GENETIC NETWORK PARAMETER ESTIMATION USING SINGLE AND MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

by

KARIM M. MORCOS

B.S., Kansas State University, 2008

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2011

Approved by:

Co-Major Professor
Sanjoy Das

Approved by:

Co-Major Professor
Stephen M. Welch

Abstract

Multi-objective optimization problems deal with finding a set of candidate optimal solutions to be presented to the decision maker. In industry, this could be the problem of finding alternative car designs given the usually conflicting objectives of performance, safety, environmental friendliness, ease of maintenance, price among others. Despite the significance of this problem, most of the non-evolutionary algorithms which are widely used cannot find a set of diverse and nearly optimal solutions due to the huge size of the search space. At the same time, the solution set produced by most of the currently used evolutionary algorithms lacks diversity.

The present study investigates a new optimization method to solve multi-objective problems based on the widely used swarm-intelligence approach, Particle Swarm Optimization (PSO). Compared to other approaches, the proposed algorithm converges relatively fast while maintaining a diverse set of solutions. The investigated algorithm, Partially Informed Fuzzy-Dominance (PIFD) based PSO uses a dynamic network topology and fuzzy dominance to guide the swarm of dominated solutions.

The proposed algorithm in this study has been tested on four benchmark problems and other real-world applications to ensure proper functionality and assess overall performance. The multi-objective gene regulatory network (GRN) problem entails the minimization of the coefficient of variation of modified photothermal units (MPTUs) across multiple sites along with the total sum of similarity background between ecotypes. The results throughout the current research study show that the investigated algorithm attains outstanding performance regarding optimization aspects, and exhibits rapid convergence and diversity.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to begin by thanking my major professor, Dr. Sanjoy Das, for his help and continuous support throughout my graduate studies at Kansas State University. Special thanks to co-major professor Stephen Welch of the Department of Agronomy at K-State, for all his advice and for making this journey enjoyable. I also would like to thank all the professors and the staff of the Electrical and Computer Engineering Department for their help, sharing of knowledge, and availability when needed, especially professor Noel Schulz.

I consider pursuing the master's degree at the ECE department an exciting step in my education and a turnover point towards my career. In addition, I would like to thank my uncle, professor Medhat M. Morcos and my aunt, professor Sharon Morcos, for all their love, help, and effort. Finally, I would like to thank every single member of my family especially my dad Mohsen, my mom Samar, and my beloved brother Mourad for their encouragement throughout my study period, and all my good friends for their strong support.

Dedication

To

my Lord and Savior Jesus Christ

Saint Mary, Theotokos

Saint George, The Prince of Martyrs

Saint Philopater Mercurius

Saint Mary Magdalene

Saint Mina, The Wonder Worker

Pope Cyril VI of Alexanderia

Pope Shenouda III of Alexanderia

CHAPTER 1 - INTRODUCTION

Classical optimization algorithms have been widely used in various real world applications. They are methods that can find an optimal solution for a given function, either minimum or maximum depending on the optimized function. Henceforth, we will refer in this thesis to this function as the objective function and without loss of generality we will assume that the optimization entails the minimization of the objective function.

Typically, real world problems cannot be handled by classical methods, as they usually tend to be too complex. An appropriate approach is to use meta-heuristic techniques, such as bio-inspired algorithms. Bio-inspired computing (BIC) has been extensively used over the past decades to solve complex real world problems. Most of BIC methods are meta-heuristic methods inspired from nature that continue to find the optimal solution of a given function iteratively. Bio-inspired computing methods are stochastic, being characterized by maintaining non-deterministic behavior towards problem solving.

1.1 Bio-inspired Computing

When it comes to complex real world problems solving, bio-inspired techniques outperform classical methods. They are characterized by many advantages over classical methods that qualify them to attain top notch title in solving such type of problems. As an advantage, they are derivative free methods which give them the capability of addressing non differentiable problems and provide simplicity in finding optimum solutions,. In addition, these techniques are population-based, allowing easy parallelization, as well as the ability to simultaneously sample different regions of the search space. Typically, in bio-inspired techniques the goodness of a solution is measured in terms of the value of the objective function. This is referred to as its fitness. Examples of such techniques are simulated annealing, artificial immune system, clonal selection principal and genetic algorithms (GAs). PSO and ant colony optimization (ACO) are techniques based on swarm intelligence. ACO is based on the ants' strategy where ants lay pheromones along their path to a food source, allowing pheromone trails to accumulate if an optimal path is chosen or evaporate if the path is least favorable by ants due to path's high cost [1]. Simulated annealing is a method inspired from the cooling of metals in which the algorithm tends to be more exploitative, after exposing their molecules to a great amount of heat that tends to make the algorithm more explorative. Artificial immune system is another adaptive and distributed approach in which the system is characterized by a memory. Such characteristic is beneficial for the system as it helps identifying patterns of the infection that have been experienced at earlier stages and allows it to produce antibodies against that infection. The memory is also useful to classify patterns learned previously by the system [2]. Clonal selection principle is similarly characterized by a memory set and cloning of antibodies that tend to be formed due to the infection of the system by a virus or bacteria and reselecting those antibodies to defeat such infection [3]. GAs are based on biological evolution where the idea is inspired from Darwin's theory of evolution. In these algorithms, evolutionary operators, such as selection, crossover, and mutation play important roles [4]. Each chromosome within the population is evaluated and assigned a fitness value that measures how good a solution is. The higher the fitness value of a chromosome, the higher the probability of the chromosome being selected for producing an offspring. The fitness values of the chromosomes are sorted and the chromosomes with higher fitness values continue to survive to the next generation while the least fit ones get eliminated. GAs operators' are specialized depending on the application itself.

1.2 Thesis Outline

The contents of this study are outlined as follows. PSO is discussed in detail in Chapter 2. In Section 2.1, standard PSO and its functionality are presented. Fully informed PSO and how it varies from standard PSO is discussed in Section 2.2. Multi-objective optimization, fuzzy ε-dominance concept, and the investigated algorithm are detailed in Chapter 3, followed by figures of simulation results of test problems. Furthermore in Chapter 4, single objective optimization of modeled GRN problem and how PSO is applied to perform parameter estimation, minimizing the coefficient of variation in *Arabidopsis thaliana* ecotypes across different planting sites are presented. Identifying confidence regions of the estimated parameters is also discussed. Figures representing range of bolting dates of the selected ecotypes throughout an entire year are presented as well. Multi-objective GRN problems and how parameter estimation was performed for a wide number of ecotypes with interline similarities is discussed in Chapter 5. Figures of the non-dominated solutions of the multi-objectives GRNs are presented. Finally, Chapter 6 concludes the research results reported throughout the thesis. Suggestions of future work that could be applied to the proposed algorithm to furthermore improve its convergence rate are documented.

CHAPTER 2 - PARTICLE SWARM OPTIMIZATION

This chapter opens by explaining basics of standard PSO, details its functionality, and the procedure of updating particles' positions. Also, fully informed PSO and various sufficient network topologies that could be adopted by the swarm to reach optimal solution of a given problem are discussed in detail.

2.1 Standard PSO

PSO is a continuous optimization method for non-linear functions, based on swarm-intelligence that first has been proposed by Kennedy and Eberhart [5]. It is a population based method that is inspired from the social behavior of bird flocks, fish schools, or any other swarm of organisms. It mimics swarm-based organisms' means of exchanging information in solving problems. PSO maintains a population of particles that explore the search space in order to find suitable optima for the optimized function. The method is adaptive in which it simulates the stochastic movements of the particles. Each solution vector of the problem is represented by a particle that explores the search space seeking global optima. Each particle is accompanied by a certain velocity that controls the movement of that particle within the search space, which is the set of all feasible solutions to that problem. The best candidate particle among the entire population is known as *global best*. Each particle in the search space keeps track of the best position visited so far using its own memory and updates that position if a better one has been found during the exploration process, and is referred to as *local best*. The *global best* candidate is updated iteratively until no better solution is found through the search process. Further improvement can be achieved by maintaining balance between exploration and exploitation [6].

PSO starts with a randomly initialized population whose values are within a certain range which control how it functions. The velocity of each particle is also initialized randomly following a uniform distribution U[0,1]. The initialized population begins exploring the entire search space. In order for the particles to converge to a suitable optimum, each particle's position is updated iteratively using the equation shown below,

$$\vec{\mathbf{v}}_{t+1} = \chi \times \vec{\mathbf{v}}_t + C_1 \times U[0,1] \otimes (\vec{\mathbf{x}}_{lb} - \vec{\mathbf{x}}_t) + C_2 \times U[0,1] \otimes (\vec{\mathbf{x}}_{gb} - \vec{\mathbf{x}}_t) \tag{2.1}$$

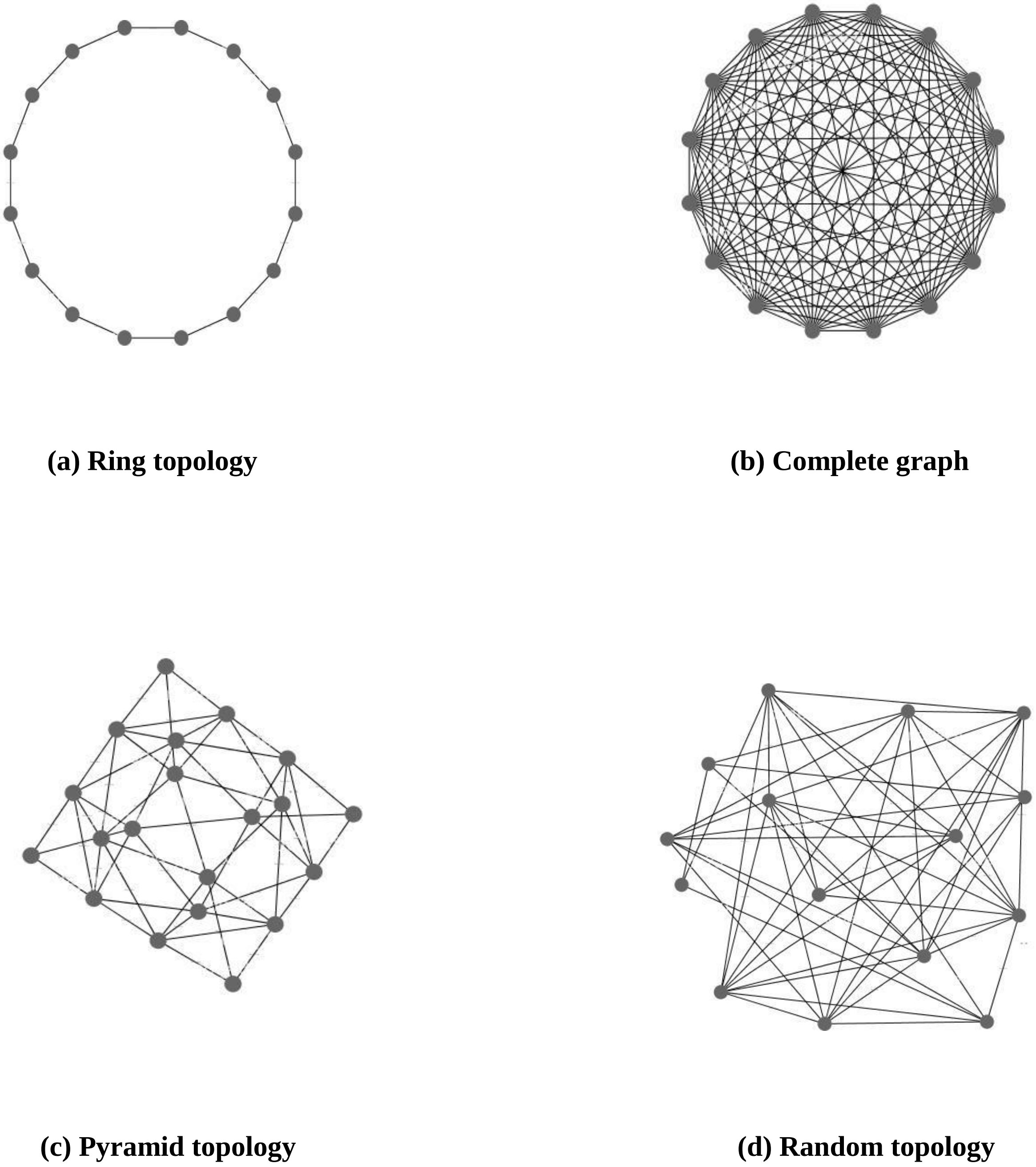
$U[0,1]$ is a multi-dimension uniformly random distribution between 0 and 1 and its dimensions are determined based on the solved problem, $\otimes$ denotes element by element multiplication while $\times$ denotes regular multiplication. In each iteration, all the particles are influenced by their current position, the particle's local best, and the population's global best. The particles' position is then updated by adding the new velocity calculated in equation (2.1) above to the particles current position which results in $\vec{\mathbf{x}}_{t+1}$,

$$\vec{\mathbf{x}}_{t+1} = \vec{\mathbf{x}}_t + \vec{\mathbf{v}}_{t+1} \tag{2.2}$$

The termination criterion of the algorithm depends on the solved problem. For instance if the problem is a benchmark problem where its global minima is known in advance, the termination criteria could be either applied by setting a specific error measure between the value of the estimated function and the global minima or limiting the number of function evaluations to a fixed value. On the other hand, when it comes to real world problems neither of the two stated criteria are applicable since a global optimum is not known a priori, and instead a number of function evaluations is determined adaptively and henceforth computational resources are saved [7].

2.2 Fully Informed PSO

Fully informed PSO on the other hand varies from the standard PSO in the sense that the particle is influenced by some or all of its neighbors rather than only the global best, depending on the topology used. The influence of the particle's neighbors does not require a specific topology or how the particles are connected together, rather it states how they interact in general [8]. Mendes et al stated that a particle could be connected to its neighbors using several topologies such as ring, complete graph, pyramid, and others. In Figure 2.1, examples of the network topologies are shown:



(a) Ring topology

(b) Complete graph

(c) Pyramid topology

(d) Random topology

Figure 2.1 Examples of different network topologies

Many topologies have been investigated by Mendes et al. [8] because according to the no-free lunch theorem (NFL) [9], there is no single approach that is better than all others in solving all the problems. The first step towards solving a problem is to accurately model the problem since finding an optimal solution to a wrong model is completely useless.

None of the network topologies investigated by Mendes *et al* performed very well on all test problems [8]; some have showed good results over others, depending on the type of the function being solved and the topology used. For instance, if we are solving for a multi-modal function, an adopted topology such as the complete graph could be deceptive and trap the particles into local minima which prevents them from reaching the desired optimum solution and results in premature convergence. Figure 2.2 shows the difference between a local minimum and a global minimum of a given function. The function has three local minima at x equal to 2, 4, and 7. However the value of the function $f(x)$ is equal to –1, –2, and –3.5 respectively, which implies that the local minimum at $x = 7$ is the lowest value and henceforth it is the global minima of the function. Some other topologies would have slow convergence rate on the population where a particle is connected to two neighbors and information is passed on in slow rate, such as ring topology.

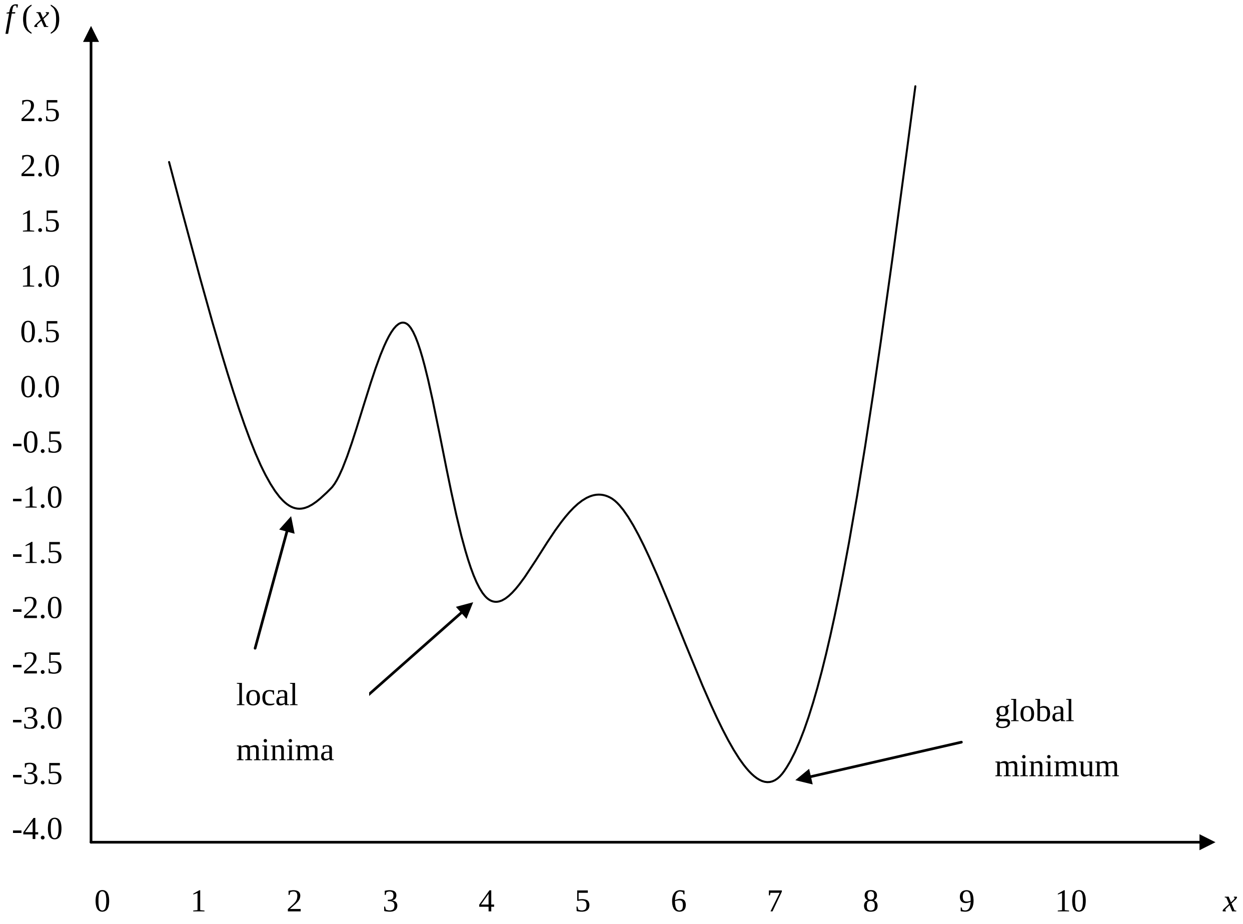


Figure 2.2 Illustration of the difference between a local minimum and a global minimum of a given multi-modal function

A random topology has been adopted by the proposed algorithm in the thesis and the topology has proven to show great results for the test problems presented in Chapter 3 and GRN problem in Chapter 5. The results and the detailed algorithm are explained in the next chapter.

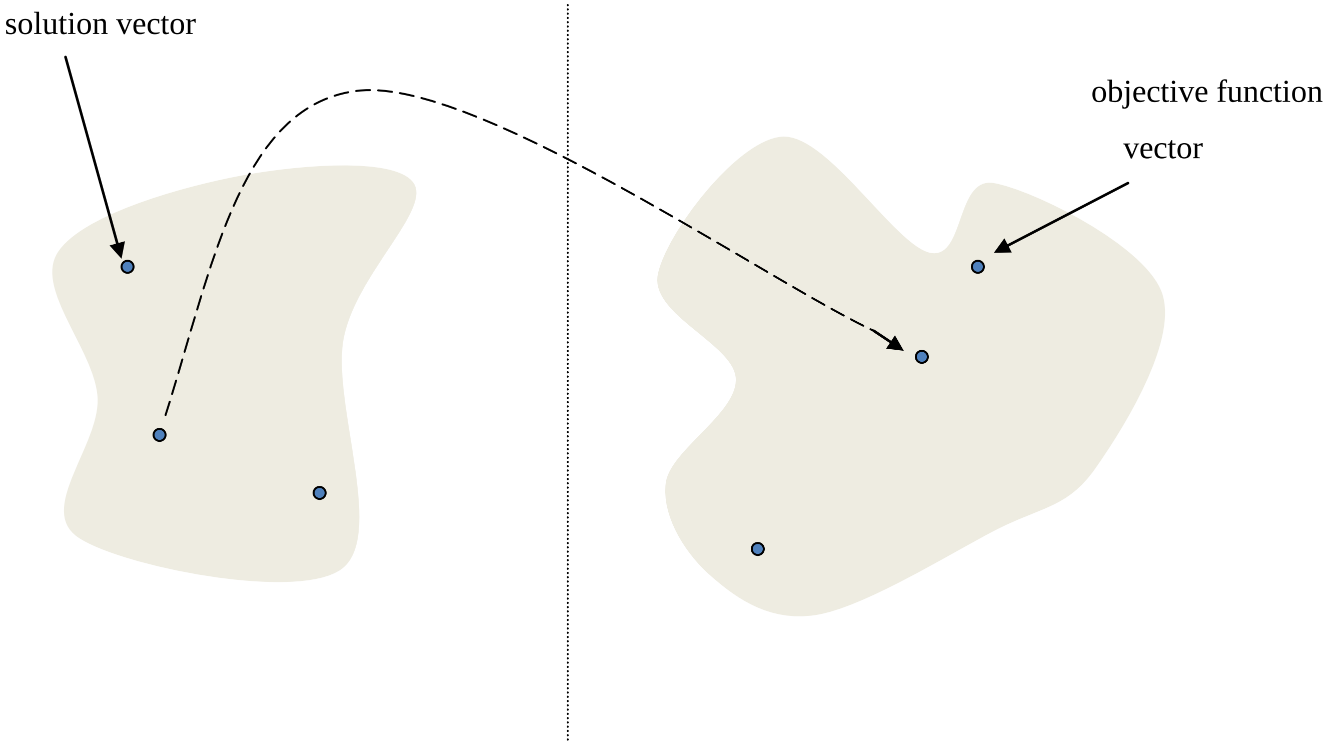
CHAPTER 3 - MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization involves solving complex real world problems with two or more objectives simultaneously. Meta-heuristic techniques have proved to solve efficiently various types of multiple-objective as well as single-objective problems. Some meta-heuristic methods are also combined with analytical methods such as gradient descent to solve such problems [10].

In multi-objective optimization a solution vector in an n-dimensional search space maps to an objective vector in the objective space. Figure 3.1 illustrates the mapping between the solution space and the objective space. When dealing with optimization problems with multiple objectives, the conventional concept of optimality does not hold [11]. Instead, the concepts of dominance and Pareto-optimality are applied. Without a loss of generality, let us assume that the multi-objective problem entails the simultaneous minimization of all M objectives, $f_i(.)$, $i = 1,\ldots,M$. Let the solution space be denoted as $\Theta \subset \Gamma^n$. A solution $\vec{\mathbf{x}} \in \Theta$ is said to *dominate* another solution $\vec{\mathbf{y}} \in \Theta$ if and only if $\forall i \in \{1,2,\ldots,M\},\quad f_i(\mathbf{x}) \le f_i(\mathbf{y})$ with at least one of the inequalities being strict, i.e., $\vec{\mathbf{x}}$ is as good as $\vec{\mathbf{y}}$ for all objectives and better for at least one. This relationship is written $\vec{\mathbf{x}} \succ \vec{\mathbf{y}}$. In the set of all feasible solutions, that subset whose members are not dominated is called the *Pareto set*. In other words, if P is the population, the Pareto set is $\mathbf{x} \in P \mid \forall \mathbf{y} \in P, \neg(\vec{\mathbf{x}} \succ \vec{\mathbf{y}})$. Its corresponding image in the space of all objective functions is known as the *Pareto front*, as shown in Figure 3.2.

Since all the solutions in the Pareto set are non-dominated, they must be treated as equally good. Therefore, the goal of an effective multi-objective optimization algorithm is to find candidate solutions whose images in the objective function space are (*i*) as close to the true Pareto front as possible, and (*ii*) as spread out and evenly spaced as possible, thereby sampling an extensive region of the Pareto front. These two conditions are usually referred to as *convergence* and *diversity,* respectively. Examples of convergence and diversity are shown in

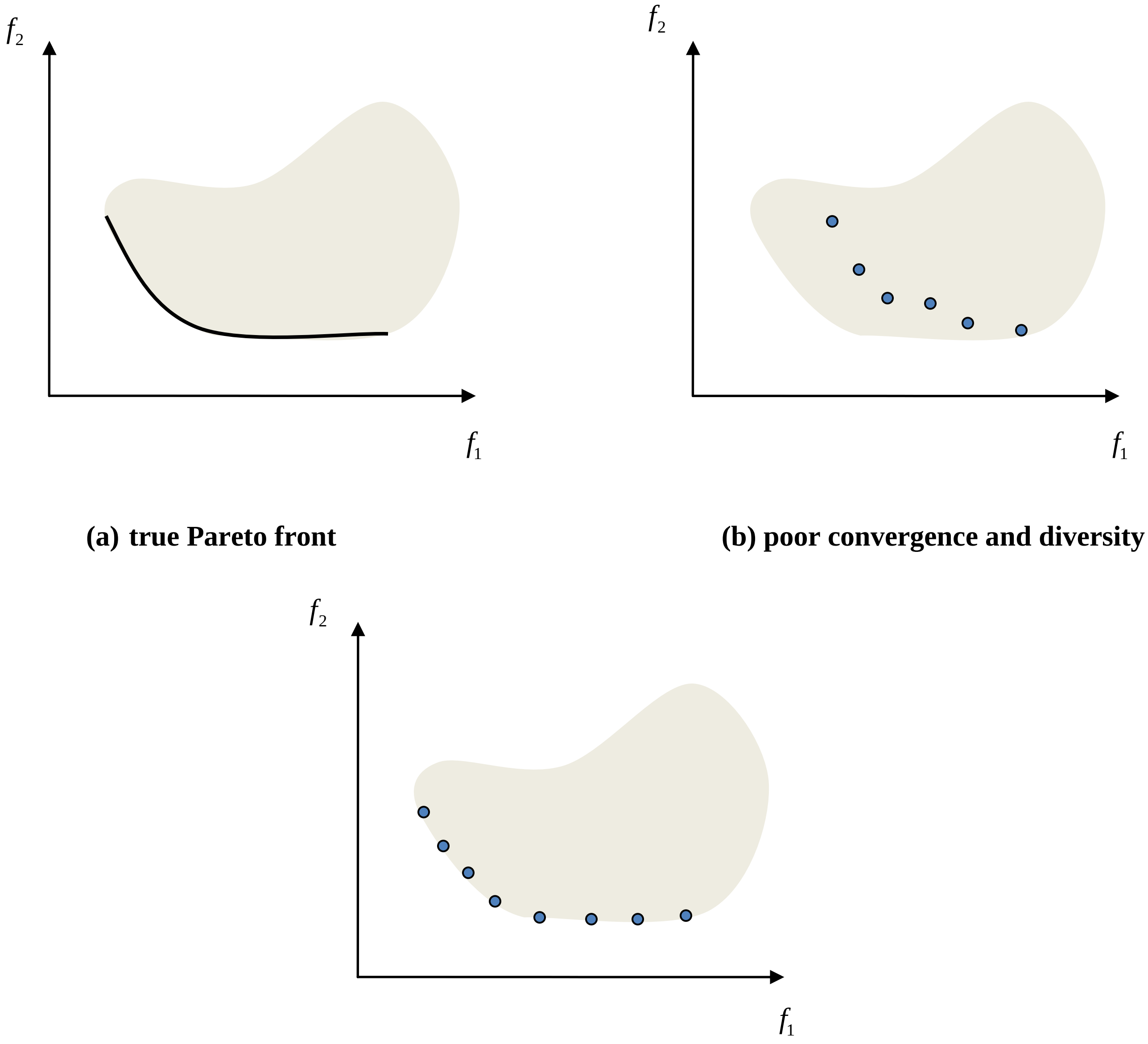
Figure 3.1, where f_1 and f_2 are to be minimized. Accomplishing good convergence and diversity are the two crucial aspects of any multi-objective optimization algorithm, including PSO. Fuzzy ε-dominance is a recently proposed scheme that combines convergence and diversity into one single measure, allowing multi-objective optimization problems to be treated as though they involved only a single objective. Fuzzy ε-dominance is an extension of fuzzy dominance that has been modified to take into account. Both are discussed next in this section.



(a) Search space

(b) Objective function space

Figure 3.1 Mapping between the solution vector in the search space to the objective function vector in the objective function space



(a) true Pareto front

(b) poor convergence and diversity

(c) good convergence and diversity

Figure 3.2 Examples of convergence and diversity concepts in multi-objective optimization

3.1 Fuzzy ε-Dominance Based PSO

A monotonically non-decreasing function $\zeta_i^{dom}(\cdot)$ ranged between [0, 1], where $i \in \{1, 2, \ldots, n\}$, a solution $\vec{\mathbf{x}} \in \Theta$ *i*-dominates another solution $\vec{\mathbf{y}} \in \Theta$ only if $f_i(\mathbf{x}) < f_i(\mathbf{y})$. The relationship is denoted as $\vec{\mathbf{x}} \succ_i^F \vec{\mathbf{y}}$ and the degree of fuzzy *i*-dominance is equal to $\zeta_i^{dom}\ f_i(\mathbf{x}) - f_i(\mathbf{y})\ \equiv \zeta_i^{dom}\ \vec{\mathbf{x}} \succ_i^F \vec{\mathbf{y}}$. The concept is regarded as a fuzzy relationship between $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$. Solution $\vec{\mathbf{x}} \in \Theta$ is said to fuzzy dominate $\vec{\mathbf{y}} \in \Theta$, for all objectives, if and only if $\forall i \in \{1, 2, \ldots, M\},\ \vec{\mathbf{x}} \succ_i^F \vec{\mathbf{y}}$ and the relationship is denoted as $\vec{\mathbf{x}} \succ^F \vec{\mathbf{y}}$. In order to compute the degree of fuzzy dominance, the concept of fuzzy intersection using *t*-norm is used,

$$\zeta^{dom}(\vec{\mathbf{x}} \succ^F \vec{\mathbf{y}}) = \bigcap_{i=1}^{M} \zeta_i^{dom}(\vec{\mathbf{x}} \succ^F_i \vec{\mathbf{y}}) \tag{3.1}$$

The membership functions $\zeta_i^{dom}(\cdot)$ in another implementation are classified to be zero for negative arguments [12]. Therefore, the degree of fuzzy dominance $\vec{\mathbf{x}} \succ_i^F \vec{\mathbf{y}}$ is necessarily evaluated to zero whenever $f_i(\mathbf{x}) > f_i(\mathbf{y})$. In the current study, we allow non-zero values in accordance with [13] [14]. Trapezoidal membership functions are used resulting in nonzero values whenever their arguments are to the right of a threshold ε, as shown in Figure 3.3,

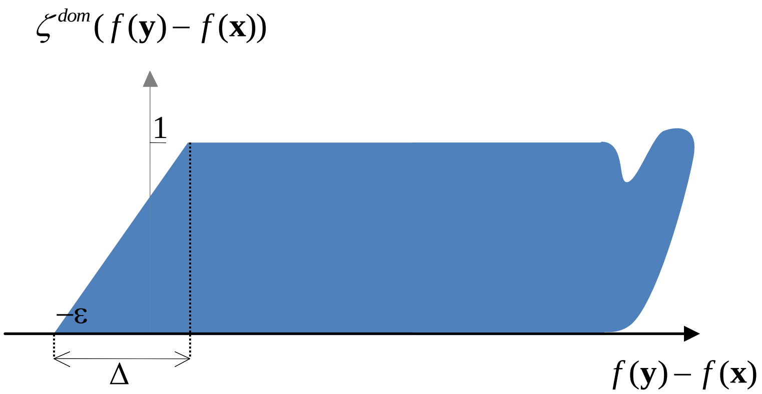


Figure 3.3 Fuzzy membership function used to compute ε-dominances

The memberships can be defined mathematically as,

$$\zeta_i^{dom}\ \Delta f_i\ = \begin{cases} 0 & \text{if} \quad \Delta f_i \leq -\varepsilon \\ (\Delta f_i)/\Delta_i & \text{if} \quad -\varepsilon < \Delta f_i < \Delta_i - \varepsilon \\ 1 & \text{if} \quad \Delta f_i \geq \Delta_i - \varepsilon \end{cases} \tag{3.2}$$

where, $\Omega f_i = f_i(\mathbf{y}) - f_i(\mathbf{x})$. In a given set of particles $P \subset \Theta$, a solution $\vec{\mathbf{y}} \in P$ is fuzzy dominated in P if and only if another solution $\vec{\mathbf{x}} \in P$ fuzzy dominates $\vec{\mathbf{y}}$. In this case, the degree of fuzzy dominance can be computed by performing a union operation over every possible $\zeta^{dom}\ \ \mathbf{x} \succ^F \mathbf{y}$, carried out using *t*-co norm as,

$$\zeta^{dom}(P \succ^F \vec{\mathbf{y}}) = \bigcup_{\mathbf{x} \in P} \zeta^{dom}(\vec{\mathbf{x}} \succ^F \vec{\mathbf{y}}) \tag{3.3}$$

In this manner, each solution can be assigned a single measure to reflect the amount it dominates others in a population. Better solutions within a set will be assigned lower fuzzy dominances, although unlike in [12] non-dominated solution may not necessarily be assigned zero values. The union and intersection operators follow the standard min and max definitions [15].

Typically, in multi-objective PSO, an external archive *A* maintains all the non-dominated solutions found. The velocities of the particles in the population are redirected towards the archived solutions. As new solutions for a given problem are discovered along the search process, the fit ones are inserted into the archive, while the least fit are discarded. The same strategy applies here. In each iteration, the population of particles is merged with the archive, and the fuzzy dominances computed. The archived solutions and the population are sorted in ascending order of their fuzzy dominances, and the best *A* size solutions are the archive for the next iteration. The global best used in equation (2.1) is the archive solution with the lowest fuzzy dominance. The local best $\vec{\mathbf{x}}_{lb}$ of particle *i*, is updated only when the *i*th particle dominates its own earlier stored local best, in which case $\vec{\mathbf{x}}_t\,(i)$ replaces $\vec{\mathbf{x}}_{lb}\,(i)$.

3.2 Fully Informed Fuzzy ε-Dominance Based PSO

The new algorithm investigated in the study is a hybrid of the two methods, fully informed and fuzzy ε-dominance PSO. Partially informed fuzzy ε-dominance (PIFD) PSO combines between how particles are informed by their neighbors to achieve convergence and using fuzzy ε-dominance concept to maintain diversity among particles. PIFD also uses an archive to store the non-dominated solutions. The best *A* size solutions are obtained by merging the non-dominated solutions of the current iteration and the archived solutions from prior generations. The solutions are sorted ascending based on their computed fuzzy dominance values.

Each particle is affected by a number of particles *n* to update its current position. The number of particles to be determined, *n*, follows a normal distribution and is denoted as $n \sim N(\mu, \sigma^2)$, where μ is the mean and σ^2 is the variance. The global best proposed in this algorithm is calculated randomly upon the determination of *n*, where the weighted sum of the *n* non-dominated particles selected from the archive is multiplied by 1 minus their fuzzy dominance values. The weights of the *n* selected particles are calculated randomly following a multi-dimensional uniform distribution $U[0,1]$ between 0 and 1. The global best proposed in this algorithm is expressed as,

$$\vec{\mathbf{x}}_{\mathbf{gb}} = \frac{\sum_{i=1}^{n}(1-\zeta_i^{dom}) \times U[0,1] \otimes A(i)}{\sum_{i=1}^{n}(1-\zeta_i^{dom}) \times U[0,1]} \tag{3.4}$$

The particles' velocities are updated using equation (2.1) as discussed in Chapter 2.

The main PIFD algorithm is detailed below.

1. Set iteration = 1.
2. Initially start with an empty archive.
3. Randomly initialize a population of *P* solutions with random velocities.

4. Set the initial archive to the current population.
5. Evaluate the objective function for each particle.
6. Update each particle's local best.
7. If neither of the two solutions dominates each other for all objectives, randomly pick one of them.
8. Merge the current population with the non-dominated solutions in archive *A*
9. Evaluate the fuzzy dominance for the merged population and update archive *A* with *P* solutions.
10. Update the population's global best.
11. Update velocity using equation (2.1).
12. Update positions of the current population according to equation (2.2).
13. Increment the value of iteration by 1.
14. If iteration $\leq$ maximum iteration go to step 5, otherwise terminate.

3.3 Test Problems

In order to test the functionality of the proposed algorithm, some of the benchmark problems found in the literature have been tested to assure the performance of PIFD. These problems are commonly used to test different multi-objective optimizers in finding the Pareto front.

KURSAWE :

$$f_1(\mathbf{x}) = \sum_{i=1}^{n-1} -10\exp(-0.2\sqrt{{x_i}^2 + {x_{i+1}}^2})$$

$$(3.5)$$

$$f_2(\mathbf{x}) = \sum_{i=1}^{n} (|x_i|^{0.8} + 5\sin(x_i)^3)$$

where n = 3 and the decision variables lie in the interval [–5, 5].

ZDT 1:

$$f_1(\mathbf{x}) = x_1$$

$$g(\mathbf{x}) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)} \tag{3.6}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})(1-\sqrt{(f_1(\mathbf{x})/g(\mathbf{x})})$$

where n = 30 and the decision variables lie in the interval [0, 1].

ZDT 3:

$$f_1(\mathbf{x}) = x_1$$

$$g(\mathbf{x}) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)} \tag{3.7}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})(1-\sqrt{(f_1(\mathbf{x})/g(\mathbf{x})} - (f_1(\mathbf{x})/g(\mathbf{x}))\sin(10\pi f_1(x)))$$

where n = 30 and the decision variables lie in the interval [0, 1].

ZDT 6:

$$f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$$

$$g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i / (n-1)\right)^{0.25} \tag{3.8}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})(1-(f_1(\mathbf{x})/g(\mathbf{x}))^2)$$

where n = 10 and the decision variables lie in the interval [0, 1].

3.4 Simulation Results

The results in this section are shown for the benchmark problems mentioned above. The figures and table of results are presented. For all the simulation runs, the population size of the PIFD algorithm is set to 100, and the external archive size *A* is set to the same value. C_1 and C_2 are both equal to 2, the constriction coefficient χ is set to 0.6. Multiple values of the constriction coefficient have been tested in order to analyze their compatibility with the new proposed method of calculating gbest, however the chosen value showed the best results. The values of μ and σ^2 are 2 and 4 respectively. The number of function evaluations for all the test problems was set to 10,000 evaluations. Figures of the Pareto front for all the problems stated above are detailed below.

As a performance measure, we used the spacing metric to measure how diverse the non-dominated solutions are in the produced Pareto front. The method was proposed by Schott [16] and is used here to measure how distributed the non-dominated solutions maintained in the archive *A*. Equation (3.9) below explains how the spacing metric is calculated,

$$SP = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(\bar{E} - E_i)^2} \tag{3.9}$$

The lower the value of *SP*, the better distributed the solutions are, with a value of 0 implying perfectly even distribution. Another quantity, E_i given by, ,

$$E_i = \min_{i,i \neq j}\left[\sum_{k=1}^{M}\left| f_k(i) - f_k(j)\right|\right]$$

(3.10)

$$\bar{E} = \frac{1}{n}\sum_{i=1}^{n} E_i$$

measures the difference between a solution *i* and all the non-dominated solutions across all the objectives and picks the minimum value of the absolute sum to consider it the distance between particle *i* and its closing neighbor.

Table 3.1 shows the spacing metric of the simulation results for 30 runs; the numeric values represent the average of the runs. In Figure 3.4, the plot shows the non-dominated solutions of the KUR test function produced by the PIFD algorithm. The figure is obtained from one of the runs.

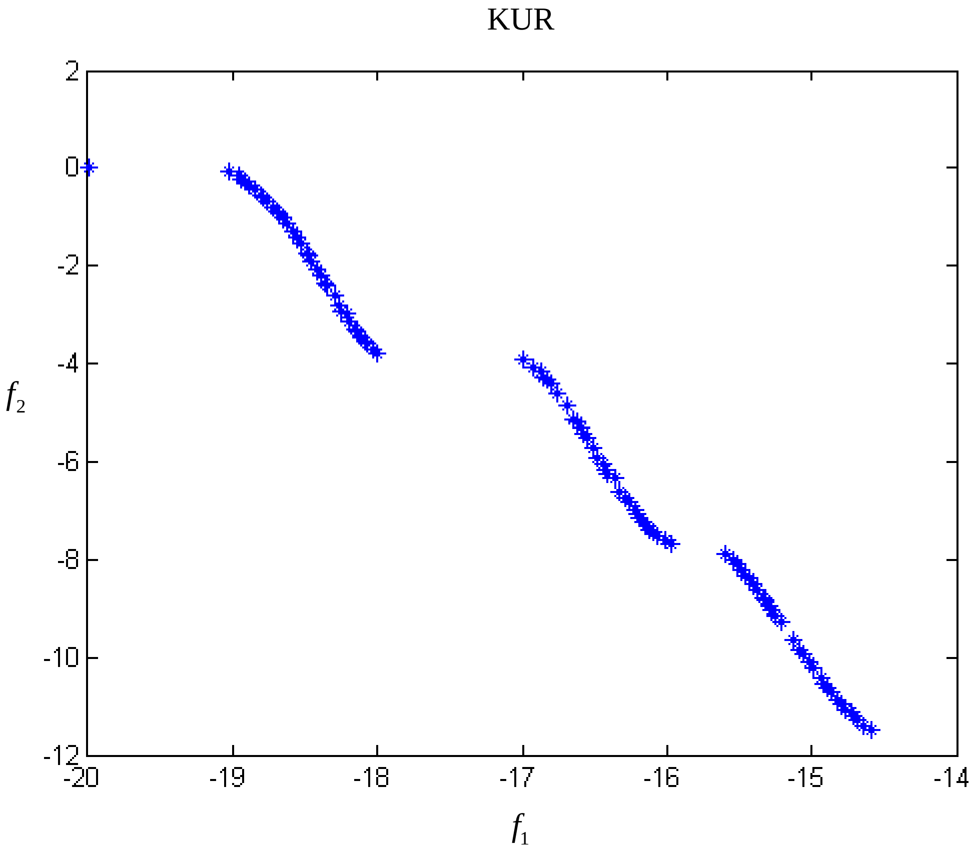


Figure 3.4 Pareto front for KUR produced by PIFD

The ZDT1 test function is shown in the figure below, which is obtained from a single run as well.

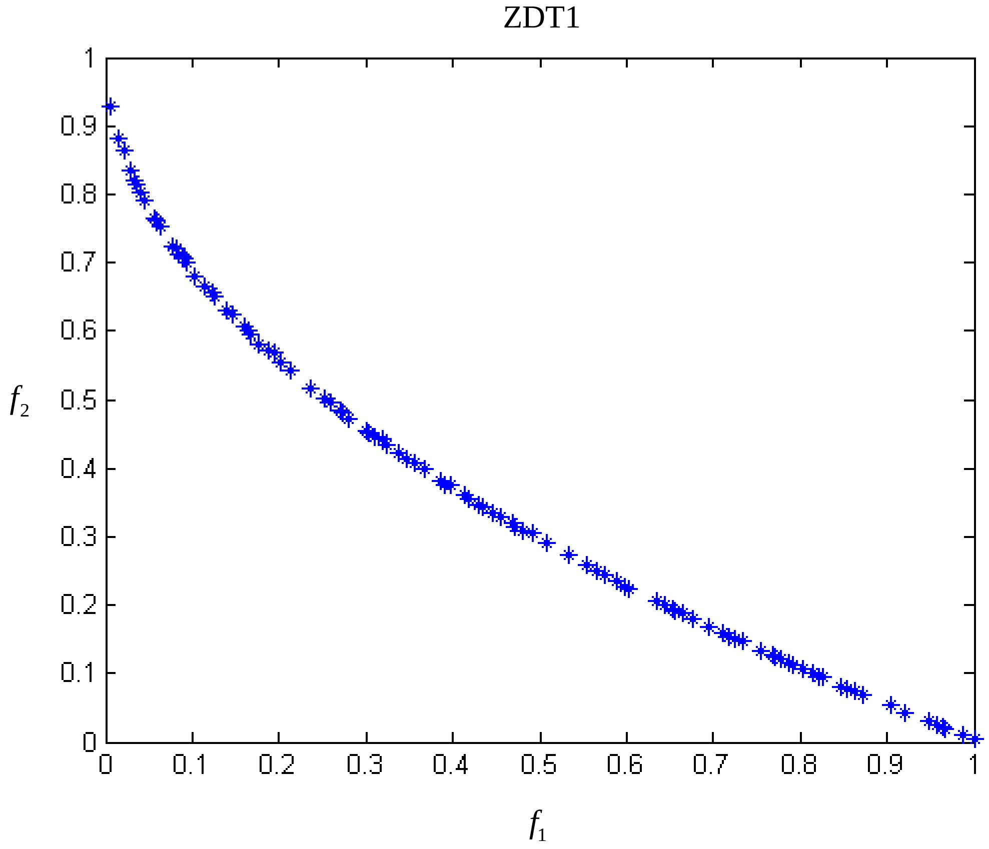


Figure 3.5 Pareto front for ZDT1 produced by PIFD

The ZDT3 test function is shown in the figure below, which is obtained from a single run as well.

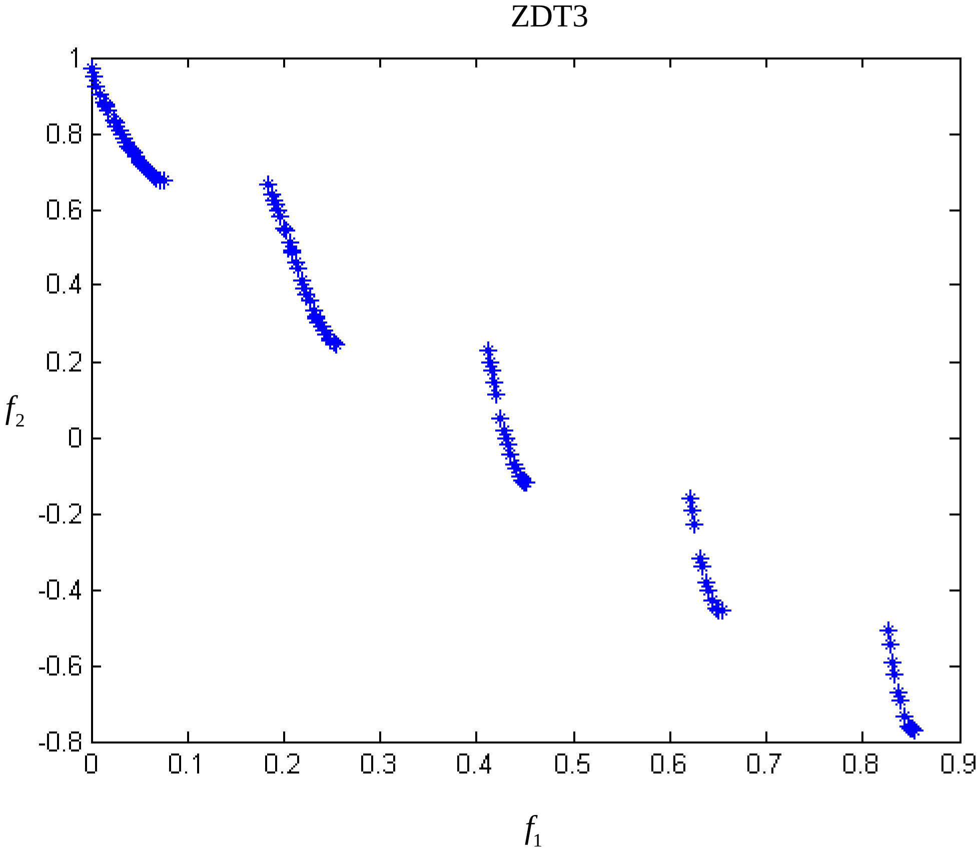


Figure 3.6 Pareto front for ZDT3 produced by PIFD

The ZDT6 test function is shown in the figure below, which is obtained from a single run as well.

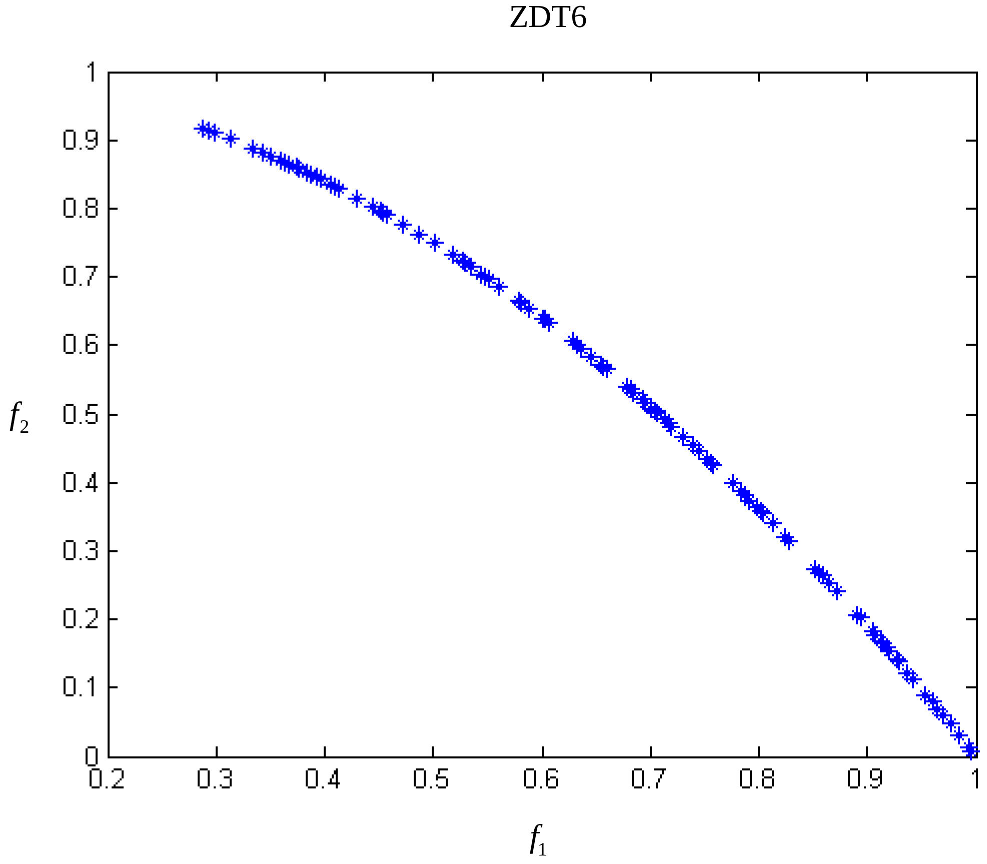


Figure 3.7 Pareto front for ZDT6 produced by PIFD

Table 3.1 Results of Spacing metric values obtained by PIFD against multiple approaches

Problem	**PIFD**	**Random Hybrid**	**MOPSO**
KUR	0.0956	0.6091	0.1904
ZDT1	0.00724	0.0686	0.0256
ZDT3	0.015	0.0541	0.0092
ZDT6	0.0068	0.1969	0.0763

The table shows the numeric mean value of the spacing metric obtained from the simulation runs for the corresponding test problem. Comparison of PIFD against multiple approaches is presented in the table, showing better results of PIFD for almost all test problems. In addition, the proposed algorithm proved to converge faster by running all test problems for a number of 10,000 function evaluations. However other listed approaches used 12,000 function evaluations and some test problems did not fully converge to the correct Pareto front.

Further applications are proposed and illustrated in Chapter 5, where PIFD is used to optimize a multi-objective GRN problem, and perform parameter estimation on an enormous number of plant ecotypes. The results produced by the algorithm tend to be very promising.

CHAPTER 4 - SINGLE-OBJECTIVE GRN

A single-objective GRN and a photothermal model to be fit to lines of *Arabidopsis thaliana* to predict flowering time are introduced in this chapter. Standard PSO was used to solve the single-objective GRN parameter estimation problem for those lines. In Section 4.1, confidence PSO (CPSO) [17] is explained and used to compute the confidence region of the estimated parameters. Simulation figures represent the confidence regions of the estimated parameters are illustrated in Section 4.2. Finally, figures are shown for several lines predicting the number of days to bolting for plants germinating on different dates. These are called "MTTY curves," where the acronym stands for "marching through the year."

Although many graphical models of *Arabidopsis thaliana* flowering time control have been proposed, there are very few quantitative models [13]. In general, three coupled genes are responsible for this floral process, *TERMINAL FLOWERING 1 (TFL1), APETALA 1 (AP1),* and *LEAFY (LFY)* [18]. *AP1* and *LFY* have a switch-like function that turns on after being sufficiently stimulated by the environment.

In order to fit different lines of *Arabidopsis thaliana* and predict their floral time, a photothermal model was created [18] which measures developmental units by accumulating a sum of environmental-determined increments on an hourly basis. Photothermal models assume that once the developmental units of a plant accumulate to a certain switching threshold, floral initiation occurs. The specific developmental units used in the [18] model are named modified photothermal units (MPTUs). MPTUs are the product of photothermal units (PTUs), which relate to daylength, temperature, and a vernalization (chilling) fraction. Flowering threshold and switching threshold will be used interchangeably in this chapter. The floral initiation process in *Arabidopsis thaliana* is controlled by a network of genes organized into four pathways (photoperiod, vernalization, gibberellin, and autonomous). The model was implemented to fit many different lines of *Arabidopsis thaliana*. However, in this project parameter estimation was applied to only five lines: *COLUMBIA* (*Col*), *GIGANTEA* (*gi*), *Col-FRIGIDA* (*Col-FRI*), *VERNALIZATION-INSENSITIVE-3* (*vin 3*), and *LUMINIDEPENDENS* as autonomous.

Naming of each line is based on gene functionality. All lines differ from each other in their behavior towards floral initiation depending on the gene's status, active or inactive. For instance, the line *gi* with lower case letters implies that the *gi* gene is inactive and is named so, while on the contrary the line *Col-FRI* with upper case letters implies that the *FRI* gene is very active. Every line comes from a different pathway background than the other, for instance *gi*, is responsible for communicating the line's diurnal clock and insensitive to long day photoperiod, while *vin3* is unable to respond to vernalization. The lines are good selection for modeling purposes and have specific defects that relate to measurement of photoperiod and vernalization. A diagram of interactions between pathways is shown in Figure 4.1. All of the mentioned lines were sowed at several different planting sites, thus experiencing various climatic conditions. These sites were Oulu, Finland; Cologne and Halle, Germany; Norwich, UK; and Valencia, Spain.

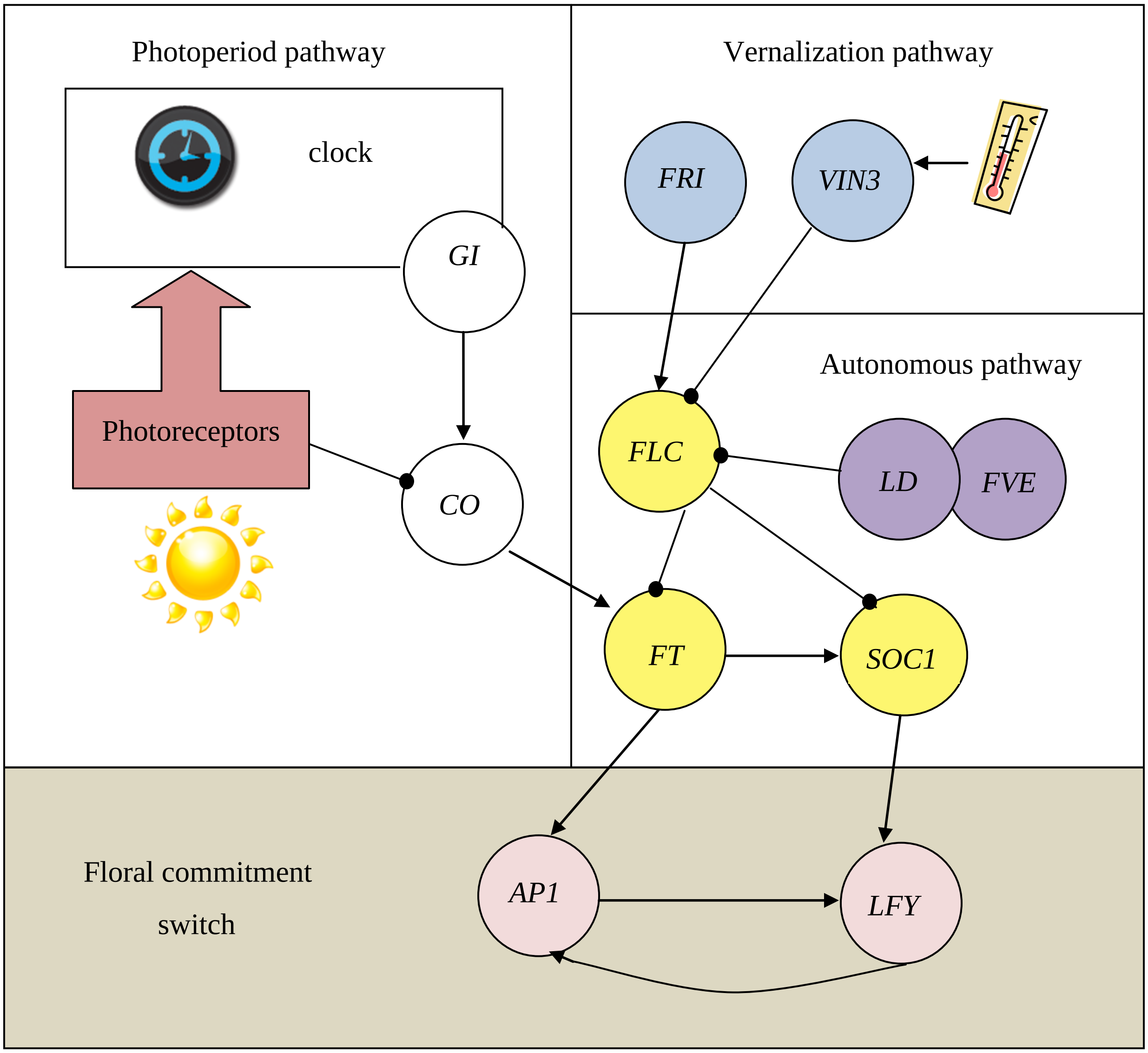


Figure 4.1 Interactions of genes among different pathways that control flowering time

For any planting, the MPTUs accumulate in one hour τ for any line and is expressed by the product of three components: vernalization, photoperiod, and temperature as depicted in Figure 4.2.

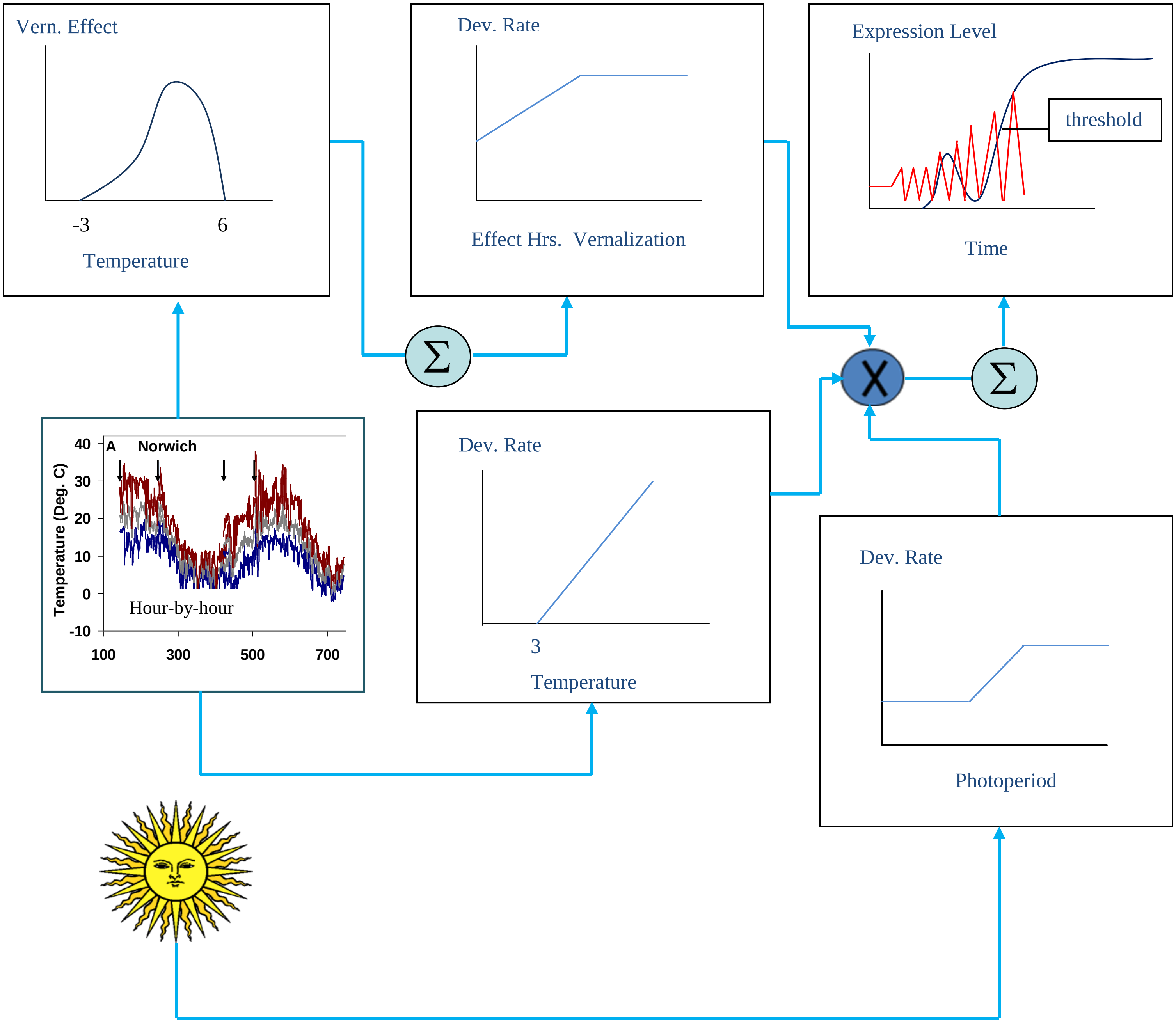


Figure 4.2 A flow diagram of the photothermal model

The following detailed set of equations from [18] defines the model. Equation (4.1) shows the factors used to calculate the MPTU,

$$mptu(\tau) = Vern(\tau)Phot(\tau)Thrm(\tau) \tag{4.1}$$

The vernalization factor is computed in equation (4.2) as,

$$Vern(\tau) = \begin{cases} F_b + Vr_h(\tau)(1 - F_b)/Vr_{sat,} & Vr_h(\tau) \le Vr_{sat} \\ 1, & \text{otherwise} \end{cases} \tag{4.2}$$

where $Vr_h(\tau)$ is expressed as,

$$Vr_h(\tau) = \sum_{s=n_1}^{\tau} \exp(\kappa)[T(s) - T_{V\min}]^{\omega}[T_{V\max} - T(s)]^{\lambda}$$

The value of $V_h(\tau)$ for line *vin3* is always set to zero, since it is insensitive to low degree temperature. The other factors, thermal and photoperiod, are given by,

$$Thrm(\tau) = \begin{cases} T(\tau) - T_{b,} & T(\tau) \ge T_b \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

$$Phot(\tau) = \begin{cases} D_{SD}, & dl(S,\tau) \le CSDL \\ D_{LD}, & dl(S,\tau) \ge CLDL \\ D_{SD} + [dl(S,\tau) - CSDL](D_{LD} - D_{SD})/(CLDL - CSDL), & \text{otherwise} \end{cases} \tag{4.4}$$

the number of MPTUs accumulates until the bolting threshold is reached. Equation (4.5) shows the accumulation of MPTUs from the sowing hour until plants bolt,

$$ST = \sum_{\tau=B_0}^{B_n} mptu(\tau) \tag{4.5}$$

This study focused on the most sensitive parameters as identified by [18]. Table 4.1 shows the values of other literature parameters that were held constant.

Table 4.1 Constant Parameters Used to Compute MPTUs

Parameter	Value
$CSDL$	10
$CLDL$	14
D_{LD}	1
V_{sat}	960
T_b	3
κ	–5.17
ω	2.23
λ	1
$T_{V\min}$	–3.5
$T_{V\max}$	6

The model assumes that when a fixed (although unknown) number of MPTUs are obtained, the plant initiates flowering. The most visible sign of floral initiation is the emergence of an inflorescence. This event is called "bolting", a term that will be used herein as synonymous with flowering. Because the switch genes had the same form in these lines, it was able to assume that this number was constant across the lines. Therefore, on the day that bolting occurred for each line, it should have accumulated a certain number of MPTUs. Therefore, a set of parameter values was sought that makes the numbers of MPTUs on the bolting dates as similar as possible across plantings – that is, that had the minimum coefficient of variation.

The standard PSO was applied in order to perform parameter estimation for the five lines to minimize the coefficient of variation of their MPTUs across all plantings. The coefficient of variation is calculated as in equation (4.6),

$$CV = \frac{\sigma}{\mu} \tag{4.6}$$

where σ and μ are the mean and the standard deviation of MPTUs across plantings. The objective function to be minimized is described in equation (4.7),

$$f(\mathbf{x}) = \min_{\mathbf{x}} CV(ST(\mathbf{x})) \tag{4.7}$$

Prior parameter estimation work with this model in [18], used Microsoft Excel Solver, obtaining a value of 10.75% for the CV. An identical CV was obtained with standard PSO, a valuable cross check on the prior results. A solution vector $\vec{\mathbf{x}}$ that consists of four parameters were to be estimated, F_b for all of the five lines and D_{SD}. However, all the lines share the same value of D_{SD} that corresponds to $\mathbf{x}(1)$, *Col* and *gi* share the same value of F_b, $\mathbf{x}(2)$, F_b for *Col-FRI* and *vin3* is common as well, $\mathbf{x}(3)$, and finally the F_b for the *autonomous* line that maps to $\mathbf{x}(4)$. These parameters all have value between zero and one.

4.1 Calculating Confidence Region Using C-PSO

This section describes the work done to calculate the confidence intervals for the parameters estimated above regarding the five lines *Col*, *gi*, *Col-FRI*, *vin3*, and *autonomous* using C-PSO [17]. At each iteration, the region is calculated based on the best CV found so far. Whenever a set of parameters results in a CV that is lower, the specific set of parameters is retained and the border is recalculated. Parameter sets that fall outside the updated border are removed. The border is defined based on a formula for confidence limits of CV estimates [19]. Points with CV values greater than the upper limit (first term in equation (4.8)) were deemed to have parameter values outside the desired 95% confidence region. Equation (4.8) shows how the confidence intervals are calculated.

$$CI = \left\{ \Lambda \left[\sqrt{\left(\frac{u_1 + 2}{\upsilon + 1} - 1 \right) \Lambda^2 + \frac{u_1}{\upsilon}} \right]^{-1/2} ,\ \Lambda \left[\sqrt{\left(\frac{u_2 + 2}{\upsilon + 1} - 1 \right) \Lambda^2 + \frac{u_2}{\upsilon}} \right]^{-1/2} \right\} \tag{4.8}$$

Normally single-objective PSO is expected to converge to single optima, so incorporating a perturbation operator is necessary to aggregate particles into a “cloud” shape that represents the confidence region. Equation (4.9) presents how this perturbation operator, called mutation, is implemented.

$$\vec{\mathbf{v}}_{t+1}^{'} = \vec{\mathbf{v}}_{t+1} + \delta \times U[-1,1] \otimes \vec{\mathbf{v}}_{t+1} \tag{4.9}$$

The perturbation process applied on the particles’ velocities acts as turbulence and creates cloud regions. However, this turbulence moved some particles outside the confidence region our region so crossover was applied to replace them with better values. The crossover operator replaces the worst particle with a new particle with better fitness using equation (4.10),

$$\vec{\mathbf{x}}_{t}^{'} = U[0.5,1] \otimes \vec{\mathbf{x}}_{t}^{w} + (1 - U[0.5,1]) \otimes \vec{\mathbf{x}}_{t}^{g} \tag{4.10}$$

If one application of crossover was insufficient to move an outlier back into the region, then the point was discarded. The uniform random number U in both equations (4.9) and (4.10) is multi-dimensional, depending on the dimensions of the objective function.

The steps of the algorithm to obtain the confidence region are detailed as follows.

1. Set iteration = 1.
2. Initially start with an empty archive.

   The archive is meant to maintain solutions that are only below the value of CI evaluated in (4.8).
3. Randomly initialize population with random positions and velocities.
4. Evaluate the objective function for each particle.

   The objective function in (4.7) is evaluated for entire population.
5. Archive only solutions that have CV less than CI computed in (4.8).

   The archived solutions form the cloud at the end of the run.
6. Replace outlier particles with fit ones using crossover in (4.10).

   The outlier particles, worst solution vectors, are replaced with better ones.
7. Update each particle’s local best and population’s global best.

8. Re-evaluate the value of CI and remove archived solutions above new CI.
9. Update velocity using equations (2.1) and (4.9).
10. Update positions of the current population according to equation (2.2).
11. Increment the value of iteration by 1.
12. If iteration $\leq$ maximum iteration go to step 4, otherwise terminate.

4.2 Simulation Results

Two different simulation runs, each of a population size of 80 particles, are presented in this section. The total number of function evaluations of the first run was set to 13,000 and the second to 10,000, with a 95% confidence interval. The reason two simulation runs were performed is twofold. First, sensitivity of the confidence limits to the amount of computational resources was assessed. Secondly, it was noted that solutions were being duplicated, indicating a waste of computational effort. Therefore runs with reduced number of iterations were performed. The results of the first run produced an archive of size 1541 solutions, and the second of 933. The figures obtained below represents the cloud of particles formed by CPSO. The solutions for both runs are plotted on the same figure for all the lines (Figures 4.3 – 4.5).

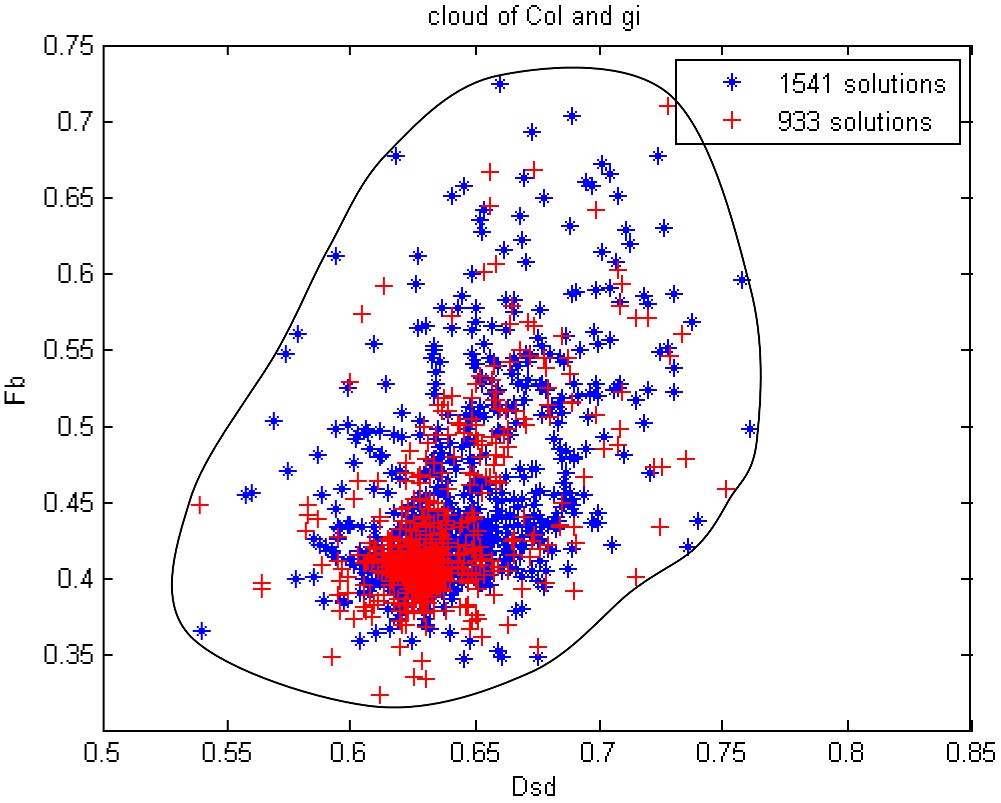


Figure 4.3 Confidence region of Col and gi represented by both populations

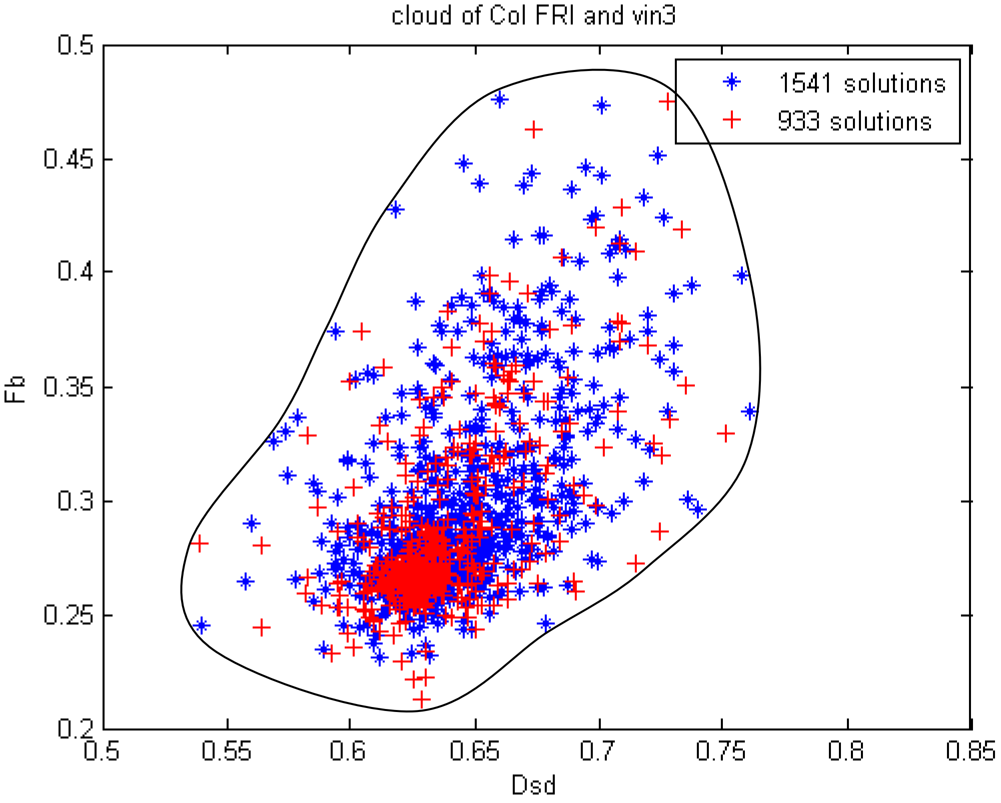


Figure 4.4 Confidence region of Col-FRI and vin3 represented by both populations

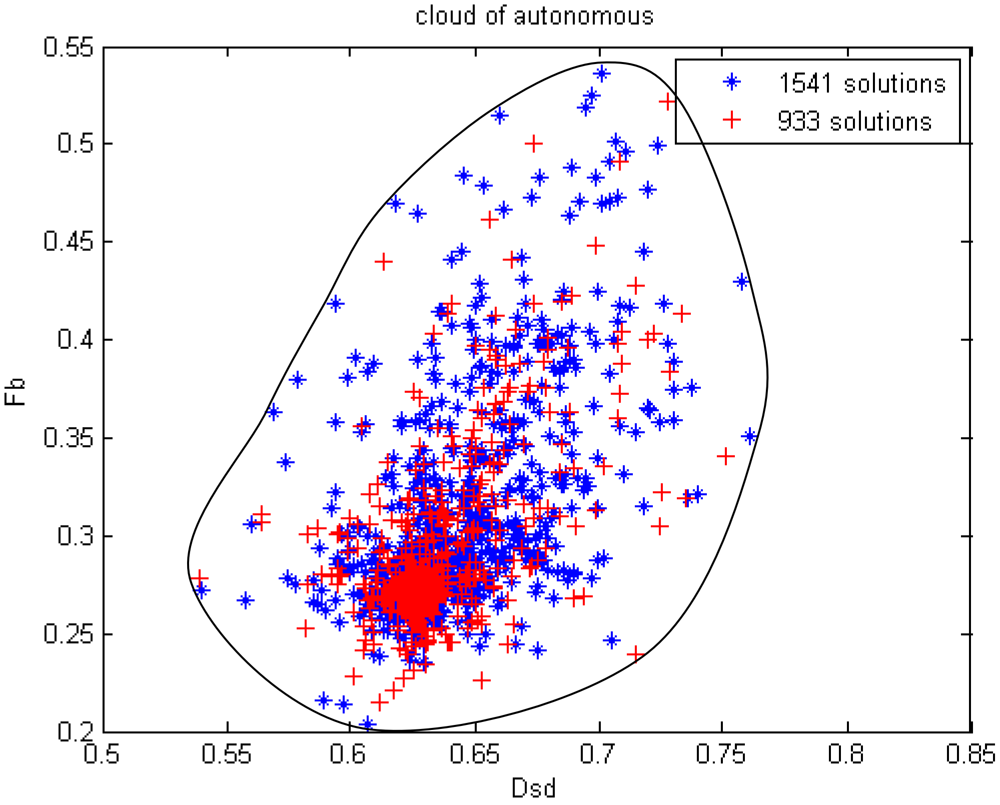


Figure 4.5 Confidence region of autonomous represented by both populations

4.3 MTTY

MTTY is a graphical representation of the predicted number of days needed to bolt, assuming that germination occurs at different times. MTTY curves were first presented in [18]; however, confidence limits were not provided for those curves. Since a slight difference in the parameter set values or a small change in the climate would result in significant variation in the number of days to bolt, it is desirable to predict the range in which bolting days may vary, instead of simply having a point estimate. The curves in Figures 4.6 – 4.10 show the upper and lower envelopes of the runs.

Archived solutions of both populations from Section 4.1 were used to propose those graphs. The curves were obtained by accumulating the MPTU values for each line on hourly basis until a threshold of 2604 units is reached, which is an approximate value at which isogenic lines bolt. Once the desired MPTU is reached, the program terminates and stores number of minimum and maximum hours needed for each line to bolt across the Norwich site.

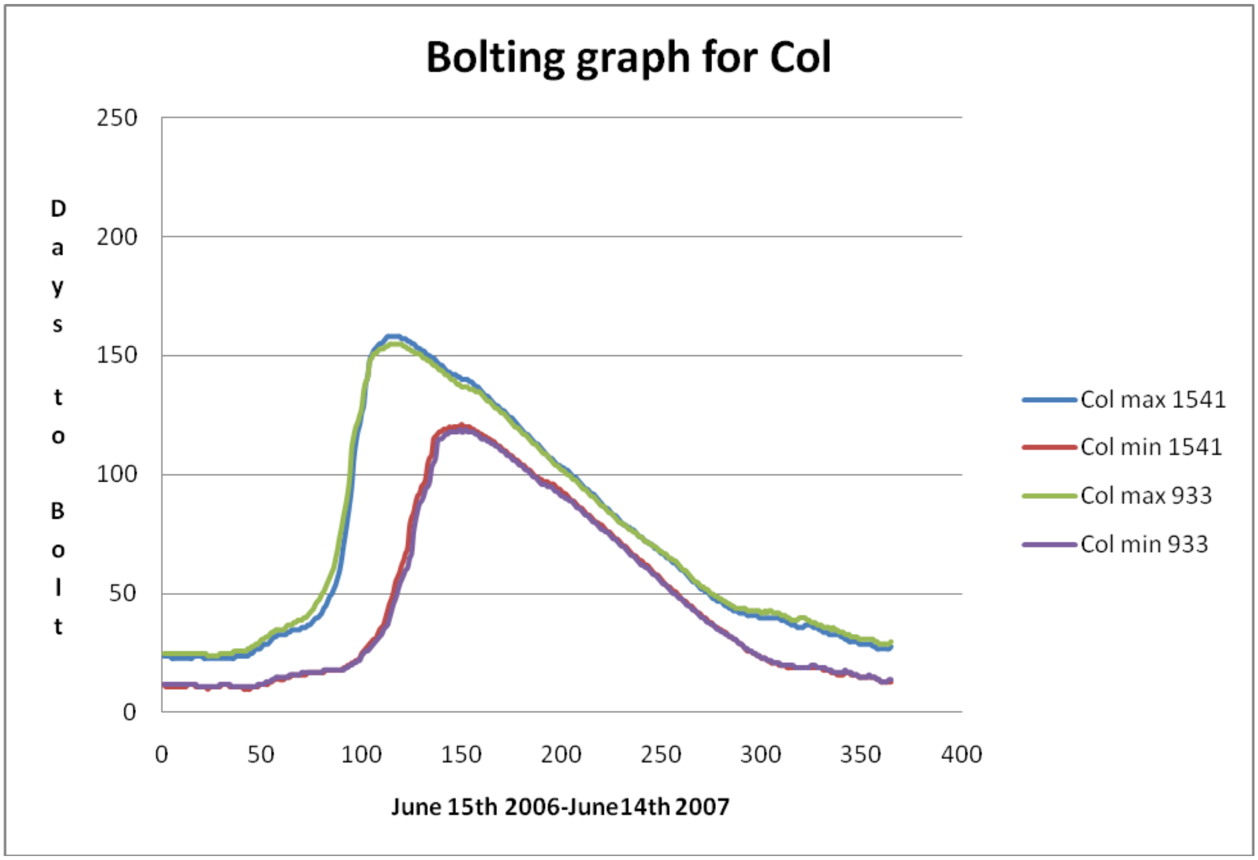


Figure 4.6 Representation of bolting days for Col through 2006

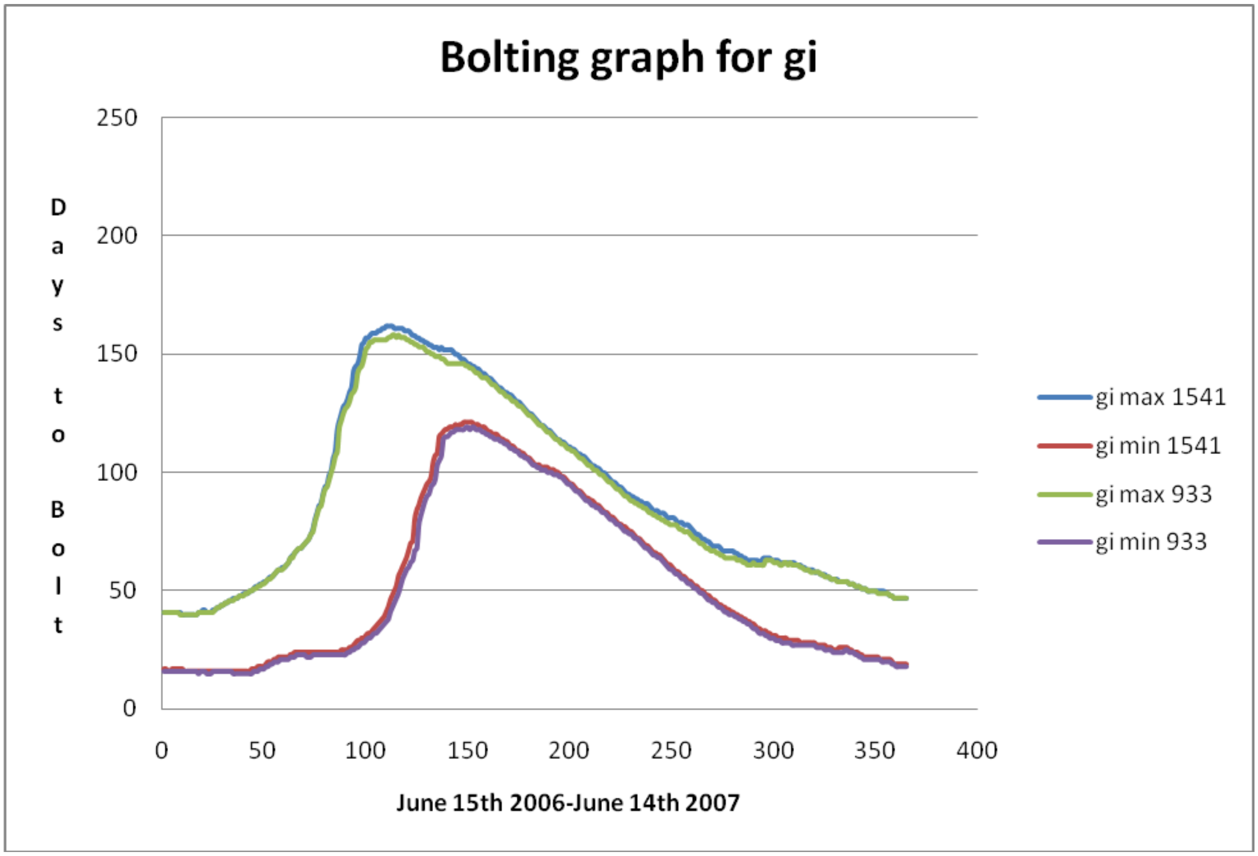


Figure 4.7 Representation of bolting days for gi through 2006

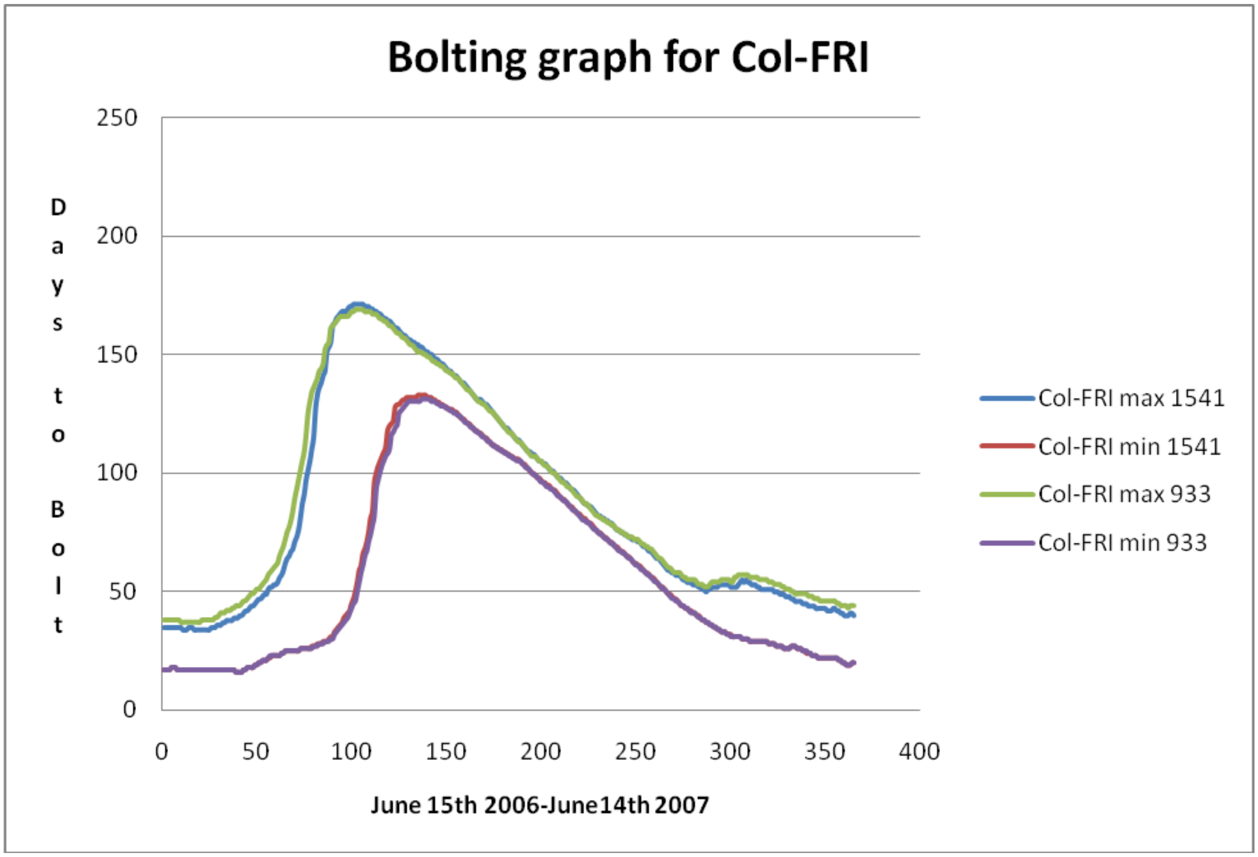


Figure 4.8 Representation of bolting days for Col-FRI through 2006

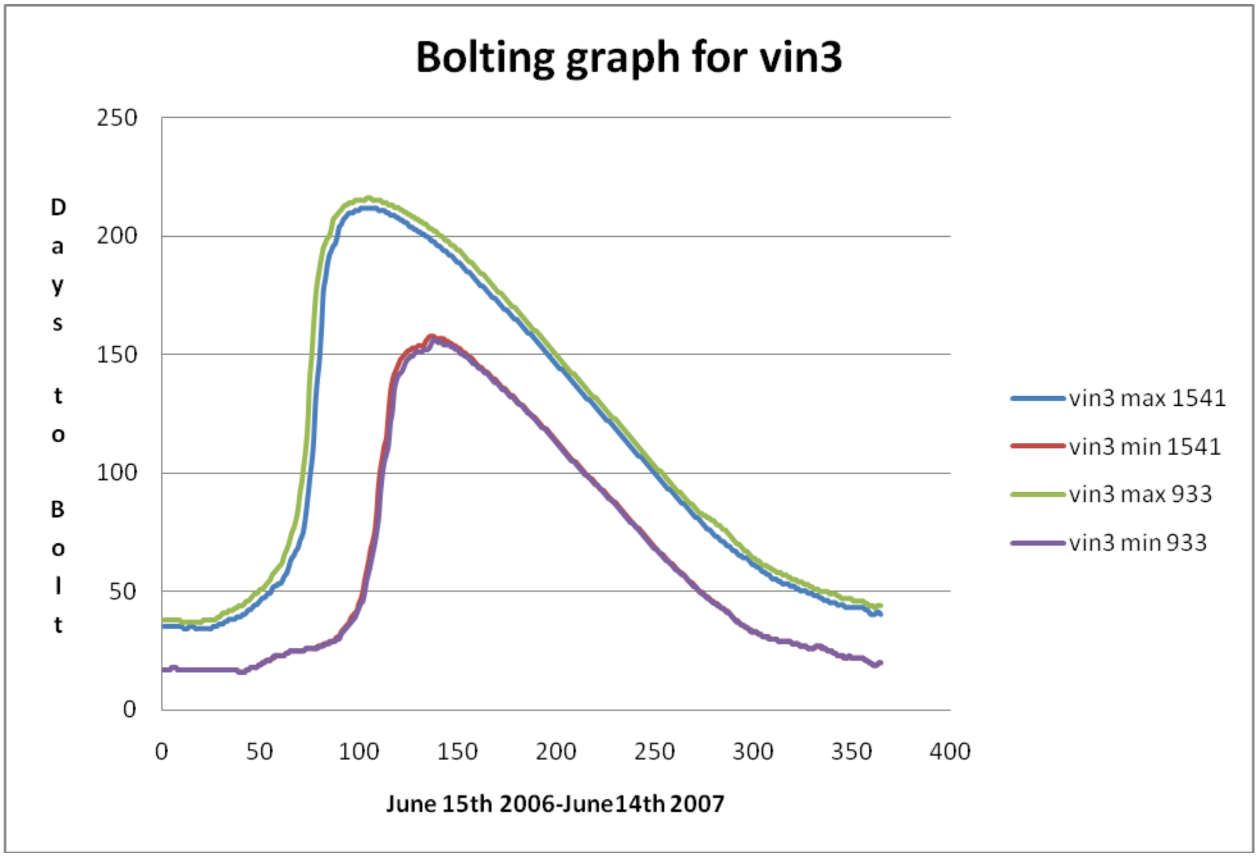


Figure 4.9 Representation of bolting days for Col through 2006

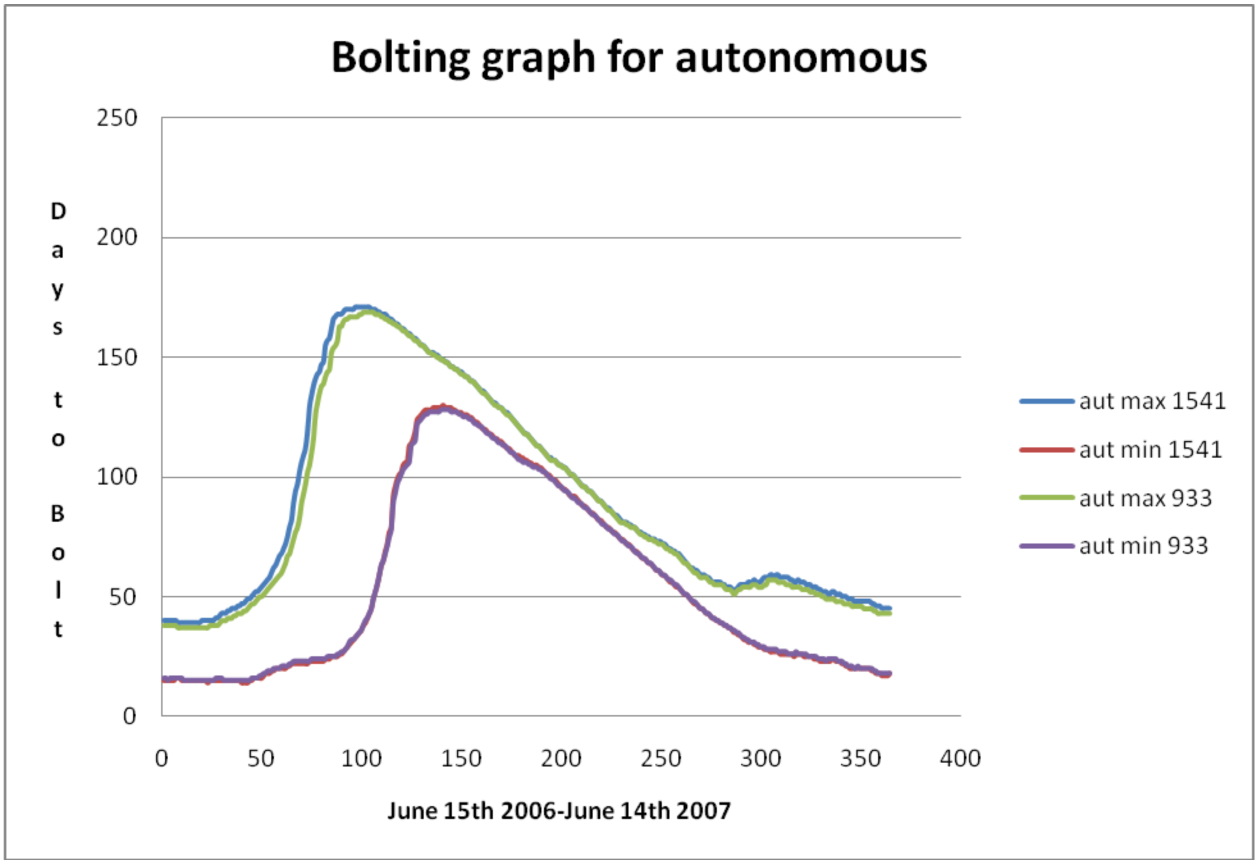


Figure 4.10 Representation of bolting days for autonomous through 2006

The variation of predicted number of bolting days is represented using the upper and lower envelopes in Figures 4.6 – 4.10. To obtain these envelopes, the model makes use of the insensitive parameters listed in table 4.1 along with the archived solutions. Then the model runs through every single solution and calculates the number of hours needed by a line for a specific day of the year. This process is done until an interval of one year is carried out.

CHAPTER 5 - MULTI-OBJECTIVES GRN

Chapter 4 discussed (*i*) how a photothermal model was used to fit different isogenic lines in *Arabidopsis thaliana* and (*ii*) how parameter estimation of single-objective GRN was implemented using PSO. This chapter discusses how the same model is used to fit 50 lines, each varying in its developmental units accumulation as its elements respond differently to weather conditions. The detailed weather data, temperature, photoperiod and vernalization that were used in Chapter 4 are used here as well. PIFD was applied to optimize the objective functions and perform parameter estimation for those lines. Two objectives were used. The first objective was to minimize the coefficient of variation of lines' MPTUs expressed in equation (4.7), while the second objective related the differences between the parameters assigned to different lines to the background genomic similarities between those lines. (The words “ecotypes” and “lines” will be used interchangeably in this chapter). Estimation of similarity measurement between isogenic lines is also explained in this chapter. Simulation results of the model and the resulting Pareto front are presented in Section 5.2 with histogram figures for the optimized lines’ individual *CV*s.

5.1 Similarity Background

It is quite possible for different sets of parameters to result in similar or identical flowering time predictions under the same weather conditions, a situation called *equifinality*. A solution to this problem would be adding additional information to the optimization in order to narrow the range of feasible estimates. One way to do this is to assume that genetically similar plants have similar parameter values, although the reverse is not necessarily true.

There are two complementary ways in which genetic similarity can be evaluated. The first is by direct comparison of the DNA sequences of genes known to influence flowering time. The advantage of this approach is the close bearing that these genes are known to have the trait of interest. The disadvantages are that there may be other yet undiscovered genes that either intermediate the effects of the known genes or affect flowering time in altogether unknown ways. Thus, a sole focus on known genes may overstate the similarities involved. An additional difficulty is that detailed sequence information for all the lines used was not readily available to the study.

The second approach, which was investigated within this study, was to look at what might be termed "background similarity," that is, general relatedness as determined by examining patterns of genetic markers distributed across the genome. Markers are distinguishable DNA sequences occurring at specific locations that can be used to categorize individuals. Given a genomic location where different marker types exist in a population, different individuals will have different markers depending on which particular sequence they may inherit from their parents. If two, closely-spaced, adjacent markers originated from the same parent, it is possible to assume that all DNA in the intervening region did as well. If two adjacent markers are from different parents, then somewhere between them will be a crossover point with DNA from one parent on one side and DNA from the second parent on the other. The greater the amount of agreement between the markers of two lines, the higher the degree of agreement one can assume between the two lines DNA.

Two sources of data were used in this study. As previously mentioned in Chapter 4, the first was flowering time data from a large, multi-site study involving nine plantings at five sites spanning the European range of *A. thaliana* (described in Wilczek et al., 2009). Each planting included between 240 and 360 distinct lines. Micrometeorological data was recorded in close proximity to the plants.

The second source was published marker data on a large set of *A. thaliana* ecotypes. Data scoring of single-nucleotide polymorphisms (SNPs) on 92 *A. thaliana* accessions was provided [20] in order to estimate the background similarity between ecotypes. *Col* and *Ler* lines were used as a reference for all ecotypes. For all accessions, SNPs were observed and either the *Ler* (*L*) allele or the *Col* (*C*) allele was recognized. The reason that these two lines were used as a reference is that they are the most common lab-strains, widely employed by researchers and enormous data amount pertaining them exists. In a few cases both marker types were present;

these were coded "*H*" for "heterozygous." In other cases a location would have neither marker (i.e., it was some other type) and these were coded "*U*" for "unknown." In addition to *Col* and *Ler*, 50 other ecotypes overlapped between the two sets of data and these were the ones used in the present study.

The similarity metric is the calculated $\mathbf{K}^*$ matrix of pairwise kinship coefficients [21]. $\mathbf{K}^*_{i,j}$ represents fraction of shared alleles between lines *i* and *j*. The formula needed to estimate $\mathbf{K}^*_{i,j}$ is given in equation (5.1),

$$\mathbf{K}^*_{ij} = G^{-1}\sum_{g=1}^{G} k_{ijg} \tag{5.1}$$

where k_{ijg} is equal to one if lines *i* and *j* have the same marker state at location *g* and, zero otherwise. *G* is the total number of known markers excluding those that are "unknown." (That is, markers whose states were not clearly established by laboratory procedures were not counted.) $\mathbf{K}^*$ is highly effective at describing multiple levels of relatedness. Each of the 52 lines was compared to every other line at all loci and a corresponding $\mathbf{K}^*_{i,j}$ value was estimated. Among the ecotypes, some were very similar to *Col*, others were very similar to *Ler*, and others were intermediate.

Multi-objective GRN problems can be categorized under two main categories in terms of their objective functions: data-oriented and energy-oriented. Data-oriented functions are based on the observed error of lines' phenotypes when they reach their floral initiation, while energy-oriented ones are based on the squared differences of the corresponding parameters on the basis of inter-line similarities. Recalling from Chapter 4, the data-oriented function is represented by the *CV* in equation (4.6); the energy-oriented is analogous to the energy content stored in metaphorical springs constraining corresponding parameters of inter-line similarities. The amount of energy resides between inter-line similarities of two different lines reflects the similarity measure between those lines. For instance, two lines with similar parameters imply that the metaphorical spring binding both lines is rather stretched and energy content tends to be of great value; however, the extreme case is unnecessarily true. Two lines bound with a shrunk spring could still possess similar parameters.

Both of the optimization problem classes involve sum of individual error terms. A decision of which terms are included in each sum is rather challenging. Due to technical reasons, separate sums over all sowings for each line × parameter would not be a good idea, as weight distribution of terms would be challenging. Another thought would be to sum all residual terms, no matter which class they belong to; however, this is rather problematic. Presumably, if all the data residual terms are represented into one sum, and all energy residual aggregated into another. The objective function can be represented by the lump sum equation below,

$$f(\mathbf{x}) = f_D(\mathbf{x}) + f_E(\mathbf{x}) \tag{5.2}$$

It is very unlikely that the right hand side terms of the equation would have the same magnitude. It is more likely that one of the terms is much larger than the other. This issue can be resolved by assigning weights to the terms of the above equation and convert the problem to a trade-off optimization problem as equation (5.3) explains,

$$f(\mathbf{x}) = \varpi_D f_D(\mathbf{x}) + \varpi_E f_E(\mathbf{x}) \tag{5.3}$$

A problem that would cross someone's mind is which values should be assigned to ϖ_D, ϖ_E. Which point would be a balance between both coefficients?

5.2 Energy-oriented Objective

A clever approach to solve this uncertainty is to deal with each objective independently as an objective. The first objective is a data-oriented one, relevant to the coefficient of variation of the lines' MPTUs (from Chapter 4), while the second objective, energy-oriented, is associated with the total sum of background similarity between ecotypes. The energy-oriented objective was initially represented by the sum of absolute difference squared of all lines' parameters ($CSDL$, F_b, and D_{SD}) multiplied by the appropriate similarity value of these two particular lines obtained from the kinship matrix. Equation (5.4) demonstrates how the energy-oriented objective is calculated,

$$f_2(\mathbf{x}) = \sum_{i=1}^{L}\sum_{j=1}^{j<i}\sum_{k=1}^{U}\mathbf{K}^{*}{}_{i,j}\;\; l_i(k) - l_j(k)\;\;{}^{2} \tag{5.4}$$

The main concern was to fit as many as ecotypes as possible in the $5-20\%$ range, henceforth different formulas of the second objective were implemented. The reason for using these formulas is twofold; first to identify the level of effectiveness of parametric differences on ecotypes and observe their reaction to that difference. Secondly to use them as being the desire to provide multiple alternatives for how influential large and small values are on the total sum. Results of all different formulas are presented in Section 5.3.

Equation (5.5) shows a different formula of the second objective,

$$f_2(\mathbf{x}) = \sum_{i=1}^{L}\sum_{j=1}^{j<i}\sum_{k=1}^{U}\mathbf{K}^{*}{}_{i,j}\left|l_i(k) - l_j(k)\right| \tag{5.5}$$

Another experiment was conducted is that to change the value of *CLDL* while fixing a different formula for the second objective. The formula in equation (5.6) calculates the sum of the absolute difference of ecotypes divided by their relative sum as given below,

$$f_2(\mathbf{x}) = \sum_{i=1}^{L}\sum_{j=1}^{j<i}\sum_{k=1}^{U}\mathbf{K}^{*}{}_{i,j}\,\frac{\left|l_i(k) - l_j(k)\right|}{l_i(k) + l_j(k)} \tag{5.6}$$

The above formula does not apply any penalties on the optimized parameters.

Finally, the last experiment conducted was to cube the absolute difference of parameter values to heavily penalized ecotypes that were genetically similar. The formula is stated in equation (5.7),

$$f_2(\mathbf{x}) = \sum_{i=1}^{L} \sum_{j=1}^{j<i} \sum_{k=1}^{U} \mathbf{K}^*{}_{i,j} \; \left| l_i(k) - l_j(k) \right|^3 \tag{5.7}$$

In Section 5.3, the results of the simulation runs are presented and figures of Pareto front are proposed. The optimizer (PIFD) was developed in *MATLAB* while the objective function was implemented in *C++*. Since computational time is very essential, the intention for this is to minimize the complexity run of the model and reduce the number of computational resources while decreasing the amount of memory space needed. *MATLAB* includes a compiler that is capable of compiling any *C++* file and outputs another *mex* file that is executed by the optimizer in *MATLAB.*

5.3 Simulation Results

In this section, the results of the simulation runs will be proposed and illustrated in details. The photothermal model was used to fit 50 ecotypes excluding *Col* and *Ler*, as previously mentioned the set of parameter for these two lines has been optimized earlier [18]. The model was run for all experiments 250 iterations and produced figures of the runs are presented below; most of the investigated energy-oriented formulas would require more iterations, such that solutions produce Pareto front figures. In Figures 5.2 – 5.12, the archived (non-dominated) solutions obtained by PIFD are presented for both objectives accompanied by histogram figures of individual *CV*s range. Figure 5.1, tends to shows that all candidate solutions converged to a point where a “single” optimum has been reached,

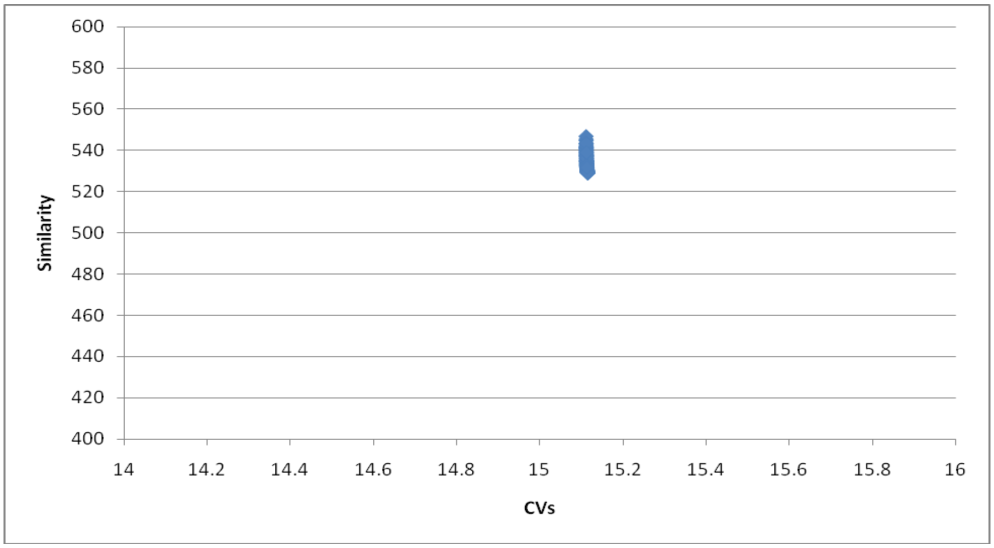
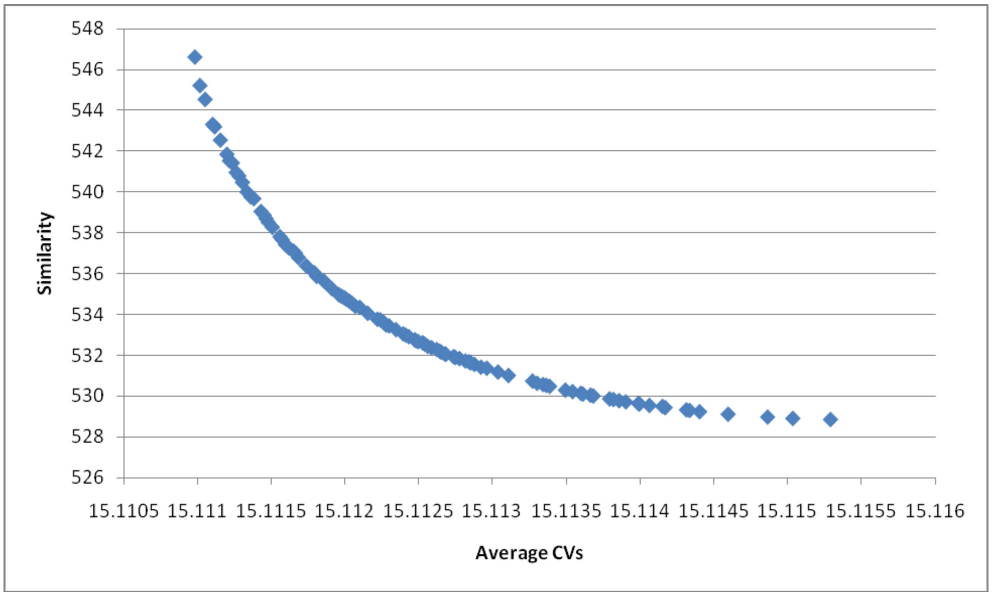
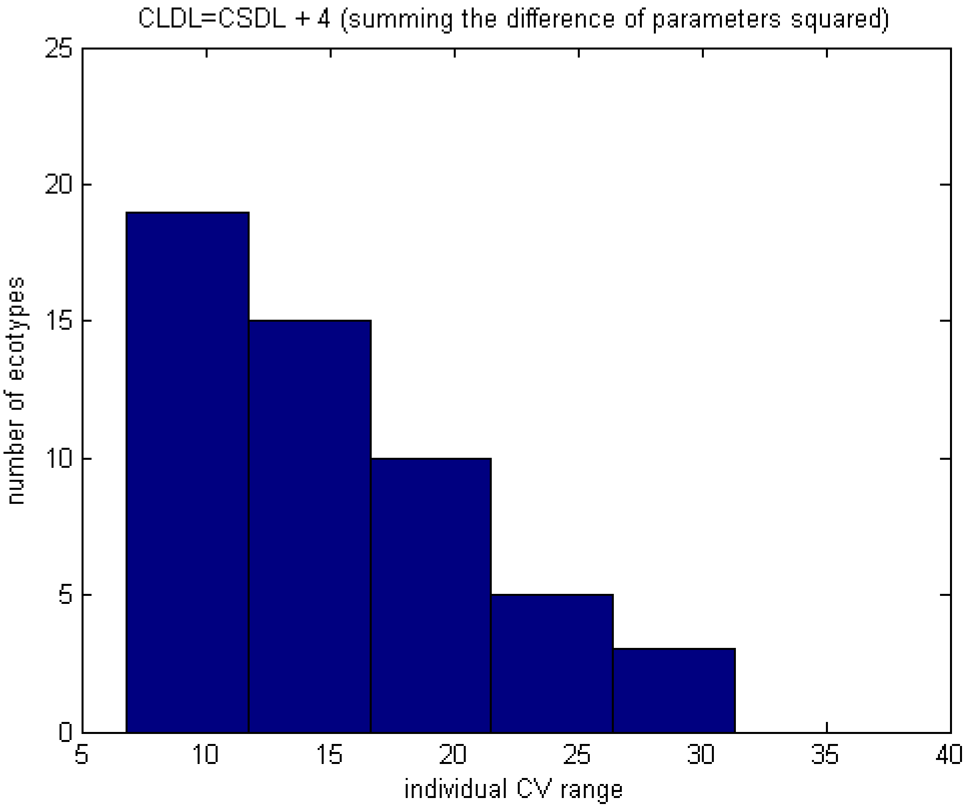


Figure 5.1 Compressed figure of Pareto front of multi-objective GRN when summing the squared difference of parameters

The above figure is detailed in Figure 5.2 to show how the diversity of solutions within the search space is maintained. Multiple values for the parameter *CLDL* were proposed in prior work where a grid search was applied in order to specify a fixed value for that particular parameter. Unfortunately, all lines' receptors differ from each other in their behavior and responding to photoperiod where some require a specific number of hours while others require less or more, depending on the ecotype's characteristic. Initially the simulation started off with a mean value for all ecotypes of 4 that was added to *CSDL*, assuming that all ecotypes would use the same number of hours. Based on that assumption, the model was run and results are shown in Figure 5.2. The figure corresponds to the formula in equation (5.4).



(a) Pareto front of multi-objective GRN where CLDL = CSDL + 4



(b) Histogram of individual CVs

Figure 5.2 Pareto front for multi-objective GRN produced by PIFD when summing the squared difference of parameters and histogram of ecotypes' individual CVs

The *x*-axis of Figure 5.2 (a) is named Average *CV*s and it represents the mean value of all individual *CV*s. The graph in Figure 5.2 (b) is right-skewed which implies that most individual *CV*s are similar; most ecotypes reside in the 5 – 20% range. The Pareto front in Figure 5.2 is detailed using the sequence of Figures 5.3 through 5.6, given the formula of equation (5.4). The sequence demonstrates the journey of search space individuals throughout the optimization procedure until convergence is achieved.

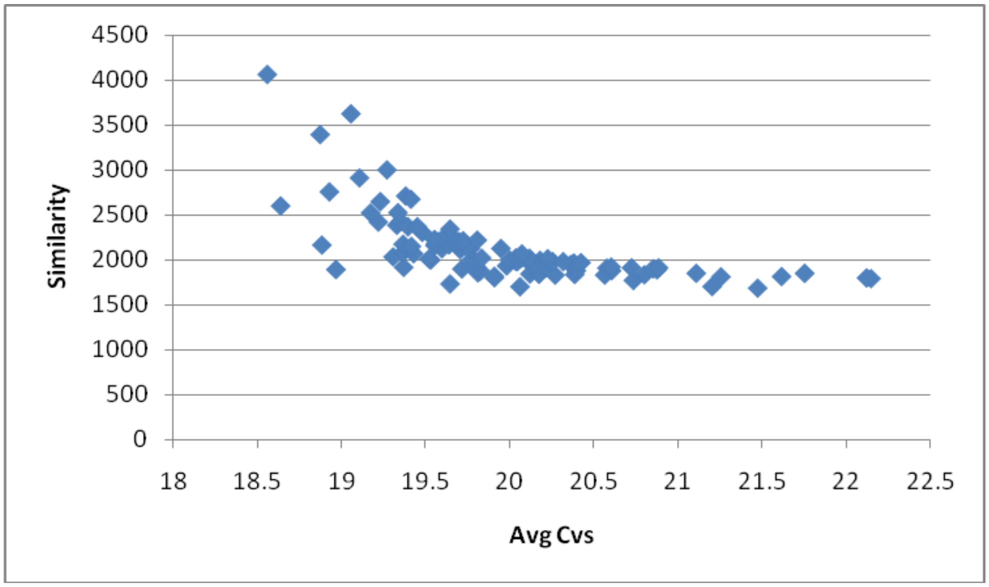


Figure 5.3 Particles formed by PFID after 25 iterations

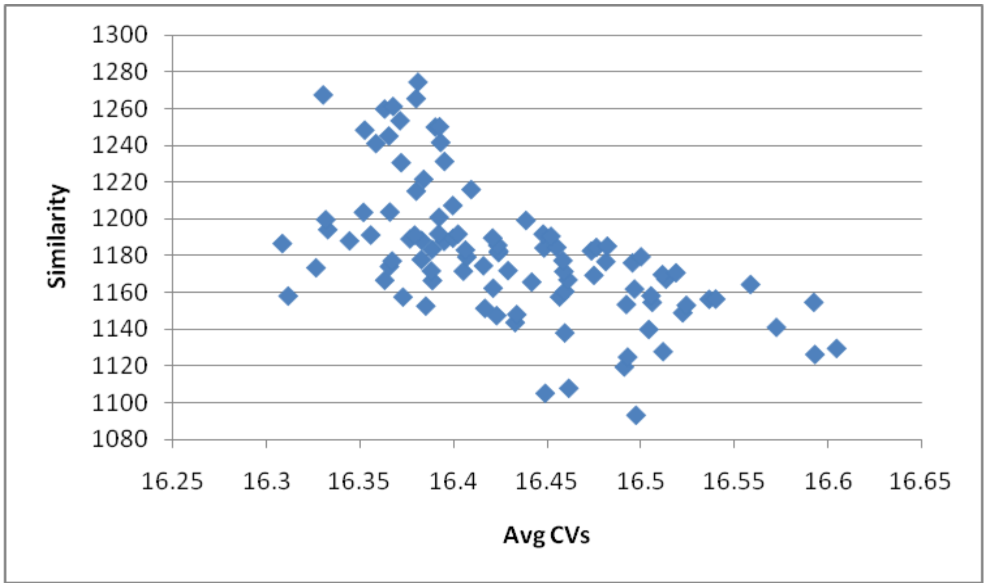


Figure 5.4 Particles formed by PFID after 50 iterations

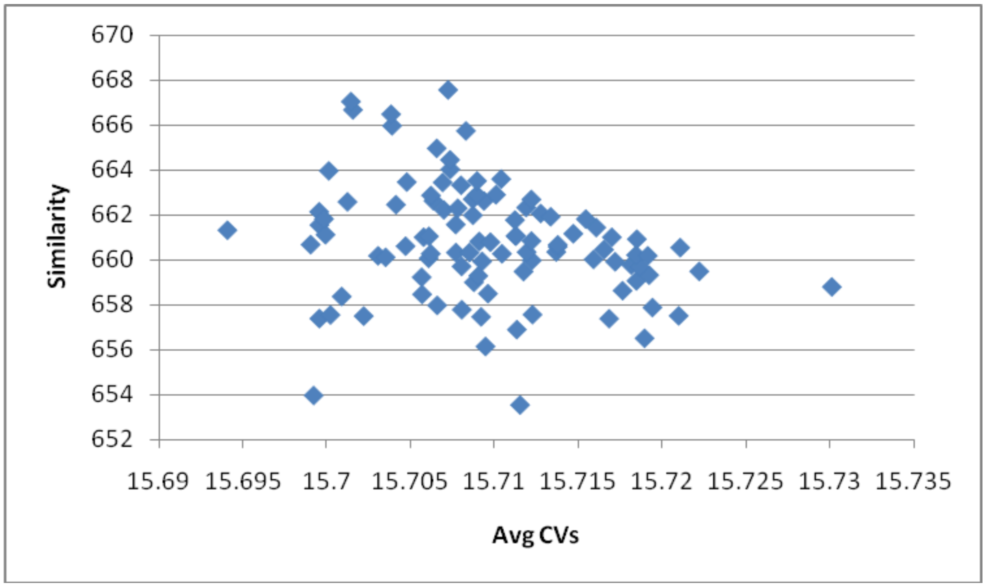


Figure 5.5 Particles formed by PFID after 100 iterations

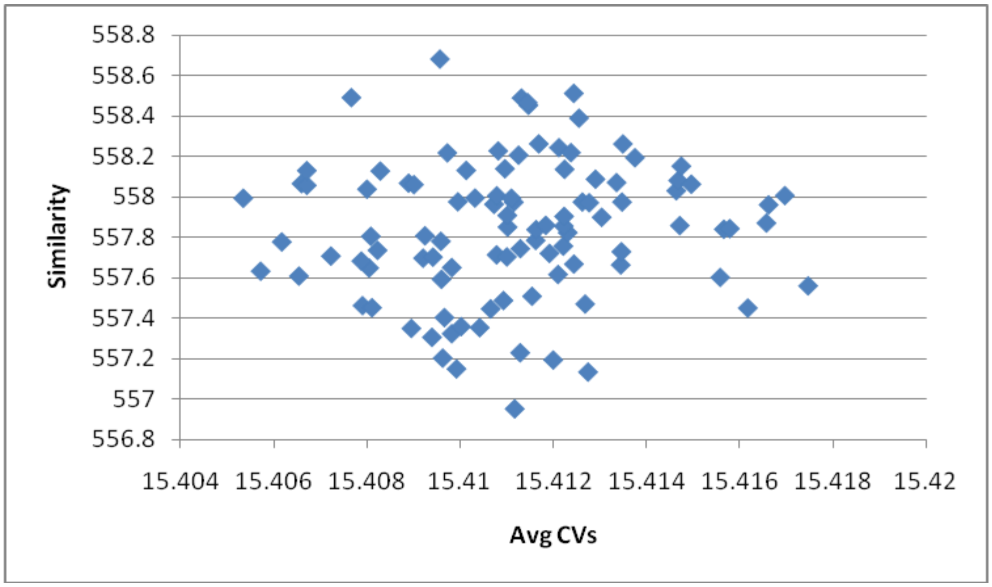


Figure 5.6 Particles formed by PFID after 150 iterations

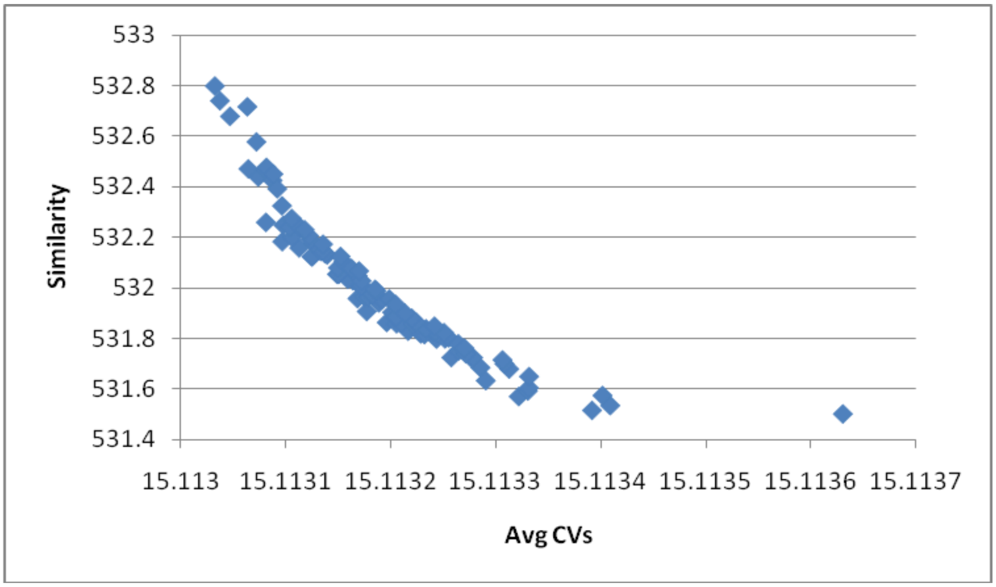
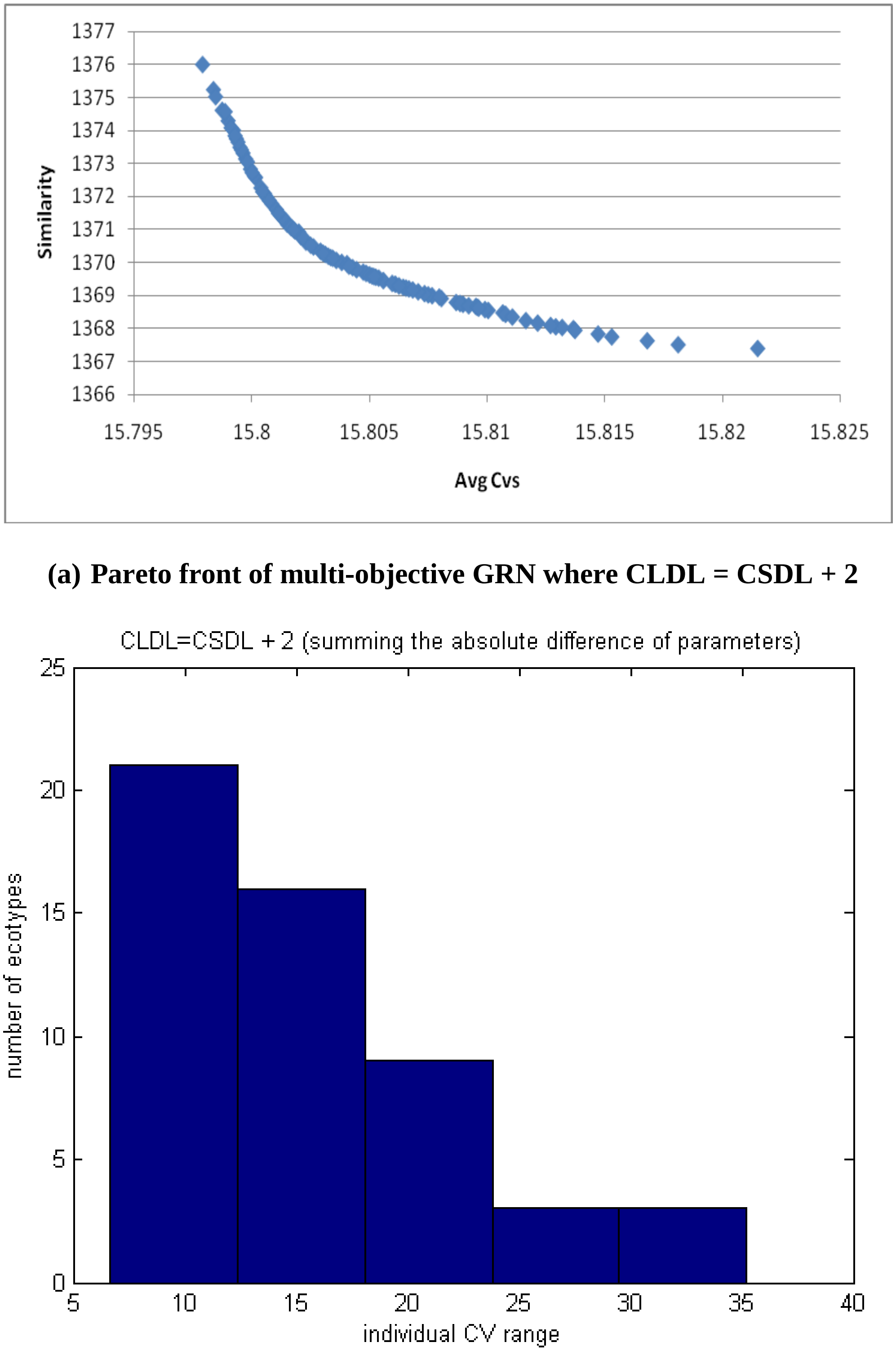


Figure 5.7 Particles formed by PFID after 200 iterations

Figure 5.8 reflects the formula stated in equation (5.5), changing the value of *CLDL* to *CSDL* + 2. The figure is presented below.

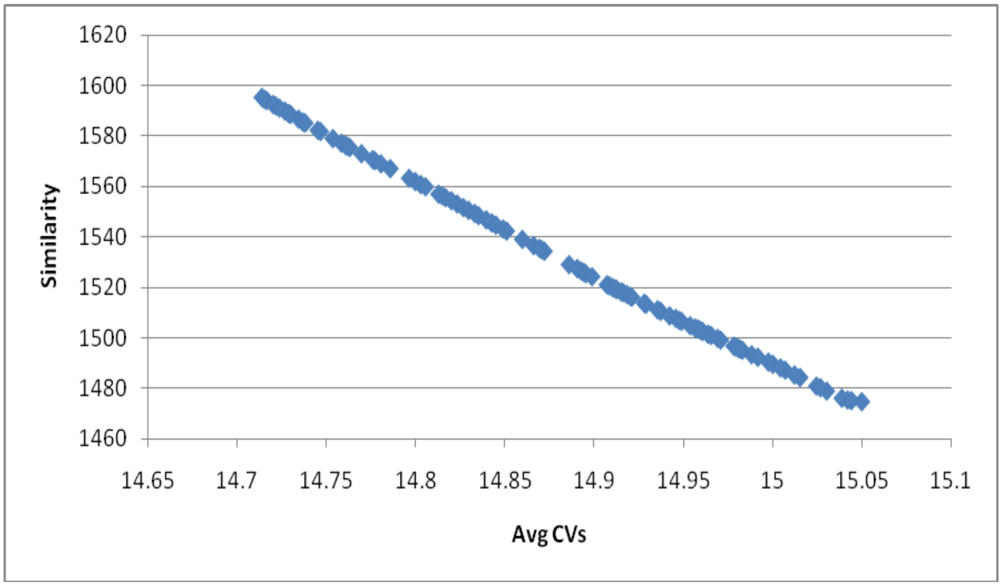


(a) Pareto front of multi-objective GRN where CLDL = CSDL + 2

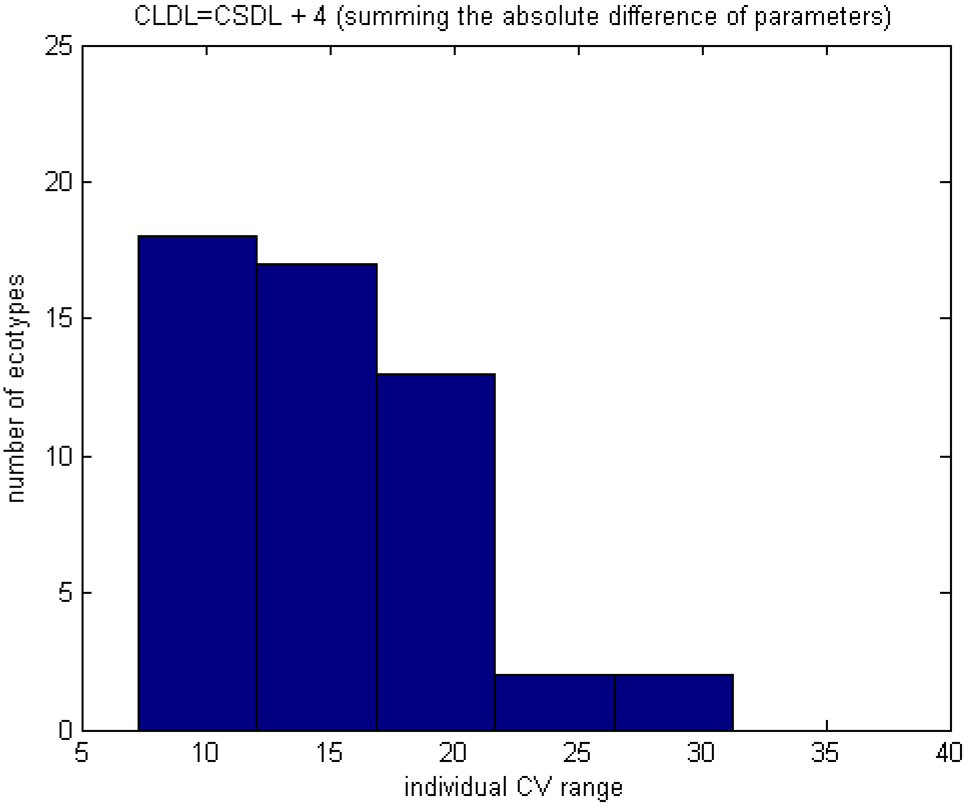
(b) Histogram of individual CVs

Figure 5.8 Pareto front for multi-objective GRN produced by PIFD when summing the absolute difference of parameters and histogram of ecotypes' individual CVs

In Figure 5.8 the value of *CLDL* was changed to *CSDL* +2. The graph appears to be right-skewed as well. In Figure 5.9 below, the *CLDL* is set back to *CSDL* + 4 while maintaining the same formula as in equation (5.5).



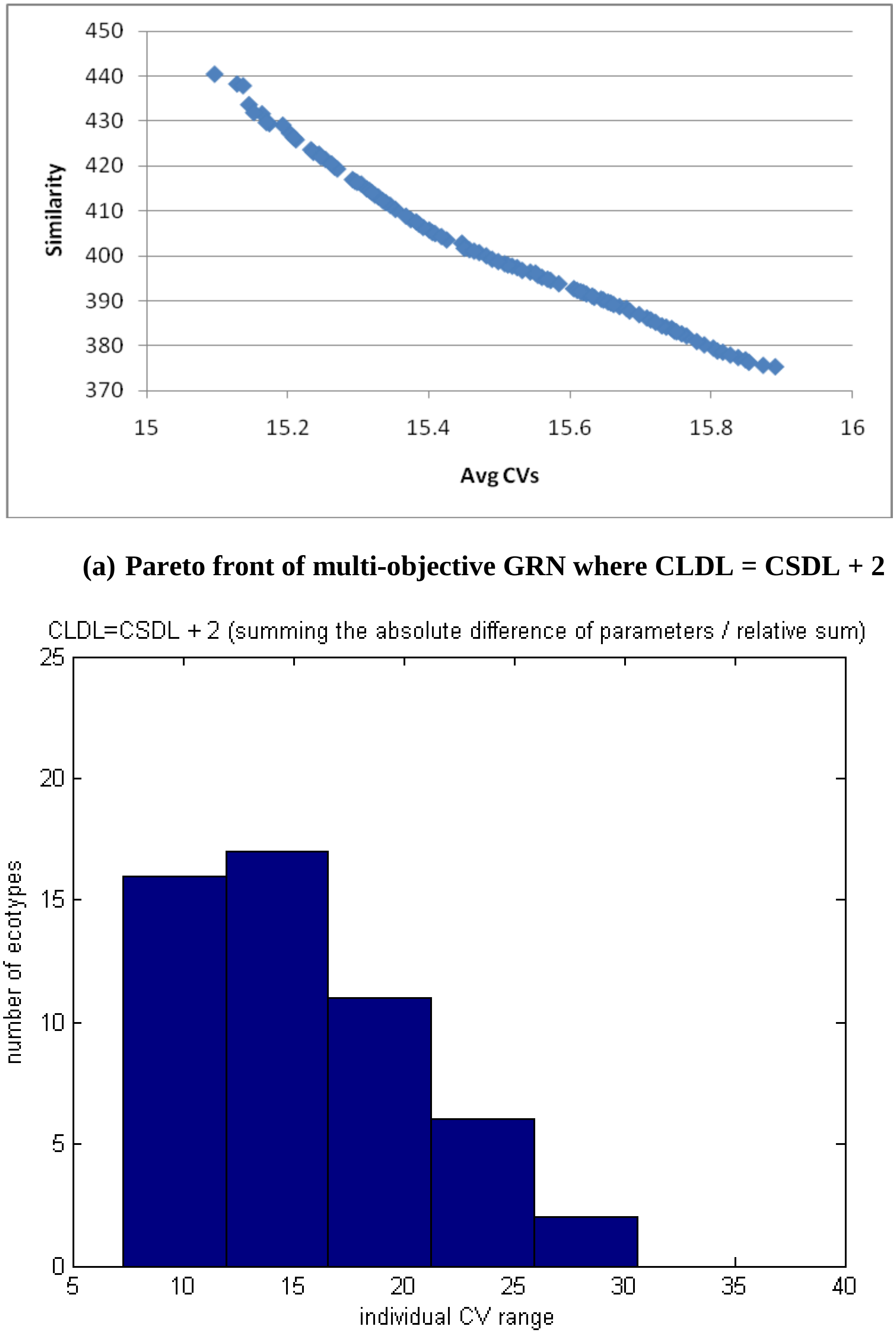
(a) Pareto front of multi-objective GRN where CLDL = CSDL + 4



(b) Histogram of individual CVs

Figure 5.9 Pareto front for multi-objective GRN produced by PIFD when summing the absolute difference of parameters and histogram of ecotypes' individual CVs

The next experiment to be conducted is changing the energy-oriented objective to the corresponding formula in equation (5.6), and setting *CLDL* to *CSDL* + 2. Figure 5.10 shows the Pareto front of the simulation run and histogram of individual *CV*s.

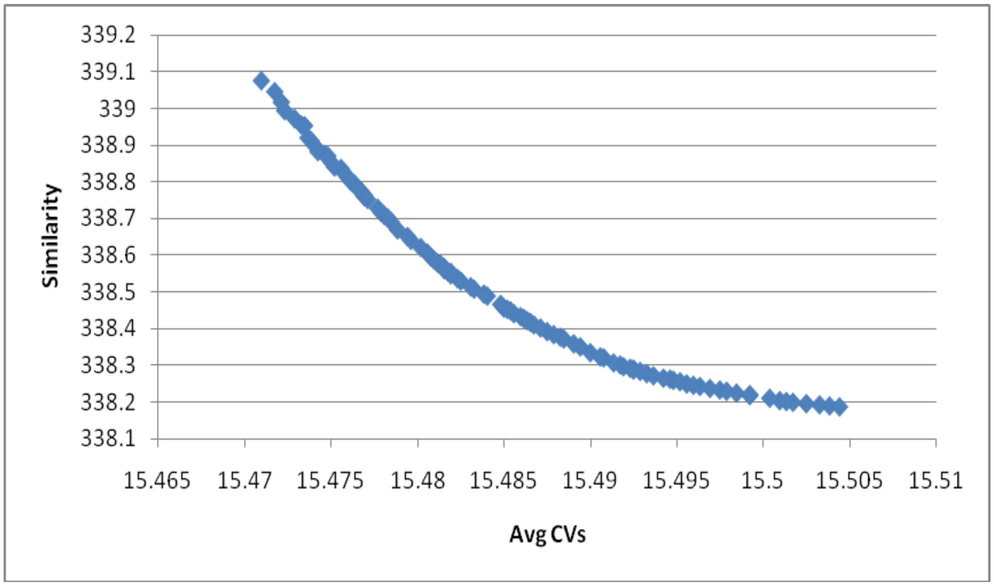


(a) Pareto front of multi-objective GRN where CLDL = CSDL + 2

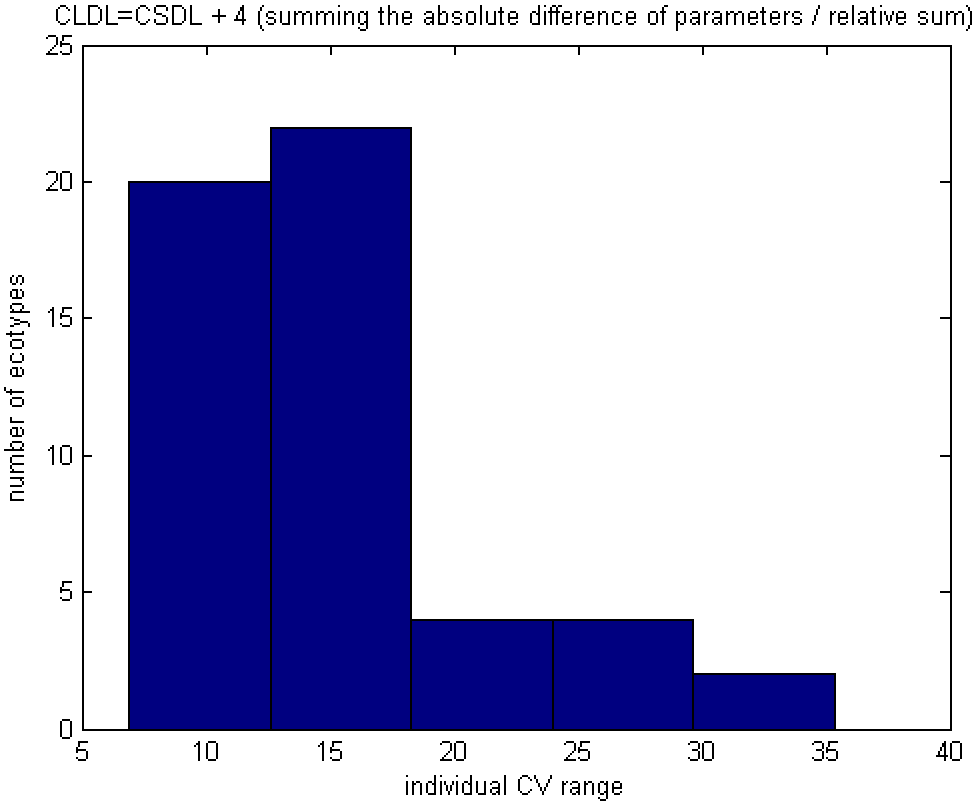
(b) Histogram of individual CVs

Figure 5.10 Pareto front for multi-objective GRN produced by PIFD when summing the absolute difference of parameters/relative sum and histogram of ecotypes' individual CVs

The individual *CV*s in the above figure also appear to be right-skewed. In the figure below, the value of *CLDL* is changed to *CSDL* + 4 using the same formula of equation (5.6).



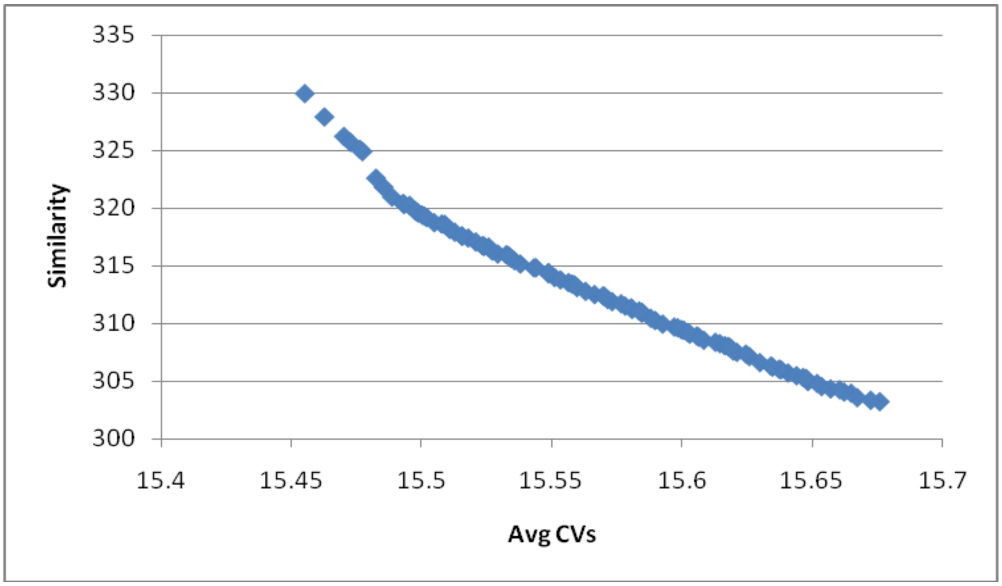
(a) Pareto front of multi-objective GRN where CLDL = CSDL + 4



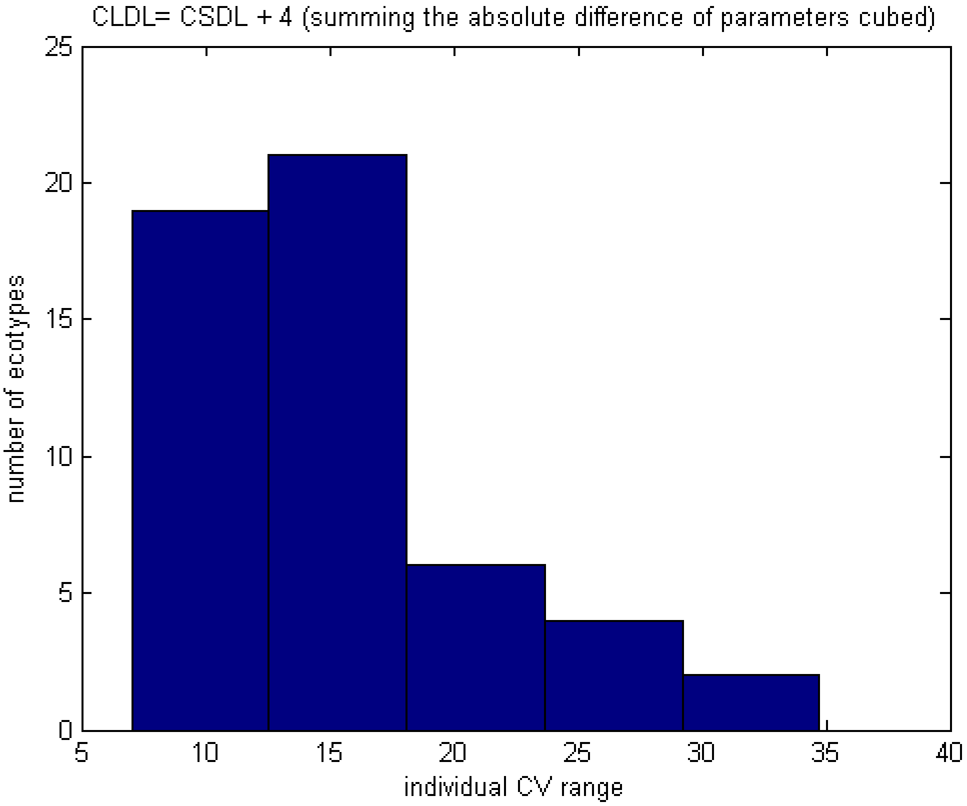
(b) Histogram of individual CVs

Figure 5.11 Pareto front for multi-objective GRN produced by PIFD when summing the absolute difference of parameters/relative sum and histogram of ecotypes' individual CVs

The results of the last run tend to be very promising as the majority of ecotypes lie in the desired range. By observing the previous figures and performing data analysis, it is believed that *CLDL* = *CSDL* + 4 outputs better results for all formulas. Finally, the last experiment brings in the energy-oriented objective described in equation 5.7, and allows for the ecotypes that were genetically similar to suffer penalties for parametric differences. Results are presented in Figure 5.12.



(a) Pareto front of multi-objective GRN where CLDL = CSDL + 4



(b) Histogram of individual CVs

Figure 5.12 Pareto front for multi-objective GRN produced by PIFD and summing the cubed difference of parameters histogram of ecotypes' individual CVs

Surprisingly, the cubing of absolute difference of parameter values did not force any penalty on similar ecotypes. Again, in the above figure the results are appeared to be right-skewed and most of ecotypes managed to remain in the desired range of *CV*s.

It is concluded that the change in the background similarity between ecotypes has no effect on their individual *CV*s. Different spectra of the energy-oriented objective have no effect on ecotypes. All figures above (Figure 5.2 (b), Figure 5.8 (b) – 5.12 (b)) are roughly right-skewed which implies that most ecotypes reside in the desired range i.e. similar.

CHAPTER 6 - CONCLUSION & FUTURE WORK

Despite the fact that classical optimization techniques solve optimization problems; they are not adequate when it comes to solving complex problems. However, evolutionary algorithms based on swarm-intelligence well address complex real-world multi-objective problems. This chapter presents in essence the overall conclusions that have been observed throughout this study.

6.1 Research Conclusion

The Partially Informed Fuzzy-Dominance (PIFD) based PSO method proposed in this study proved to maintain a diverse set of solutions along with a remarkable improvement in convergence rate. Diversity among candidate solutions is maintained by the adoption of fuzzy dominance concept, leading to only archive non-dominated individuals while discarding non-qualified ones. Convergence is implemented by making use of a dynamic network topology where particles are guided by a randomly generated number of elite individuals selected from the archive. This group of particles follows a normal distribution with specific mean and variance.

Testing of the proposed algorithm has been implemented as discussed in Chapter 3 using four benchmark problems. All four test problems, KUR, ZDT1, ZDT3, and ZDT6 have shown remarkable results in terms of convergence rate and diversity among individuals. Their corresponding Pareto fronts have been presented and explained as well. They clarify how rapidly individuals converge given the test problems, using fewer number of fitness function evaluations. Performance measures such as spacing metric have been used to measure how spread the solutions are in the produced Pareto fronts. A table of readings shows an outstanding results regarding diversity aspect.

In Chapter 4, a photothermal model was used to fit wide number of ecotypes in *A. thaliana* plant. It controls flowering time of *A. thaliana* in which three entangled genes are responsible for the floral process. The model implies that once a plant’s developmental units accumulate to a certain threshold, floral initiation occurs. Estimates of the confidence regions of ecotypes’ parameters were performed by making use of C-PSO, along with the discussed model. The algorithm produced two populations of individuals and whose confidence intervals for five ecotypes are shown in Section 4.2, each of a different pathway background. C-PSO was also used to minimize the coefficient of variation of single-objective GRN across the nine plantings used in this study. Within the same chapter, the produced populations were used to calculate ecotypes’ envelope curves by running the model and making use of the archived individuals. Maximum and minimum number of hours needed for each ecotype to bolt throughout the year experiencing various weather data plotted in Figures 4.6 through 4.10, which are known as MTTY curves.

Chapter 5 defines what similarity background is and how it is calculated among ecotypes. The chapter presents Pareto front figures of multi-objective GRN that branches to two objectives, data-oriented and energy-oriented. Figures were produced using the photothermal model and RFID algorithm to optimize the multi-objective GRN problem. Many formulas of the energy-oriented objective were investigated in order to determine effectiveness variation of parametric differences on ecotypes. All formulas of energy-oriented objective produced right-skewed histograms of individual CVs implying that penalty or freedom of ecotypes that are genetically similar for parametric differences has no consequences. The same conclusion holds for the total sum of ecotypes where parametric differences of genetically similar ecotypes have no role.

6.2 Future Work

In addition to the results presented throughout this study for single and multi-objective GRN using the investigated algorithm, further experiments can be conducted in future research opportunities that include the following.

- Study the effect of applying dynamic network topologies of neighbors on search space individuals using different probability distributions.

- Incorporate algorithmic improvements to maintain diversity among solutions, and analyze the rate of convergence of the algorithm with reduced number of fitness function evaluations.
- Optimize insensitive parameters used to compute MPTUs, using an appropriate method.
- Compare the obtained parameters to older ones and observe the effect of change on MPTU values.
- Further study of the effect of change on the coefficient of variation and total sum of energy-oriented objective after plugging in the new parameters, observing the ones that contribute to the change.

REFERENCES

[1] Ahuja, A., Das, S. and Pahwa, A., (2007). “An AIS-ACO Hybrid Algorithm for Multi-objective Distribution System Configuration Problem,” IEEE Transactions on Power Systems, Vol. 22, No. 3, pp. 1101 – 1111.

[2] S. Das, Min Gui, and Pahwa, A., (2008). “Artificial immune systems for self-nonself discrimination: Application to anomaly detection,” Studies in Computational Intelligence Vol. 116, pp. 231 – 248.

[3] Das, S., Natarajan, B., Stevens, D., and Koduru, P., (2008). “Multi-objective and constrained optimization for DS-CDMA code design based on the clonal selection principle,” Elsevier Science Publishers, Vol. 8, No. 1, pp. 788 – 797.

[4] Koduru, P., Das, S., Welch, S.M., and Roe, J., (2004). “Fuzzy Dominance Based Multi-objective GA-Simplex Hybrid Algorithms Applied to Gene Network Models,” The Genetic and Evolutionary Computation Conference Vol. 1, pp. 356 – 367.

[5] Kennedy, J. and Eberhart, R.C., (1995). “Particle swarm optimization,” Proc. IEEE Int'l. Conf. on Neural Networks, IEEE Service Center, Piscataway, NJ pp. 1942 – 1948.

[6] Koduru, P., Das, S. and Welch, S. (2006). “A particle swarm optimization-nelder mead hybrid algorithm for balanced exploration and exploitation in multidimensional search space,” International Conference on Artificial Intelligence, Vol. 2, pp. 457–464.

[7] Zielinski, K. and Laur, R., (2007). “Stopping criteria for a constrained single-objective particle swarm optimization algorithm,” Informatica Vol. 35, pp. 51 – 59.

[8] Mendes, R., Kennedy, J. and Neves, J., (2004). "The fully informed particle swarm: Simpler, maybe better," IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, pp. 204 – 210.

[9] Wolpert, D.H. and Macready, W.G., (1995). "No free lunch theorems for search," Santa Fe Institute, Technical Report SFI-TR-95-02-010, [Online]. Available: citeseer.nj.nec.com/wolpert95no.html.

[10] Das, S., Morcos, K. and Welch, S.M., (2009). "Combining fuzzy dominance based PSO and gradient descent for effective parameter estimation of gene regulatory networks," Proc. Multi Conference on Computer Science and Information Systems, Algarve, Portugal, (Ed. Antonio Palma dos Reis), pp. 3 – 10.

[11] Das, S., and Panigrahi, B.K., (2008). "Multi-objective evolutionary algorithms," Encyclopedia of Artificial Intelligence, (Eds. J. R. Rabuñal, J. Dorado & A. Pazos), Idea Group Publishing, Vol. 3, pp. 1145 – 1151.

[12] Koduru, P. *et al.*, (2008). "Multi-objective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks," IEEE Transactions on Evolutionary Computation, Vol. 12, No. 5, pp. 572 – 590.

[13] Koduru, P., Das, S., and Welch, S.M., (2007). "Multi-objective and hybrid PSO using ε-fuzzy dominance," Proc. Genetic and Evolutionary Computing Conference, London, UK, pp. 853 – 860.

[14] Koduru, P., Das, S. and Welch, S.M., (2009). "A hybrid PSO algorithm for single and multi-objective optimization," IEEE Transactions on Evolutionary Computation, revised and resubmitted.

[15] Mendel, J.M., (1995). "Fuzzy logic systems for engineering: A Tutorial," Proc. IEEE, Vol. 83, No. 3, pp. 345 – 377.

[16] Schott, J. R., (1995). “Fault tolerant design using single and multicriteria genetic algorithm optimization,” M.S. thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. Of Technology, Cambridge, MA.

[17] Koduru, P., Welch, S.M., and Das, S., (2007). “A particle swarm optimization approach for estimating confidence regions,” Proc. Genetic and Evolutionary Computing Conference, London, UK, pp. 70 – 77.

[18] Wilczek A. *et al.*, (2009). “Effects of genetic perturbation on seasonal life history plasticity,” Science Express DOI: 10.1126/science.1165826, Vol. 323. No. 5916, pp. 930–934.

[19] Vangel, M. G., (1996). “Confidence Intervals for a normal coefficient of variation,” The American Statistician, Vol. 50, No. 1, pp. 21 – 26.

[20] El-Lithy M. *et al.*, (2006). “New Arabidopsis Recombinant Inbred Line Populations Genotyped Using SNPWave and Their Use for Mapping Flowering-Time Quantitative Trait Loci,” Genetics, Vol. 172, No. 3: pp. 1867 – 1876, Supp. Table 1.

[21] Zhao K., Aranzana M. J., Kim S., Lister C., Shindo C., et al. (2007). “An Arabidopsis example of association mapping in structured samples,” PLoS Genet 3(1): e4. doi:10.1371/journal.pgen.0030004.

Appendix A - Notations

χ	Constriction coefficient that maintains particles' stability
C_1	Cognitive constant
C_2	Social constant
D_{SD}	Short day development rate
D_{LD}	Long day development rate
$CSDL$	Critical short day length in hours
$CLDL$	Critical long day length in hours
T_b	Base temperature
F_b	Baseline repression level
U	Number of line parameters
k_{ijg}	Value of gene *g* for lines *i* and *j;* one if identical, zero if not
G	Total number of markers present within an ecotype excluding unknown markers
κ	Overall scaling factor
$T_{V\min}$	Minimum vernalizing temperature
$T_{V\max}$	Maximum vernalizing temperature
ε	Constant value of 0.01
ω	Exponent on difference from $T_{V\min}$
λ	Exponent on difference from $T_{V\max}$
L	Total number of ecotypes
M	Number of objectives in a given function

$\bar{E}$	Mean of all the E_i
E_i	Minimum difference between a solution *i* and its closing neighbor for all objectives
SP	Spacing metric for measuring diversity between non-dominated solutions
ST	Threshold value at which plants bolt
Vr_{sat}	Number of hours needed to saturate vernalization
Λ	Updated value of CV estimate after each iteration
υ	Degrees of freedom for calculating a specific CI
Δf_i	Difference between fitness function values of two particles for objective *i*
Δ_i	Difference between maximum and minimum values for objective *i*
$U[0,1]$	Uniformly distributed random number between 0 and 1
$N(\mu,\sigma^2)$	Normally distributed random number with mean μ and variance σ^2
P	Population of solutions
A	External archive that holds non-dominated solutions
$f(i)$	Fitness value of particle *i*
δ	Turbulence factor
Θ	Solution space
$\vartheta_{i,j}$	Similarity value that corresponds to lines *i* and *j*
β_i	Marker presented in a specific line *i*
τ	Instant in time
B_0	Sowing hour of a specific gene
B_n	Bolting hour of a specific gene
CI	Confidence interval of an estimated parameter

f_D	Data-oriented objective
f_E	Energy-oriented objective
ϖ_D	Relative weight to data-oriented objective
ϖ_E	Relative weight to energy-oriented objective
l_i	Denotes line *i*
u_1	Lower bound of the confidence interval
u_2	Upper bound of the confidence interval
ζ_i^{dom}	Membership function value of particle *i*
$(\vec{\mathbf{x}} \succ_i^F \vec{\mathbf{y}})$	Solution vector $\vec{\mathbf{x}}$ dominates solution $\vec{\mathbf{y}}$ for objective *i*
$(\vec{\mathbf{x}} \succ^F \vec{\mathbf{y}})$	Solution vector $\vec{\mathbf{x}}$ dominates solution $\vec{\mathbf{y}}$ for all objectives
$\vec{\mathbf{x}}_t$	Particle’s current position at t time
$\vec{\mathbf{x}}_{lb}$	Particle’s best position during search process, local best
$\vec{\mathbf{x}}_{gb}$	Population’s best candidate, global best
$\vec{\mathbf{v}}_t$	Current velocity at t time
$\vec{\mathbf{v}}_{t+1}^{'}$	Velocity after applying turbulence
$\vec{\mathbf{x}}_t^w$	Outlier particle with worst fitness value
$\vec{\mathbf{x}}_t^g$	Best fitted particle used in crossover to replace $\vec{\mathbf{x}}_t^w$
$\vec{\mathbf{x}}_t^{'}$	Newly created particle that replaced $\vec{\mathbf{x}}_t^w$
$\mathbf{K}^*$	Matrix of pairwise kinship coefficients