Band Selection for Hyperspectral Images Based on Parallel Particle Swarm Optimization Schemes

Yang-Lang Chang and Jyh-Perng Fang
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan

Lena Chang
Department of Communications Navigation and Control Engineering
National Taiwan Ocean University
Keelung, Taiwan

Jon Atli Benediktsson
Department of Electrical and Computer Engineering
University of Iceland
Reykjavik, Iceland

Hsuan Ren and Kun-Shan Chen
Center for Space and Remote Sensing Research
National Central University
Chung-Li, Taiwan

***Abstract*— *Greedy modular eigenspaces* (GME) has been developed for the band selection of *hyperspectral images* (HSI). GME attempts to greedily select uncorrelated feature sets from HSI. Unfortunately, GME is hard to find the optimal set by greedy operations except by exhaustive iterations. The long execution time has been the major drawback in practice. Accordingly, finding an optimal (or near-optimal) solution is very expensive. In this study we present a novel parallel mechanism, referred to as *parallel particle swarm optimization* (PPSO) band selection, to overcome this disadvantage. It makes use of a new particle swarm optimization scheme, a well-known method to solve the optimization problems, to develop an effective parallel feature extraction for HSI. The proposed PPSO improves the computational speed by using parallel computing techniques which include the *compute unified device architecture* (CUDA) of *graphics processor unit* (GPU), the *message passing interface* (MPI) and the *open multi-processing* (OpenMP) applications. These parallel implementations can fully utilize the significant parallelism of proposed PPSO to create a set of near-optimal GME modules on each parallel node. The experimental results demonstrated that PPSO can significantly improve the computational loads and provide a more reliable quality of solution compared to GME. The effectiveness of the proposed PPSO is evaluated by *MODIS/ASTER airborne simulator* (MASTER) HSI for band selection during the Pacrim II campaign.**

I. Introduction

The benefits of high dimensional spectral images makes use of the advances of sensor technologies and a growing number of spectral bands. High-dimensional remote sensing obtained from multispectral, hyperspectral and even ultraspectral sensors generally provide huge spectral information for data analysis. It covers a tremendous of applications from satellite remote sensing imaging and surveillance monitoring systems to industrial product inspections and medical imaging examinations. World-wide researchers all report the difficulties regarding its intrinsic characteristics of the data complexities. Consequently, determining the most useful and valuable information has become more essential. In response, a technique known as *greedy modular eigenspaces* (GME) [1] has been developed for the band selection of hyperspectral datasets. GME attempts to greedily select a near-optimal unique band (feature) set from *hyperspectral images* (HSI). Unfortunately, it is hard to find the most optimal set by greedy reordering operations of GME except by exhaustive iterations. The long execution time of these exhaustive iterations has been the major drawback in practice. Accordingly, finding a near-optimal solution is very expensive.

Recently, a new *simulated annealing band selection* (SABS) [2] based on GME is reported to improve to the performances of GME. It selects sets of non-correlated hyperspectral bands according to *simulated annealing* (SA) algorithm and utilizes the inherent separability of different classes in HSI to reduce dimensionality. As a common technique in metallurgy, SA denotes the slow-cooling melt behavior in the formation of hardened metals. It is a general purpose optimization technique which can find optimal or near-optimal solutions. SABS is an optimization technique based on SA to select an optimal set of feature bands from HSI.

However, the long execution time of SABS has been the major drawback in practice. Numerous studies have been devoted to *parallel* SA algorithms. Due to the constrain of a single *Markov chain* (MC) needed to be adjusted in the parallel mechanism, only a limited SA parallelism was exploited [3]. Currently, a new *parallel* SA method, referred to as a *parallel simulated annealing* (PSA) band selection approach [4], was introduced to overcome this disadvantage. It can improve the computational speed by using parallel computing techniques. It allows multiple MC to be traced simultaneously and fully utilizes the significant parallelism embedded in SABS to create a set of PSA modules on each parallel node.

In this paper we further present a novel parallel band selection approach, referred to as *parallel particle swarm optimization* (PPSO), for HSI. The approach is based on the *particle swarm optimization* (PSO) [5] scheme. PSO is a relatively new swarm intelligence algorithm originally designed to solve the nonlinear function optimization and neural network training problems [5]. It's inspired by social behavior of organisms such as bird flocking. PSO scheme is currently a most well-known method to solve the heuristic optimization

problems.

PPSO can enhance the memory lacks of PSA heuristic parameters and improve the limited searches and convergences of PSA. Each move of PSO particles is deeply affected by its current positions and its memory of previous useful parameters, and by the cooperation and group knowledge of the swarm [6]. These properties make PPSO a powerful method for solving optimization problems. Compared to PSA, PPSO outperforms in terms of efficiency of dimension reductions. The computational speed of PPSO is improved by parallel techniques which include *compute unified device architecture* (CUDA) of the *graphics processor unit* (GPU), *message passing interface* (MPI) and *open multi-processing* (OpenMP) applications. These parallel implementations can fully utilize the significant parallelism of PPSO and create a set of near-optimal GME modules on each parallel node.

The efficiency of PPSO is evaluated by *MODIS/ASTER airborne simulator* (MASTER) HSI for land cover classification during the Pacrim II campaign. The experimental results demonstrated that PPSO can significantly improve the computational loads and provide a more reliable quality of solution compared to GME in terms of the efficiency of dimension reductions and feature extractions. The rest of this paper is organized as follows. In Section II, the proposed PPSO is described in details. In Section III, a set of experiments is conducted to demonstrate the feasibility and utility of the proposed approach. Finally, in Section IV, several conclusions are presented.

II. Methodology

A drawbacks of the PSA method is its lack of memory. It limits the searches and convergences in the solution space. The particles proposed in PPSO have individual memory to cooperate each others forwarding to a global (near-optimal) solution. It uses only one velocity operator to deal with the search process for finding the global optima. The details of PSO are available in Ref. [5]. A brief description is as follows:

A. PSO

Assume there is a swarm of I particles. X_i is the position vector of the i^{th} particle in the *D-dimensional* search space. Each particle $X_i = (x_{i1}, \ldots x_{id})$ representing a potential solution is defined as the i^{th} particle in the d^{th} dimension of the *D-dimensional* solution space. $V_i = (v_{i1}, \ldots v_{id})$ is the velocity vector of the i^{th} particle as shown in Fig. 1 and Fig. 2. T represents the total number of iterations. $P_i = (P_{i1}, \ldots P_{id})$ is the local best position the i^{th} particle had reached and P_{gd} is the global best position for all the particles in the *D-dimensional* solution space. The velocity and position of each PSO particle is adjusted by the following equations:

$$v_{id}(t+1) = \omega \times v_{id}(t) + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times rand() \times (p_{gd} - x_{gd}), \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (2)$$

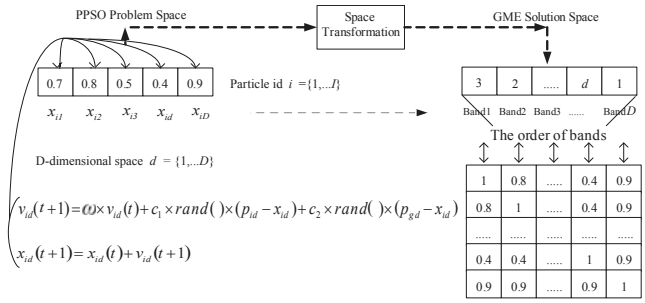


Fig. 1. The space transformation mechanism of the proposed PPSO approach.

where the function $rand()$ is a random number between 0 and 1. ω is called the *inertia weight* [7]. c_1 and c_2 are the acceleration constants, also known as *cognitive confidence coefficients*. They determine how much the particle is influenced by its best location and by the best position found by the swarm. The time-step $t + 1$ is usually set to one time unit.

B. Space Transformation

Before applying PSO to GME band selection problem, a space transformation is need to transfer the model of PPSO problem space to the model of GME solution space. In Fig. 1, on the left hand side, the PSO problem space described in previous subsection is mapped to GME solution space. Each position vector X_{id}, where $d = 1, \ldots D$, representing a potential solution in PPSO problem space is corresponding to a solution in GME solution space. The order of bands $(Band_1, \ldots Band_D)$ in GME solution space is consistent with the position vector X_i of the i^{th} particle. If the position vector X_{id} has the highest value than the others, its corresponding band should be the first band in the band list, and so on and so forth. For any arbitrary vector X_{id}, there must be one and only one GME band permutation corresponding to it. The mapping must exactly apply the relation within the vector X_{id} to the corresponding band permutation of GME. If $X_{ix} > X_{iy}$ then the permutation order of $Band_x$ should be on the front of $Band_y$. These two mapping strategies of space transformation must be satisfied to ensure the proposed PPSO band selection can be properly implemented.

C. PPSO

The natural parallelism of proposed PSO is in the face that each particle can be regarded as an independent agent. Parallel computation benefits the algorithm by providing each agent with one of the parallel processors [8]. The intrinsic parallel characteristics embedded in PSO can be therefore suitable for a parallel implementation. As shown in Fig. 2, the flowchart of the proposed PPSO scheme is illustrated. Three main loops are deployed in PPSO. They are the generation,

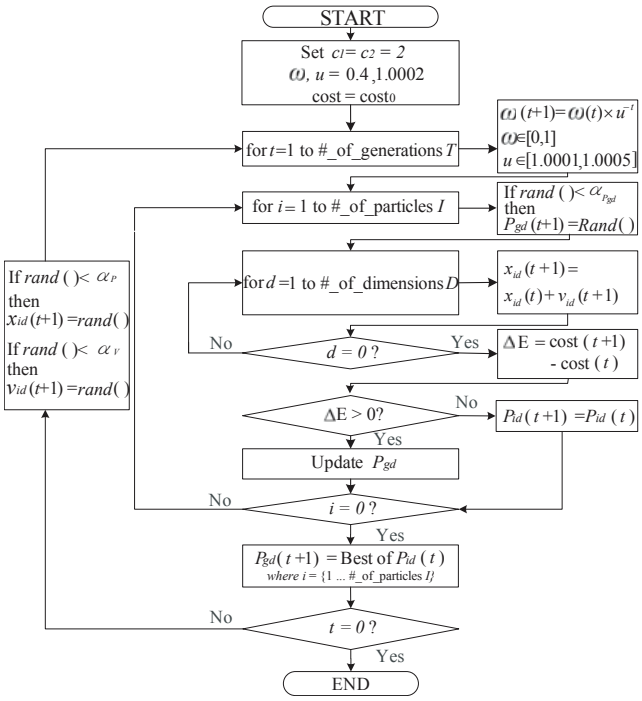


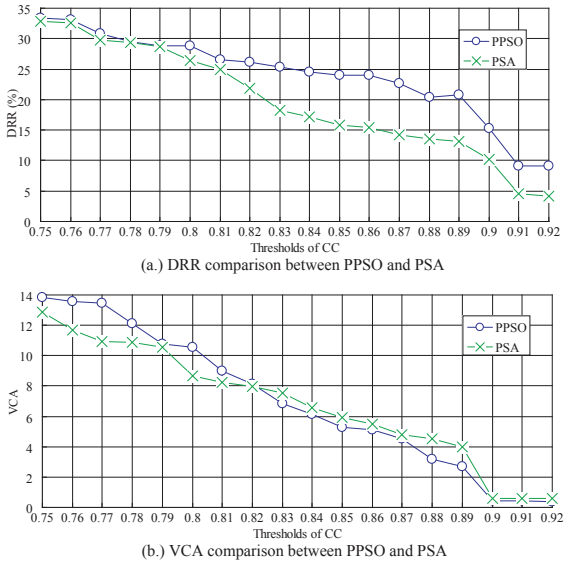
Fig. 2. The flowchart of the proposed PPSO scheme.

particle and dimension loops. The initial statuses are also shown in Fig. 2. After initialization, the cost can be produced by the permutation. The cost function, also known as the fitness function, of the GME [1] is defined as:

$$cost = \sum_{l=1}^{n_k} {m_l}^2 \times c^2, \tag{3}$$

where c, m_l and n_k represent the values of the corresponding *correlation coefficient* (CC), the number of bands (feature spaces) in modular eigenspaces Φ_l^k, and the total number of modular eigenspaces of a GME set Φ^k, i.e. $l \in \{1, \ldots, n_k\}$ respectively.

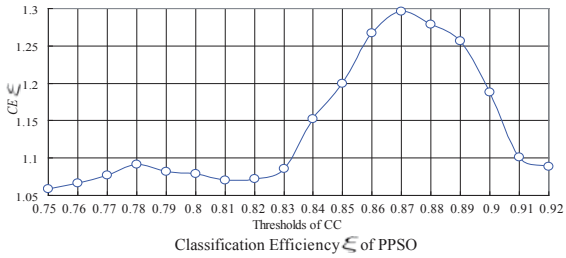
The constants $\alpha_{P_{gd}}$, α_P and α_V are set as constants to exclude PPSO from falling into local optima prematurely. Three major loops are activated to calculate the moving distances for different dimensions d and particles i. A new set of the local best P_{id} is then compared to produce an updated global best P_{gd} for preparing a new permutation and a new calculation of cost function for next generation t. In order to shorten the communication time and obtain the same computational loads on each parallel node, the loads of PPSO are balanced by using a load estimator to evenly distribute the HSI dataset for a better computational performance. The master-slave approach is hence practical to construct the PPSO architecture. Three parallel PPSO implementations inspired by parallel SA [3], namely non-interacting PSO (NIPSO), periodic exchange PSO (PEPSO) and asynchronous PSO (APSO), are conducted to evaluate the proposed PPSO scheme.



(a.) DRR comparison between PPSO and PSA

(b.) VCA comparison between PPSO and PSA

Fig. 3. The comparisons between PPSO and PSA for (a.) DRR and (b.) VCA with different thresholds of CC.



Classification Efficiency ξ of PPSO

Fig. 4. The efficiency comparisons between PPSO and PSA with different thresholds of CC.

III. Experimental Results

A plantation area in Au-Ku on the east coast of Taiwan was chosen for investigation. The image data was obtained by the PacRim II project. The proposed PPSO was applied to 44 bands. The criterion for calculating the classification accuracy of experiments was based on exhaustive test cases. Eighteen correlation coefficient thresholds, $CC = 0.7 \sim 0.92$ with a offset of 0.01, were selected to carry out the experiments. CUDA, MPI and OpenMP were used to implement PPSO. The parameters obtained by PSA are as the same as reported in Ref. [4]. The parameters used for PPSO are initialized as follows. The *inertia weight* ω is set as 0.4. The acceleration constants (*cognitive confidence coefficients*) c_1 and c_2 are both equal to 2. All of the multiple combinations of parameters stated above are averaged to obtain the experimental results. The *variance of classification accuracy* (VCA) and a *dimension reduction rate* (DRR), defined in Eq. 4, are used to compare the performances between PPSO and PSA.

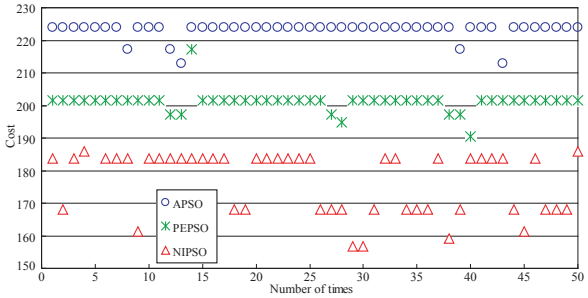


Fig. 5. The cost comparisons among NIPSO, PEPSO and APSO with 50 operations.

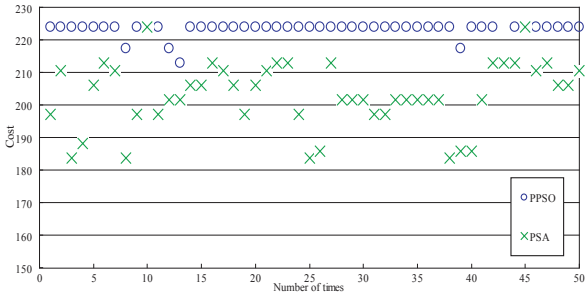


Fig. 6. The cost comparisons between PPSO and PSA with 50 operations.

$$DRR = \frac{m_t - n_k}{m_t} = \frac{\sum_{l=1}^{n_k}(m_l - 1)}{m_t}, \quad (4)$$

where m_l and n_k represent the number of bands in modular eigenspaces Φ_l^k, and the total number of modular eigenspaces in a GME set. m_t is the total number of original bands (i.e. $m_t = \sum_{l=1}^{n_k} m_l$) [1].

Fig. 3 summarizes the evaluation results of qualities of solutions (costs) to illustrate the validity of proposed PPSO. The criteria for the evaluation in Fig. 3 (a.) and (b.) are based on the same quality of solutions and the same period of time experimented with different benchmarks. Furthermore, an evaluation of classification efficiency $(CE = \xi)$,

$$\xi = \frac{DRR}{VCA}, \quad (5)$$

as shown in Fig. 4, is also designed to validate the significant contributions of proposed PPSO. The results appeared in Fig. 4 show that an efficient critical point around $CC = 0.87$ can be reached to obtain a high quality DRR along with a lower VCA impact.

Interestingly, a cost comparison among proposed NIPSO, PEPSO and APSO is illustrated in Fig. 5. The APSO has the best performance than the two others. A comparison of solution quality between PPSO and PSA is also measured in the experiment. The experimental results also disclose that PPSO outperforms PSA in terms of both the qualities of solutions and the trend to get the global best solutions as shown in Fig. 6.

IV. Conclusions

This paper presents a novel PPSO technique using parallel computing techniques for band selections of HSI. The PPSO is introduced to overcome the lack of memory of PSA method and further extend the search and convergence abilities in the solution space by delivering parallel computing on demand in a swarm to reach a global optimal or near-optimal solution. The intrinsic parallel characteristics embedded in PPSO is very suitable for a parallel implementation. In this paper, three parallel PSO implementations, namely NIPSO, PEPSO and APSO, are carried out by the CUDA GPU computing, MPI library and OpenMP API techniques. The proposed PPSO uses only one velocity operator to deal with the search process compared to traditional heuristic optimization techniques.

Encouraging experimental results showed that the proposed PPSO can significantly improve the computational loads and provide a more reliable quality of solution compared to the PSA method. The proposed CE provides an objective evaluation to determine a suitable value of CC, and to obtain a high quality DRR accompanied with a lower VCA impact. Besides the subjects discussed in this paper, how to find the best tradeoff among global search, accuracy, and computational cost will be the issues of our future studies.

Acknowledgment

This work was supported by the National Science Council, Taiwan, under Grant No. NSC 97-2116-M-027-002.

References

[1] Y. L. Chang, C. C. Han, K. C. Fan, K. S. Chen, C. T. Chen, and J. H. Chang, "Greedy modular eigenspaces and positive boolean function for supervised hyperspectral image classification," *Optical Engineering* **42, no. 9**, pp. 2576–2587, 2003.

[2] Y. L. Chang, J. P. Fang, H. Ren, W. Y. Liang, and J. N. Liu, "A simulated annealing feature extraction approach for hyperspectral images," in *IEEE Int. Geosci. Remote Sensing Symposium (IGARSS'07)*, pp. not–available, 2007.

[3] S.-Y. Lee and K. G. Lee, "Synchronous and asynchronous parallel simulated annealing with multiple markov chains," *IEEE Trans. Parallel and Distributed Systems* **7, no. 10**, pp. 993–1008, 1996.

[4] Y.-L. Chang, J.-P. Fang, H. Ren, L. Chang, W.-Y. Liang, and K.-S. Chen, "Parallel simulated annealing approach to band selection for hyperspectral imagery," in *IEEE Int. Geosci. Remote Sensing Symposium (IGARSS'08)*, pp. not–available, 2008.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. of the IEEE, Int. Conf. on Neural Networks*, pp. 1942–1948, 1995.

[6] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, JMorgan Kaufmann, San Francisco, CA, 2001.

[7] Y. H. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," *Proc. of the 7th Annual Conference on Evolutionary Programming VII, San Diego*, pp. 591–600, 1998.

[8] N. Jin and Y. Rahmat-Samii, "Parallel particle swarm optimization and finite-difference time-domain (pso/fdtd) algorithm for multiband and wide-band patch antenna design," *IEEE Trans. Antennas Propag.* **53, no. 11**, pp. 3459V–3468, 2005.