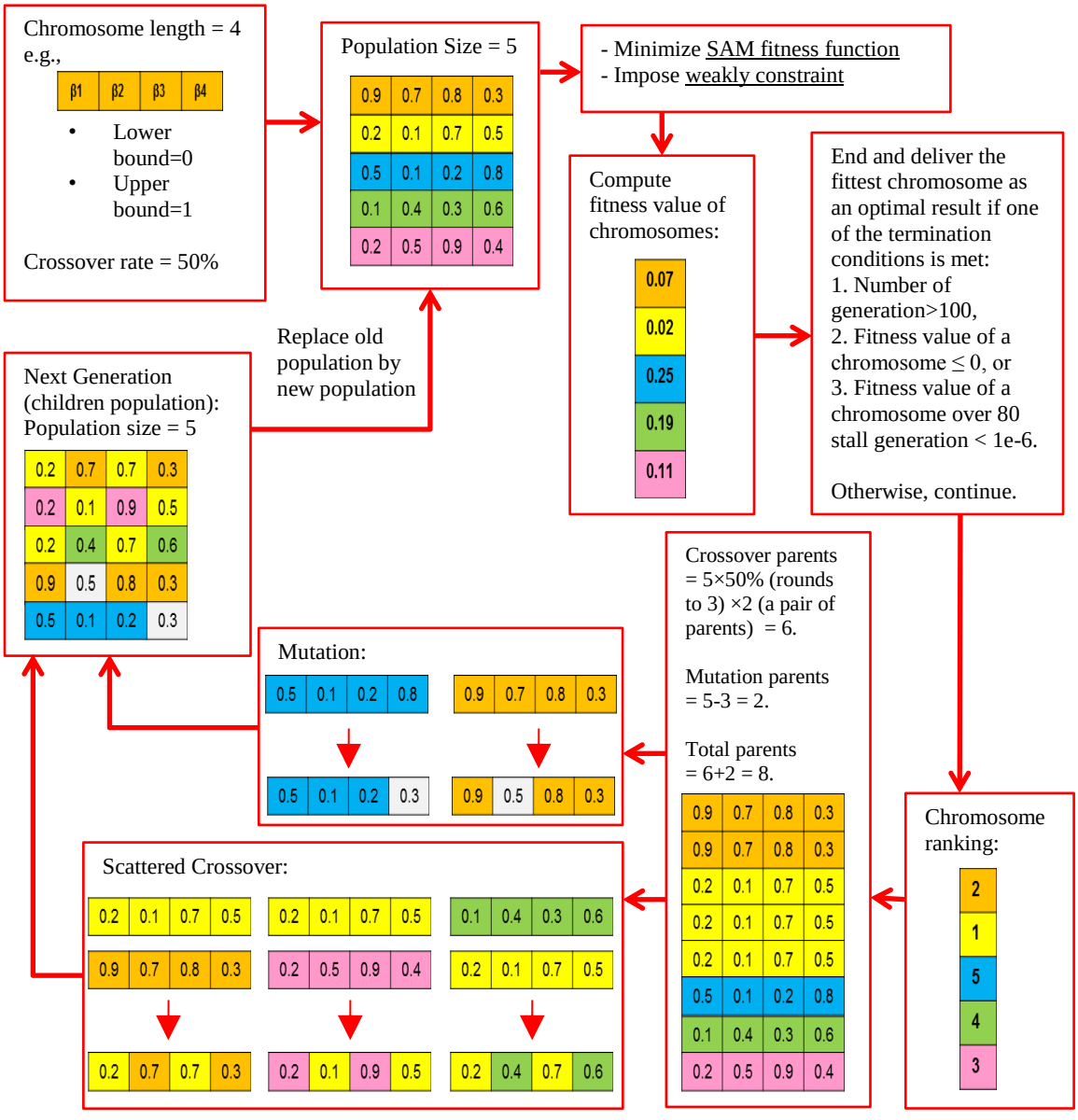


Figure 13: An example of the working mechanisms of the proposed GA.

4.4.1 Encoding

The values of Ω are arranged sequentially as genes in a chromosome. The size of each chromosome, *p,* is the same as the number of endmembers. In order to evaluate the efficiency and robustness of the GA, Bekiroğlu *et al.* (2009) compared four types of encoding: binary, value, quaternary, and octal. The authors concluded that a real value encoding is more suitable than other encoding approaches when the database size of genes decrease (smaller number of genes). Encoding plays an important role to keep the GA efficient and robust (Mathworks, 2011). As abundances are real values, value encoding is used in this study (Farzam *et al.*, 2008). The length of a chromosome is the same as the length of Ω in Eq. (15). For example, in Figure 13, a chromosome of size 5 with double data type or real values is represented as [0.9 0.7 0.8 0.3].

4.4.2 Fitness Value Computation

The GA minimizes a fitness function to solve the constrained optimization problem. The fitness function performs the quality assessment of a chromosome (Huang *et al.*, 2006). The fitter chromosomes with higher probability, based on their fitness value evaluation, are selected into the recombination pool for the reproduction (Mathworks, 2011). In this study, the GA finds the optimal values of Ω by minimizing the SAM fitness function in Eq. 27 imposing Eq. 28. For example, a chromosome, [0.2 0.1 0.7 0.5], as illustrated in Figure 13, satisfying the bound in Eq. 28 achieves the minimum fitness value of 0.02.

4.4.3 Population Size

A population size, S, represents the number of chromosomes in each generation. A large population size is more likely to obtain global minimum by searching the solution space more thoroughly (Mathworks, 2011). However, it might make the algorithm run very slow. A very small population size might be stuck into a local minimum. Therefore, the population size is adjusted in this study to keep the balance between time and optimal result. The adjustment of S is discussed in section 4.3. For example, population size of 5 is shown in Figure 13.

4.4.4 Initialization of the population

The population is initiated using an equally distributed random initial population satisfying the condition in Eq. 28 (Mathworks, 2011). For example, an initial population of 5 × 4 is generated with random values satisfying Eq. 28.

4.4.5 Chromosome Ranking

The fitness value of each chromosome is calculated using Eq. 27. It ranks the raw fitness values, R, of each chromosome as $1/\sqrt{R}$, an operational series of values (Mathworks, 2011). For example, in Figure 13, the fitness values of 5 chromosomes from top to bottom were 0.07, 0.02, 0.25, 0.19, and 0.11 respectively. After applying the fitness scaling rank function to these fitness values, the rank of the 5 chromosomes from top to bottom became 2, 1, 5, 4, and 3, respectively.

4.4.6 Parent Selection

The chromosomes from current generation, i.e., parents, are selected to contribute to the population of the next generation according to their ranking of the fitness value using a stochastic uniform selection (Mathworks, 2011). The uniform selection selects parents for crossover and mutation proportional to the top ranking chromosomes, where the top ranking chromosomes are copied multiple times. Crossover combines a pair of parents from the population based on the amount of crossover fraction. Mutation makes the random changes in a single parent, and these types of children are called mutation children. For example, in Figure 13, to produce 5×50% = 3 (rounded) crossover children with 50% crossover rate, the GA requires crossover parents, CP = 5×50%×2 = 6 (rounded), since 2 parents are required to produce each crossover child. The mutation parents, MP = 5-3 = 2, are required to produce 2 mutation children. The total number of parents required to be selected are CP+MP = 6+2 = 8 but the total number of chromosomes = 5. In this case, 8 parents are selected along a selection line proportional to the ranking of each chromosome (Mathworks, 2011). For that reason, the chromosomes with rank 1 was copied twice and rank 2 was copied once to make the total number of parents = 8 (Figure 13).

4.4.7 Scattered Crossover

Each pair of parent chromosome participates in crossover by exchanging each other's genes to produce children chromosomes for the next generation as illustrated in Figure 13. The scattered crossover is used in the GA, where a child chromosome is produced through a random binary vector that exchanges genes of each parent

chromosome, where 1 represents the genes from one parent and 0 represents the genes from another parent. For example, in Figure 13, a pair of parent chromosomes [**0.2** 0.1 **0.7** 0.5] and [0.9 **0.7** 0.8 **0.3**] exchange genes between them based on a random binary vector [1 0 1 0] and produces a child chromosome [**0.2 0.7 0.7 0.3**].

4.4.8 Adaptive Mutation

Random changes to the genes of the parent chromosomes, mutation, are applied to produce children satisfying Eq. 28, where the direction of mutation depends on the previous successful or failed generation (Mathworks, 2011). The mutation operation alters genes and promotes diversity among populations. For example, a parent chromosome [0.5 0.1 0. 2 0.8] in Figure 13 produces a mutation child [0.5 0.1 0. 2 **0.3**].

4.4.9 Termination Conditions

Children population from crossover and mutation replaces the parent population from the previous generation and goes through the fitness evaluation as shown in Figure 13. The algorithm terminates when one of the following criteria are met (Mathworks, 2011).

a) The algorithm stops when it reaches the number of generations defined by the user;
b) The algorithm stops when it reaches the time limit defined by the user;
c) The algorithm stops when the best fitness value in the current population is less than or equal to the fitness limit defined by the user;

d) The algorithm stops when the weighted average change in the fitness value over stall generation is less than the function tolerance;

e) The algorithm stops if there is no improvement in the fitness function during an interval of time, also called the stall time limit; and

f) The algorithm stops if there is no feasible solution found within the nonlinear constraint tolerance.

An example of these termination criteria is shown in Figure 13.

CHAPTER 5: VALIDATION

The accuracy assessment procedure of the unmixing results for the synthetic and real datasets is discussed in this chapter. A series of experiments was conducted for the synthetic datasets by applying the proposed GA-based method and two advanced approaches, the SAC and the NNSLO. The SAC approach was implemented in the Matlab platform using Eqs. 17, 18, and 19 (Rabah *et al*., 2011), while the NNSLO algorithm was executed from an open source Matlab-based toolbox called the Hyperspectral Image Analysis Toolbox (HIAT; Rosario-Torres *et al.*, 2007), which is, for example, used by NASA's Jet Propulsion Laboratory, the Canada Border Services Agency, and the US Army. The result of each unmixing method was compared against the true results in the case of the synthetic data and a classification map in the case of real data to assess the accuracy of estimated abundances.

5.1 Validation Process for Synthetic Data

The accuracy assessment between the true abundances (i.e., known abundances used to create the synthetic data) and the estimated abundances were performed using a statistical measure, the Index of Agreement (IA), which calculates the correspondence between the true and estimated values (Wilmott, 1982; Champagne, 2001).

The IA was calculated using the following formula:

$$IA = 1 - \left[\sum_{h=1}^{t} \sum_{i=1}^{p} (w_{ih} - \alpha_{ih})^2 \Big/ \sum_{h=1}^{t} \sum_{i=1}^{p} (\left|u_{ih}\right| - \left|d_{ih}\right|)^2 \right], \tag{29}$$

where t is the total number of pixels, w_{ih} and α_{ih} are respectively the true and estimated abundances for the i^{th} endmember of the h^{th} pixel, u_{ih} is the difference between w_{ih} and the mean of the true abundance over all pixels for the i^{th} endmember of the h^{th} pixel, and d_{ih} is the difference between α_{ih} and mean of the estimated abundance over all pixels for the i^{th} endmember of the h^{th} pixel. The IA is measured on a scale between 0 (0% agreement) and 1 (100% agreement).

In addition, the CORrelation (COR) and Root Mean Square Error (RMSE) were also used to measure the accuracy (He *et al.*, 2008). They were calculated as follows:

$$COR = \sum_{h=1}^{t} \sum_{i=1}^{p} \left(w_{ih} \times \alpha_{ih} \right) \Big/ \sqrt{\sum_{h=1}^{t} \sum_{i=1}^{p} \left(w_{ih} \right)^2} \sqrt{\sum_{h=1}^{t} \sum_{i=1}^{p} \left(\alpha_{ih} \right)^2} \tag{30}$$

and

$$RMSE = \sqrt{\sum_{h=1}^{t} \sum_{i=1}^{p} \left(w_{ih} - \alpha_{ih} \right)^2 \Big/ p} \, . \tag{31}$$

In order to investigate the impact of illumination fluctuation, noise, and spectral signature variability on each unmixing method, 12 synthetic data cubes were generated with a combination of SNR (90, 60, 30, and15 dB), spectral signature variability (0%, 5%, and 10%) and random uniform illumination fluctuation (between 0 and 1.28) as discussed in chapter 3. The entire validation process is illustrated in Figure 14.

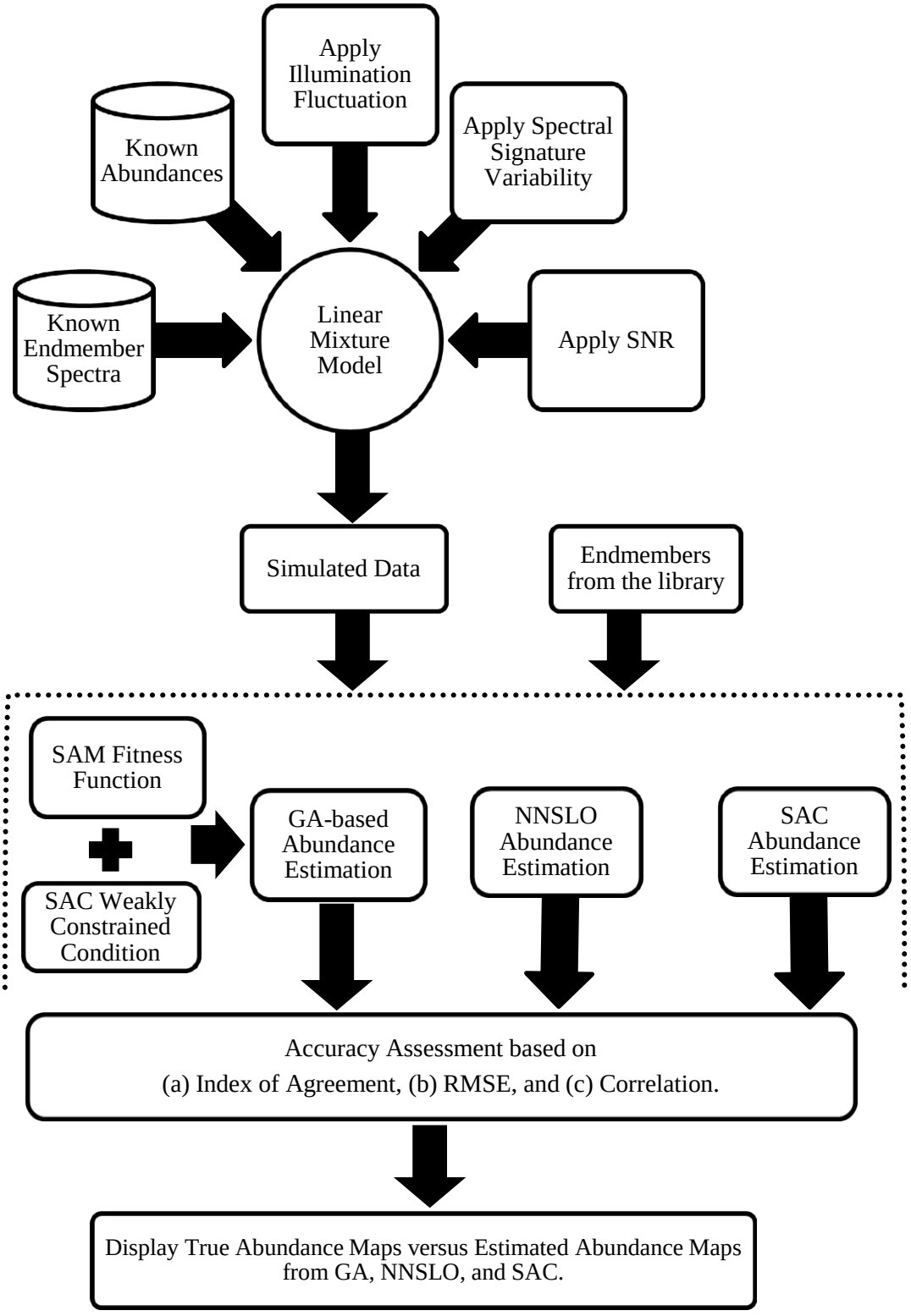


Figure 14: The process of synthetic data generation, abundance estimation, comparison, and validation.

5.2 Validation Process for Real Data

The ground reference abundances were not available to verify those estimated ones of the Cuprite dataset. For that reason, the Cuprite mineral classification map produced by the USGS Tricorder 3.3 (Clark *et al.*, 1996) shown in Figure 15 was used as a reference to evaluate the unmixing result derived classification map. The unmixing results of each unmixing methods were converted to a classification map. Each classification map was produced based on the dominant abundance in each pixel, where the dominant abundance was assigned to a class (Shrestha *et al.*, 2005; Heldens *et al.*, 2009). No threshold (e.g., abundance > 50%) was used in this case as each unmixing method might have a different value for the dominant abundance.

The USGS Cuprite mineral map was registered with the AVIRIS Cuprite image (Figure 9) in the ESRI[9] ArcMap software platform. It was registered by selecting 12 Ground Control Points (GCP) using the First-Order Polynomial as shown in Figure 16, where the total RMSE was 1.72 or < 2 pixels because of lack of proper GCP (e.g., road intersection) in the classification map. The Nearest Neighbour resampling method was used to resize the USGS Cuprite classification map to the extent (400 pixels × 350 pixels) of the AVIRIS Cuprite image. The Region of Interest (ROI) polygons were created for eight endmembers out of nine used for the unmixing that are present in the USGS Cuprite classification map. These are Muscovite-medlow-Al, Muscovite-medhi-Al, Montmorillonite-Na, Kaolinite KGa-2 pxl, Kaolinite KGa-1 wxl, Chlorite+Muscovite,

[9] ESRI – ArcGIS for Desktop – Advanced Desktop GIS Mapping / GIS Editing Software. Accessed Oct 07, 2012: http://www.esri.com/software/arcgis/arcgis-for-desktop

Chalcedony, and Alunite GDS82. No ROI was selected for the Hematite.02+Quartz.98 endmember as no ground reference data were available for it. This endmember was selected to test the influence of an unknown endmember in unmixing results.

The USGS Cuprite mineral map from the Red Green Blue (RGB) colour BitMaP (BMP) file format was converted to a gray scale image using a Matlab function, where each gray scale value corresponds to each class. In the ENVI software, the gray scale values of eight endmember classes were converted to eight ROIs (Figure 17). The Alunite ROI was created by combining all Alunite classes in the USGS mineral map, because only one Alunite endmember was used for unmixing the scene, where the multiple Alunite endmembers were present in the USGS classification map (Figure 15).

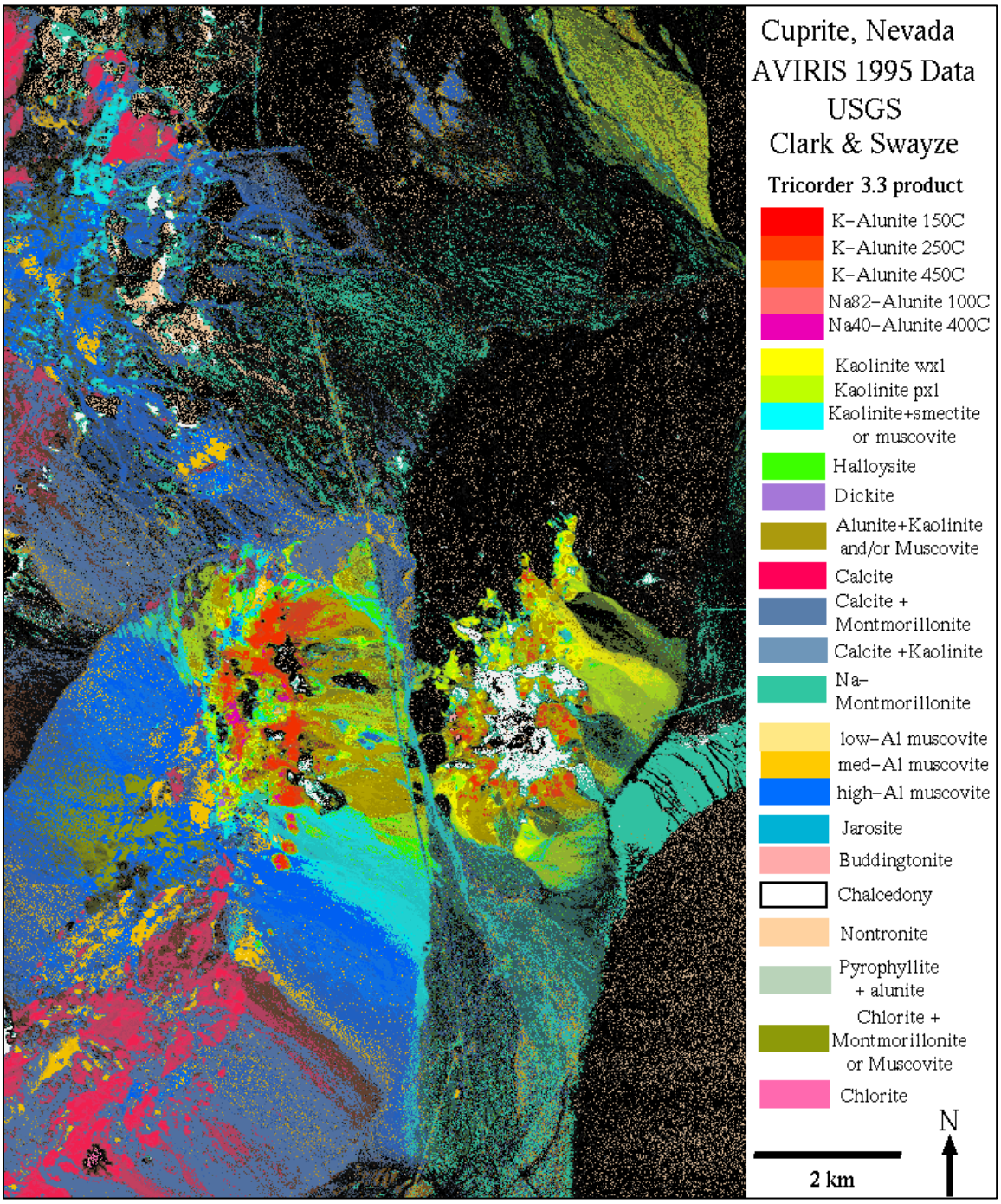


Figure 15: The USGS Tricorder 3.3 classification map of Cuprite minerals taken from Clark et al. (1996).

Link	X Source	Y Source	X Map	Y Map	Residual
1	517.139846	-630.150967	373.698491	111.605403	1.74690
2	313.889865	-506.494778	172.956635	228.312509	1.41703
3	288.922351	-644.043428	148.349058	95.552259	0.75672
4	457.180753	-673.081320	314.906813	70.893581	1.35304
5	210.214843	-413.557986	74.751749	317.819932	2.95149
6	455.407995	-405.473221	311.878736	331.946621	2.16628
7	459.723244	-411.553304	316.952565	326.071661	1.39133
8	335.021622	-635.526640	194.582600	103.952551	1.16048
9	343.958271	-668.473223	203.495181	73.083976	1.65853
10	182.776889	-689.681337	41.232378	48.703159	1.65019
11	206.998360	-705.735310	64.545288	33.481672	2.09955
12	237.954413	-623.756917	97.483859	112.526589	1.16167

Auto Adjust Transformation: 1st Order Polynomial (A Total RMS Error: 1.71710

Figure 16: Ground control points selected and associated individual registration errors (Residual) and total error (RMSE) to match the Cuprite mineral map to AVIRIS data (screenshot captured from the ESRI ArcMap software).

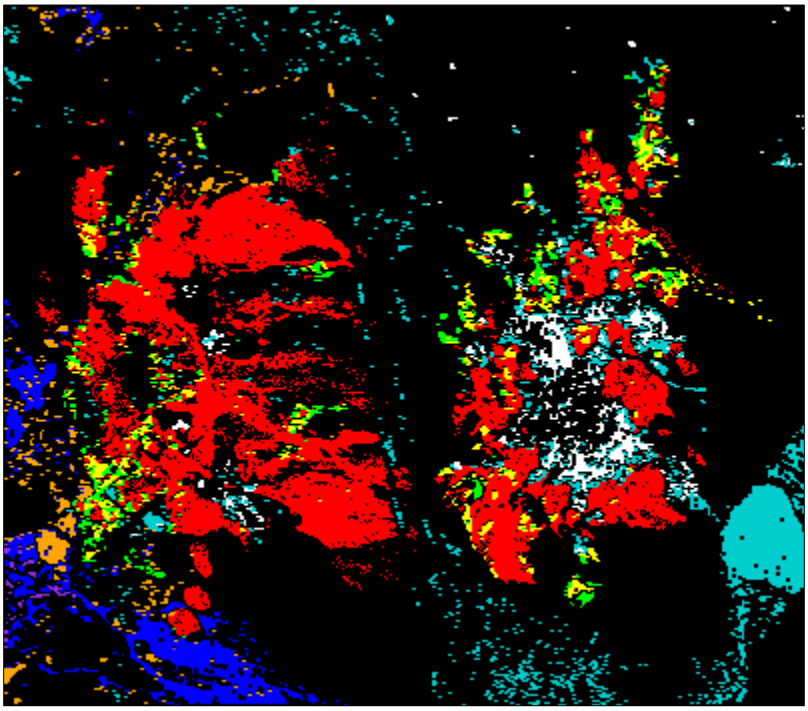


Figure 17: Eight ROI polygons: Muscovite-medhi-Al, Muscovite-medlow-Al, Montmorillonite-Na, Kaolinite pxl, Kaolinite wxl, Chlorite+Muscovite, Chalcedony, and Alunite derived from the USGS Cuprite classification map shown in Figure 15.

The accuracy of the classification map derived from the estimated abundances was calculated using an error matrix (Congalton, 1991; Mallinis *et al*., 2012; Im *et al*., 2012), where the ROI polygons were used as the ground reference. The error matrix includes overall accuracy, user accuracy (commission error), producer accuracy (omission error), and Kappa coefficient. User accuracy is the percentage of endmember pixels in the class that matches with the ROI polygon. Producer accuracy is the percentage of endmember pixels in the ROI polygon that matches with the class. The commission error occurs when endmember pixels assigned to a particular class actually belong to other classes, while the omission error represents the percentage of endmember pixels that are absent in the class but present in the ROI polygon. In addition, the Kappa coefficient provides better comparison of multiple classification maps with error matrices to check if the accuracy is better than other results (Cohen, 1960; Congalton, 1991). Furthermore, overall accuracy calculates the percentage of all pixels in the class that are classified correctly. It is calculated by dividing the total number of correctly and incorrectly classified endmember pixels in all classes by the total number of correctly classified endmember pixels in all classes.

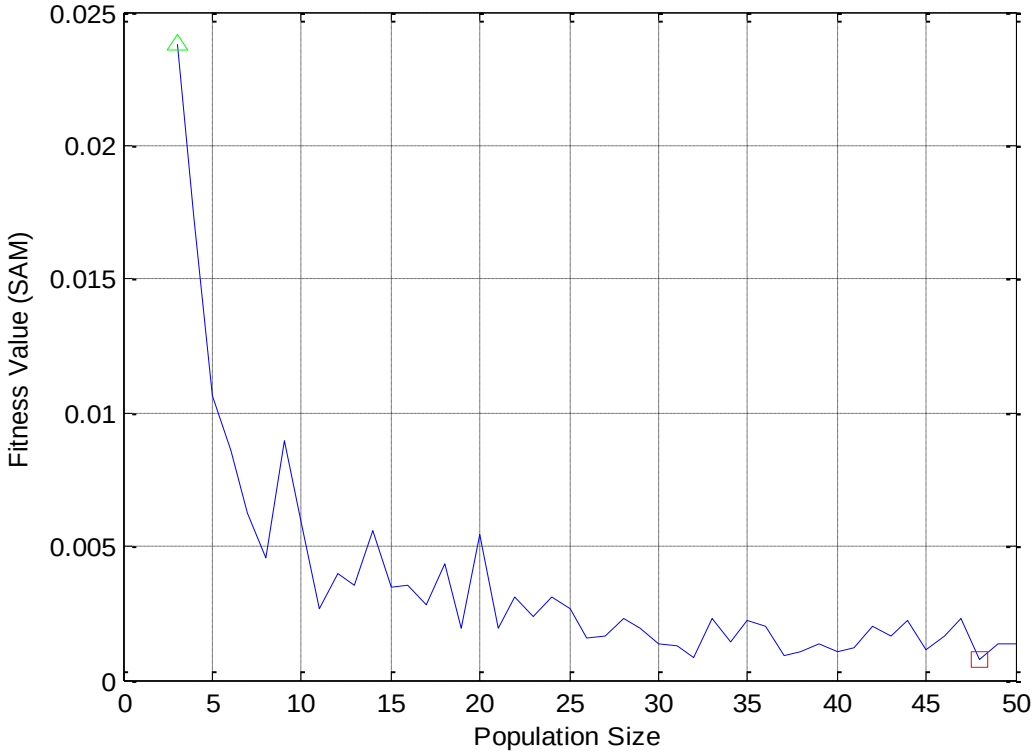
In addition, the estimated abundance maps were visually compared against the USGS Cuprite mineral map shown in Figure 15.

CHAPTER 6: RESULTS AND DISCUSSION

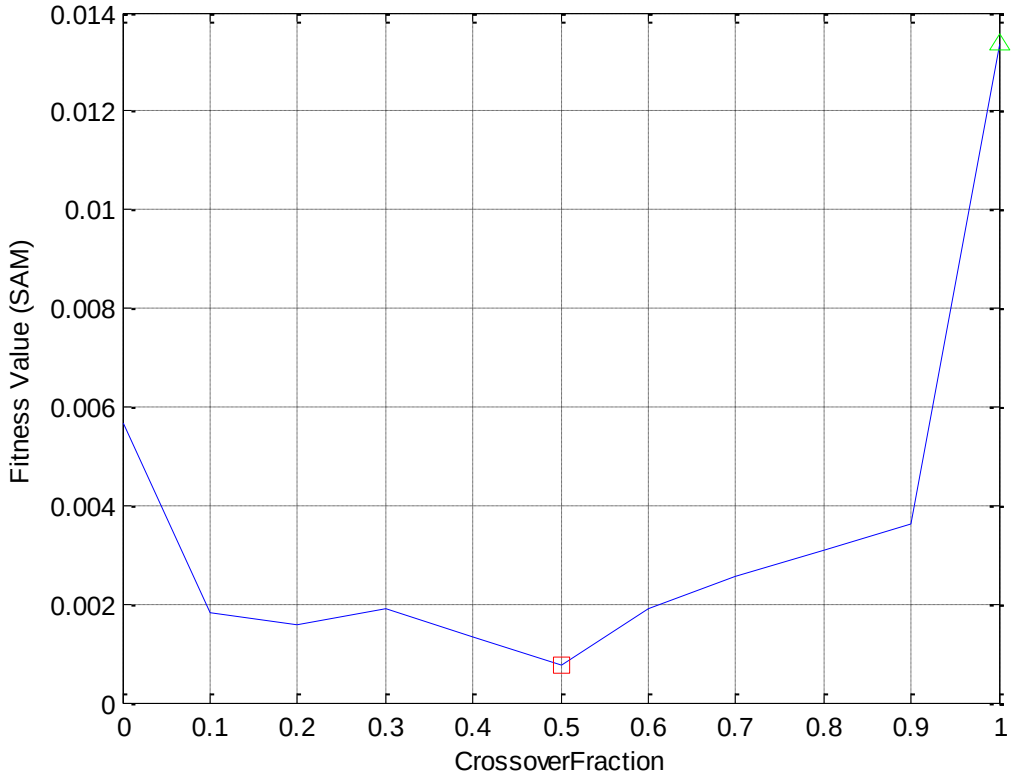
In this chapter, the comparison and interpretation of the GA, SAC, and NNSLO estimated abundance maps of synthetic and real datasets are discussed.

6.1 Unmixing Results of Synthetic Datasets

As discussed in Chapter 3, the synthetic datasets were created using a linear mixture with fully constraint condition, where the true abundances are known. To estimate the abundances of each synthetic data, the proposed GA was adjusted with population size = 48 (Figure 18a), crossover rate = 0.5 (Figure 18b), elite count = 0, number of generations = 100, and stall generation = 80.



(a)



(b)

Figure 18: A population size = 48 (top) and crossover rate = 0.5 (bottom) shown in red rectangle achieved the minimum fitness value of a random pixel from synthetic data cube.

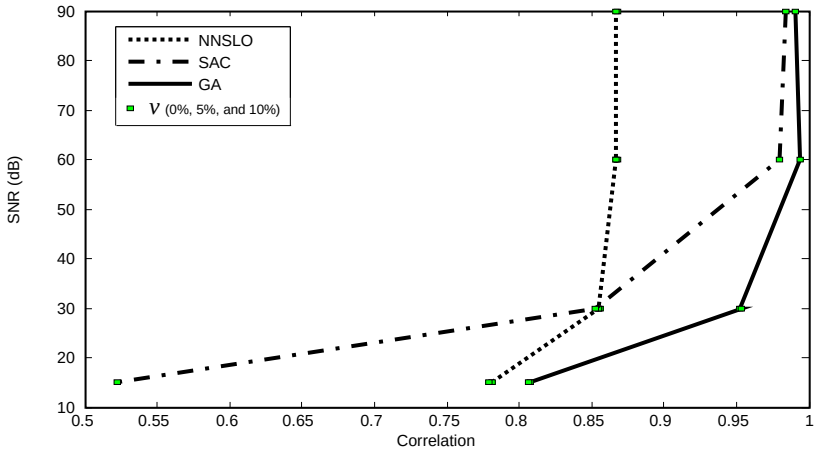
The validation results shown in Table 1 represent RMSE, COR, and IA of the GA, SAC, and NNSLO estimated abundances of a synthetic data in particular SNR, signature variability, and illumination fluctuation.

*Table 1: Validation results of the GA, SAC, and NNSLO with **SNR**, **spectral signature variability**, and random uniform **illumination fluctuation** between 0 and 1.28.*

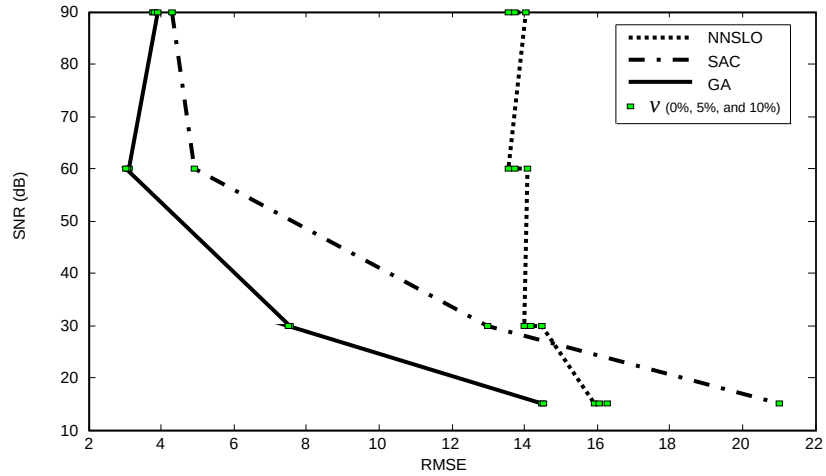
SNR (dB)	Sig. var. (%)	GA RMSE	SAC RMSE	NNSLO RMSE	GA COR	SAC COR	NNSLO COR	GA IA	SAC IA	NNSLO IA
90	10	3.7973	4.2858	13.5486	0.9908	0.9846	0.8674	0.9684	0.9598	0.5979
90	5	3.8249	4.2858	13.7361	0.9907	0.9846	0.8680	0.9680	0.9598	0.5867
90	0	3.9053	4.2858	14.0566	0.9902	0.9846	0.8669	0.9666	0.9598	0.5672
60	10	3.0972	4.9360	13.5607	0.9935	0.9794	0.8671	0.9790	0.9466	0.5972
60	5	3.0641	4.9360	13.7480	0.9937	0.9794	0.8678	0.9794	0.9466	0.5860
60	0	3.0453	4.9360	14.0682	0.9938	0.9794	0.8666	0.9797	0.9466	0.5665
30	10	7.5386	12.9870	13.9950	0.9527	0.8527	0.8549	0.8755	0.6306	0.5710
30	5	7.5282	12.9870	14.1738	0.9529	0.8527	0.8558	0.8759	0.6306	0.5599
30	0	7.5245	12.9870	14.4816	0.9529	0.8527	0.8549	0.8760	0.6306	0.5406
15	10	14.4758	21.0239	15.9081	0.8076	0.5225	0.7809	0.5410	0.0318	0.4457
15	5	14.5290	21.0239	16.0477	0.8061	0.5225	0.7814	0.5376	0.0318	0.4359
15	0	14.5161	21.0239	16.2864	0.8065	0.5225	0.7795	0.5384	0.0318	0.4190

The validation results from Table 1 are interpreted in Figure 19. In case of both GA and SAC, the COR, RMSE, and IA results were similar for various levels of signature variability such as 0%, 5%, and 10%; however, RMSE and IA results from the NNSLO were different. For all noise levels, the GA has higher COR and IA and lower RMSE with respect to the true abundances than the SAC and the NNSLO. The COR, RMSE, and IA differences between the GA and the SAC increased for noise levels below an SNR of 60 dB. However, the difference of these evaluation measures between the GA

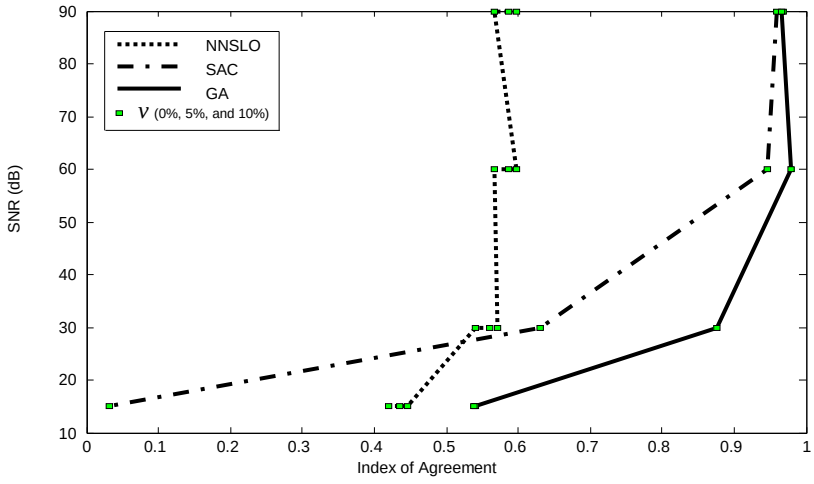
and the NNSLO decreased significantly for noise levels below an SNR of 30 dB. The GA achieved overall the best result for an SNR of 60 dB, instead of SNR = 90 dB. In case of SNR = 90 dB, illumination fluctuation and signature variability dominated the linear mixture model in Eq. 22. It is clear from the results that the SAC is very sensitive to noise and the NNSLO is affected by the illumination fluctuation and signature variability. High-noise levels affect all three unmixing approaches of which the GA achieved the best results. Although both GA and SAC are less sensitive to illumination fluctuation and signature variability in case of low to moderate noise, the GA improves the accuracy significantly compared to the SAC in case of high noise. Therefore, it can be concluded from Table 1 that the GA has the lowest average RMSE of 7.2372 compared to the SAC average RMSE of 10.8082 and NNSLO average RMSE of 14.4676, the highest average COR of 0.9360 compared to the SAC average COR of 0.8348 and NNSLO average COR of 0.8426, and the highest average IA of 0.8405 compared to SAC average IA of 0.6422 and NNSLO average IA of 0.5395. In this controlled experiment, considering most of the possible real-world scenarios with amplitude errors and noise, the GA-based unmixing approach indicates a significant improvement of abundance estimation accuracy compared to the SAC and NNSLO approaches based on an improvement of the average IA by 19.83% and 30.10%, respectively, an improvement of the average COR by 10.12% and 9.34%, respectively, and an improvement of the average RMSE by 3.57 and 7.23, respectively.



(a)



(b)



(c)

Figure 19: Comparison of validation results of the GA, SAC, and NNSLO based on the true abundances for various SNR levels as shown in (a) Correlation versus SNR, (b) RMSE versus SNR, and (c) Index of Agreement versus SNR. The signature variability of 0%, 5%, and 10% were used for each SNR level, respectively.

The signature variablity of 0%, 5%, and 10% have almost no influence in the validation results of unmixing approaches except the NNSLO. Therefore, a comparison of estimated mineral abundances of four cases out of twelve in Table 1 with 5% signature variability are shown in Figures 20, 21, 22, and 23. The scatterplot results in these Figures show that the estimated abundances of GA has a better linear relationship (positive correlation) with true abundances in most of the cases compared to the SAC and the NNSLO. The outliers in the GA scatterplot (e.g., Figure 20b) represents underestimated abundances that could not be modeled by the SAM fitness function with a weakly constraint condition as the algorithm terminates after a certain number of generation. They occurred due to the very high illumination fluctuation and signature variability; in addition, the outliers in high to very high noise (e.g., Figures 22b and 23b) occurred due to high noise. The SAC estimated abundaces for moderate noise level (SNR of 60dB in Figure 21c) overestimated and underestimated abundances between 0 and 1 because of imposing fully constrained condition. In addition, the SAC seems to underestimate abundances for the high noise level (SNR of 30 dB in Figure 22c) and shows no correlation with the true abundances for very high noise level (SNR of 15 Figure 23c). The NNSLO scatterplots underestimated abundances because of its sensitivity to illumination fluctuation and signature variability applied with all noise levels (SNR of 90, 60, 30, and 15 dB in Figure 20d, 21d, 22d, 23d, respectively). In the case of higher noise levels, SNR=30 and 15 dB, the GA estimated abundances (Figures 22b and 23b) still showed a linear relationship with true abundances compared to the SAC (Figures 22c and 23c) and the NNSLO (Figures 22d and 23d), where the NNSLO

estimated abundances show better agreement with the true abundances compared to those retrieved using the SAC.

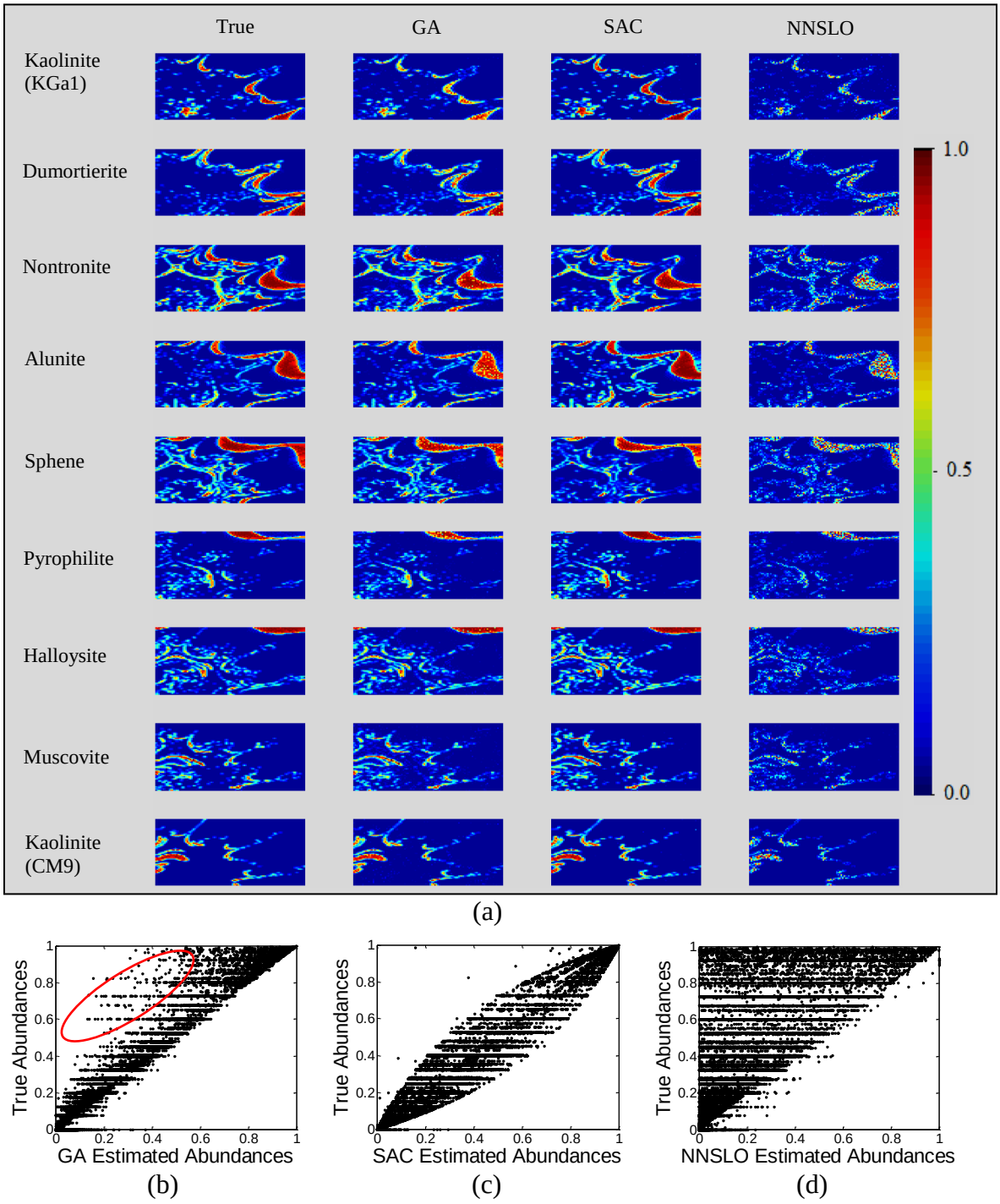


Figure 20: (a) Mineral abundance maps generated using the GA, SAC, and NNSLO unmixing methods for nine mineral endmembers with an ***SNR level of 90 dB (low noise), signature variability of 5%*** *and random uniform illumination fluctuation between 0 and 1.28 are compared against the true result. Three scatter plots of true versus estimated abundances of all endmembers of each unmixing method are shown in (b), (c), and (d). An example of outliers (circled in red) is shown in (b).*

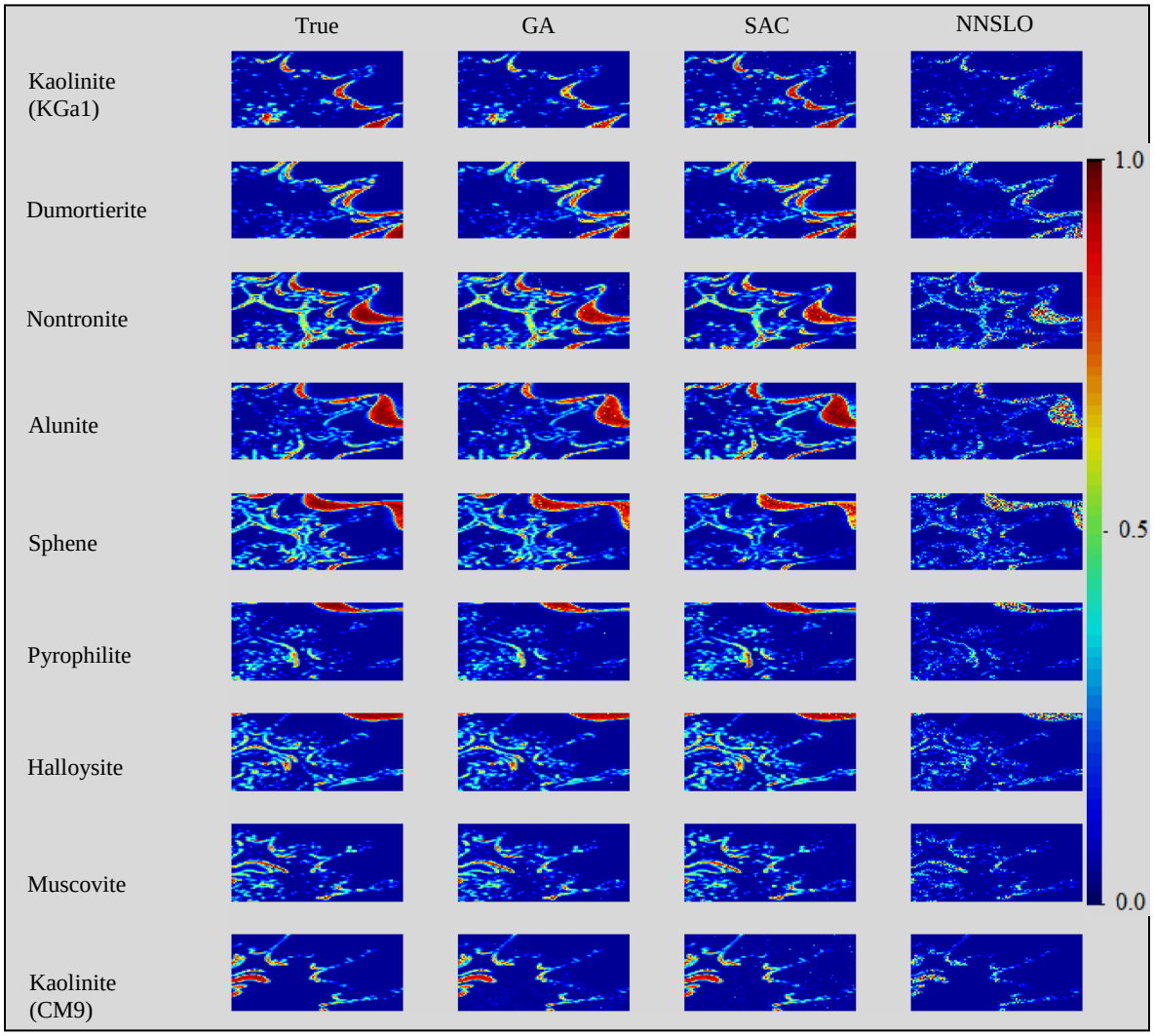


Figure 21: (a) Mineral abundance maps generated using the GA, SAC, and NNSLO unmixing methods for nine mineral endmembers with an ***SNR of 60 dB (moderate noise), signature variability of 5%*** *and random uniform illumination fluctuation between 0 and 1.28 are compared against the true result. Three scatter plots of true versus estimated abundances of all endmembers of each unmixing method are shown in (b), (c), and (d).*

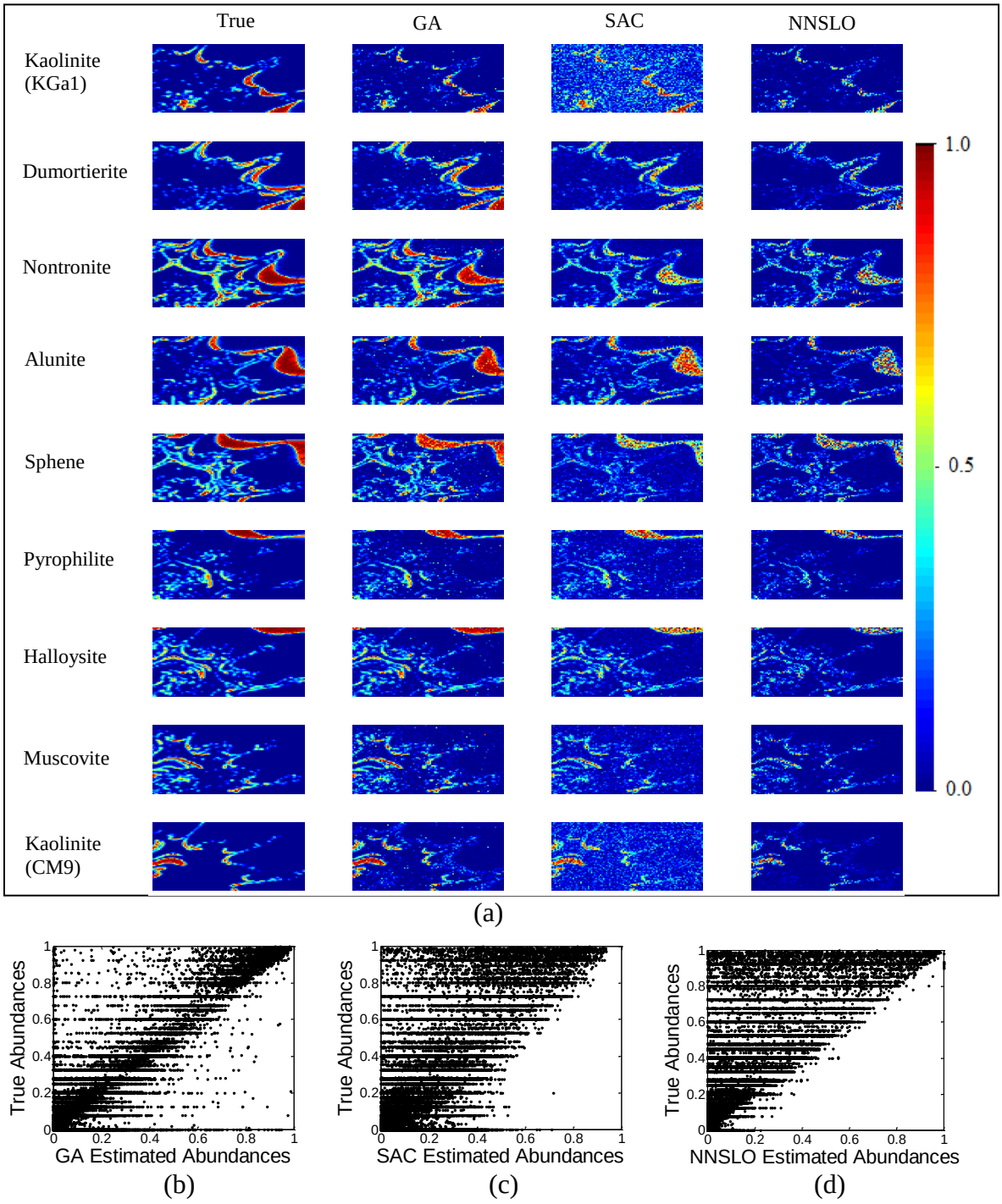


Figure 22: (a) Mineral abundance maps generated using the GA, SAC, and NNSLO unmixing methods for nine mineral endmembers with an ***SNR of 30 dB (high noise), signature variability of 5%*** *and random uniform illumination fluctuation between 0 and 1.28 are compared against the true result. Three scatter plots of true versus estimated abundances of all endmembers of each unmixing method are shown in (b), (c), and (d).*

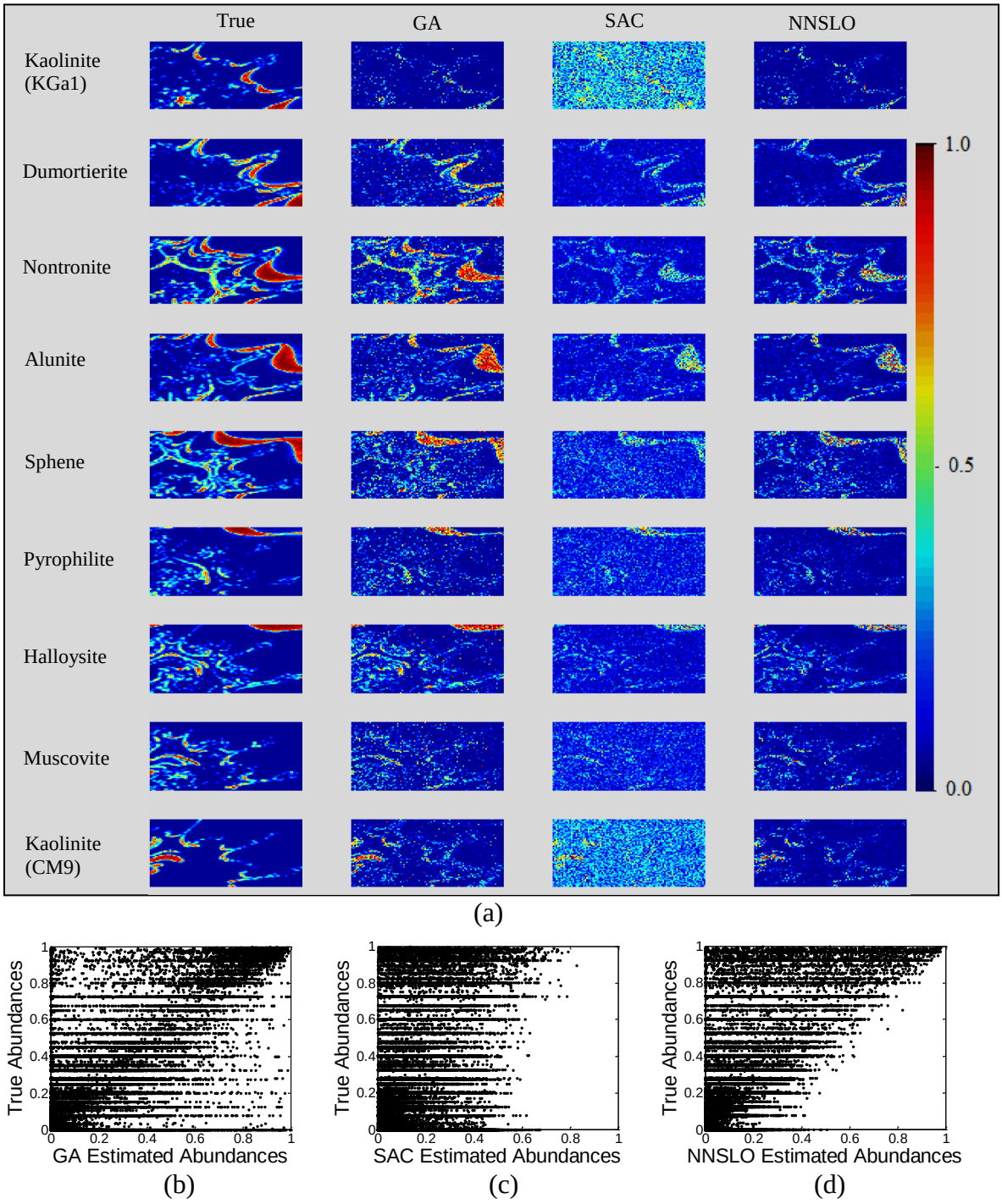


Figure 23: (a) Mineral abundance maps generated using the GA, SAC, and NNSLO unmixing methods for nine mineral endmembers with an ***SNR of 15 dB (very high noise), signature variability of 5%*** *and random uniform illumination fluctuation between 0 and 1.28 are compared against the true result. Three scatter plots of true versus estimated abundances of all endmembers of each unmixing method are shown in (b), (c), and (d).*

6.1.1 Accuracy versus Time and Population Size of the GA

A GA-based unmixing experiment was conducted using a (10 pixels × 10 pixels × 221 bands) synthetic data cube with an SNR of 30dB, signature variability of 5%, and random uniform illumination fluctuation between 1 and 1.28. The purpose was to observe the dependency of the GA-based abundance estimation accuracy on the most important GA parameter, the population size. It is clear from the three accuracy measures (IA, COR and RMSE) in Figure 24 that after a particular population size (i.e., population size = 10) there is very little improvement in the accuracy. The population size also affects the computation time. A linear relationship between population size and CPU (Intel® Core™ i5 processor at 3.46 GHz) time is shown in Figure 24d, where it takes more time to produce unmixing result if the population size increases. Therefore, it is ideal to set the population size in such a way so that the GA produces a near-optimal result within a reasonable computation time. For example, setting the population size to10 requires 20 seconds to unmix 100 pixels, where a better validation result is achieved with an approximate IA of 0.83, COR of 0.95, and RMSE of 0.77.

Furthermore, the processing time of the GA-based unmixing method increases with the increment of number of pixels. In addition, the increment of number of endmembers and number of spectral bands usually requires the increment of population size, which increases processing time.

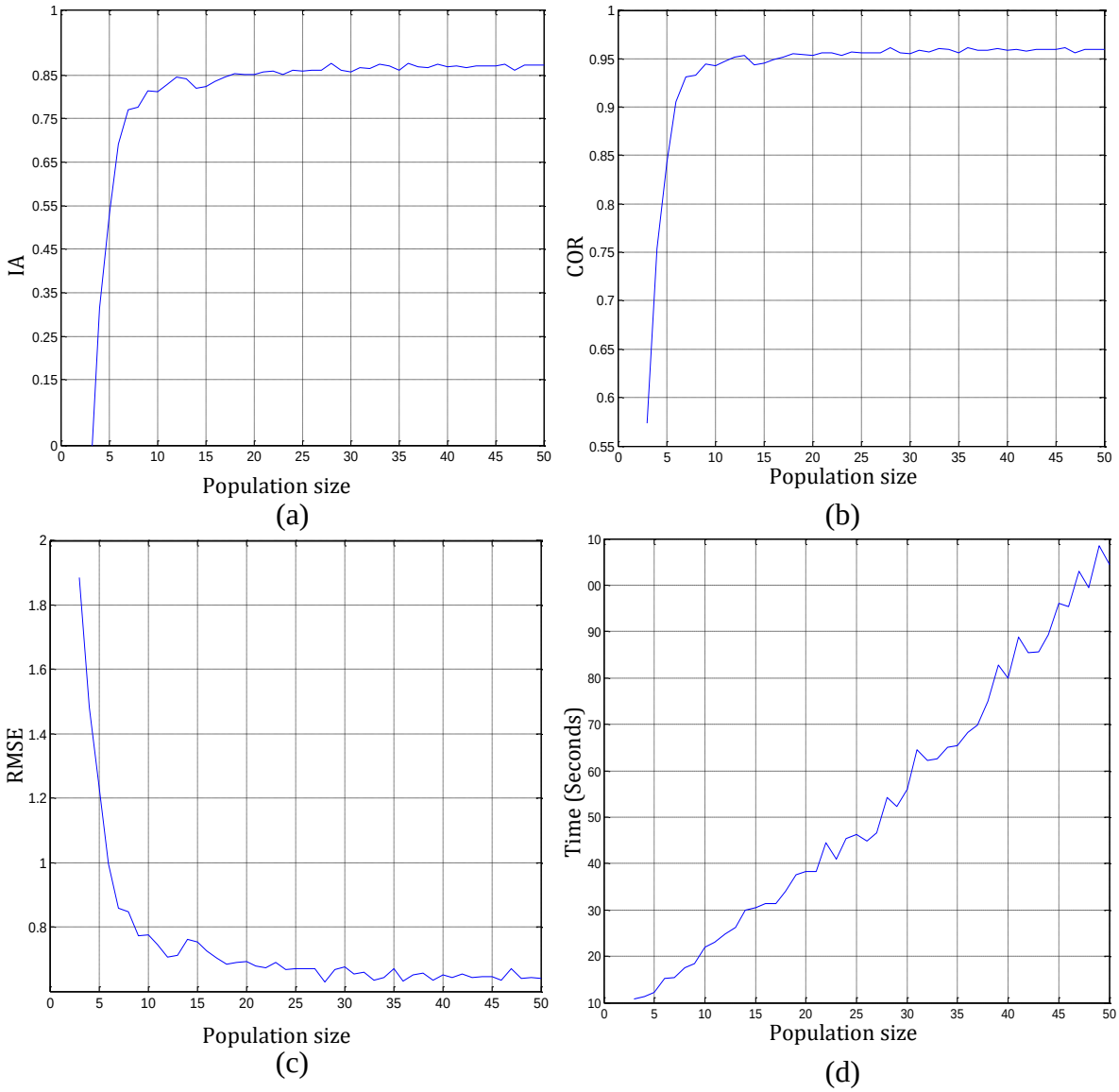


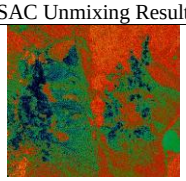
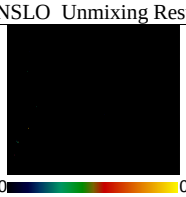
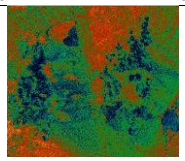
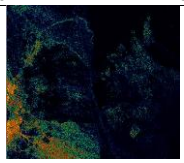
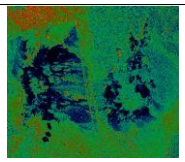
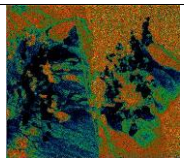
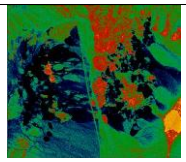
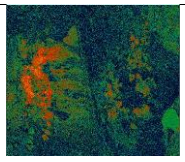
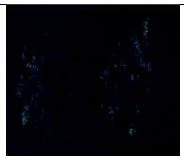
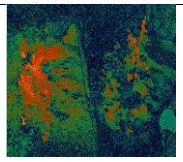
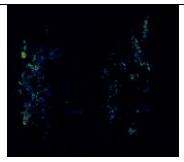
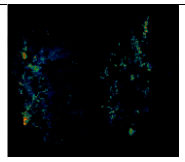
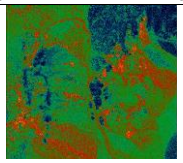
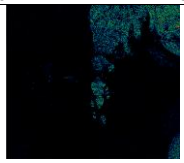
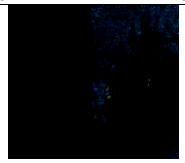
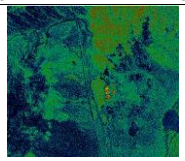
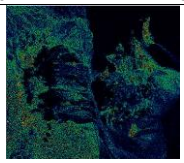
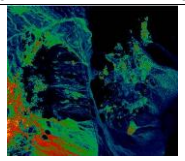
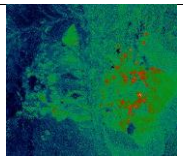
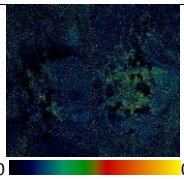
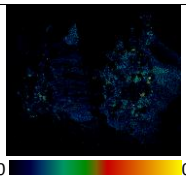
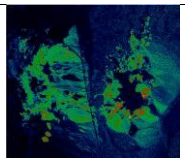
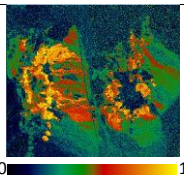
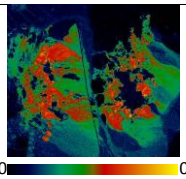
Figure 24: Population size is plotted against (a) IA, (b) COR, (c) RMSE, and (d) CPU time.

6.2 Unmixing Results of Real Data

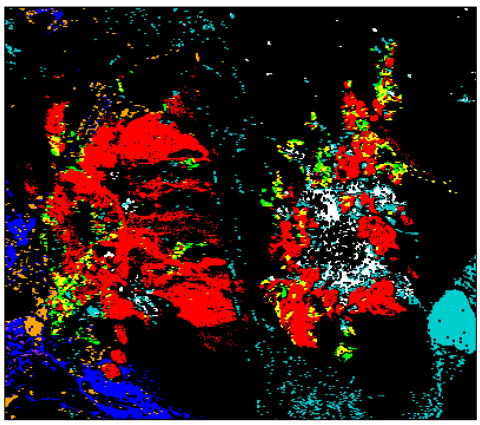
The GA-based unmixing was run on the Cuprite data with a set of adjusted parameters: population size =17, elite count = 0, crossover fraction = 0.10, number of generations = 91, and stall generation = 11. In addition, the SAC and the NNSLO unmixing methods were also applied to the Cuprite data. The unmixing result of each

method with abundance range is shown in Table 2 and was compared visually and quantitatively. In case of the SAC, the maximum abundances of all endmembers except Montmorillonite-Na are less than 50%. On the other hand, the maximum abundances of all endmembers calculated by the GA are greater than or equal to 50%. The NNSLO calculated the maximum abundances greater than 50% for Muscovite-medhi-Al, Montmorillonite-Na, Chlorite+Muscovite, and Alunite GDS82 endmemers only. If a classification map is produced based on the dominant abundances greater than 50% of each endmember, the GA will classify more endmember classes compared to the SAC and NNSLO. In order to keep the GA, SAC, and NNSLO comparison process unbiased, no threshold was used with the dominant abundances to produce the classification maps (discussed in the section 5.2). The abundance maps in Table 2 can be visually compared with the ground reference map in Figure 25a. The GA abundance maps of the Muscovite-medhi-Al, Kaolinite KGa-2 pxl, Kaolinite KGa-1 wxl, Chalcedony, and Alunite GDS82 seem to have better match with the ground reference map compared to the SAC and the NNSLO. An example of the visual comparison is discussed in the next paragraph.

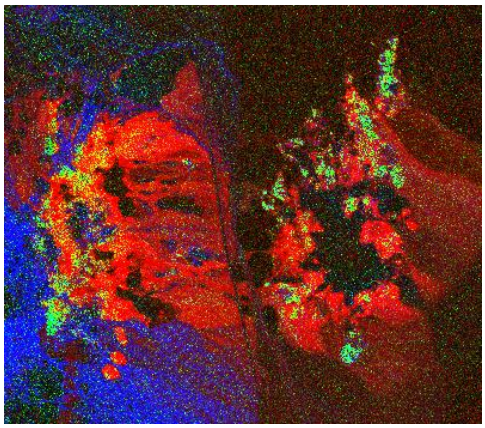
Table 2: Visual comparison of the GA, SAC, and NNSLO estimated abundance maps of nine mineral endmembers (no stretching was applied to the unmixing results).

	SAC Unmixing Result	GA Unmixing Result	NNSLO Unmixing Result
Muscovite-medlow-Al	 0.00 0.32	 0.00 0.73	 0.00 0.13
Muscovite-medhi-Al	 0.00 0.35	 0.00 0.91	 0.00 0.53
Montmorillonite-Na	 0.00 0.59	 0.00 0.96	 0.00 0.96
Kaolinite KGa-2 pxl	 0.00 0.40	 0.00 0.52	 0.00 0.26
Kaolinite KGa-1 wxl	 0.00 0.43	 0.00 0.50	 0.00 0.28
Hematite .02+Quartz .98	 0.00 0.32	 0.00 0.69	 0.00 0.29
Chlorite+Muscovite	 0.00 0.15	 0.00 0.71	 0.00 0.78
Chalcedony	 0.00 0.31	 0.00 0.59	 0.00 0.48
Alunite GDS82	 0.00 0.39	 0.00 1.00	 0.00 0.71

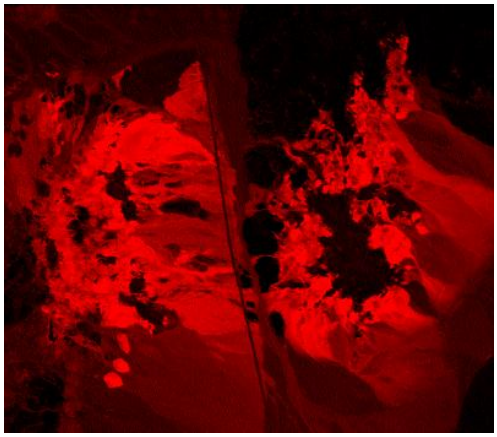
In Figure 25, the visual comparison indicates that the GA unmixing results better agrees with the ground reference. Linear 2% stretching was applied to improve the display quality in Figures 25b, c, and d. It also shows that the NNSLO calculated almost no abundances for the Muscovite-medhi-Al and Kaolinite pxl due to the similar spectral shape between Muscovite-medhi-Al and Chlorite+Muscovite as well as Kaolinite pxl and Kaolinite wxl (spectral signatures shown in Figure 10). The SAC seem to overcalculate or undercalculate abundances of Muscovite-medhi-Al, Kaolinite pxl, and Kaolinite wxl because of imposing fully constrained condition.



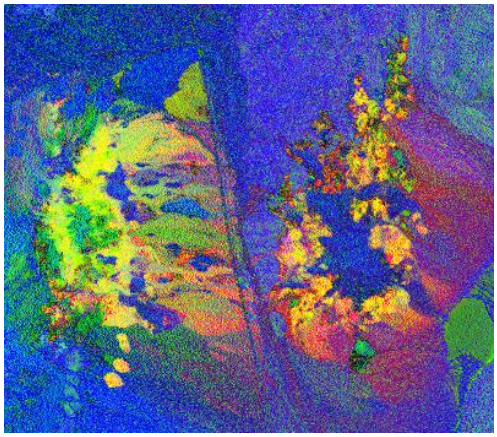
(a) Ground reference



(b) GA



(c) NNSLO



(d) SAC

Figure 25: Comparison of unmixing results: (a) the ground reference map derived from the USGS Tricorder 3.3 classification, (b), (c), and (d) show the GA, NNSLO, and SAC results. The RGB composite represents abundance maps in Red: Alunite, Green: Kaolinite KGa2 pxl, and Blue: Muscovite-medhi-Al. (Legend of (a): Muscovite-medhi-Al, Muscovite-medlow-Al, Montmorillonite-Na, Kaolinite pxl, Kaolinite wxl, Chlorite+Muscovite, Chalcedony, and Alunite).

A classification map was produced for each unmixing method based on dominant abundances. The Hematite .02+Quartz .98 class for each classification map was kept unclassified, as no ground reference ROI was available for it. These classification maps were validated using error matrices based on the ground reference. The error matrices of the GA, NNSLO, and SAC unmixing approaches are shown in Tables 3, 4, and 5, respectively.

Table 3: The error matrix of the GA classification map.

GA: Ground Reference (Pixels)

Overall Accuracy = (24662/36077) 68.3593%
Kappa Coefficient = 0.5191

Class	Kaol. KGa-2 pxl	Kaol. KGa-1 wxl	Chalc.	Musco. medhi-Al	Musco. medlow-Al	Chlo.+Musco.	Montm.-Na	Alun.	Total
Kaolinite KGa-2 pxl	**9**	13	0	0	0	0	1	7	30
Kaolinite KGa-1 wxl	12	**15**	0	0	0	0	0	20	47
Chalcedony	3	3	**336**	4	4	0	121	29	500
Muscovite-medhi-Al	240	122	13	**3271**	698	205	50	199	4798
Muscovite-medlow-Al	2	2	0	1	**7**	0	1	1	14
Chlorite+ Muscovite	245	195	41	639	184	**23**	44	160	1531
Montmoril-lonite-Na	107	60	2590	465	328	2	**3898**	669	8119
Alunite	999	1567	367	124	212	1	665	**17103**	21038
Total	1617	1977	3347	4504	1433	231	4780	18188	**36077**

GA: Ground Reference (Percent)

Class	Kaol. KGa-2 pxl	Kaol. KGa-1 wxl	Chalc.	Musco. medhi-Al	Musco. medlow-Al	Chlo.+Musco.	Montm.-Na	Alun. GDS82	Total
Kaolinite KGa-2 pxl	**0.56**	0.66	0.00	0.00	0.00	0.00	0.02	0.04	0.08
Kaolinite KGa-1 wxl	0.74	**0.76**	0.00	0.00	0.00	0.00	0.00	0.11	0.13
Chalcedony	0.19	0.15	**10.04**	0.09	0.28	0.00	2.53	0.16	1.39
Muscovite-medhi-Al	14.84	6.17	0.39	**72.62**	48.71	88.74	1.05	1.09	13.30
Muscovite-medlow-Al	0.12	0.10	0.00	0.02	**0.49**	0.00	0.02	0.01	0.04
Chlorite+ Muscovite	15.15	9.86	1.22	14.19	12.84	**9.96**	0.92	0.88	4.24
Montmoril-lonite-Na	6.62	3.03	77.38	10.32	22.89	0.87	**81.55**	3.68	22.50
Alunite	61.78	79.26	10.97	2.75	14.79	0.43	13.91	**94.03**	58.31
Total	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**

Table 4: The error matrix of the NNSLO classification map.

NNSLO: Ground Reference (Pixels)

Overall Accuracy = (21294/36079) 59.0205%
Kappa Coefficient = 0.4093

Class	Kaol. KGa-2 pxl	Kaol. KGa-1 wxl	Chalc.	Musco. medhi-Al	Musco. medlow-Al	Chlo.+Musco.	Montm.-Na	Alun.	Total
Kaolinite KGa-2 pxl	**0**	0	0	0	0	0	0	0	0
Kaolinite KGa-1 wxl	1	**2**	0	0	0	0	0	7	10
Chalcedony	0	0	**12**	0	0	0	19	8	39
Muscovite-medhi-Al	4	0	0	**21**	27	1	0	1	54
Muscovite-medlow-Al	0	0	0	0	**0**	0	0	0	0
Chlorite+ Muscovite	719	521	118	4074	952	**228**	133	677	7422
Montmoril-lonite-Na	118	84	3001	374	325	2	**4393**	858	9155
Alunite	775	1370	216	35	129	0	236	**16638**	19399
Total	1617	1977	3347	4504	1433	231	4781	18189	**36079**

NNSLO: Ground Reference (Percent)

Class	Kaol. KGa-2 pxl	Kaol. KGa-1 wxl	Chalc.	Musco. medhi-Al	Musco. medlow-Al	Chlo.+Musco.	Montm.-Na	Alun.	Total
Kaolinite KGa-2 pxl	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kaolinite KGa-1 wxl	0.06	**0.10**	0.00	0.00	0.00	0.00	0.00	0.04	0.03
Chalcedony	0.00	0.00	**0.36**	0.00	0.00	0.00	0.40	0.04	0.11
Muscovite-medhi-Al	0.25	0.00	0.00	**0.47**	1.88	0.43	0.00	0.01	0.15
Muscovite-medlow-Al	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00
Chlorite+ Muscovite	44.47	26.35	3.53	90.45	66.43	**98.70**	2.78	3.72	20.57
Montmoril-lonite-Na	7.30	4.25	89.66	8.30	22.68	0.87	**91.88**	4.72	25.37
Alunite	47.93	69.30	6.45	0.78	9.00	0.00	4.94	**91.47**	53.77
Total	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**

Table 5: The error matrix of the SAC classification map.

SAC: Ground Reference (Pixels) Overall Accuracy = (8656/34158) 25.3411% Kappa Coefficient = 0.1570									
Class	**Kaol. KGa-2 pxl**	**Kaol. KGa-1 wxl**	**Chalc.**	**Musco. medhi-Al**	**Musco. medlow-Al**	**Chlo.+Musco.**	**Montm.-Na**	**Alun.**	**Total**
Kaolinite KGa-2 pxl	**54**	17	23	45	36	4	129	1401	1709
Kaolinite KGa-1 wxl	556	**941**	216	296	228	17	188	11527	13969
Chalcedony	1	7	**178**	0	0	0	90	58	334
Muscovite-medhi-Al	29	25	720	**1557**	128	141	241	650	3491
Muscovite-medlow-Al	33	29	297	6	**6**	0	163	334	868
Chlorite+ Muscovite	0	0	0	0	0	**0**	0	0	0
Montmoril-lonite-Na	709	470	1689	2597	1023	69	**3816**	1226	11599
Alunite	16	43	11	0	0	0	14	**2104**	2188
Total	1398	1532	3134	4501	1421	231	4641	17300	**34158**
SAC: Ground Reference (Percent)									
Class	**Kaol. KGa-2 pxl**	**Kaol. KGa-1 wxl**	**Chalc.**	**Musco. medhi-Al**	**Musco. medlow-Al**	**Chlo.+Musco.**	**Montm.-Na**	**Alun.**	**Total**
Kaolinite KGa-2 pxl	**3.86**	1.11	0.73	1.00	2.53	1.73	2.78	8.10	5.00
Kaolinite KGa-1 wxl	39.77	**61.42**	6.89	6.58	16.05	7.36	4.05	66.63	40.90
Chalcedony	0.07	0.46	**5.68**	0.00	0.00	0.00	1.94	0.34	0.98
Muscovite-medhi-Al	2.07	1.63	22.97	**34.59**	9.01	61.04	5.19	3.76	10.22
Muscovite-medlow-Al	2.36	1.89	9.48	0.13	**0.42**	0.00	3.51	1.93	2.54
Chlorite+ Muscovite	0.00	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00
Montmoril-lonite-Na	50.72	30.68	53.89	57.70	71.99	29.87	**82.22**	7.09	33.96
Alunite	1.14	2.81	0.35	0.00	0.00	0.00	0.30	**12.16**	6.41
Total	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**

The producer accuracy from the error matrices (Tables 3, 4, and 5) is compared in Figure 26. The results from the error matrices show that the GA classification map achieved a higher overall accuracy of 68.4% compared to the NNSLO with one of 59.0%, and the SAC with one of 25.3%. The reason is the GA achieved relatively high producer

accuracy in four larger classes (Chalcedony, Muscovite-medhi-Al, Montmorillonite-Na, and Alunite) that contains more than 3000 pixels. Inspite of the higest overall accuracy, the GA has significantly low producer accuracy of 0.49% for the Chlorite+Muscovite class compared to the NNSLO producer accuracy of 98.70% because Muscovite-medhi-Al abundances mostly dominated on Chlorite+Muscovite abundances. In addition, the GA producer accuracy of 0.56% for the Kaolinite wxl class is significantly lower compared to the SAC producer accuracy of 61.42%, because the Alunite abundances mostly dominated the Kaolinite wxl abundances. The NNSLO has a lower overall accuracy because of the lowest producer accuracy (< 0.5%) for the Chalcedony and Muscovite-medhi-Al classes. In addition, the SAC has the lowest overall accuracy due to its lowest producer accuracy in the largest class, i.e., Alunite class. In general, the GA was able to classify at least a few pixels of the four smaller classes, Kaolinite pxl, Kaolinite wxl, Muscovite-medlow-Al, and Chlorite+Muscovite. In addition, the GA abundance maps (Table 2) of these four smaller classes seem to have greater than or equal to 50% maximum abundances with a better visual match with the reference map (Figure 25a) compared to the SAC and NNSLO. In this case, the SAC could not classify any Chlorite+Muscovite pixels, while the NNSLO could not classify any Kaolinite pxl and Muscovite-medlow-Al pixels. The overall accuracy of all three unmixing approaches could be higher if the endmember spectral signature used for unmixing were the same as the endmember spectral signatures used to produce the USGS Cuprite mineral map. The image-to-image registration errors (< 2 pixels) also influenced the accuracy of the classification maps.

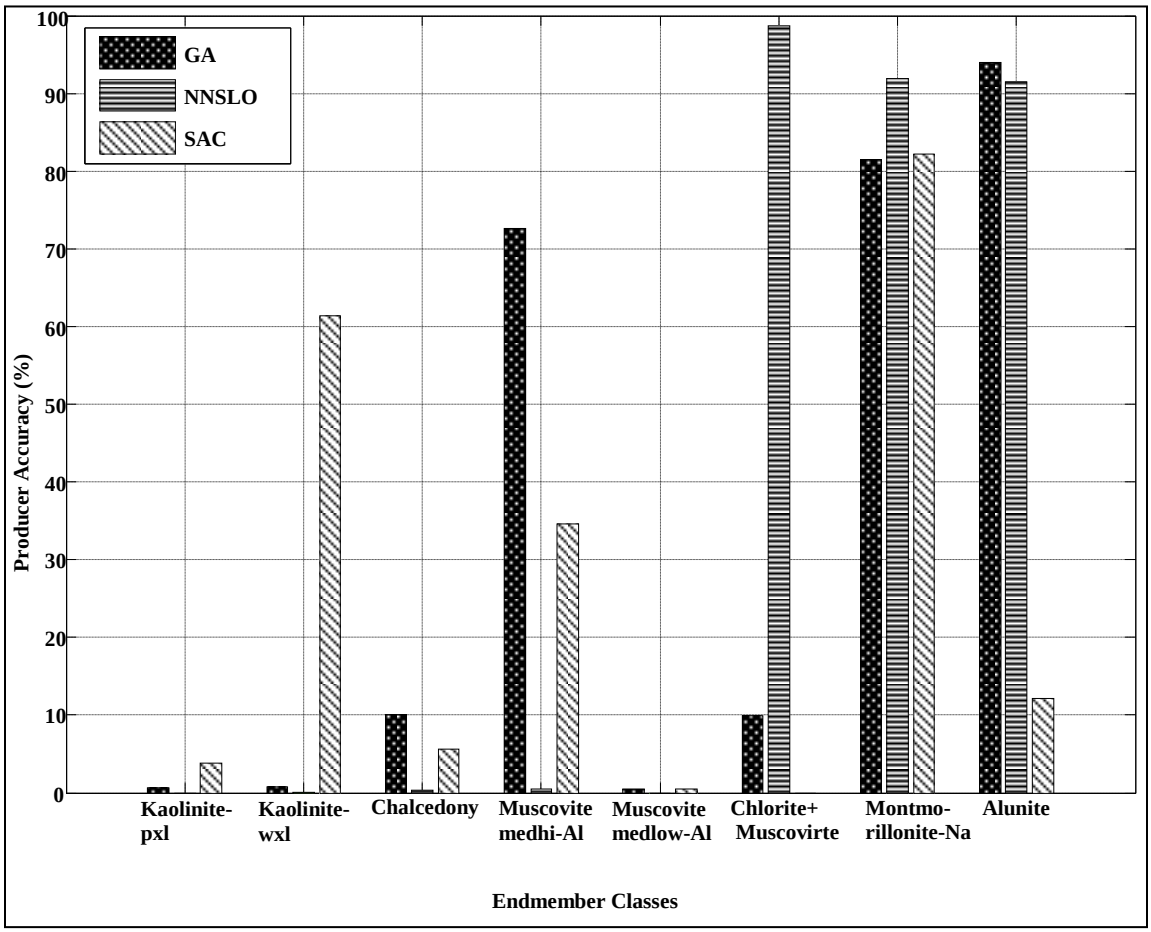


Figure 26: Comparison of producer accuracy of the GA, SAC, and NNSLO methods in each class.

CHAPTER 7: CONCLUSION AND FUTURE WORK

The purpose of this study was to develop an unmixing method to improve the accuracy of abundance maps by reducing the impact of illumination fluctuation, signature variability, and noise. For that purpose, a new GA-based unmixing with a properly designed fitness function (SAM) and a weakly constrained condition based on the concept of the SAC was presented in this work. In the case of synthetic data, a combination of random uniform illumination fluctuation between 0 and 1.28, spectral signature variability of 0%, 5%, and 10%, and SNR of 90, 60, 30, and 15 dB were used to create 12 datasets. It can be concluded from the unmixing results of these synthetic datasets that the GA was more robust to illumination effects, signature variability, and noise by improving the average index of agreement between the true and estimated abundances by 19.83% and 30.10% compared to the SAC and NNSLO, respectively. In addition, the GA improved the average correlation between the true and estimated abundances by 10.12% and 9.34% compared to the SAC and NNSLO, respectively. Furthermore, GA reduced the average RMSE between the true and estimated abundances by 3.57 and 7.23 compared to the SAC and the NNSLO, respectively.

In this study, a series of controlled experiments with synthetic hyperspectral datasets was performed to compare the unmixing performance of the GA against the SAC and the NNSLO. The results obtained from these experiments showed that the GA is less sensitive to illumination fluctuation, signature variability, and noise compared to the SAC and the NNSLO due to integration of SAM in the fitness function and the constraint

condition. Due to the evolutionary nature towards the global optima, the GA-based unmixing became less sensitive to noise.

From the experiment with the Cuprite hyperspectral data it was concluded that the GA unmixing result generally has a better visual and quantitative agreement with the ground reference data compared to the SAC and the NNSLO. For this purpose, a pre-classified mineral map from USGS was used as a ground reference for validating each unmixing result by converting the result to a classification map. The classification map produced from the GA unmixing results improved the overall accuracy by 43.1% and 9.4% compared to the SAC and NNSLO, respectively.

It can be difficult to achieve the ground reference abundances for real data. The availability of spectral signatures and more ground reference data of the desired endmembers present in the scene would be more useful to validate the unmixing results. As endmember selection plays an important role to improve the unmixing accuracy, an automated endmember selection method could be incorporated with the GA-based unmixing approach in the future. In addition, spatial information of endmembers with more constraints can be incorporated with the GA-based unmixing.

In general, the GA-based methods are time-consuming (Farzam *et al.*, 2008). The proposed GA was designed to achieve optimal unmixing results within a realistic timeline. For that reason, the GA-based unmixing was executed using a local multi-core parallel processing approach (Mathworks, 2011), where an individual core processor handled each group of pixels. In the future, it is possible to increase the processing speed significantly if more processors are added to a processor network.

The GA parameter adjustment (e.g., settings of population size, crossover fraction, and number of generation) presented in this work was only performed using a random pixel of the scene, where the GA obtained a near-optimal solution only for a new constraint-optimization problem. In the future, the near-optimal solution can be improved significantly, if the GA parameters are adjusted for every single pixel of the scene using a faster programming platform such as C++.

The GA-based unmixing method overcame some of the major limitations of the SAC and improved the accuracy of the estimated abundances compared to advanced unmixing methods (e.g., NNSLO and SAC) currently available.

REFERENCES

Abrams, M.J., Ashley, R.P., Rowan, L.C., Goetz, A.F.H., and Kahle, A.B. (1977). Mapping of hydrothermal alteration in the Cuprite Mining District, Nevada, using aircraft scanner images for the spectral region 0.46–2.36µm. *Geology*, 5:713–718.

Adams, J. B. and Smith, M. O. (1986). A new analysis of rock and soil types at the viking lander. *Journal of Geophysics Research*, 91(B8): 8098-8112.

Adams, J. B. and Gillespie, A R. (2006). *Remote Sensing of Landscapes with Spectral Images: A Physical Modeling Approach*. Cambridge; New York, Cambridge University Press, pp. 3-19.

Adams, J.B., Smith, M.O., and Gillespie, A.R. (1993). Imaging spectrometry: interpretation based on spectral mixture analysis, In Pieters CM, Englert P, editors. *Remote geochemical analysis: elemental and mineralogical composition.* New York: Cambridge University Press, 7: 145-166.

Adler-Golden, S.M., Levine, R.Y., Matthew, M.W., Richtsmeier, S.C., Bernstein, L.S., Gruninger, J., Felde, G., Hoke, M., Anderson, G.P. and Ratkowski, A. (2001). Shadow-Insensitive Material Detection/Classification with Atmospherically Corrected Hyperspectral Imagery. *Proceedings of SPIE*, Orlando, FL, Vol. 4381.

Alcântara, E., Barbosa, C., Stech, J., Novo, E., and Shimabukuro, Y. (2009). Improving the Spectral Unmixing Algorithm to Map Water Turbidity Distributions. *Environmental Modelling Software*, 24(9): 1051-1061.

Ball, J, Kari, S, and Younan, N. (2004). Hyperspectral Pixel Unmixing using Singular Value Decomposition. *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS'04)*, Alaska, USA, 5: 3253-3256.

Bekiroğlu, S., Dede, T., and Ayvaz, Y. (2009). Implementation of different encoding types on structural optimization based on adaptive genetic algorithm. *Finite Elem. Anal. Des. 45* (11): 826-835.

Bertram, J. (2000). The Molecular Biology of Cancer. *Mol. Aspects Med*., 21 (6): 167-223.

Bioucas-Dias, J. (2009). A Variable Splitting Augmented Lagrangian Approach to Linear Spectral Unmixing. *First Workshop on Hyperspectral Image and Signal Processing Evolution in Remote Sensing 2009 (WHISPERS'09)*, 1(11): 1-4.

Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.

Boardman, J. (1993). Automated Spectral Unmixing of AVIRIS Data using Convex Geometry Concepts. *Proceedings of the Fourth JPL Airborne Geoscience Workshop*, pp. 11-14.

Boardman, J. W. (1998). Post-ATREM polishing of AVIRIS apparent reflectance data using EFFORT: a lesson in accuracy versus precision. *Summaries of the Seventh JPL Airborne Earth Science Workshop*, JPL Publication 99-17, pp. 1:53.

Bowles, J., Antoniades, J., Baumback, M., Grossman, J., Haas, D., Palmadesso, P., and Stracka, J. (1997). Real-time Analysis of Hyperspectral Data Sets Using NRL's ORASIS Algorithm. *Proceedings of the SPIE, Imaging Spectrometry III*, San Diego, CA, USA, 3118: 38-45.

Champagne, C. M. (2001). Remote sensing of plant water content for precision agriculture: the potential for hyperspectral modelling. *University of Ottawa*, Ottawa, Ontario, Canada, pp. 44-45.

Chang, C.I. (2007). *Hyperspectral Data Exploitation: Theory and Applications*. Wiley-Interscience, pp. 230-231.

Chang, C. I., Wu, C. C., Liu, W. W. and Ouyang, Y. C., 2006. A new growing method for simplex-based endmember extraction algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 44: 2804-2819.

Clark, R.N., Swayze, G.A., Gallagher, A.J., King, T.V.V., and Calvin, W.M. (1993a). *The U. S. Geological Survey, Digital Spectral Library: Version 1: 0.2 to 3.0 microns.* U.S. Geological Survey Open File Report 93-592, 1340.

Clark, R.N., Swayze, G.A., and Gallagher, A. (1993b). *Mapping Minerals with Imaging Spectroscopy*. U.S. Geological Survey, Office of Mineral Resources Bulletin, 2039: 141-150.

Clark, R.N. and Swayze, G.A. (1996). Evolution in imaging spectroscopy and sensor signal-to-noise: An examination of how far we have come. *In Summaries of the 6th Annual JPL Airborne Earth Science Workshop*, pp. 49-53.

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37(1): 35-46.

De-Asis, A. M., Omasa, K., Oki, K., and Shimizu, Y. (2008). Accuracy and Applicability of Linear Spectral Unmixing in Delineating Potential Erosion Areas in Tropical Watersheds. *International Journal of Remote Sensing*, 29(14):4151-4171.

De-Carvalho, O.A. and Meneses, P.R. (2000). Spectral correlation mapper (SCM): an improvement on the spectral angle mapper (SAM). *In: Summaries of the 9th JPL airborne earth science workshop, Pasadena, CA: JPL Publication 00–18*, Vol. 9.

Deep, K., and Dipti, K. (2008). A Self Organizing Migrating Genetic Algorithm for Constrained Optimization. *Applied Mathematics and Computation, 198*: 237-250.

Farzam, M., Beheshti, S., and Raahemifar, K. (2008). Calculation of Abundance Factors in Hyperspectral Imaging using Genetic Algorithms. *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE)*, Niagara Falls, Ontario, pp. 837-842.

Foppa, N., Wunderle, S., and Hauser, A. (2002). Spectral Unmixing of NOAA-AVHRR Data for Snow Cover Estimation. *Proceedings of EARSeL-LISSIG Workshop Observing our Cryosphere from Space*, Berne, Switzerland, 156:155-163.

Gao, B.C.; Goetz, A.F.H. (1990). Column atmospheric water vapor and vegetation liquid water retrievals from airborne imaging spectrometer data. *Journal of Geophysical Research*, 95: 3549-3564.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison Wesley Reading MA, Addison-Wesley*, pp. 412.

Green, A.A., Berman, M., Switzer, P., and Craig, M.D. (1988). A Transformation for Ordering Multispectral Data in terms of Image Quality with Implications for Noise Removal. *IEEE Transactions on Geoscience and Remote Sensing*, 26: 65-74.

He, M., and Mei, S. (2008). Fully Constrained Oblique Projection Approach to Mixed Pixel Linear Unmixing. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Congress Beijing*, 7084:70840C.

Healey, G. and Slater, D. (1999). Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions. *IEEE Trans. Geosci. Remote Sens.*, 37(6): 2706-2717.

Heldens, W., Heiden, U., Bachmann, M., Esch, T., Müller, A., and Dech, S. (2009). Scaling issues in validation of abundance maps derived from HyMAP data of an urban area. *6th EARSeL SIG IS Workshop on Imaging Spectroscopy*, Tel Aviv, Israel.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. *Ann Arbor MI University of Michigan Press*, 34:228.

Hook, S.J., Rast, M. (1990) Mineralogic mapping using Airborne Visible Infrared Imaging Spectrometer (AVIRIS), Shortwave Infrared (SWIR) data acquired over Cuprite, Nevada. In: Green, R.O. Ed. *Proceedings of the Second Airborne Visible Infrared Imaging Spectrometer (AVIRIS) Workshop*, JPL Publication 90-54, pp. 199-207.

Hostert, P., Roder, A., and Hill, J. (2003). Coupling spectral unmixing and trend analysis for monitoring of long-term vegetation dynamics in Mediterranean rangelands. *Remote Sensing of Environment*, 87: 183-197.

Huang, C., and Wang, C. (2006). A GA-based Feature Selection and Parameters Optimization for Support Vector Machines. *Expert Systems with Applications, Elsevier Ltd*., 31(2), 231-240.

Im J., Jensen J.R., Jensen R.R., Gladden J., Waugh J., and Serrato, M. (2012). Vegetation Cover Analysis of Hazardous Waste Sites in Utah and Arizona Using Hyperspectral Remote Sensing. *Remote Sensing*, 4(2): 327-353.

Ja'fari, A. and Moghadam, R. H. (2012). Integration of ANFIS, NN and GA to determine core porosity and permeability from conventional well log data. *Journal of Geophysics and Engineering*, 9(5): 473.

Jensen, J. R. (2004). *Introductory Digital Image Processing: A Remote Sensing Perspective* (3 ed). Upper Saddle River, NJ: Pearson Education, Inc., pp. 431-462.

Keshava, N., and Mustard, J. (2002). Spectral Unmixing. *IEEE Signal Processing Magazine*, 19: 44-57.

Konak, A., Coit, D., and Smith, A. (2006). Multi-objective Optimization using Genetic Algorithms: A Tutorial. *Reliability Engineering System Safety*, 91(9):992-1007.

Kruse, F., Lefkoff, A., Boardman, J., Heidebrecht, K , Shapiro, A., Barloon, P., and Goetz, A. (1993). The spectral image processing system (SIPS) - interactive visualization and analysis of imaging spectrometer data. *Remote Sensing of Environment*, 44: 145-163.

Lawson, C. and Hanson, R. (1974). *Solving Least Square Problems*. Prentice-Hall, Inc.

Lee, T.A., Headley, L.M., and Hardy, J.K. (1991). Noise Reduction of GC/MS Data Using Principal Component Analysis. *Analytical Chemistry*, 63: 357-360.

Liu, J., Miller, J. R., Haboudane, D., Pattey, E., and Hochheim, K. (2008). Crop Fraction Estimation from Casi Hyperspectral Data using Linear Spectral Unmixing and Vegetation Indices. *Canadian Journal of Remote Sensing*, 34(Suppl. 1): S124-S138.

Mallinis, G. and N. Koutsias (2012). Comparing ten classification methods for burned area mapping in a Mediterranean environment using Landsat TM satellite data. *International Journal of Remote Sensing*, 33(14): 4408-4433.

Mathworks. (2011). *Global Optimization Toolbox: User's Guide* (r2011b).

Mehta, K. and S. Bhattacharyya (1999). Combining rules learnt using genetic algorithms for financial forecasting. Evolutionary Computation, 1999. CEC 99. *Proceedings of the 1999 Congress on*, 2: 1245-1252.

Miao, L., Qi, H., and Szu, H. (2007). A Maximum Entropy Approach to Unsupervised Mixed-Pixel Decomposition. *IEEE Transactions on Image Processing*, 16(4): 1008-1021.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press, pp. 4-5.

Nascimento, J., and Bioucas-Dias, J. (2005). Does Independent Component Analysis Play A Role in Unmixing Hyperspectral Data. *IEEE, Transactions on Geoscience and Remote Sensing*, 43(1): 175–187.

National Research Council (U.S.). Committee on Remote Sensing for Agricultural Purposes. (1970). *Remote sensing with special reference to agriculture and forestry / Committee on Remote Sensing for Agricultural Purposes, Agricultural Board, National Research Council.* National Academy of Sciences, Washington, D.C. pp. 237-238.

Neville, R.A., Staenz, K., Szeredi, T., Lefebvre, J., and Hauff, P. (1999). Automatic Endmember Extraction from Hyperspectral Data for Mineral Exploration. *Proceedings of the Fourth International Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing*, Ottawa, Ontario, Canada, 2: 891-897.

Omata, K., S. Kobayashi, et al. (2012). Kinetic model of K–Ni/α-Al2O3 catalyst for oxidative reforming of methane determined by genetic algorithm. *Applied Catalysis A: General 425–426*, (0): 170-177.

Orr, H.A. (2009). Fitness and its role in evolutionary genetics. *Nat. Rev. Genet*. 10 (8): 531–9.

Parente, M., Bishop, J. L., and Bell III, J. F. (2009). Spectral Unmixing for Mineral Identification in Pancam Images of Soils in Gusev Crater, Mars. *International Journal of Solar System Studies*, 203(2):421-436.

Park, B., Windham, W. R., Lawrence, K. C., and Smith, D. P. (2007). Contaminant classification of poultry hyperspectral imagery using a spectral angle mapper algorithm. *Biosystems Engineering*, 96(3): 323-333.

Parra, L., Spence, C., Sajda, P., Ziehe, A., and Müller, K.-R. (2000). Unmixing Hyperspectral Data. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 13:942-948.

Pearlmutter, B.A., and Parra, L.C. (1997). Maximum Likelihood Blind Source Separation: A Context-Sensitive Generalization of ICA M. C. Mozer, M. I. Jordan, & T. Petsche, eds. *Advances in Neural Information Processing Systems*, 9: 613-619.

Peddle, D.R., F.G. Hall and E.F. LeDrew. (1999). Spectral Mixture Analysis and Geometric Optical Reflectance Modeling of Boreal Forest Biophysical Structure. *Remote Sensing of Environment*, 67(3): 288-297.

Peddle, D. R., and Smith, A. M. (2005). Spectral Mixture Analysis of Agricultural Crops: Endmember Validation and Biophysical Estimation in Potato Plots. *International Journal of Remote Sensing*, 26(22): 4959-4979.

Plaza, A., Martinez, P., Perez, R., and Plaza, J. (2002). Spatial/spectral endmember extraction by multidimensional morphological operations. *IEEE Transactions on Geoscience and Remote Sensing*, 40: 2025-2041.

Plaza, J., Hendrix, E. M. T., García, I., Martín, G., and Plaza, A. (2011). On Endmember Identification in Hyperspectral Images Without Pure Pixels: A Comparison of Algorithms. *Journal of Mathematical Imaging and Vision*, 42(2-3): 163-175.

Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). Section 2.10. QR Decomposition. *Numerical Recipes: The Art of Scientific Computing* (3rd ed.), New York: Cambridge University Press.

Quirin, A., and Korczak, J. (2005). Representation of genetic individuals for unmixing multispectral data. *Congress on Evolutionary Computation*, 2: 1325-1331.

Rabah, Z. B., Farah, I. R., and Ghezala, H. B., and Solaiman, B. (2011). A New Method to Change Illumination Effect Reduction Based on Spectral Angle Constraint for Hyperspectral Image Unmixing. *IEEE Geosci. Remote Sensing Lett.*, 8(6): 1110-1114.

Riaño, D., Chuvieco, E., Salas, J., and Aguado, I. (2003). Assessment of Different Topographic Corrections in Landsat-TM Data for Mapping Vegetation Types. *IEEE Transactions on Geoscience and Remote Sensing*, 41 (5): 1056-1061.

Richter, N., Staenz, K., and Kaufmann, H. (2008). Spectral Unmixing of Airborne Hyperspectral Data for Baseline Mapping of Mine Tailings Areas. *International Journal of Remote Sensing*, 29(13): 3937-3956.

Roberts, D. A.; Halligan, K.; Dennison, P. *VIPER Tools User Manual (Version 1.5)*. (2007). Santa Barbara: University of California, Santa Barbara, pp. 53-58.

Roberts, D. A., Gardner, M., Church, R., Ustin, S., Scheer, G., Green, R.O. (1998). Mapping chaparral in the Santa Monica Mountains using multiple endmember spectral mixture models. *Remote Sensing of Environment*, 65(3): 267-279.

Rogge, D. M., Rivard, B., Zhang, J., and Feng, J. (2006). Iterative Spectral Unmixing for Optimizing Per-Pixel Endmember Sets. *IEEE Geoscience and Remote Sensing*, pp. 3725-3736.

Rosario-Torres, S. (2004). Iterative Algorithms for Abundance Estimation on Unmixing of Hyperspectral Imagery. *University of Puerto Rico at Mayagüez. 0-7803-8742-2/04 (C) IEEE 3278*, pp. 1-67.

Rosario-Torres, S. and Vélez-Reyes, M. (2005). An algorithm for fully constrained abundance estimation in hyperspectral unmixing. *In Proceedings of SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, 5806: 711-719.

Rosario-Torres, S., Vélez-Reyes, M., Hunt, S. D., and Jiménez, L. O. (2007). New Developments and Application of the UPRM MATLAB Hyperspectral Image Analysis Toolbox. *In Proceedings of SPIE: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII*, Vol. 6565.

Rouse, D. and Trussell, H. (2005). Estimation of Spatially Varying Illumination in Hyperspectral Images. *Proceedings of the IEEE International Conference on Image Processing*, Genoa, Italy, pp. 11-14.

Rowan, L.C., Hook, S. J., Abrams, J. M., Mars, J. C. (2003). Mapping Hydrothermally Altered Rocks at Cuprite, Nevada, Using the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), a New Satellite-Imaging System. *Economic Geology*, 98(5): 1019-1027.

Sanchez, S., Martin, G., Plaza, A., and Chang, C.-I. (2010). GPU Implementation of Fully Constrained Linear Spectral Unmixing for Remotely Sensed Hyperspectral Data Exploitation. *Proceedings of the SPIE*, 7810.

Sastry, K. (2007). Single and Multiobjective Genetic Algorithm Toolbox in C++. *University of Illinois at Urbana-Champaign*, Urbana, IL, IlliGAL Report No. 2007016.

Shaw, G. and Burke, H. (2003). Spectral Imaging for Remote Sensing. *Lincoln Lab. J.*, 14(1): 3–28.

Shang, J., Neville, R., Staenz, K., Sun, L., Morris, B., and Howarth, P. (2008). Comparison of Fully Constrained and Weakly Constrained Unmixing through Mine-Tailing Composition Mapping. *Canadian Journal of Remote Sensing*, 34(S1):S82-S91.

Shepp, L., and Vardi, Y. (1982). Maximum Likelihood Reconstruction for Emission Tomography. *In IEEE Transactions on Medical Imaging*, 1(2):113-122.

Shrestha, D. P., D. E. Margate, Meer, F., V. D., and Anh, H. V. (2005). Analysis and classification of hyperspectral data for mapping land degradation: An application in southern Spain. *International Journal of Applied Earth Observation and Geoinformation*, 7(2): 85-96.

Sirkeci, B. (2001). Total Least Squares Approaches for Spectral Unmixing of Hyperspectral Imagery. *Northeastern University*, pp. 1-70.

Srinivas, M., and Patnaik, L. (1994). Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on System, Man and Cybernetics*, 24(4):656–667.

Staenz, K, Nadeau, C, Secker, J, and Budkewitsch, P. (2000). Spectral Unmixing applied to Vegetated Environments in the Canadian Arctic for Mineral Mapping. *Proceedings of the XIXth congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Amsterdam, the Netherlands, pp. 8-16.

Staenz, K., Schwarz, J., Vernaccini, L., Vachon, F. and Nadeau, C. (1999). Classification of Hyperspectral Agricultural Data with Spectral Matching Techniques. *Proceedings of the International Symposium on Spectral Sensing Research*, Las Vegas, Nevada, pp. 12-20.

Teillet, P. M., Guindon, B., and Goodeonugh, D. G. (1982). On the slope-aspect correction of multispectral scanner data. *Canadian Journal of Remote Sensing*, 8: 84-106.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra* (SIAM).

Tsoulos, I. (2009). Solving constrained optimization problems using a novel genetic algorithm. *Applied Mathematics and Computation*, 208(1):273-283.

Tyo, J. S., Robertson, J., Wollenbecker, J, and Olsen, R. C. (2000). Statistics of target spectra in HSI scenes.*Proceedings of SPIE Conference*. Imaging Spectrometry VI.

Velez-Reyes, M., Puetz, A., Hoke, M. P., Lockwood, R. B., Rosario, S. (2003). Iterative Algorithms for Unmixing of Hyperspectral Imagery. *Proceeding of SPIE 5093*, vol. 418.

Vidal, T., Crainic, T.G., Gendreau M., Prins, C., 2011. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. *Tech. Rep., CIRRELT 2011-61*.

Wilmott, C.J. (1982). Some Comments on the Evaluation of Model Performance. *Bulletin of the American Meteorological Society*, 63(11): 1309-1313.

Winter, M. E. (1999). N-finder: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In: Image Spectrometry V, *Proceedings of SPIE*, Vol. 3753: 266- 277.

Yao, H., and Tian, L. (2003). A Genetic-Algorithm-based Selective Principal Component Analysis (GA-SPCA) Method for High-Dimensional Data Feature Extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 41(6):1469-1478.

Zhang, J, Chung, H, and Lo, W. L. (2007). Clustering-based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, 11(3): 326–335.

Zhang, J., Rivard, B and Rogge, D. M. (2008). The Successive Projection Algorithm (SPA), An Algorithm with A Spatial Constraint for the Automatic Search of Endmembers in Hyperspectral Data. *Sensors*, 8(2): 1321–1342.

Zhao, J., L. Wang, Zeng, P., Fan, W. (2012). An effective hybrid genetic algorithm with flexible allowance technique for constrained engineering design optimization. *Expert Systems with Applications*, 39(5): 6041-6051.

APPENDIX

Simple Genetic Algorithm (SGA)

A SGA is the most widely used method to solve many single-objective constrained-handling problems. For instance, Farzam *et al.* (2008) utilized the SGA functionalities in MATLAB with the Least Square Error (LSE) fitness function (Eq. 7) to optimize the ASC constrained optimization problem (Eq. 4). In addition, the Illinois Genetic Algorithms Laboratory[10] at the University of Illinois developed an open source GA toolbox in C++ that uses a SGA to solve constrained optimization problems of a single-objective function (Sastry, 2007). The Kanpur Genetic Algorithms Laboratory[11] at the Indian Institute of Technology also developed an open source SGA in C to solve single objective constrained-handling problems.

The basic steps of a SGA are as follows (Goldberg, 1989; Konak *et al.*, 2006; Sastry, 2007):

1. **Initialization:** Set i = 1. Create an initial population set P_i from randomly generated solutions S and evaluate the fitness of the solutions in P_i;
2. **Crossover:** Produce an offspring population P_o by choosing two solutions S_1 and S_2 from P_i according to their fitness values and add them to P_i;
3. **Mutation:** Apply the mutation rate to mutate each solution in P_o;

[10] "Single and Multiobjective Genetic Algorithm Toolbox in C++". Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois, Urbana-Champaign, USA. Accessed Jan 07, 2011: http://illigal.org/2007/06/05/single-and-multiobjective-genetic-algorithm-toolbox-in-c.

[11] "Single-objective GA code in C (for Windows and Linux)". Kanpur Genetic Algorithms Laboratory (KANGAL), Indian Institute of Technology, Kanpur, India. Accessed Jan 07, 2011: http://www.iitk.ac.in/kangal/codes.shtml.

4. **Fitness:** For each solution in P_o, assign a fitness value depending on its objective function value and infeasibility;
5. **Selection:** Select solutions S from P_o according to their fitness and copy S to $P_i + 1$;
6. **Termination:** If the termination conditions (e.g., maximum number of generation) are satisfied, stop the search and go back to the current population. Otherwise, set i=i+1 and go to step 2; and

Self-organizing Migrating Genetic Algorithm (C-SOMGA)

Deep *et al.* (2008) constructed a penalty parameter-free hybrid algorithm, C-SOMGA, for solving the constrained nonlinear optimization problems. The advantages of this algorithm are: 1) It does not require any penalty parameters for constrained handling; 2) it can work with a small population size; and 3) it requires less time to evaluate an objective function in the entire run. The authors have found that the results of the C-SOMGA algorithm is more robust in terms of how close the algorithm gets to the global minimum, the number of function calls required to converge to the minimum, and the number of successful runs compared to the Constrained GA (C-GA) and the C-SOMA.

The steps of C-SOMGA are as follows:

1. Produce an initial population;
2. Evaluate all chromosomes from the population;
3. Apply a tournament-based selection process for constrained optimization to select better chromosomes for next generation;

4. Generate new child chromosomes by applying the crossover operator with probability P_c;
5. Evaluate new child chromosomes;
6. Apply a mutation operator with probability P_m on each chromosome;
7. Evaluate mutated chromosomes;
8. Find the fittest individual of the population;
9. Create a new population with size N for each active chromosome using the positions of the active individuals towards the fittest chromosomes;
10. Sort the new population according to their fitness in decreasing order;
11. Check the feasibility criteria for each individual in the sorted population;
12. Replace active chromosomes with the new position by sorting them in an order if the feasibility criteria are satisfied. Otherwise, go to step 11;
13. Select the fittest individuals based on the tournament selection for the next generation from previous and current generations;
14. If the termination criteria are not satisfied, go to step 3; otherwise, go to step 15; and
15. The best chromosome is considered as the final optimal solution.

Novel Genetic Algorithm (NGA)

Tsoulos (2009) developed a new constrained optimization GA called NGA and compared it with the above-mentioned C-SOMGA. His algorithm performed better than the C-SOMGA as he uses modified crossover and mutation operators to preserve the feasibility of the solutions. It is a local search procedure, which selects chromosomes

randomly and uses a stopping rule based on asymptotic considerations. His algorithm initializes a set of chromosomes inside upper and lower bound of the objective function and iteratively applies modified crossover and mutation operators until the stopping criteria are met. Penalty strategy is used to penalize infeasible solutions according to the constrained violations. The crossover and mutation operators preserve the feasibility of chromosomes by rejecting chromosomes that are outside the constrained boundary. The local search procedure used in his algorithm improves the fitness of the chromosomes and speeds up the convergence of the algorithm to find the global minimum. The algorithm terminates when the global minimum is achieved from the recorded variance of the best-discovered value in every generation.

The steps of his algorithms are as follows:

1. Set the following parameters:
    a) The number of chromosomes, i.e., population size ,
    b) The maximum number of generations,
    c) The selection rate,
    d) The mutation rate, and
    e) The local search rate.
2. Start iteration loop.
3. Initialize the chromosomes and store them. Every chromosome is initialized randomly inside the feasible region.
4. Evaluate the fitness for every chromosome. This evaluation is performed using a penalty technique.

5. Apply the modified genetic operations of crossover and mutation to the population.
6. Select randomly some of the chromosomes from the population and apply them to the local search procedure.
7. End iteration loop.
8. Terminate if the termination criteria is satisfied. Otherwise, go to step 4.