Guiding Multi Objective Evolutionary Algorithms Towards Interesting Regions

Jürgen Branke, Thomas Kaußler, Hartmut Schmeck
Institute AIFB, University of Karlsruhe
D-76128 Karlsruhe, Germany
Email:{branke|schmeck}@aifb.uni-karlsruhe.de

Abstract

Many real world design problems involve multiple, usually conflicting optimization criteria. Often, it is very difficult to weigh the criteria exactly before alternatives are known. Multi-Objective Evolutionary Algorithms based on the principle of pareto optimality are designed to produce the complete set of nondominated solutions, which allows the user to choose among many alternatives. However, although it is very difficult to exactly define the weighting of different optimization criteria, usually the user has some notion as to what range of weightings might be reasonable. In this paper, we present a novel, simple and intuitive way to integrate the user's preference into the evolutionary algorithm. On a number of test problems we show that the proposed algorithm efficiently guides the population towards the interesting region, allowing a faster convergence and a better coverage of the this area of the pareto optimal front.

Keywords: evolutionary algorithm, multiple objectives, pareto optimal, preferences

1 Introduction

Many real-world design problems involve multiple objectives that have to be optimized simultaneously. As these objectives are usually conflicting, no single solution may exist that is best regarding all criteria considered. Therefore, at some stage of the problem solving process, the decision maker (DM) has to articulate his/her preferences about the objectives. Following a classification by Veldhuizen [7], the articulation of preferences may be done either before (a priori), during (progressive) or after (a posteriori) the optimization process.

A priori approaches are usually not practicable, since they require the user to explicitly and exactly weigh the different criteria and to effectively turn the multi-objective optimization problem into a single objective problem prior to knowing possible alternatives.

A posteriori approaches may use an evolutionary algorithm (EA) to produce as many different pareto-optimal[1] solutions as possible, and then allow the DM to choose among them.

With progressive approaches, decision making and optimization are intertwined.

In general, a posteriori approaches are most convenient, since they do not require the user to interact with the process, and the DM can delay any choice until the alternatives are known. Unfortunately, searching for all pareto optimal solutions is a difficult and time-consuming process. Besides, generally the DM has some intuition about what trade-offs might be sensible, and this knowledge should be used during the search. For example, speeding up production by 1 day would be valued between $ 500 and $800 by the DM. A solution speeding up production another day at the expense of several million dollars might be pareto optimal, but totally unacceptable from the perspective of the DM.

Recently, two approaches have been published aimed at focusing the search towards interesting regions as defined by the DM. Deb [3] uses an analogy from goal programming and allows the DM to define a goal (a single desired combination of characteristics) towards which the search is directed. Although this seems very appealing to a DM, when the pareto optimal front is not know, the effect of such a goal on the search

[1] a solution is called pareto optimal if, when compared to any other solution, it is superior in at least one criterion.

process is difficult to assess. For example, if a goal is set that lies within the feasible region, it hinders search to even better solutions, if it lies too far away from the feasible region, it has basically no effect at all.

Cvetkovic and Parmee [1] allow the DM to articulate fuzzy preferences, like "criterion 1 is much more important than criterion 2". Again, the exact effect on the obtained solutions is difficult to assess.

With the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) presented in this paper, we allow the DM to specify maximal and minimal acceptable weightings for one criterion over the other, and use these to guide the EA towards pareto-optimal solutions within these boundaries.

As we will show, this not only speeds up convergence, but also leads to a better coverage of the interesting part of the pareto optimal front. Also, the DM will know exactly what kind of solutions he/she can expect, the approach is very clear.

The paper's outline is as follows: First, we will explain our algorithm in detail in Section 2. Then, in Section 3, the approach is tested on a number of common benchmark problems and compared to other approaches. The paper concludes with a summary.

2 Guided Multi Objective Evolutionary Algorithm

The basic idea of our G-MOEA is that although the DM is unable to define an exact utility function resp. weighting between the different criteria, he/she has some idea about reasonable trade offs. This knowledge should be used in order to guide the search towards the interesting area.

While in this article we are only considering linear combinations (trade-offs) of two objectives $(f_1(x), f_2(x))$, in general it should be possible to extend our approach to non-linear combinations and more objectives.

The DM's intuition is specified in terms of maximal and minimal linear weighting functions $w_{min}(f_1(x), f_2(x))$ and $w_{max}(f_1(x), f_2(x))$. They represent statements of the form "one unit of a is worth at most x units of b" (and vice versa), and correspond directly to slopes of the utility function, or the tangent to the pareto-optimal curve.

The following two tasks are crucial to our EA:

1. It has to ensure that the search population converges to the true pareto-optimal front in the, for the decision maker, interesting part of the search space.

2. Diversity has to be maintained along the interesting part of the pareto-optimal front to provide a maximal set of diverse alternatives.

G-MOEA operates on a modified concept of nondomination levels, a new fitness sharing function and binary tournament selection. The modified ranking procedure by a nondomination analysis of the population is the means to guide the search towards the area that is interesting for the DM.

The core of our algorithm is the guided nondomination ranking which proceeds as follows: For both, the maximum and the minimum weighting function, the optimal solution in the current population is determined (so-called corner individuals, cf. Figure 1) and given rank 1. Then all nondominated individuals are evaluated w.r.t both linear weighting functions. Of these individuals all that have a better fitness than either of the two border individuals according to at least one of the two linear weights are assigned the same rank (this corresponds to all individuals inside the parallelogram as shown in Figure 1) . All thus ranked individuals are removed from the population, and the process starts anew with the remaining individuals, this time the next highest rank is awarded. This procedure is repeated iteratively until each individual has been assigned a rank.

This modified nondomination analysis can be performed efficiently by a transformation of the objective values and a succeeding "standard" nondomination analysis. Instead of a pareto analysis of the individuals based on their objective values $(f_1(x), f_2(x))$ they are ranked by an examination of their minimum and maximum weighting function values $(w_{min}(f_1(x); f_2(x)), w_{max}(f_1(x); f_2(x)))$. This transformation allows to compute the guided ranking as described above as efficiently as the "standard" nondominated ranking.

Note that with $w_{min}(x) = f_1(x)$ and $w_{max}(x) = f_2(x)$ the case of no a priori preference articulation by the DM can be handled. The pareto analysis then corresponds to the normal nondomination ranking without guidance.

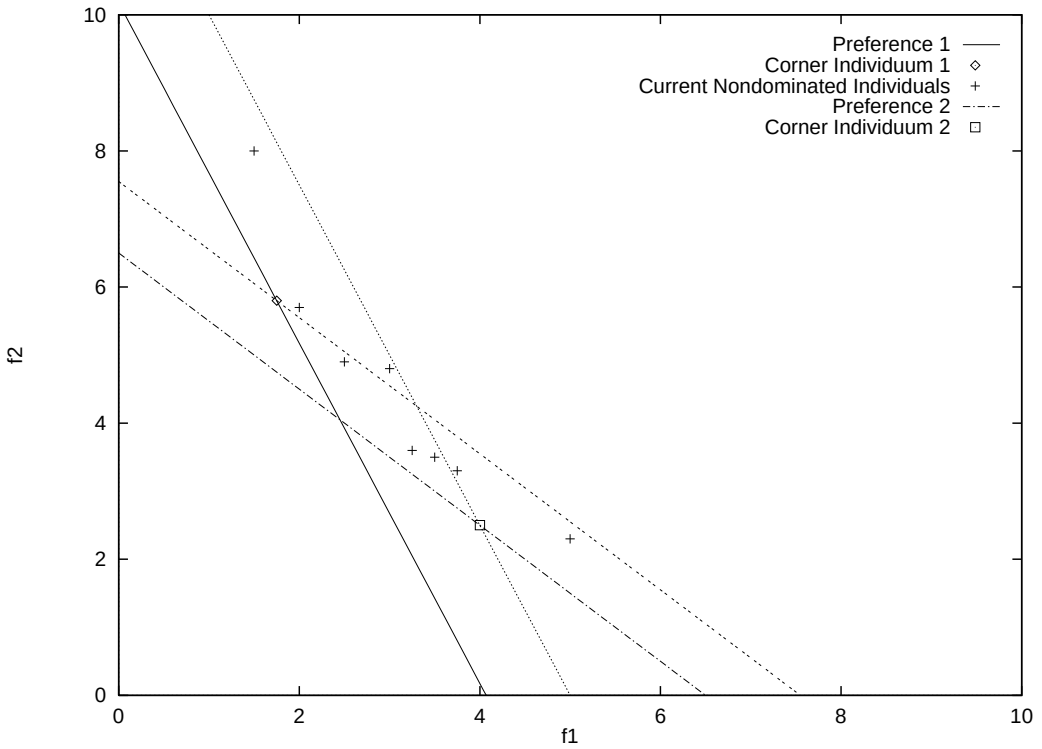


Figure 1: Only the nondominated solutions within the parallelogram defined by the corner individuals and the weightings are assigned the same rank. The plot assumes a minimization problem.

The guided nondomination analysis guides the population towards the true pareto-optimal front in the, for the decision maker, interesting part of the search space. Diversity along the pareto optimal front is supported by fitness sharing which takes into account all individuals and works on the objective space. Like in the standard sharing procedure for each individual i the niche count m_i is calculated [6]. The shared fitness f_i' for each individual is then calculated as $f_i' = f_i + (m_i - \min_i(m_i)/(\max_i(m_i) - \min_i(m_i))$. That way, no individual can get a worse shared fitness than any individual with an inferior rank. It is ensured that the current pareto front converges quickly to the true pareto front while the individuals are nevertheless well distributed. The population based approach as opposed to pareto level based sharing was chosen since the goal is not to maintain diversity in each pareto level but in the overall population. It is, for example, not desired that the second level converges towards the true pareto front in the same area as does the current pareto front, and that more interesting regions not covered by the current front are neglected.

For the experiments reported below, standard settings were generational replacement, tournament selection with tournament size 2, and the use of Simulated Binary Crossover as introduced by Deb and Beyer [4]. Further problem dependent settings are mentioned in the next section.

3 Experimental Results

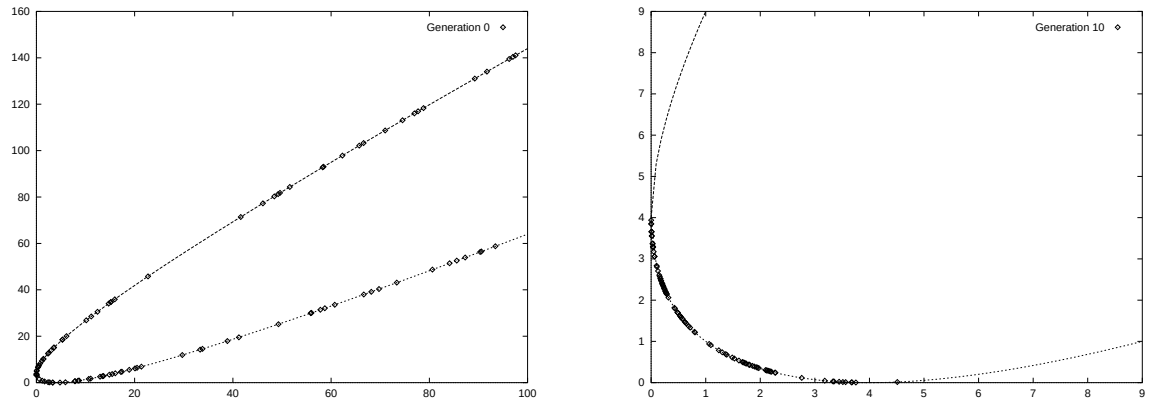
In this section the G-MOEA is applied to four test problems that were already examined or introduced in related work.

Test Problem 1

First we examine the algorithms convergence properties using Schaffer's f2 Function. This is a very simple test function introduced by Schaffer in his dissertation [5]. Nevertheless nearly every multi-objective EA is applied to it to test whether the population becomes and remains well distributed along the pareto optimal front. In the tests reported here, no guiding was used (i.e. no transformation).

Parameter settings used were population size of 100, $\sigma_{share} = 0.1$, crossover probability of 1.0 and no mutation. The mutation probability is set to zero to observe the effectiveness of G-MOEA alone.

As the results of a typical run show, the population quickly converges to the pareto front and distributes the individuals fairly evenly over the pareto-optimal front. The G-MOEA is moreover able to maintain the diversity on the long run (not shown here). These results prove that our implemented fitness sharing as described above works fine.

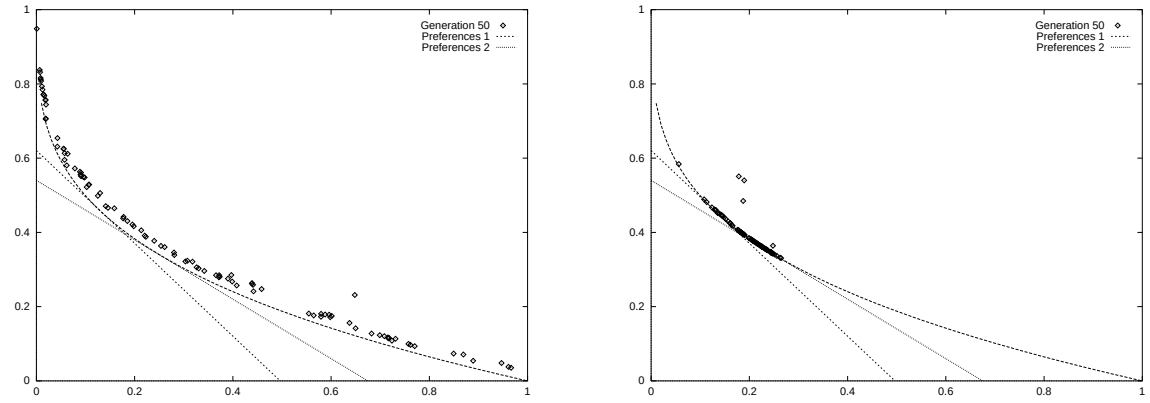


(a) (b)

Figure 2: Without any preferences, the G-MOEA converges quickly and evenly to the pareto optimal front. (a) shows the initial population, (b) the situation after 10 generations.

Test Problem 2

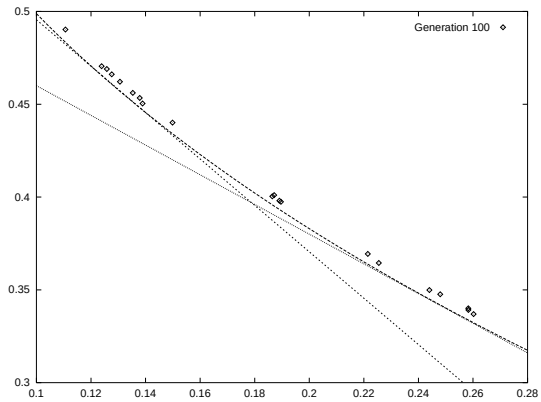
The effect of guidance is tested on Deb's convex test problem [2] with $\alpha = 0.3$ (function specific parameter). The EA used a population size of 100, $\sigma_{share} = 0.5$, crossover probability of 0.8 and mutation probability of 0.01. As minimum and maximum weighting, $\frac{f_1}{f_2} = 0.8$ resp. $\frac{f_1}{f_2} = 1.25$ are used.



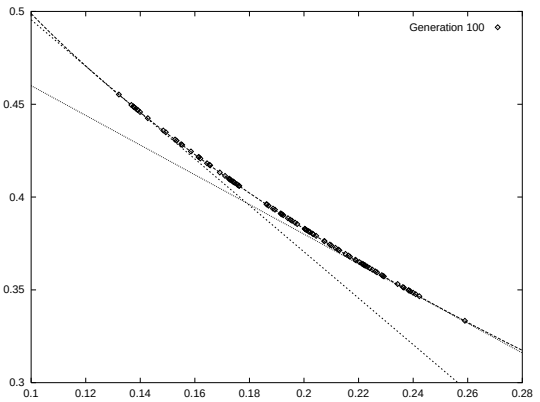
(a) (b)

Figure 3: After 50 generations, G-MOEA (b) clearly outperforms the unguided version (a) in terms of convergence and coverage of the interesting area.

As can be seen in Figure 3, G-MOEA is not only capable of guiding its search towards the desired region of the pareto optimal front (area between the two corner individuals), but moreover converges much faster



(a)



(b)

Figure 4: When the intersting region is enlarged, the better performance of G-MOEA (b) over the unguided version (a) is even more obvious. The plots show the population in generation 100.

to the true pareto optimal front.

In Figure 4, the interesting region is enlarged. One can clearly see that this region of the pareto optimal front is much better explored by G-MOEA than by the unguided variant, in terms of convergence as well as in terms of diversity or numbers of individuals.

Some additional tests show the impact of the weighting preferences on the G-MOEA. The plot in Figure 5 has been produced using the minimum and maximum weighting $\frac{f_1}{f_2} = 1.0$ resp. $\frac{f_1}{f_2} = 3.0$, while in Figure 6, the weightings $\frac{f_1}{f_2} = 0.5$ resp. $\frac{f_1}{f_2} = 0$ were used.

As can be seen, G-MOEA adapts its focus accordingly.

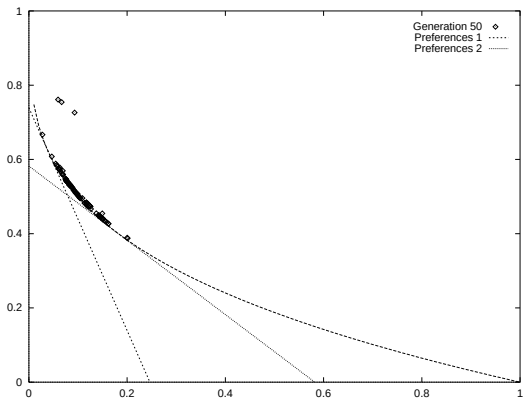


Figure 5: By stressing criterion 2 over criterion 1, the region of convergence is shifted accordingly.

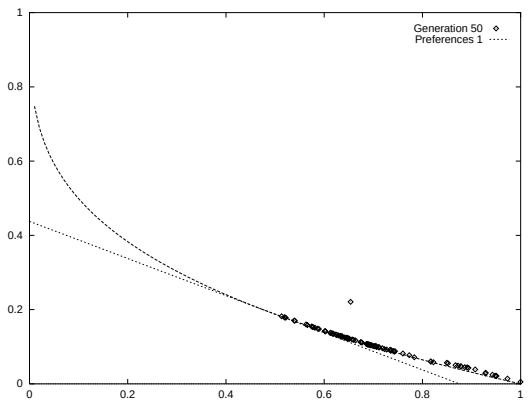


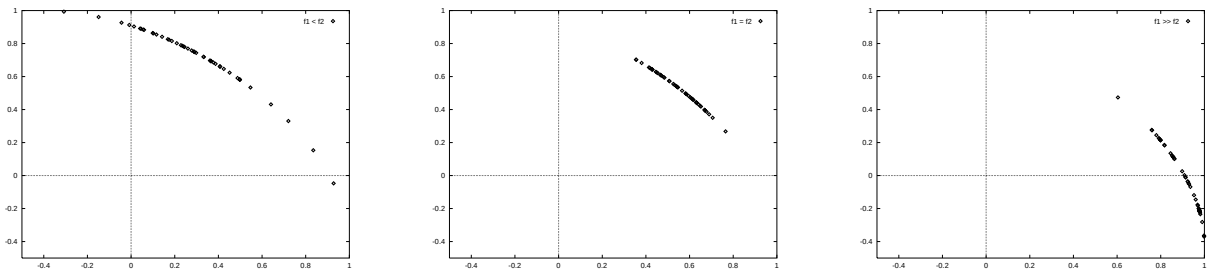
Figure 6: Strongly stressing criterion 1 over criterion 2 shifts the region of convergence accordingly.

Test Problem 3

Cvetkovic and Parmee [1] introduced the following simple test function to test their preference based MOEA.

In the following tests, G-MOEA was run with population size of 50, crossover probability of 0.8, mutation probability of 0.02, and $\sigma_{share} = 0.5$ for 15 generations. Weightings used were $\frac{f_1}{f_2} = (0.4, 1.0); (0.8, 1.25); (1.5, \infty)$.

Figure 7 again shows how the focus of G-MOEA can be adapted by setting proper preferences. The results match those reported in Cvetkovic and Parmee [1].



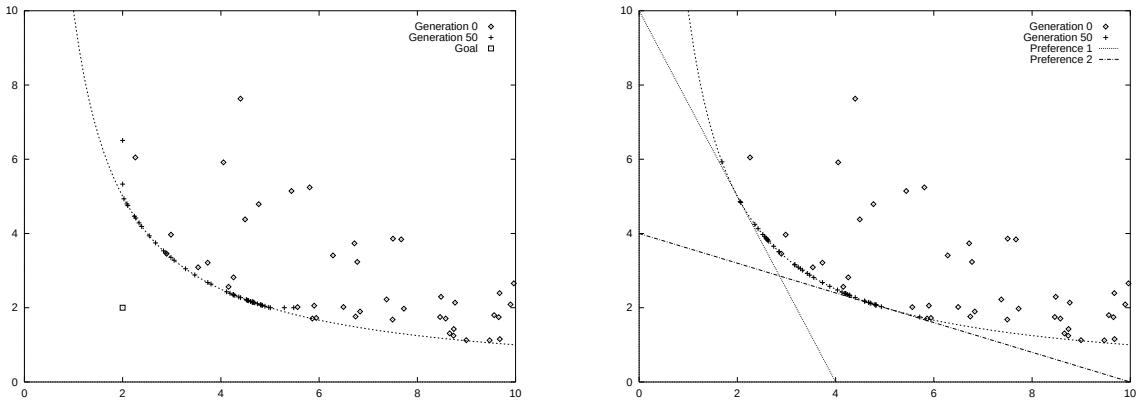
(a) (b) (c)

Figure 7: Population after 15 generations with focus on criterion 1 (a), equal importance of both criteria (b), and strong focus on criterion 2 (c).

Test Problem 4

Finally, we compared our approach to the goal programming inspired MOEA presented by Deb [3] on his test problem P1.

Parameter settings were population size of 50, crossover probability of 0.8, mutation probability of 0.02, and $\sigma_{share} = 0.5$ run for 50 generations. Weighting used was $\frac{f_1}{f_2} = (0.4, 2.5)$. As goal for the approach suggested in [3], the point (2.0;2.0) was used.



(a) (b)

Figure 8: G-MOEA (b) can lead to equivalent results when compared to the goal programming approach (a).

The results from Figure 8 show that the G-MOEA is able to provide solutions that are matching the ones obtained with goal programming.

The two approaches differ mainly in the way the preferences are expressed. The suitability of one approach over the other may depend on the application. But generally, the goal programming approach is sensitive to a good choice of the goal, while in our approach, the DM can explicitly define the interesting range of trade-offs without any knowledge of the pareto optimal front. In some problems it therefore may be easier/more convenient for the DM to give initial preferences than to define a goal.

4 Conclusion

We have presented a simple, intuitive and powerful way to include a decision maker's vague preferences of one criterion over the other into the optimization process of Multi-Objective Evolutionary Algorithms.

The proposed Guided Multi-Objective Evolutionary Algorithm (G-MOEA) provides the user with the possibility to guide the search process towards a specified region of the pareto optimal front. With guidance, the algorithm converges much faster towards the optimal front, which may save significant computational efforts. Moreover, it covers the interesting region much better than the unguided version.

Compared to recently presented approaches with similar intentions, G-MOEA is able to match their reported performance. At the same time, G-MOEA seems to be more flexible and more intuitive to use than the other approaches, as it allows to explicitly set the minimum and maximum acceptable trade-offs.

References

[1] D. Cvetkovic and I. C. Parmee. Use of preferences for ga-based multi-objective optimization. In W. Banzhaf et al., editor, *Genetic and Evolutionary Computation Conference*, volume 2, pages 1504–1509. Morgan Kaufmann, 1999.

[2] K. Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. In *Evoluationary Algorithms in Engineering and Computer Science*, 1999.

[3] K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Congress on Evolutionary Computation*, volume 1, pages 77–84. IEEE, 1999.

[4] K. Deb and H.-G. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. Technical Report CI-61/99, Department of Computer Science/XI, University of Dortmund, Germany, 1999.

[5] J. D. Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms.* PhD thesis, Vanderbilt University, 1984.

[6] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[7] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.