

Hdl-specific oligonucleotides. *Hdl* state variable data (equations 1,2) are presented as dimensionless $\Delta\Delta Ct$ values normalized by *OsGAPDH* expression levels as a control.

2. THE MULTI-OBJECTIVE HYBRID APPROACH

2.1 Multi-objective Optimization

Evolutionary algorithms have emerged as one of the most popular approaches for the complex optimization problems. They draw upon Darwinian paradigms of evolution to search through the solution space (the set of all possible solutions). Starting with a set (or population) of solutions, in each generation of the algorithm, new solutions are created from older ones by means of two operations, mutation and crossover. Mutation is accomplished by imparting a small, usually random perturbation to the solution. In a manner similar to the Darwinian paradigm of survival of the fittest, only the better solutions are allowed to remain in a population, the degree of optimality of the solution being assessed through a measure called fitness.

When dealing with optimization problems with multiple objectives, the conventional concept of optimality does not hold well [9, 19, 11]. Hence, the concepts of dominance and Pareto-optimality are applied. Without a loss of generality, if we assume that the optimization problem involves minimizing each objective $e_i(\cdot)$, $i = 1..M$, a solution u is said to dominate over another solution v iff $\forall i \in \{1, 2, \dots, M\}$, $e_i(u) \leq e_i(v)$ with at least one of the inequalities being strict, i.e. for each objective, u is better than or equal to v and better in at least one objective. This relationship is represented as $u \succ v$. In a population of solution vectors, the set of all non-dominating solutions is called the Pareto front. In other words, if S is the population, the Pareto Front Γ is given by,

$$\Gamma = \{u \in S \mid \forall v \in S, \neg(u \succ v)\} \quad (3)$$

The simplistic approach of aggregating multiple objectives into a single one often fails to produce good results. It produces only a single solution. Multi-objective optimization on the other hand involves extracting the entire Pareto front from the solution space. In recent years, many evolutionary algorithms for multi-objective optimization have been proposed [12, 13].

2.2 Fuzzy Dominance

Assume an overall minimization problem involving M objective functions $e_i(\cdot)$, $i = 1 \dots M$. The solution space is denoted as $\Psi \subset \mathfrak{R}^n$. Given a monotonically non-decreasing function $\mu_i^{dom}(\cdot)$, whose range is in $[0, 1]$, $i \in \{1, 2, \dots, n\}$ such that $\mu_i^{dom}(0) = 0$, solution $u \in \Psi$ is said to i -dominate solution $v \in \Psi$, if and only if $e_i(u) < e_i(v)$. This relationship can be denoted as $u \succ_i^F v$. If $u \succ_i^F v$, the degree of fuzzy i -dominance is equal to $\mu_i^{dom}(e_i(v) - e_i(u)) \equiv \mu_i^{dom}(u \succ_i^F v)$. Fuzzy dominance can be regarded as a fuzzy relationship $u \succ_i^F v$ between u and v . Solution $u \in \Psi$ is said to fuzzy dominate solution $v \in \Psi$ if and only if $\forall i \in \{1, 2, \dots, M\}$, $u \succ_i^F v$. This relationship can be denoted as $u \succ^F v$. The degree of fuzzy

dominance can be defined by invoking the concept of fuzzy intersection and using a t -norm,

$$\mu^{dom}(u \succ^F v) = \bigcap_{i=1}^M \mu_i^{dom}(u \succ_i^F v) \quad (4)$$

Given a population of solutions $S \subset \Psi$, a solution $v \in S$ is said to be fuzzy dominated in S iff it is fuzzy dominated by any other solution $u \in S$. In this case, the degree of fuzzy dominance can be computed by performing a union operation over every possible $\mu^{dom}(u \succ^F v)$, carried out using t -co norms as,

$$\mu^{dom}(S \succ^F v) = \bigcup_{u \in S} \mu^{dom}(u \succ^F v) \quad (5)$$

In this manner, each solution can be assigned a single measure to reflect the amount it dominates others in a population. Non-dominated individuals within a solution will be assigned zero fuzzy dominance, as for any non-dominated individual $\mu_i^{dom}(u \succ_i^F v)$.

Further details of this concept can be found in [7]. In the present work, the membership $\mu_i^{dom}(u \succ_i^F v)$ is piecewise linear, and given by,

$$\mu_i^{dom}(\Delta e_i) = \begin{cases} 0, & \Delta e_i \leq 0 \\ (\Delta e_i) / \Delta_i, & 0 < \Delta e_i < \Delta_i \\ 1, & \Delta e_i \geq \Delta_i \end{cases} \quad (6)$$

where, $\Delta e_i = e_i(v) - e_i(u)$. The union and intersection operators follow the standard min and max definitions [14].

2.3 The Fuzzy Simplex Genetic Algorithm

Fuzzy dominance time makes it possible to assign a single measure of fitness to multiple individuals. Computing the mutual fuzzy dominance in a population of individuals enables local gradient descent based techniques to be applied in a multi-objective framework. In [7] a strategy was proposed that applied a local search procedure, the Nelder-Mead algorithm, in conjunction with a genetic algorithm.

A simplex in n -dimensions consists of $n+1$ solutions u_k , $k = \{1, 2, \dots, n+1\}$ which are its vertices. In a plane, this corresponds to a triangle. The solutions are evaluated in each step and the worst solution w is identified. The centroid of the simplex is then evaluated, excluding the worst solution and the worst point is reflected along the centroid. If $c = \sum_k u_k - w$ is the

centroid, the reflected solution is

$$r = c + (c - w) \quad (7)$$

Usually, the worst point w is replaced with the reflected point r in the simplex, but if the r is better than any solution in the simplex, the simplex is further expanded as,

$$r_e = c + \eta(c - w) \quad (8)$$

where η is called the expansion coefficient. However, if the reflected solution r is worse than w , the simplex is contracted and the reflected solution is placed on the same side of the centroid. When solution r is not worse than w , but worse than any other solution in the simplex, the simplex is still contracted,

but the reflection is allowed to remain on the other side of the simplex's centroid. Reflection is carried out as follows,

$$r_c = c \pm \kappa(c - w) \quad (9)$$

In the above equation, κ is called the contraction coefficient. Solution w is replaced with the new one, r , r_e , or r_c in the next step. The simplex algorithm is allowed to run for multiple steps before it converges.

Fuzzy dominance is the objective function to which Nelder-Mead is applied. This allows the Nelder-Mead algorithm to push solutions towards regions of lower dominance, i.e. the Pareto front.

In the earlier technique, the Nelder-Mead simplex search was incorporated within a multi-objective genetic algorithm. Within each generation, $n + 1$ solutions were selected randomly from the population, and a fixed number of iterations of the Nelder-Mead procedure were applied. Selecting points from the population of solutions for the Nelder-Mead algorithm, either randomly, or using some other ordering technique such as picking fitter solutions, has been used routinely in the literature on GA-simplex hybrid algorithms [7, 15, 16, 17]. Unfortunately, the Nelder-Mead algorithm fails to reach a local minimum when the corners of the simplex are spread too far apart in the search space. This is because in a complex landscape, a planar approximation of a large region of the front, that the simplex attempts to fit, may not be feasible. This research proposes the use of the k -means clustering algorithm to select the corners of the simplex [7, 15, 16, 17].

Using K -means clustering, a total of K cluster centers, $c_1 \dots c_K$ are generated randomly in the search space. Then a two-step iterative process adjusts the positions of the cluster centers. In the first step the algorithm computes the cluster center that lies closest to each point in the population. Any point u_i is said to belong to cluster c_k iff $k = \arg \min_{k'} (\|u_i - c_{k'}\|)$. The second step replaces the cluster center with the average of all the points, i.e., $c_k = \sum_{u_i \in c_k} u_i$.

K -means clustering breaks up the population into closely spaced clusters. We have applied the K -means algorithm to identify such clusters before applying the Nelder-Mead algorithm to each cluster. Specifically, at the beginning of each generation of the hybrid algorithm, the entire population is clustered. A few clusters are picked for the application of the Nelder-Mead algorithm. When there are more than $n + 1$ solutions, the vertices are selected at random from within the cluster. On the other hand, when the cluster size is too small, the cluster is simply ignored. A fraction of the offspring population is obtained from the Nelder-Mead algorithm, while the rest of the offspring is obtained by the genetic operators of selection, crossover and mutation.

A binary tournament selection is implemented in the genetic algorithm that selected two individuals at random from the population with replacement, and picks the one with the least fuzzy dominance. An offspring t , was computed from two parents u and v in the following manner,

$$t = \zeta u + (1 - \zeta)v \quad (10)$$

where ζ is a uniformly distributed random number in $[0, 1]$.

Solutions were mutated with a probability of β , by adding a random number with zero mean, that followed a Gaussian distribution with a spread σ , according to,

$$u = u + N(0, \sigma) \quad (11)$$

The fuzzy logic parameters in [7] were fixed at preset values that was a fraction of the range of values that the function throughout the population. Although the algorithm was able to converge rapidly towards the Pareto front of the solution space, the spread of the final front obtained was highly localized. Much of the effort in current multi-objective optimization is targeted towards finding solutions that are not only close to the Pareto front, but also that sample the front at approximately regular intervals. The present work rectifies this shortcoming by applying a co-evolutionary method to adapt the fuzzy logic parameters; $\Delta_i, i = 1, \dots, M$.

While there is considerable scope for improvement in spacing individuals along the front, speedup in the algorithm's convergence can also be accomplished. In [7] where values for the parameters associated with membership functions (Δ_i s) were held constant, it has been argued that for simple monotonic landscapes the simplex is "flipped" in a direction orthogonal to the non-dominated front. In the present research these values can co-evolve. Therefore the direction along which the "flipping" occurs can be controlled. Furthermore, co-evolving these parameters aids the genetic algorithm as well. As the population advances towards the Pareto front, the parameters adapt so that the advance can be made in the direction which allows faster progress towards the Pareto front. This is depicted in Figure 1 for a two-dimensional objective function space. The co-evolutionary approach is discussed in the following subsection.

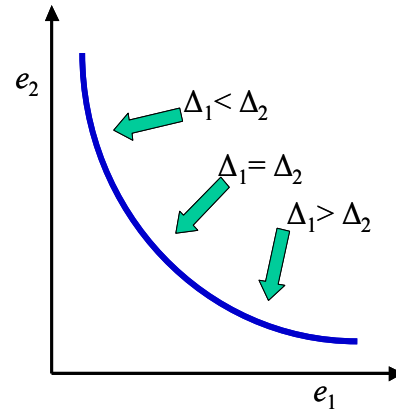


Figure 1. Influence of the fuzzy parameters on the direction of progress of the front.

3. THE CO-EVOLUTIONARY ALGORITHM

The co-evolutionary approach treats the set of parameters of the hybrid algorithm as its individual solutions. However, to distinguish between the individuals of the main algorithm, we will call the individuals of the co-evolutionary process as simply the parameter vectors. Because of the low dimensionality of the search space, a simple (1+1)-ES (evolutionary strategy) has been considered for co-adapting the parameter vectors. The fitness function of each vector is designed to take into account both the spread along the Pareto front of the main algorithm, as well as its rate of convergence.

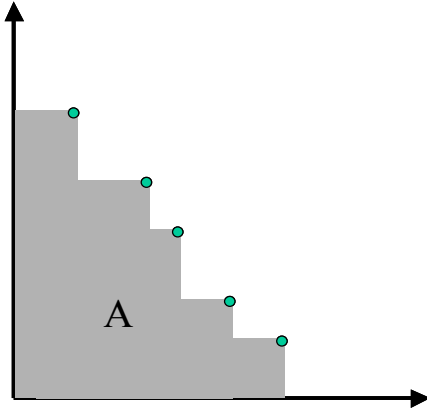


Figure 2. Hyper-area covered by a non-dominated set of solutions

For each solution in the non-dominated front, the hyper-area of the hyper-rectangle formed by the solution and the origin as its extremities can be a rough measure of the proximity of the closeness of the solution to the origin. In [18] the union of the hyper-area of all the non-dominated solutions is considered as a measure of proximity of the entire front to the origin. In minimization problems such as ours, where the ideal point is the origin, this hyper-area $A(S)$ covered by a set of non-dominated solutions S , shown in Figure 2 is a good measure of quality.

In order to measure the quality of the distribution of a non-dominated front, the algorithm borrows a feature from [19], where a method is suggested to break up a set of solutions into groups based on the angles they subtend with the axes. For the two-dimensional case that this paper considers, a scalar measure is obtained for each solution v ,

$$\sigma(v) = \frac{e_1^2(v) - e_2^2(v)}{e_1^2(v) + e_2^2(v)} \quad (12)$$

It can be shown that the loci of all points with constant values of σ will be straight lines intersecting the origin as shown in Figure 3.

After the σ 's of the non-dominated set are computed, the algorithm divides the set into different intervals as shown in Figure 3. Each interval corresponds to a range of values of σ . When dividing the range into L intervals, the l^{th} interval would contain solutions whose σ 's lie within the range $[(\frac{1}{2}L-l)/L, (\frac{1}{2}L-l+1)/L]$. Following this division, the number of solutions in each range is computed. If the number of solutions in the l^{th} interval is n_l , the entropy of the set is computed as,

$$\eta(S) = - \sum_l \frac{n_l}{n} \log\left(\frac{n_l}{n}\right) \quad (13)$$

where S is the set of non-dominated solutions whose size is n . This entropy gives an effective measure of the distribution obtained by the algorithm. A higher value of $\eta(S)$ reflects a better spread of the solutions. This new measure is an innovation of the present research.

This article uses the ratio, $\eta(S)/A(S)$ as a measure of how good a set S is. In order to measure the effectiveness of a parameter vector, the improvement in this ratio in a single iteration is taken

as the fitness. The co-evolutionary process maintains only a single parameter vector that it perturbs and applies to the main algorithm. The η/A improvement is computed after one generation of the main algorithm. If it produces more improvement than the one recorded by the parent vector, the latter gets replaced by the mutant; otherwise, the mutated vector is discarded.

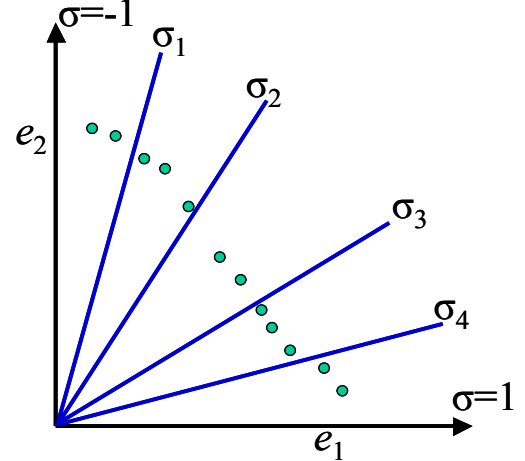


Figure 3. Lines of constant σ in the objective function space

4. RESULTS

The proposed algorithm was implemented to predict the Heading date (Hd1) at 28°C and 20°C for Short Day (SD) period by evolving the parameters of equation (1). The results obtained were compared with NSGA-II [13]. For both the algorithms a population size of 100 was used. The mutation and crossover probabilities were fixed at 0.1 and 0.7 respectively. For the co-evolution in the proposed method (the Fuzzy Simplex Genetic Algorithm, or FSGA), a mutation probability of 0.1 was used. Two fitness functions were defined as a function of mean squared error (MSE) between the experimental values and predicted values of Hd1 data at 28°C and 20°C, making the problem a multi-objective problem with two objectives. These two objectives have to be simultaneously minimized by the multi-objective algorithms. The model for the prediction uses a temperature dependency, hence making it possible to predict for both the temperatures simultaneously. The actual fitness function used is a modification of MSE as given below,

$$e = f_1 * f_2 * f_3 \quad (14)$$

where,

$$f_1 = \| Hd1_{\text{exp}} - Hd1_{\text{sim}} \| \quad (15)$$

$$f_2 = \text{count}(Hd1_{\text{sim}} \leq 0) \quad (16)$$

and,

$$f_3 = 1 + \left(\frac{\| \text{area}(Hd1_{\text{exp}}) - \text{area}(Hd1_{\text{sim}}) \|}{\text{area}(Hd1_{\text{exp}})} \right)^{\frac{1}{2}} \quad (17)$$

The error function has three different terms. The first term is the MSE between the experimental data and simulated data by the evolved equation. The second part is the number of times the simulated curve of Hd1 changes its sign. Without using this term in the fitness measure during evolution of the parameters, the solutions were trying to minimize the MSE by fitting an oscillating

curve. The best fit curve was not a model of Hd1 but high frequency noise data that happened to be close to actual data points giving it a low MSE. From a biological stand point we know Hd1 cannot have negative values, so in order to avoid solutions which are oscillatory we multiply the fitness function by 'n' which is number of times the curve takes predicted value of Hd1 less than zero. The third part is the function of difference between the areas under the curve of experimental data and simulated data between the same intervals. Without this term the algorithm fits a curve that is nearly a straight line between the experimental data available and is not actually trying to fit the form of the curve. Using the difference of area under the curve also in fitness forces the algorithm to find solutions that are having the form and reject those which do not conform to the shape.

With the aforementioned fitness function both FSGA and NSGA-II were run on same initial populations and allowed to converge to a good Pareto front. The FSGA converged to the front in about 7500 function evaluations, while NSGA-II needed much higher number of function evaluations. Figure 4 shows the convergence of FSGA after 2478, 4977 and 7477 function evaluations. As FSGA utilizes simplex in its runs at each generation and depending on how simplex flips the points the number of function evaluations at the end of each generation varies. So we show the Pareto front for the nearest function evaluations at 2478, 4977 and 7477 function evaluations. Figure 5 shows the convergence of

NSGA-II after 2500, 5000 and 7500 function evaluations. Comparing the two plots we observe that FSGA has a faster convergence compared to NSGA-II. Figure 6 shows fronts obtained by both the algorithms after 7500 function evaluations. Figure 7 shows the evolution of the parameters of fuzzy dominance with each generation in one of the runs. To compare the performance of the algorithm without co-evolution we run the same population without co-evolution and using fixed values for both the variables at 0.5 and 1.0 for two different runs. Table 1 shows the entropy, area under the curve and fitness values of the Pareto front calculated as explained earlier. We note that co-evolution resulted in a better performance compared to runs with fixed values.

Table 1. Comparison of non-dominated solution sets obtained after without co-evolution and with co-evolution.

Δ_1, Δ_2	η	A	η/A
0.5*range	1.3683	413.9992	.0033
1.0*range	1.0055	1.0380	0.9687
co-evolved	2.0479	0.9834	2.0825

Figure 8 and 9 show the time series prediction of one of the solutions of the Pareto front compared with the experimental data. An inset within each plot close-ups showing the accuracy in prediction at specified dates.

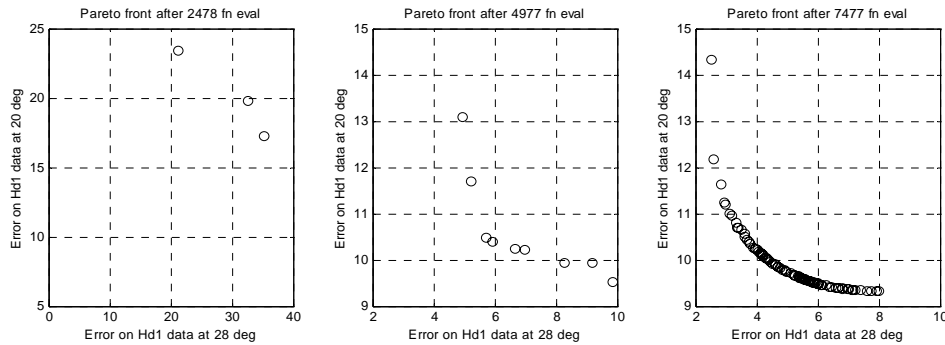


Figure 4. Subplots showing the non-dominated solutions front for FSGA after 2478, 4977 and 7477 function evaluations

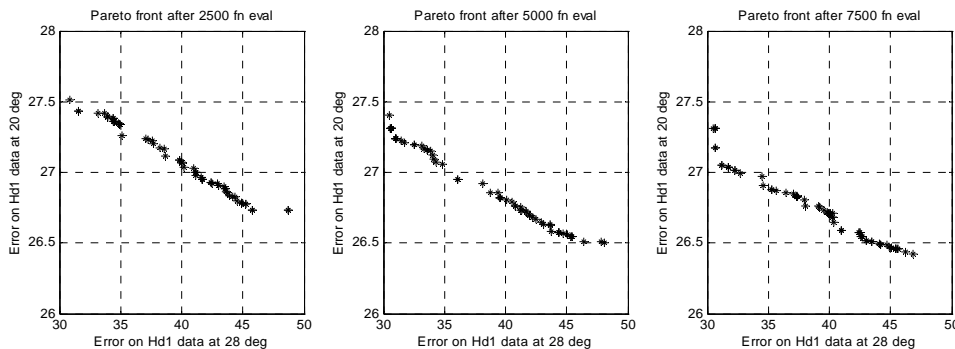


Figure 5. Subplots showing the non-dominated solutions front for NSGA-II after 2500, 5000 and 7500 function evaluations (Note the change in scale from Figure 4)

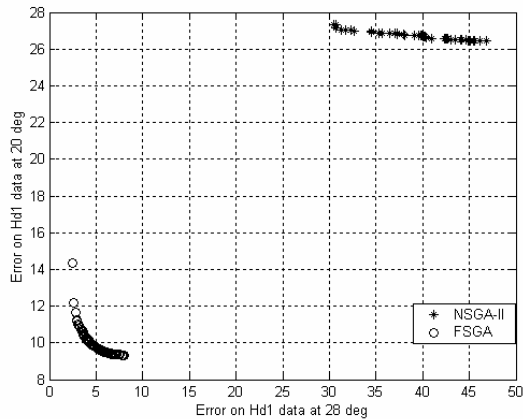


Figure 6. Comparison of the non-dominated solutions set (Pareto-front) after 7500 function evaluations for both NSGA-II and FSGA

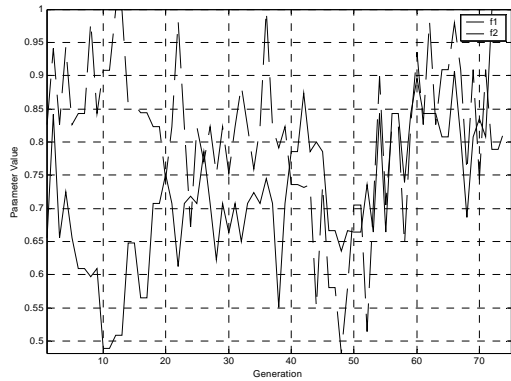


Figure 7. Coevolved parameters variation with generation for FSGA algorithm.

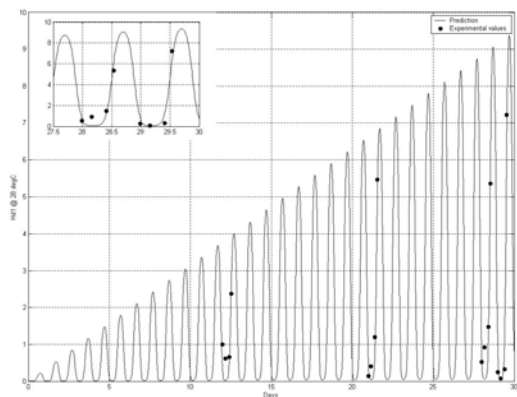


Figure 8. Plot showing prediction and experimental values of Hd1 at 28°C. The inset shows a close-up of the fit for the data during days 27.5 to 30.

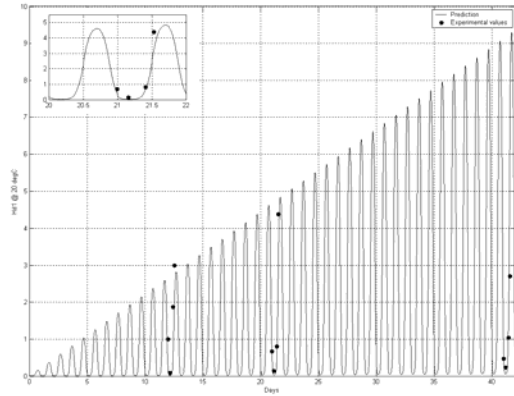


Figure 9. Plot showing prediction and experimental values of Hd1 at 20°C. The inset shows a close-up of the fit for the data during days 20 to 22.

REFERENCES

- [1] <http://ergo.integratedgenomics.com/GOLD/>
- [2] Cooper, M., Chapman, S.C., Podlich, D.W. & Hammer, G.L. *In Silico Biol.* 2 (2002), 151-164.
- [3] Welch, S.M., Roe, J.L., and Dong, Z.: A genetic neural network model of flowering time control in *Arabidopsis thaliana*. *Agron. J.* Vol 95 (2003) 71-81.
- [4] Welch, S.M., Dong, Z., and Roe, J.L.: Modelling gene networks controlling transition to flowering in *Arabidopsis*. 4th International Crop Science Congress, Brisbane, Au. Sep 26 – Oct 1, 2004a. (www.cropscience.org.au/icsc2004/symposia/3/2/index.htm#TopOfPage).
- [5] Dong, Z. Incorporation of genomic information into the simulation of flowering time in *Arabidopsis thaliana*. Ph.D. dissertation, Kansas State University (2003).
- [6] Welch, S.M., Roe, J.L., Das, S., Dong, Z., R. He, M.B. Kirkham.: Merging genomic control networks with soil-plant-atmosphere-continuum (SPAC) models. *Agricultural Systems* (2004) (*In press*).
- [7] P. Koduru, S. Das, S. M. Welch, J. L. Roe, "Fuzzy Dominance Based Multi-objective GA-Simplex Hybrid Algorithms Applied to Gene Network Models", Lecture Notes in Computer Science: Proceedings of the Genetic and Evolutionary Computing Conference, Seattle, Washington, 2004.
- [8] Kojima, S., Y Takahashi, Y. Kobayashi, L. Monna, T. Sasaki, T. Araaki, and M. Yano. 2002. Hd3a, a rice ortholog of the *Arabidopsis* FT gene, promotes transition to flowering downstream of Hd1 under short-day conditions. *Plant Cell Physiol.* 43:1096-1105.
- [9] Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, Vol. 3, no. 1 (1995) 1-16.
- [10] Coello Coello, C.A. : A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems*, Vol. 1, no. 3 (Aug 1999) 269-308.

- [11] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, Vol. 7, no. 2 (Apr. 2003) 117-132.
- [12] Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, Vol. 3, no. 4 (Nov. 1999) 257-271.
- [13] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2): 182-197 (2002).
- [14] Mendel, J.M.: Fuzzy logic systems for engineering: A tutorial, *Proceedings of the IEEE*, Vol 83, No. 3 (March 1995) 345-377.
- [15] Renders, J.M., Flasse, S.P.: Hybrid methods using genetic algorithms for global optimization, *IEEE Transactions on Systems, Man and Cybernetics Part-B*, Vol. 28, no. 2 (Apr. 1998) 73-91.
- [16] Yen, J., Liao, J.C., Lee, B., Randolph, D.: A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method, *IEEE Transactions on Systems, Man and Cybernetics Part-B*, Vol. 7, no. 1 (Feb. 2003) 243-258.
- [17] Robin, F.; Orzati, A.; Moreno, E.; Homan, O.J.; Bachtold, W.: "Simulation and evolutionary optimization of electron-beam lithography with genetic and simplex-downhill algorithms", *IEEE Transactions on Evolutionary Computation*, pp. 69-82, Vol. 7, no. 1, Feb. 2003.
- [18] Metrics for quality assessment of a multiobjective design optimization solution set, Jin Wu, Shapour Azarm, *Transactions of ASME*, Vol. 123, March, 2001, pp- 18-25.
- [19] S. Mostaghim and J. Teich, Strategies for finding good local guides in Multi-objective particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, Pages 26-33, 2003.