

Self-Organizing GA for Crop Model Parameter Estimation using Multi-resolution Satellite Images

Akhter, S.,¹ Sakamoto, K.,¹ Chemin, Y.,² and Aida, K.,¹

¹National Institute of Informatics, Tokyo, Japan and Tokyo Institute of Technology, Yokohama, Japan
E-mail: shamimakter@gmail.com

²International Centre of Water for Food Security, Charles Sturt University, Wagga Wagga, NSW 2650, Australia

Abstract

We present a methodology for estimating the parameters for crop assimilation studies from satellite images. The procedure is optimized with an evolutionary search technique. A Genetic Algorithm (GA) operates well in high-dimensional non-linear domains. However, its parameters must be set in advance. In this paper, we use a self-organizing GA, in which the initial parameters are generated and assigned automatically. Numerical experiments were conducted to analyze the performance of the methodology, and our method's effectiveness on both synthetic and real satellite data was proven. This study shows that the self-organizing GA methodology is better than the conventional GA approach in estimating crop assimilation.

1. Introduction

Information related to a crop such as its growth, water stress, duration, and date of emergence are useful for monitoring agricultural activities. Satellite data provides useful information over a large area. Besides obtaining and using more direct observable data (e.g., land cover, leaf area index, elevation, and evapotranspiration), a challenge for the future is how to obtain non-visible data (e.g., soil characteristics, groundwater depth, and irrigation practices) from satellite images. Ines (Ines and Droogers, 2002a) (Ines and Droogers, 2002b) proposed an inverse modeling scheme to obtain non-visible data through assimilation of crop model data with satellite-observable data. (Chemin and Honda, 2006) implemented real-coded genetic algorithms using data assimilation to fuse spatial content of higher spatial resolution (HSR) with temporal content of lower spatial resolution (LSR) satellite images. The genetic algorithm (GA) is an evolutionary search technique (Holland, 1975) that uses the mechanism of natural selection to search a decision space for optimal solutions. One of its advantages is that it can operate well in high-dimensional non-linear domains. However, its main difficulty is in deciding an appropriate set of parameters (Crepinsek et al., 2000) including population size, number of generations, selection pressure, crossover probability, mutation probability, etc. Several metaevolutionary approaches (Freisleben and Metz, 1996) (Grefenstette, 1986) (Lee and Takagi, 1994) have been used to determine the GA parameters for finding solutions of different evolutionary problems.

The hierarchical GA approach (Abrams, 2003) generates a self-organizing GA (SOGA) (Jeong and Lee, 1998) (Zhang et al., 2009) that modifies the mutation probability, crossover probability, and population size of each generation. Parameter-less GA (PLGA) (Harik and Lobo, 1999) (Lobo and Goldberg, 2004) (Pelikan and Lobo, 1999) is a subclass of SOGA (Mitchell, 2005). It has no mutation, 50% crossover probability, and selection pressure, and it doubles the population size on a given signal of fitness from a competing population running simultaneously. PLGA assigns GA parameters automatically. The work in reference (Chemin and Honda, 2006) puts one GA inside each iteration of another GA. Parameterizing these two intertwined GAs appears to be an evolutionary problem. PLGA meets the entailed requirements and is the basis of a new assimilation methodology. The new methodology implements PLGA with the existing crop assimilation method proposed in ref. (Chemin and Honda, 2006), for automatic generation of suitable GA parameters. We conducted several experiments and found that our method was better than the conventional GA approach in assimilation with synthetic and real satellite data. The experiments used the dataset from ref. (Chemin and Honda, 2006) on a rice field crop, grown from December 2001 to April 2002, and located in Suphan Buri, Central plain, Thailand.

2. Related Work

Many studies have used data assimilation techniques to solve different parameterization search problems

concerning land surface process modeling (Bach and Meuser, 2003) (Rudiger et al., 2007) and crop parameter estimation (Witt and Diepen, 2007) (Olioso et al., 2005). Crop data assimilation with GA has been used for estimating soil hydraulic functions (Ines and Droogers, 2002a) and quantifying irrigation characteristics (Honda and Ines, 2004) (Ines and Honda, 2005). Crop data assimilation with parameter estimation (PEST) has been used for farm irrigation scheduling (Dorji, 2003) and estimating soil hydraulic properties (Jhorar et al., 2002). These studies combined actual evapotranspiration (ETa) data extracted from satellite images with simulated ETa data. Leaf Area Index (LAI) data can also be used to perform assimilation (Ines and Honda, 2005). One problem with this kind of research is lack of satellite data. Usually, LSR data are available on a daily basis and HSR data on a monthly basis. LSR data contains mixed information in one pixel area. Ines proposed an inverse modeling scheme to obtain non-visible data through the assimilation of RS data into the crop growth model (Ines and Droogers, 2002a) (Ines and Droogers, 2002b). Chemin used real-coded GA (Chemin and Honda, 2006) as a merging scheme in the data assimilation algorithm to fuse LSR and HSR satellite images. However, the model contained two intertwined GAs, a pixel GA and an evaluation GA. The pixel GA evaluates the best crop assimilation parameters for an HSR pixel, and the evaluation GA evaluates the best combination of HSR pixel individuals to form an LSR pixel (Chemin and Honda, 2006). Thus, choosing optimal parameters for those GAs is an important issue. Various metaevolutionary approaches and adaptive algorithms have been devised for finding optimal or sub optimal GA parameters (Freisleben and Metz, 1996) (Grefenstette, 1986) (Lee and Takagi, 1994). The metaevolutionary approach was used in ref. (Grefenstette, 1986) to determine the population size, cross-over probability, mutation rate, generation gap, scaling window, and selection strategy. The metaevolutionary approach was also used in ref. (Freisleben and Metz, 1996) for studying the effect of dynamically adaptive population size, crossover, and mutation rate on De Jong's set of test functions. The PLGA approach we present here should be useful for solving the GA parameter selection problems in the existing work (Chemin and Honda, 2006).

2.1 The Simulation Model

The simulation model in ref. (Chemin and Honda, 2006) can monitor country-wide agricultural activity parameters derived from satellite images. Specifically, it uses a crop model and assimilation to

get crop parameters. Actual evapotranspiration (ETa) values are a combination of soil evaporation and plant transpiration values. ETa data are used as known values and can be derived from satellite images by using the Surface Energy Balance Algorithm (SEBAL) (Bastiaansen, 1995). ETa data for the same location on various dates enable crop parameters to be predicted for that location. Satellite images with pixel sizes larger than 1 km generally have a daily return period over the same area of the world, thus enabling ETa maps with such pixel size to be made everyday. Hence, we can analyze agricultural activities in that one-pixel area. However, a smaller pixel size is still required for a smaller area. In that case, we need high-resolution satellite images of the same location on various dates. However, it is a difficult task to get high-resolution images on even a weekly basis. To overcome this problem, the simulation model (Figure 1) uses a fusion of HSR and LSR images. The ETa of an LSR pixel is regarded as the average ETa of HSR images.

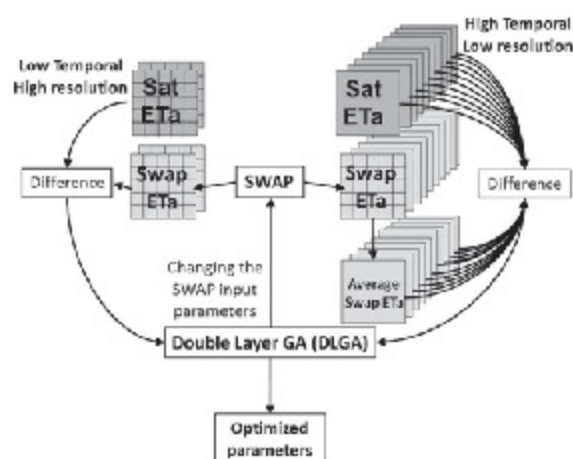


Figure 1: Existing Crop Assimilation Model (Chemin and Honda, 2006)

Thus, the LSR images can be filled with the HSR images. Each HSR pixel generates different individuals with different crop parameters. All individuals are put into the SWAP crop model (Van Dam et al., 1997) and produce a simulated ETa (SimETa) for the satellite image ETa (SatETa) of each corresponding date. Thereafter, each SimETa is compared with SatETa data and the average difference is taken to be the assimilation performance.

$$C_{\text{assim}} = \frac{1}{q} \sum_i^q (ETa_{\text{sat},i} - ETa_{\text{sim},i})^2 \quad \text{Equation 1}$$

$$C_{LSR} = \frac{1}{n} \sum_D \left(\left(ETa_{LSR,D} - \frac{1}{p^2} \sum_{x,y} (ETa_{SWAP_{LSR,D}}) \right)^2 \right) \quad \text{Equation 2}$$

$$F = \frac{1}{C_{LSR} + \frac{1}{p^2} \sum_{x,y} C_{Sim}} \quad \text{Equation 3}$$

Equations (1) and (2) are the cost functions of HSR and LSR. In (Equation 1), $ETa_{HSR,d}$ and $ETa_{SWAP_{HSR,d}}$ respectively denote the SatETa and SimETa obtained from an HSR image of location coordinate (x,y) on date $d \in [d_1, \dots, d_n]$. In (Equation 2), $ETa_{LSR,D}$ indicates the SatETa obtained from an LSR image taken on date $D \in [D_1, \dots, D_n]$. p is the number of rows or the number of columns in one HSR image. An HSR image contains p^2 pixels for an area presented by one pixel in an LSR image. Thus, $\frac{1}{p^2} \sum_{x,y} (ETa_{SWAP_{LSR,D}})$ computes the average SimETa of p^2 HSR images. The total fitness, F , is defined in (Equation 3). One pixel GA is evaluated for each HSR pixel, and (Equation 1) is used to calculate the cost. However, a problem arises when calculating the cost in (Equation 2). SimETa for LSR contains p^2 pixels, and each of these pixels contains k individuals. Hence, we need to select the best combination of p^2 individuals from kp^2 individuals. The evaluation GA solves the selection problem by calculating the cost (Equation 2) and fitness of all individuals (Equation 3). The pixel GA recreates individuals using GA operators based on the fitness. Good individuals, i.e., those with higher fitness, are selected for the next generation.

$$Max_fitness = \max(B(s_y, I), B(s_{y+1}, I)) \quad \text{Equation 4}$$

$$N \rightarrow \left[\left(\left(\max_fitness \leftrightarrow s(s_{y+1}, I) \right) \wedge s(s_y, I) \right) \rightarrow s(s_{y+N}, I+1) \right] \quad \text{Equation 5}$$

2.2 Parameter-Less GA (PLGA)

Setting the GA parameters such as population size, selection rate, crossover rate, and mutation rate usually requires a lot of experiments. To search for suitable GA parameters, we use the rules from ref. (Lobo, 2000), where the preset and fixed selection pressure s is 0.4 and the crossover probability P_c is 0.5. The typical relationship between population size and computational time is shown in (Figure 2). PLGA is helpful in this situation because it simultaneously runs a number of GAs with fixed

selection pressure and crossover probability values and maintain certain predefined rules.

A simple GA with a small population converges faster than one with a large population. However, the quality of the solution is weaker with smaller populations. Conversely, a GA with a larger population needs more computational time. Thus, a suitable or optimal balance between computation time and population size should be found (Lobo, 2000).

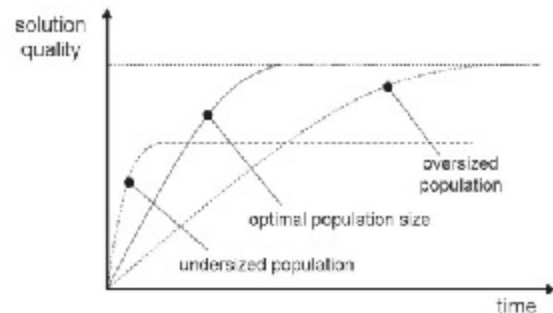


Figure 2: The Relationship between Population Size and Computation Time (Lobo, 2000)

The way to select a suitable population size is thus to set a small size initially and increase it by using the following heuristics. Let G be the group of GA processes using similar GA parameters and i be the indicator of the specific G . t is a positive integer and $t > 2$. $F_{avg}G_i$ is the average fitness of G_i .

- 1) If the population size of G_i is N_i , set the population size of G_{i+1} to $2N_i$.
- 2) Increment the t -array counter at each step (Lobo, 2000). Table 1 shows an example of the counter increment system (with $t=2$).
- 3) The position of the most significant digit that changed during the increment operation indicates which G should be run.
- 4) If $F_{avg}G_{i+1} > F_{avg}G_i$, delete G_i .

Table 1: Example of t -array Counter Algorithm

Counter (base $t=2$)	Most Significant Digit Changed	Action
0		
1	1	Run 1 generation of G_1
10	2	Run 1 generation of G_2
11	1	Run 1 generation of G_1
100	3	Run 1 generation of G_3
101	1	Run 1 generation of G_1
110	2	Run 1 generation of G_2
111	1	Run 1 generation of G_1
1000	4	Run 1 generation of G_4
...

The rules can eliminate the time penalty that occurred for the oversized population in (Figure 2).

However, they are not applicable when a mutation operator is used.

Thus, our PLGA implementation does not use mutation operator in order to ensure convergence of the populations (Lobo, 2000).

3. Methodology

In the assimilation model presented in ref. (Chemin and Honda, 2006), it is relatively difficult to find a suitable population size and generation number in a few experiments. Finding the best combination on all GA parameters is a challenging issue when two GAs are run hierarchically. Thus, the proposed methodology runs the existing assimilation model (Chemin and Honda, 2006) but automatically generates suitable GA parameters (population size and generation numbers). PLGA is applied to the pixel GA. Our preliminary experiments show that the evaluation GA's performance is not significantly affected by the selection of GA parameters. Let S be a set of parameters {popsize, generation} for PLGA and let it be assigned to a specific CAM (an existing Crop Assimilation Model (Chemin and Honda, 2006)) for evaluation and $S_y = \{2^y, x^*\}$. As a load-balancing strategy for the CAMs, the generation number (x^*) in S , is assigned such that the product of the population size and generation remains the same for all PLGA populations. N is the total number of populations in PLGA. We can implement a PLGA population replacement mechanism as in (Equation 5). It tells us that at the I^{th} iteration, for each consecutive population's pair, the new population $B(S_{y+1}, I+1)$ for the next iteration ($I+1$) will be created only if the maximum average fitness population, Max_{pop} , is $B(S_{y+1}, I)$ and the minimum average fitness population $B(S_y, I)$ is deleted. The $Max()$ function in (Equation 4) gives the maximum fitness population, and the $R_{\text{del}}()$ function removes the population (with the minimum fitness). The logical operator \Leftrightarrow is the "biconditional" operator, and \wedge is the logical "and" operator. \rightarrow is the "implies" symbol and represents that only a true expression on the left side causes the expression on the right side to be true.

Our PLGA scheme (Figure 3) runs three consecutive CAMs, where $i=1, 2$, and 3. The steps are as follows:

- 1) First, the input files for each CAM_{*i*} are created. The input files contain the GA parameters according to the rules of PLGA.
- 2) Each CAM_{*i*} runs several times (five times in our case) and the average fitness is generated.

3) All CAMs create output files when they finish their computations. The output files contain the results of CAM_{*i*}, such as the fitness of their populations, the gene values of all individuals, etc. The average fitness values ($F_{\text{avg}}\text{CAM}_i$) are compared. The CAM_{*i*}, to be run or to be eliminated, is selected, and a new set CAM_{*i+1*} is generated. Table 2 lists the four cases for comparing average fitness values and their corresponding decisions. The generation number in CAM_{*i*} ($1, x$, and x^2) is assigned from largest to smallest population.

Table 2: The Comparison cases and decisions

Comparison	Decision
$F_{\text{avg}}\text{CAM}_1 > F_{\text{avg}}\text{CAM}_{i+1} > F_{\text{avg}}\text{CAM}_{i+2}$	No Deletion. Continuously run all the CAMs again with their previous assigned generation number.
$F_{\text{avg}}\text{CAM}_1 > F_{\text{avg}}\text{CAM}_{i+1} < F_{\text{avg}}\text{CAM}_{i+2}$ or $F_{\text{avg}}\text{CAM}_1 < F_{\text{avg}}\text{CAM}_{i+1} < F_{\text{avg}}\text{CAM}_{i+2}$	Delete CAM _{<i>i</i>} and CAM _{<i>i+1</i>} . Additionally create two new CAMs as CAM _{<i>i+3</i>} , CAM _{<i>i+4</i>} with 2^{i+2} and 2^{i+3} population size correspondingly.
$F_{\text{avg}}\text{CAM}_1 < F_{\text{avg}}\text{CAM}_{i+1} > F_{\text{avg}}\text{CAM}_{i+2}$	Remove CAM _{<i>i</i>} and create CAM _{<i>i+3</i>} with 2^{i+2} population size.

4. Results

4.1 Experiment 1: Setting up the PLGA

We experimented on eight GA parameters used in both the pixel GA and evaluation GA (Chemin and Honda, 2006): population size per pixel (popsize), generation limit (generation), probability of crossover (pcrossover), probability of mutation (pmutation), and in the evaluation GA, population size (s popsize), generation limit (s generation), probability of crossover (s pcrossover), and probability of mutation (s pmutation). Figures 4 and 5 show the evaluation GA and the pixel GA fitness values of changing one GA parameter when all other above parameters (except the one to be changed) are fixed. It is clear from the figures that the fitness values are affected by the parameters.

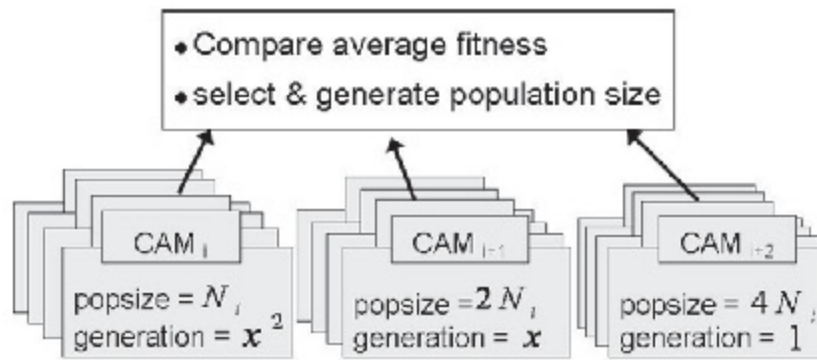


Figure 3: Proposed CAM with PLGA

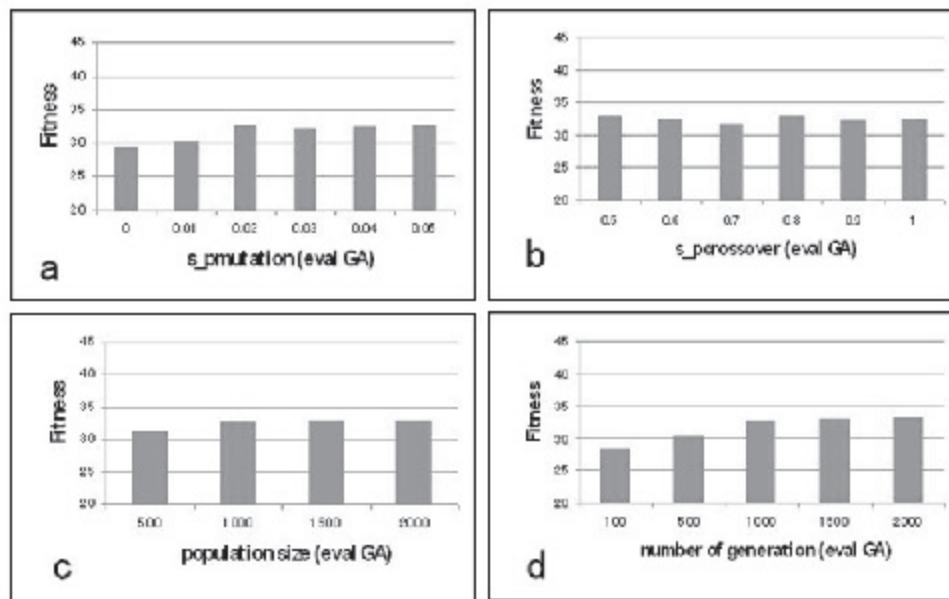


Figure 4: Exploring Evaluation GA Parameters and Their Effect on the Fitness

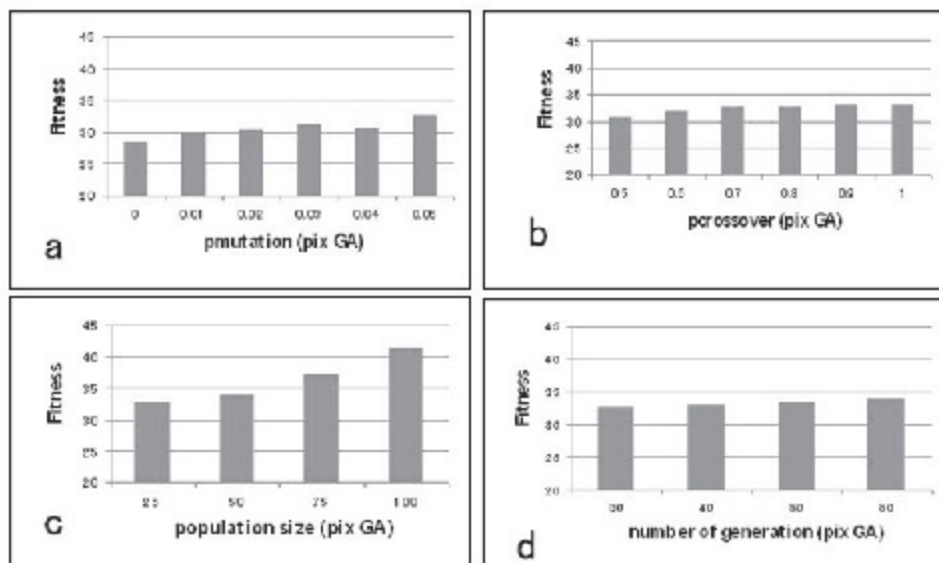


Figure 5: Exploring Pixel GA Parameters and Their Effect on the Fitness

Determining the best GA parameter combination is also an evolutionary problem. Thus, PLGA is very helpful in determining the best GA parameter combination. For instance in (Figure 4c) (popsize of the pixel GA is fixed to 25), the fitness value does not change significantly as a result of changing the popsize of the evaluation GA. However, in (Figure 5c) (popsize of the evaluation GA is fixed to 1000), the fitness value is significantly affected by the popsize of the pixel GA. These findings led us to implement PLGA only in the pixel GA.

$$simSta_{LSR}(Date) = \frac{1}{n_pix} \sum_{pix=0}^{n_pix-1} simSta_{HSR}(Date, pix) \quad \text{Equation 6}$$

$$simSta_{with\ noise}(Date) = simSta(Date) + \alpha \cdot simSta(Date) \\ \alpha = \beta \cdot Rand(-1, 1) \quad \text{Equation 7}$$

$$AE_{gene} = \frac{1}{n_gene} \sum_{k=0}^{n_gene-1} \left(\frac{1}{n_pix} \sum_{pix=0}^{n_pix-1} \|Sim(pix, k) - Syn(pix, k)\| \right) \quad \text{Equation 8}$$

4.2 Experiment 2: CAM with PLGA on Synthetic Data

Four pixels worth of synthetic data were created for this experiment. Three different sets of parameters for Ground Water Level in January (GWJan), Ground Water Level in December (GWDec), Date of Emergence of Crop (DEC), Start irrigation (STS), Time Extent of Crop (TBC) were generated from the SWAP model (Table 3). The units for GWJan and GWDec are cm, and the rest use Date of Year (DOY). A given gene corresponds to an unknown crop parameter in the GA evaluation. The other parameters of the SWAP model were taken from ref. (Chemin and Honda, 2006). The SWAP model was run with three sets of input data (Set¹, Set², Set³), and it generated three sets of yearly ETa (ETa¹, ETa², ETa³) output data. We selected 14 dates and their corresponding ETa data from January to April (Table 3). Three sets (A, B, and C) of four-pixel HSR images on four different dates (2002/2/28, 2002/3/15, 2002/3/31, and 2002/4/21) were created from the input data (Table 3). In set A, all pixels (pix 0, pix 1, pix 2, pix 3) were made from ETa¹. In set B, pix 0 and pix 1 were made from ETa¹, while pix 2 and pix 3 were made from ETa². In set C, pix 0 and pix 1 were made from ETa¹, pix 2 from ETa², and pix 3 from ETa³. One-pixel LSR images were created in the same way (one image for set A, one for set B, and one for set C). The LSR pixel was produced from the corresponding HSR image pixels by using (Equation 6).

To make the data more realistic, a small amount of noise was added to the HSR and LSR images by using (Equation 7). n_pix in (Equation 6) is the total number of HSR pixels, and n_gene in (Equation 8) is the available number of genes. $Sim(pix, k)$ represents the simulated value of gene k in pix number pixel and $Syn(pix, k)$ represents the synthetic value of gene k in pix number pixel. The simulated assimilation curves for all sets (A, B, C) for the HSR and LSR images are impressive. In all three cases, the fitness values meet the convergence requirement (0.1 cm/day). However, the suitable popsize and generation are different in all cases.

Table 3: Crop parameters and ETa values

Input Data	Set ¹	Set ²	Set ³
GWJan (cm)	60	90	73
GWDec (cm)	88	75	52
DEC (DOY)	8	10	12
STS (DOY)	25	30	62
TBC (DOY)	108	120	142
Date	ETa ¹	ETa ²	ETa ³
2002/1/08	0.328	0.332	0.332
2002/1/15	0.044	0.044	0.044
2002/1/17	0.035	0.035	0.035
2002/1/27	0.084	0.065	0.049
2002/2/08	0.236	0.222	0.205
2002/2/16	0.340	0.332	0.325
2002/2/23	0.342	0.339	0.338
2002/2/28	0.366	0.364	0.365
2002/3/13	0.397	0.395	0.396
2002/3/15	0.396	0.426	0.396
2002/3/29	0.385	0.402	0.415
2002/3/31	0.449	0.474	0.496
2002/4/14	0.079	0.100	0.113
2002/4/21	0.035	0.041	0.053

The suitable popsize is 1600 for set A, 800 for set B, and 400 for set C. Next, the synthetic pixels were generated from Set¹, Set² and Set³. We ran the simulation model under the assumption that its accuracy would be proven if the simulated outputs and the Set¹, Set², Set³ values were similar. The simulated results (for unknown parameters values) with sets A, B and C are presented in Figure 6. Equation 7 was used to calculate the average error (AE_{gene}). In the case of set A, the AE_{gene} of each gene value is 20. However, in the case of set B (set C), AE_{gene} is 6 (5.5). The LSR image covers a large area (1000 m x 1000 m) with one pixel.

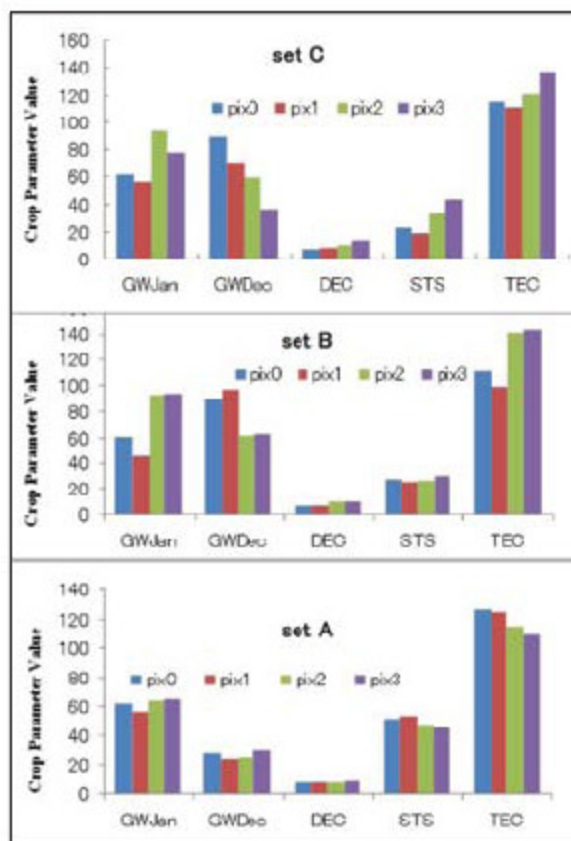


Figure 6: Simulated Output Data (set A, B, C)

Hence, it is impossible in practice that the four pixels would have similar input values. Thus, a situation like that of set A would rarely occur. However, the outcomes for the other two cases are reasonable and establish the model's accuracy. Additionally, the proposed CAM with the PLGA model gives better feedback for finding every gene value in comparison the previous CAM without the PLGA model. The average genes mean square error deviations of the proposed and previous CAM is 26.4%.

4.3 Experiment 3: CAM with PLGA on Real Data

The total area of Suphan Buri province is 5,358 square kilometers. Its map contains around 5,358 pixels at 1000 meter resolution. We used ETa data taken from ref. (Chemin and Honda, 2006). The LSR image was a Terra-MODIS (1000 m x 1000 m) eight-day product from 2002. The HSR images were Landsat 7 ETM+L1B (60m x 60m) standard one-day products taken on 08th January 2002, 14th April 2002 and Terra-ASTER (90 m x 90 m) one-day products taken on 16th February 2002. The SWAP model input files were the same as those used in ref. (Chemin and Honda, 2006).

Following the suggestions in ref. (Chemin and Honda, 2006), the minimum population size was 1000 and the maximum generation was 1000 in the evaluation GA, for CAMs with and without the PLGA model. CAM with the PLGA model started with a population size of 25 for the pixel GA and a crossover probability of 0.5. CAM without the PLGA model used a fixed population of 25, generation number of 30, crossover probability of 0.8, and mutation probability of 0.05. Proper LSR pixel assimilation depends on the HSR pixels' assimilation and the evaluation procedure to select the perfect combination of HSR individuals to form an LSR pixel. Regarding the existing crop assimilation model, the experiments on HSR pixels had consistent values for the simulated ETa data. We believe that the crossover and mutation values used in the evaluation GA were not properly considered in the previous implementation (Chemin and Honda, 2006). In the current experiments we varied the crossover and mutation probabilities in order to obtain enough diversity as in actual satellite ETa data. Figures 7, 8 and 9 present the effect of CAM with PLGA (population of 200 and 30th generation) on the ETa assimilation of the 25 pixel HSR images.

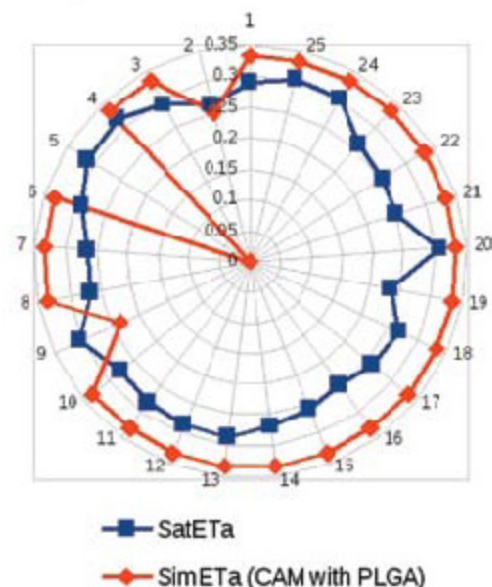


Figure 7: The 25 HSR pixel (08 January 2002)

The assimilation results in Figure 7 and Figure 8 seem more accurate than those in Figure 9. The rainy season, which started in the last week of April, and the absence of proper ground water information for these periods made it hard for the evaluation model to estimate the appropriate assimilation in Figure 9. Figure 10 shows the ETa assimilation of the LSR pixel.

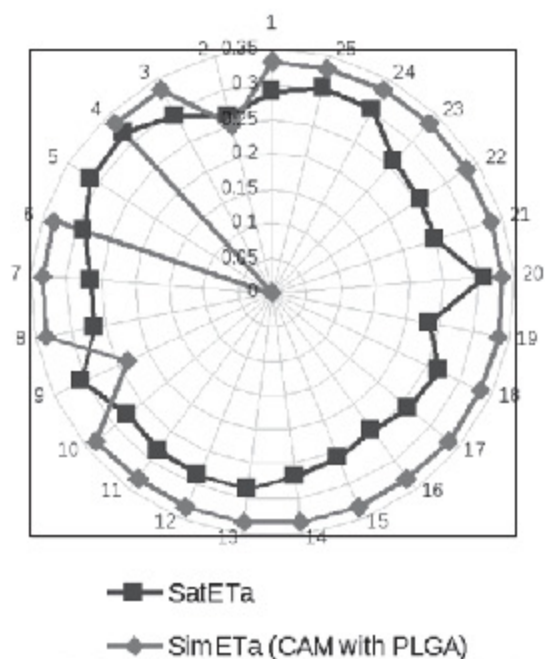


Figure 8: The 25 HSR pixel (16 February 2002)

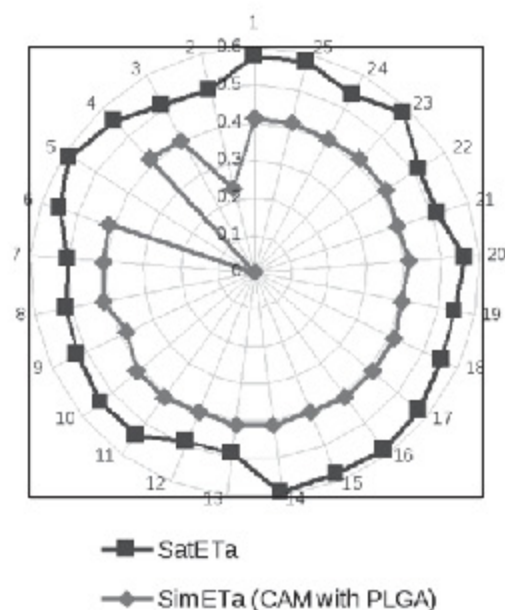


Figure 9: The 25 HSR pixel (14 April 2002)

The crop assimilation with PLGA performs better. According to ref. (Chemin and Honda, 2006), the LSR differences were not satisfactory in CAM without the PLGA model with only two HSR images because the rainy season started in the last week of April and fields that were underwater after sowing in January. However, the situation could be remedied with an additional HSR image that provided more information for finding more

accurate LSR ETa values. Figure 10 shows the output when an HSR image from 14 April 2002 was added to the input. The assimilation is better in this case. Another problem is how to determine an appropriate population and generation number to simulate the GA. PLGA can solve this problem by finding an appropriate population and generation number within a given time frame.

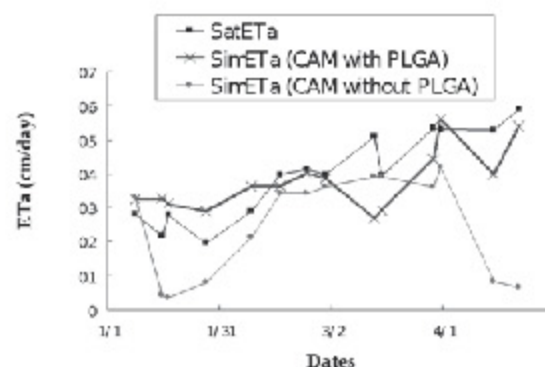


Figure 10: The LSR ETa assimilation curves by CAM

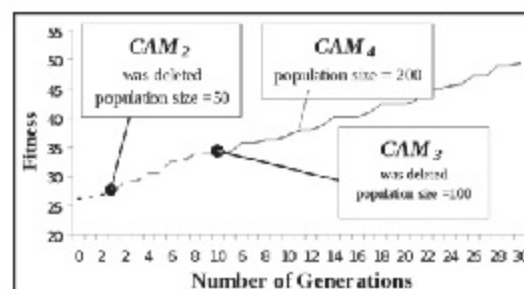


Figure 11: The Process of Increasing Fitness

Figure 11 presents the process by which fitness was increased. When CAM₃ was 10 generations, its average fitness $F_{avg}CAM_3$ was exceeded by $F_{avg}CAM_4$, the average fitness of CAM₄. Therefore, CAM₃ was eliminated and CAM₄ was run. The program stopped when CAM₄ reached 30. $F_{avg}CAM_4$ was never overtaken by the other average fitnesses. Therefore, the population of 200 was the best for the given time of the experiment. We could not know that information until we ran PLGA. Figure 12 presents the outcomes of CAM with PLGA with a population 200 and 30 generations. The values are similar to those in ref. (Chemin and Honda, 2006). Regarding the optimized parameters of the depth to ground water, GWJan is about 60 cm and GWDec is about 50–80 cm; these values are typical of rice cropping areas. The optimized Date of Emergence of Crop (DEC) is within the range of 2–9 DOY.

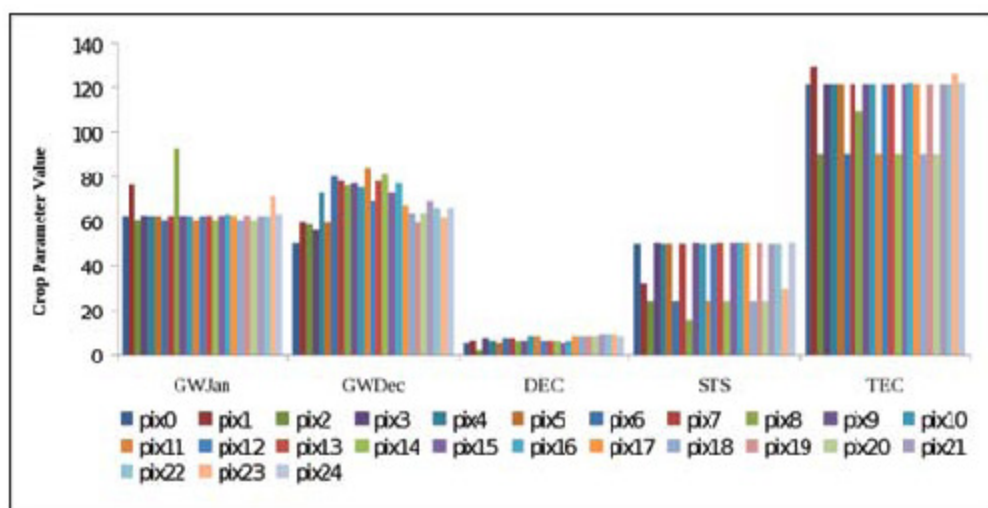


Figure 12: Target Outputs Generated from Proposed CAM with PLGA

The range of sowing dates was from the middle to the end of December 2001. Thus, a 10 day distribution is possible. Start Irrigation Scheduling (STS) values were from January 15 to February 19 (15~50 DOY). The TEC value was between 90~129 DOY. DOYs of three and half months are typical in rice cultivation. However, sometimes the crop times can be extended. Hence, these TEC values are reliable for rice crop areas.

4.4 Run Time Analysis

The computation time of the CAM without PLGA (Chemin and Honda, 2006), T_1 , is determined by (Equation 9). Here, $\text{ind}(i)$ is the number of individuals in CAM_i , and it is computed by multiplying the number of HSR pixels with the population size in the pixel GA. $\text{CAM}(t)$ denotes the computation time of the CAM process, or the evaluation of SimETa for one individual. We used a computer with a Xeon E5410 Quad core 2.33 GHz x 2 and Linux (kernel 2.6.18). The $\text{CAM}(t)$ simulation took from 0.2 ~ 1.5 [sec]. The computational time of the evaluation GA, $\text{inGA}(t)$, maintains the steady behavior because of the fixed parameters. Our preliminary experiments show that $\text{inGA}(t)$ takes about 50~60 [sec] with a population size of 1000 and 1000 generations. $\text{gen}(i)$ is the number of generations in the pixel GA. The total CAM computation time of one generation is $\text{CAM}(t) \cdot \text{ind}(i)$. the CAM in ref. (Chemin and Honda, 2006) was executed N_{times} times, and T_{avg} in (Equation 10) is the total run time to get the average fitness.

$$T_i = \text{CAM}(t) \cdot \text{ind}(i) + \text{inGA}(t) \cdot \text{gen}(i) \quad \text{Equation 9}$$

$$T_{\text{avg}} = N_{\text{times}} \cdot T_i \quad \text{Equation 10}$$

$$E = \frac{T_1 / T_R}{n} = \frac{1684000 / 13230}{150} = 74\% \quad \text{Equation 11}$$

Let N^* be the population size required by the GA to reach a certain target solution. In the worst case, the time spent by the CAM with PLGA in order notation is $O(T_{\text{avg}}, N^*)$ (Lobo, 2000), which is about N^* times slower than a CAM without PLGA that starts with an optimal population size. We implemented our CAM with PLGA (three consecutive existing CAM methods) with an initial population of 25 in the pixel GA and a population of 1000 and 1000 generations in the evaluation GA. After 15 iterations, PLGA finds that the population of 200 gets the best fitness within a run time of around two weeks. This time is too long for real-time applications. However, the fitness evaluations of each population size (2, 4, 8, ..., N^*) are independent in CAM with PLGA. Thus, a parallel or distributed implementation can reduce the total computational time and the evaluation GA can be replaced with another PLGA. To confirm whether this method is practical, the CAM with PLGA was implemented on a parallel computing testbed, called InTrigger (InTrigger, 2009), with a total of 150 CPU cores. The run time fell to the order of an hour (about four hours). Thus, from (Equation 11), we can say that the parallel implementation (T_n) with 150 CPU cores gains a 74% performance improvement (E) over that of one CPU (T_1). Thus, the evaluation time bottleneck for CAM with PLGA

can be removed by using high performance computing (HPC) platforms. Additionally, the SWAP executions of each CAM without PLGA are also independent. Thus, a hierarchical parallel implementation (Akhter et al., 2008) would efficiently reduce the execution time of CAM with PLGA, and we plan to implement one in the near future.

5. Discussions

Most GA researchers focus on the operators' (crossover, mutation etc.) probability and their adaptation but not so much on population size (Lobo, 2000). However, population size is one of the most crucial parameters. Additionally, in the existing CAM without PLGA, it is possible to trace the suitable crossover probability [0.1~0.9] and mutation probability [0.1~0.9] values throughout the experiment. However, it is relatively difficult to trace the suitable population size and generation numbers within a few experiments. Finding the best combination of all is a challenging issue when two GAs are run hierarchically. That's the main reason not to apply conventional Meta-GA concepts. In adaptive population size GA (APGA), the population size parameter is eliminated at the cost of introducing two new ones, the minimum and maximum lifetimes of newborn individuals (Black et al., 2000). It seems better than PLGA in performance (Eiben and Schut, 2008). However, if the GA's performance is sensitive to these new parameters, such a parameter replacement can make things worse, and it is still unknown what their effects are in the existing CAM without PLGA implementation. The PLGA algorithm is simple and more user friendly. It keeps the GA out of a local optimum and helps it reach the global optimum solution without any difficulty. Additionally, PLGA works well with fixed selection pressure, crossover probability, and no mutation. Moreover, as was proved in ref. (Lobo, 2000), the PLGA approach is good for finding solutions in complex search problems, such as the minimum Steiner tree problem, without having to worry about the GA parameter settings. This paper applies PLGA to the problem of crop assimilation. However, PLGA was used in place of pixel GA since the fitness value was shown to be affected more by the population size of the pixel GA. The evaluation GA was not changed, and hence, its contribution to the time complexity remained the same in the experiment. CAM with PLGA overcomes the assimilation problem of CAM without PLGA (Chemin and Honda, 2006) with an additional HSR image from the end of the season (Figure 10). Thus, the use of more HSR images will help the LSR to converge around the date.

The weight of the assimilation fitness is more geared towards HSR than to LSR. In the future, we plan to study a fusion fitness equation that uses weights to balance the HSR and LSR contributions. Our approach works well with the pixel GA and is helpful for finding a suitable population size and generation number to find out the best fitness within a given amount of time, and we believe the known values can be set to the existing crop assimilation model for every re-assimilation purpose.

6. Conclusion

PLGA can find suitable GA parameters for an application. The CAM with PLGA model's performance was evaluated with both synthetic and real data and the experimental study shows that the model can provide relatively better results on the assimilation process. Experiments with three different sets of synthetic data proved that the optimal population size is unreliable due to the input image value changes. CAM with PLGA can find the optimal population size and make the convergence more efficient. Its accuracy with synthetic data is good enough for practical applications. CAM with PLGA performs impressively with real data, and the outcomes are in an acceptable range. However, the total computational time with this model is much longer than that of CAM without PLGA with known optimal GA parameters. A parallel and distributed implementation can solve this problem. However, PLGA still requires a proper load balance with a suitable distribution mechanism. Grid computing (Grid Computing, 2009) with its large computational resources can reduce the run time of a province-sized experiment to just a few days.

References

- Ahram, J. P., 2003, A Hierarchical Genetic Algorithm for the Traveling Salesman Problem, *Honors Project*, Winter, Carleton University, Computer Science.
- Akhter, S., Osawa, K., Nishimura, M., and Aida, K., 2008, Experimental Study of Distributed SWAP-GA Models on the Grid, *IPSJ Transactions on Advanced Computing Systems*, 1, 193-206.
- Black, T., Eiben, A. E., and van der Vaart, N. A. L., 2000, An Empirical Study on GAs Without Parameters, *Proceedings of the 6th Conference on Parallel Problem Solving from Nature*, 1917 in Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, New York, 315-324.

- Bach, H., and Mauser, W., 2003, Methods and Examples for Remote Sensing Data Assimilation in Land Surface Process Modeling", *IEEE Transaction on Geoscience and Remote Sensing*, Vol. 41, No 7, 0196-2891/03.
- Bestiassensen, W. G. M., 1995, Regionalization of Surface Flux Densities and Moisture Indicators in Composite Terrain. A Remote Sensing Approach under Clear Skies in Mediterranean Climates, *Agric. Res. Dept., Wageningen, The Netherlands*, Report 109.
- Chemmin, Y., and Honda, K., 2006, Spatiotemporal Fusion of Rice Actual Evapotranspiration with Genetic Algorithms and an Agrohydrological Model," *IEEE Transactions on Geoscience and Remote Sensing*, 44(11), 3462-3469.
- Crepinsek, M., Mernik, M., and Zumer, V., 2000, A Metaevolutionary Approach for the Travelling Salesman Problem, *22nd Int. Conf. Information Technology Interfaces III 2000*, Pula, Croatia.
- Dorji, M., 2003, Integration of SWAP Model and SEBAL for Evaluation of on - Farm Irrigation Scheduling with Minimum Field Data, *Enschede, ITC*, 100.
- Eiben, A. E., and Schut, M. C., 2008, New Ways to Calibrate Evolutionary Algorithms, Siarry, P. and Michalewicz, Z. (eds.), *Advances in Metaheuristics for Hard Optimization*, Natural Computing Series, Springer 153-177.
- Freisleben, B., and Merz, P., 1996, New Genetic Local Search Operators for the Traveling Salesman Problem, in *proceedings of Parallel Problem Solving from Nature IV*, 890-899.
- Grefenstette, J. J., 1986, Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man & Cybernetics SMC-16*, No. 1, 122-128.
- Grid Computing, 2009, http://en.wikipedia.org/wiki/Grid_computing
- Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley.
- Harik, G., and Lobo, F., 1999, A Parameter-Less Genetic Algorithm, *IlligAL Report No. 99009*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
- Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press.
- Honda, K., and Ines, A. V. M., 2004, Genetic Algorithms in Quantifying Water Management and Agricultural Practices at the Sub-Pixel Level, *Proceedings of the 6th International Conference on Hydroinformatics*, 2, 1319-1325.
- Ines, A. V. M., and Droogers, P., 2002a, Inverse Modeling in Estimating Soil Hydraulic Functions: A Genetic Algorithm Approach, *Hydrology and Earth System Sciences*, 6(1), 49-65.
- Ines, A. V. M., and Droogers, P., 2002b, Inverse Modeling to Quantify Irrigation System Characteristics and Operational Management, *Irrigation and Drainage Systems* 16, 233-252.
- Ines, A. V. M. and Honda, K., 2005, On Quantifying Agriculture and Water Management Practices from a Low Spatial Resolution RS Data using Genetic Algorithms: A Numerical Study for Mixed-Pixel Environment, *Advances in Water Resources* 28, 856-870.
- InTrigger, 2009, <http://www.intrigger.jp/wiki/index.php/InTrigger>
- Jeong, I., and Lee, J., 1998, A Self-Organizing Genetic Algorithm for Multimodal Function Optimization, *Artificial Life and Robotics Journal*, Springer Japan, Vol. 2, No. 1, 48-52.
- Jhorar, R. K., Bastiassensen, W. G. M., Feddes, R. A., and Van Dam, J. C., 2002, Inversely Estimating Soil Hydraulic Functions using Evapotranspiration Fluxes, *Journal of Hydrology*, 258, 198-213(16).
- Lee, M., and Takagi, H., 1994, A Framework for Studying the Effects of Dynamic Crossover, Mutation and Population Sizing in Genetic Algorithms, *Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms, lecture notes in Artificial Intelligence* 101, 111-126.
- Lobo, F. G., 2000, The Parameter-Less Genetic Algorithm: Rational and Automated Parameter Selection for Simplified Genetic Algorithm Operation, *Doctoral Dissertation*, Universidade Nova de Lisboa, Lisboa.
- Lobo, F. G., and Goldberg, D. E., 2004, The Parameter-Less Genetic Algorithm in Practice, *Information Sciences*, 167, 217-232.
- Mitchell, G., 2005, Quality-Time Tradeoff in a Distributed Parameter-Less Genetic Algorithm", *AIA*, http://elm.eeng.dcu.ie/~alife/gmit/pubs/TSP_AI_A_2005a.pdf
- Olioso, A., Inoue, Y., Ortega-Farisa, S., Demarty, J., Wigneron, J. P., Braud, I., Jacob, F., Lecharpentier, P., Otile, C., Calvet, J. C., and Brisson, N., 2005, Future Directions for Advanced Evapotranspiration Modeling: Assimilation of Remote Sensing Data into Crop Simulation Models and SVAT Models", *Irrigation and Drainage Systems*, 19, 377-412.

- Pelikan, M., and Lobo, F., 1999, Parameter-Less Genetic Algorithm: A Worst-Case Time And Space Complexity Analysis, *HHGAL Report No. 99014*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
- Rudiger, C., Calvet, J., Mahfour, J., Jarlan, L., Balsamo, G., and Sabater, J. M., 2007, Assimilation of Land Surface Variables for Model Initialisation at Meteo-France", *Proceedings of the International Congress on Modelling and Simulation (MODSIM)*, 1716-1722.
- Van Dam, J. C., Huygen, J., Wesseling, J. G., Feddes, R. A., Kabat, P., Van Walsum, P. E. V., Groenendijk, P., and Van Diepen, C. A., 1997, Theory of SWAP Version 2.0: Simulation of Water Flow and Plant Growth in the Soil-Water-Atmosphere-Plant Environment, *Technical Document 45*. Wageningen Agricultural University and DLO Winand Staring Centre. The Netherlands.
- Witt, A. J. W. D., and Diepen, C. A. V., 2007, Crop Model Data Assimilation with the Ensemble Kalman Filter for Improving Regional Crop Yield Forecasts", *Agricultural and Forest Meteorology*, Elsevier Publication, 146, 38-56.
- Zhang, J., Zhuang, J., Du, H., and Wang, S., 2009, Self-Organizing Genetic Algorithm Based Tuning of PID Controllers, *Journal of Information Sciences*, Elsevier Publication, 179, Issue 7, 1007-1018.