Multiobjective Evolutionary Path Planning via Fuzzy Tournament Selection

Gerry Dozier, Shaun McCullough, Abdollah Homaifar, Eddie Tunstel, Loretta Moore

Abstract— **This paper introduces a new selection algorithm that can be used for evolutionary path planning systems. This new selection algorithm combines fuzzy inference along with tournament selection to select candidate paths (CPs) to be parents based on: (1) the Euclidean distance from origin to destination, (2) the sum of the changes in the slope of a path, and (3) the average change in the slope of a path.**

In this paper, we provide a detailed description of the fuzzy inference system used in the new fuzzy tournament selection algorithm (FTSA) as well as some examples of its usefulness. We use 12 instances of our FTSA to rank a population of CPs using the above criteria. Based on its path ranking capability, we show how the FTSA can obviate the need for the development of an explicit multiobjective evaluation function. Finally, we use the FTSA to enhance the performance of an existing evolutionary path planning system called GEPOA.

I. Introduction

RECENTLY , there has been a growing number of successful applications of Evolutionary Path Planners [1, 2, 3, 6, 8, 10, 11, 13, 14]. However, many of these systems are primarily concerned with finding the shortest path between a starting point and a destination for a robot to traverse.

In path planning, the shortest path may not always be the most efficient means of getting from start to destination. There are many other attributes of a path that may be desirable in addition to distance. One example would be the smoothness of a path. The development of an effective multiobjective closed-form fitness equation for evolutionary path planning may be difficult. Therefore, our concentration has been placed on the development of a multiobjective selection method based on fuzzy inference [16]. By concentrating on selection, one only need be concerned with how individuals of a population compare relative to one another.

In this paper, we introduce a new tournament selection method, called fuzzy tournament selection, that can be used with evolutionary path planning systems. The fuzzy tournament selection algorithm (FTSA) described in this paper selects candidate paths (CPs) to be parents and undergo reproduction based on: (1) the Euclidean distance of a path from the origin to its destination, (2) the sum of the changes in the slope of a path, and (3) the average change in the slope of a path. The remainder of this paper is organized as follows. In Section II we provide a brief overview of multiobjective optimization. In Section III we briefly describe some related work that inspired the development of our FTSA. In Section IV, we discuss our new FTSA in detail by 'walking through' an example fuzzy tournament between two CPs. In Section V, we present our experiments and results, and in Section VI, we present our conclusions. In Section VII, we discuss our ongoing work.

Gerry Dozier is with the Department of Computer Science & Engineering, Auburn University, 109 Dunstan Hall, Auburn University, Auburn, AL 36849 gvdozier@eng.auburn.edu

Shaun McCullough is with the NASA Center for Autonomous Control Engineering, North Carolina A&T State University, Greensboro, NC 27411, shaunm@ncat.edu

Abdollah Homaifar is with the NASA Center for Autonomous Control Engineering, North Carolina A&T State University, Greensboro, NC 27411, homaifar@ncat.edu

Jet Propulsion Laboratory, Robotic Vehicles Group, Pasadena, CA 91109, tunstel@robotics.jpl.nasa.gov

Loretta Moore is with the Department of Computer Science & Engineering, Auburn University, 115 Dunstan Hall, Auburn University, Auburn, AL 36849 moore@eng.auburn.edu

II. Multiobjective Optimization

The multiobjective optimization problem (MOP) can be stated as follows. Given a set of objective functions $f = \{f_1, f_2, \ldots, f_m\}$, find a point $x = \{v_1, v_2, \ldots, v_n\}$ such that f is minimized (or maximized). In order to effectively discriminate between two points x_0 and x_1 it is important to impose some type of preference structure on f [15], which defines the relevance of each objective function in f. A candidate solution to the MOP, x_0, is said to dominate another candidate solution, x_1, if x_0 is preferred based on some preference structure $\mathcal{P}$.

In [15], Yu introduces three preference structures for multiobjective optimization: value function preference, Pareto preference, and lexicographic preference. In value function preference, a function g is defined on f such that $g(x_0) < g(x_1)$ if and only if x_0 is preferred to x_1. In the above case, x_0 is said to dominate x_1 and x_1 is said to be dominated by x_0.

Perhaps the most widely used preference structure used in evolutionary multiobjective optimizers is the Pareto preference structure [9, 12]. Using Pareto preference, x_0 is said to dominate x_1 if $\forall_i f_i(x_0) \leq f_i(x_1) \wedge \exists_i f_i(x_0) < f_i(x_1)$ [12]. The Pareto optimal set of a MOP is the set of all nondominated points.

Another type of preference structure is known as lexicographic preference. In this type of preference an order is imposed on f and a point x_0 is said to dominate another point, x_1 if $f_k(x_0) < f_k(x_1)$ and $f_i(x_0) = f_i(x_1)$ *for* $i = 1, \ldots, k-1$. Our FTSA can be viewed as a generalization of lexicographic preference. In fact, it may best be referred to as fuzzy lexicographic preference (FLP). In FLP, the $<$, $>$, and $=$ operators are fuzzy sets which return a membership grade. This membership grade is a value within the interval [0..1], where 0 means strictly false, 1 means strictly true, and a value in between the two extremes represents some possibility of truth.

III. Related Work

The development of our FTSA was inspired, in part, by [5, 10, 11]. In [10], Xiao, Michalewicz, Zhang and Trojanowski present a value preference structure for optimizing motion plans based on path length, path smoothness and clearance (feasibility). In [5], Fujimura uses Pareto preference for discovering a set of pareto-optimal motion plans based on path length and energy consumption. In [11], Shibata and Fukuda use a “fuzzy” value preference structure for developing motion plans based on time and load objectives (constraints).

IV. Fuzzy Tournament Selection

The overall objective of our FTSA is to allow evolutionary path planners to evolve CPs that have: (1) minimal distances from start to destination, (2) minimal sums of the changes in slope (SCS), and (3) minimal average changes in slope (ACS). Our FTSA takes six inputs: the distances, the SCS , and the ACS of two CPs (CP_1 and CP_2) that are randomly chosen from the current population and returns one output in the continuous interval [-1, 1]. This output corresponds to the CP which should be selected to be a parent. Any output that is less than zero means that CP_1 is to be selected while any output greater than zero means that CP_2 is to be selected.

Implementation of a fuzzy inference system requires assigning membership functions for both inputs and outputs by partitioning the respective universes of discourse using fuzzy subsets. With knowledge of the membership functions in place, the fuzzy system performs three primary operations — fuzzification of input variables, inference via a set of fuzzy rules that map fuzzy inputs to fuzzy outputs, and defuzzification of aggregated fuzzy outputs. In what follows, we describe these three attributes as implemented in our FTSA.

A. Fuzzification

Each CP chosen randomly from the current population has three attributes: (1) the Euclidean distance of the CP from start to destination, (2) the SCS along the CP, and (3) the ACS along the CP. For example, consider two CPs. Let $d_1 = 24.80$, $s_1 = 17.96$, $a_1 = 1.80$, $d_2 = 25.37$, $s_2 = 11.46$, and $a_2 = 1.91$, represent the distance, SCS, and ACS of CP_1 and CP_2. We convert the six inputs into 3 derived parameters, d, s, and a, whose values are in [-1,1] as follows:

$d = \frac{d_1-d_2}{d_1+d_2} = \frac{24.80-25.37}{24.80+25.37} = -0.01$,

$s = \frac{s_1-s_2}{s_1+s_2} = \frac{17.96-11.46}{17.96+11.46} = 0.22$, and

$a = \frac{a_1-a_2}{a_1+a_2} = \frac{1.80-1.91}{1.80+1.91} = -0.03$.

Notice that for values of d, s, and a, which are less than zero, the more desirable attribute belongs to CP_1 and vice-versa for CP_2. Upon closer inspection of d, s, and a one can see that CP_1 has a slightly shorter distance and ACS than CP_2 while CP_2 has a smaller SCS than CP_1.

Each of the derived inputs d, s, a, has a domain partitioned by three fuzzy subsets defined using overlapping membership functions. Figure 1 shows the three membership functions for x. They are described as follows. If the value of x is non-positive then it is a member of set LT with degree $\mu_{LT}(x)$. Members of LT represent the set of all tuples (x_1, x_2) such that $x_1 < x_2$. That is x_1 is less than x_2. Similarly, values of x that are non-negative are members of set GT with degree $\mu_{GT}(x)$ and represent the set of tuples (x_1, x_2) such that $x_1 > x_2$.

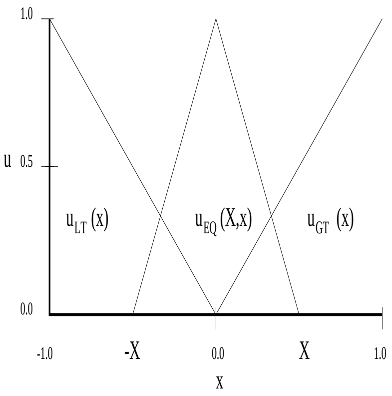

Fig. 1. The Membership Functions for LT, $EQ(X)$, and GT

All values of $|x| < X$ are members of set EQ with degree $\mu_{EQ}(X, x) = 1 - \frac{x}{X}$ if $X > x \geq 0.0$ and $\mu_{EQ}(X, x) = 1 + \frac{x}{X}$ if $-X < x \leq 0.0$. Members of EQ represent the set of all tuples (x_1, x_2) such that $x_1 \approx x_2$. That is based on X, x_1 is approximately the same as x_2. By changing X the FTSA has the ability to adapt its focus on optimizing an objective.

For the continuation of our example, let $D = S = A = 0.15$. Therefore,

$d = -0.01$ has membership degree 0.01 in LT, degree 0.93 in $EQ(D)$, and degree 0.0 in GT.

$s = 0.22$ has membership degree 0.00 in LT, degree 0.00 in $EQ(S)$, and degree 0.22 in LT.

$a = -0.03$ has membership degree 0.03 in LT, degree 0.80 in $EQ(A)$, and degree 0.0 in GT.

B. Fuzzy Rulebase and Inference

The fuzzy rulebase is formulated as shown in Figure 2. Actually our rulebase consists of 3 rules. Rules 1-3 can be viewed as one rule with three ‘ORed’ antecedents. Rule 4 could be viewed as a second rule and Rules 5-7 can be viewed as the third and final rule. For each of the seven rules, P represents the singleton consequent of a rule. If the consequent of a rule is $P = -1$, then the rule has specified that CP_1 should be selected to be a parent. Similarly if a rule’s consequent is $P = 1$ then it has specified that CP_2 should be selected.

The rulebase shown in Figure 2 is an example of a zero-order Sugeno fuzzy rulebase [7]. This is because each consequent of a rule is a zero-degree polynomial of the input variables (of the rule). In an n-order Sugeno fuzzy rulebase the consequent of each rule would be n^{th} degree polynomial of the input variables.

In continuation of our example, we can compute the fire strength of each of the antecedents by using the *multiplication* operator for conjunction and the *max* op-

```
Rule 1:   If d is LT                                 Then P = -1
Rule 2:   If d is EQ(D) and s is LT                  Then P = -1
Rule 3:   If d is EQ(D) and s is EQ(S) and a is LT    Then P = -1
Rule 4:   If d is EQ(D) and s is EQ(S) and a is EQ(A) Then P =  0
Rule 5:   If d is EQ(D) and s is EQ(S) and a is GT    Then P =  1
Rule 6:   If d is EQ(D) and s is GT                  Then P =  1
Rule 7:   If d is GT                                 Then P =  1
```

Fig. 2. The Fuzzy Rulebase

erator for disjunction. Let f_i denote the fire strength of Rule i. The fire strengths are as follows (where $d = -0.01$, $s = 0.22$, $a = -0.03$, and $D = S = A = 0.15$):

$f_1 = \mu_{LT}(-0.01) = 0.01$,
$f_2 = \mu_{EQ}(D, -0.01) * \mu_{LT}(0.22) = 0.93 * 0.00$,
$f_3 = \mu_{EQ}(D, -0.01) * \mu_{EQ}(S, 0.22) * \mu_{LT}(-0.03) = 0.93 * 0.00 * 0.00$,
$f_4 = \mu_{EQ}(D, -0.01) * \mu_{EQ}(S, 0.22) * \mu_{EQ}(A, -0.03) = 0.93 * 0.00 * 0.80$,
$f_5 = \mu_{EQ}(D, -0.01) * \mu_{EQ}(S, 0.22) * \mu_{GT}(-0.03) = 0.93 * 0.00 * 0.00$,
$f_6 = \mu_{EQ}(D, -0.01) * \mu_{GT}(S, 0.22) = 0.93 * 0.22$, and
$f_7 = \mu_{GT}(-0.01) = 0.00$.

Notice that only two rules fire; Rule 1 fires with $f_1 = 0.01$ and Rule 6 fires with $f_6 = 0.20$. The other five rules have fire strengths of 0.0.

At this point, our rules have three consequences. Let the first consequent be $P = -1$, the second consequent be $P = 0$, and the third consequent be $P = 1$. Also let F_k be the 'ORed' fire strengths of all rules associated with the kth consequent. Thus, $F_1 = max(f_1, f_2, f_3) = 0.01$, $F_2 = f_4 = 0.00$, and $F_3 = max(f_5, f_6, f_7) = 0.20$.

C. Defuzzification

The defuzzification technique used is the mean of the maxima [7], [11]. Let P_k represent the consequent for F_k. Thus, $P_1 = -1$, $P_2 = 0.0$ and $P_3 = 1.0$. Our defuzzification function is $O = \frac{\Sigma_i F_i * P_i}{\Sigma_i F_i} = 0.90$, where O represents the output of the our zero-order Sugeno fuzzy inference system. Therefore, in conclusion of our example CP_2 would be selected to be a parent.

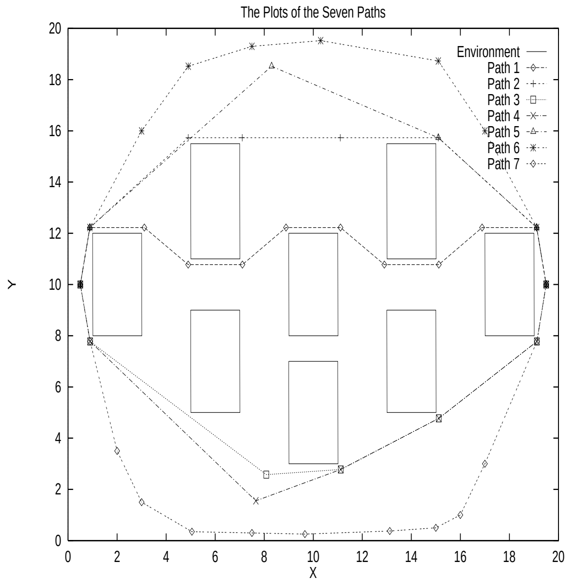

Fig. 3. The Plots of the Seven Paths

Path	*Distance*	*SCS*	*ACS*
Path 1	24.79	17.96	1.79
Path 2	25.36	11.45	1.90
Path 3	25.90	11.46	2.29
Path 4	26.84	11.46	2.29
Path 5	26.92	11.45	2.86
Path 6	30.03	11.45	1.43
Path 7	32.00	11.46	1.04

Fig. 4. The Attributes of the Seven Paths

V. Experiments and Results

A. Experiment I

In order to test our FTSA, we have chosen seven paths shown in Figure 3. The paths all have the same start (located at (0.5, 10.0)) and destination (located at (19.5,10.0)). These paths could have been created by any of the aforementioned evolutionary planning systems. The attributes for each of the seven paths are shown in Figure 4. Notice that the paths are sorted based on their distances with path1 being the shortest and path7 being the longest. Having the paths arranged in this manner will help one to see how the FTSA goes about selection based on distance, SCS, and ACS.

In this experiment, we allowed each of the seven paths to compete in a match with every other path (a total of 21 matches in all). The FTSA was used to select the better of the two paths. When a path won a match, a counter corresponding to the winning path was incremented. We performed the above procedure twelve times; one for each of the values of D taken from the set $\{0.0, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. We kept S and A constant at 0.15.

B. Experiment I Results

The results of our first experiment are shown in Figure 5. Notice that when $D = 0.0$ the ranking of the paths is based purely on distance. This corresponds to how a conventional evolutionary path planner using tournament selection would assign wins. As we increase the range of similarity for the path lengths, D, the longer but smoother paths are increasingly preferred over the shorter paths. This is

Path	$D = 0.0$	0.025	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Path 1	6	4	2	2	2	1	1	0	0	0	0	0
Path 2	5	6	6	6	6	6	5	5	4	4	4	4
Path 3	4	5	5	5	4	3	3	3	3	3	3	3
Path 4	3	3	4	3	2	2	2	2	2	2	2	2
Path 5	2	2	3	2	1	1	1	1	1	1	1	1
Path 6	1	1	0	2	3	4	4	4	5	5	5	5
Path 7	0	0	1	1	3	4	5	6	6	6	6	6

Fig. 5. Path Rankings vs. D, with $S = 0.15$ and $A = 0.15$

indicated by the fact that the longer, smoother paths receive an increasing number of wins as D is increased to 0.35.

Notice that when $D = 0.25$ Path 2 (the second shortest path) and Path 7 (the longest path) have the same number of wins. This is interesting because they both lost one tournament to another path.

A surprising and unexpected result can be seen upon closer observation of Figure 5! Notice that number of wins assigned to Path 1 and Path 7 decreases/increases monotonically as D is increased. However, this is not the case for Paths 2-6. To better see this phenomenon, observe Figure 6.

In Figure 6, the number of wins versus the corresponding value of D is plotted for each of the seven paths. These results are both shocking and encouraging! It seems that even though our FTSA is simple and is composed of three rules it still exhibits some complex behavior. This behavior we see as a result of changing the values of only one of three parameters.

In Section 3, one may have thought that Rule 4 was extraneous. Actually this is not the case at all. Rather than having Rule 4's consequent tied to 0.0 one could instead use the consequent to signal for the values of D, S, and A to be modified. In other words, Rule 4 can be used to indicate the convergence of the population. If Rule 4 fires regularly then this is a sign the population is converging. By modifying the values of D, S, and A, the FTSA causes an evolutionary path planner to place greater emphasis on other attributes. How D, S, and A are effectively modified is a topic for future research.

C. Experiment II

In Experiment II, we used our FTSA in a hybrid evolutionary/visibility-based motion planning and obstacle avoidance system called GEPOA [3]. GEPOA uses steady-state reproduction, flat crossover [4] with gaussian mutation, and uniform mutation in an effort to develop feasible, minimal distance paths. In each generation two parents are selected using tournament selection with a tournament size of 2. If the first parent selected represents an infeasible path, it is repaired 50% percent of the time. If the first parent selected is feasible then the two parents create one offspring which replaces the worst individual in the

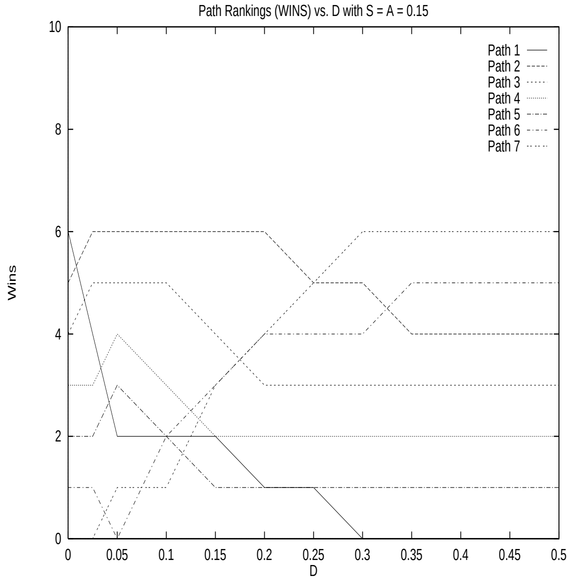

Fig. 6. The Plots of Wins vs. D

population. For FTS to be effectively used in this type of system, it must be able to adequately rank individuals of a population as was shown previously. For the remainder of this paper, let GEPOA+FTS denote GEPOA with fuzzy tournament selection.

Using the environment shown in Figure 3, we compared the paths developed by GEPOA and GEPOA+FTS. The parameters for each of these algorithms were as follows: the population size was 20, the flat crossover with gaussian mutation (standard deviation = 4.0) usage rate was 0.66, and the uniform mutation rate was 0.34. After the initial population was created, both algorithms were allowed to run for 500 generations, thus, creating a total of 520 individuals. For GEPOA+FTS, we set D = 0.15, S = 0.15, and A = 0.15.

D. Experiment II Results

Figures 7 and 8 show the initial populations that were randomly generated by GEPOA and GEPOA+FTS respectively. Since GEPOA and GEPOA+FTS use a visibility-based algorithm to repair infeasible paths, it is not uncommon to find feasible (but sub-optimal) paths in the initial population.

Figure 9 shows the population of paths developed by GEPOA after 500 steady-state generations. Notice that GEPOA has converged upon the two equal and shortest paths; however, these paths are quite rugged. By contrast, Figure 10 shows the population of paths developed by GEPOA+FTS after 500 steady-state generations. First of all notice that GEPOA+FTS has converged upon a number of good paths. Notice also in Figure 10 that the shortest path is still represented. The fact that it is infeasible is not a major concern because it has a chance of being repaired! Not only does FTS allow evolutionary search to converge upon the best path but it also seems to allow for a great deal of valuable, much needed diversity.

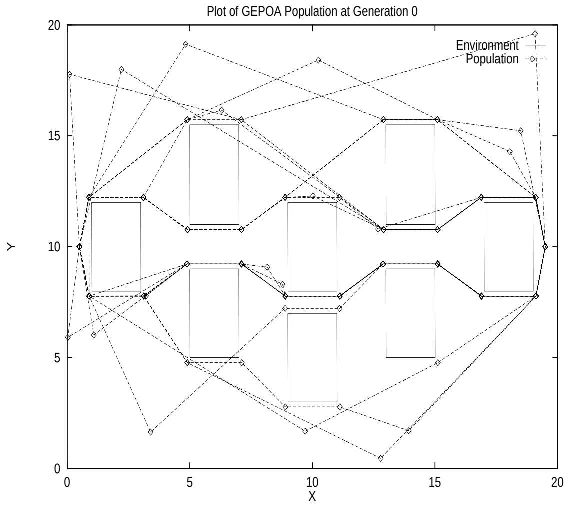

Fig. 7. Population of GEPOA at Generation 0

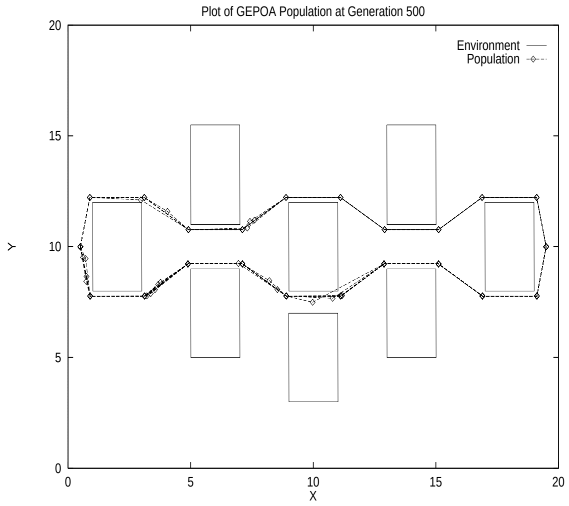

Fig. 9. Population of GEPOA after 500 Generations

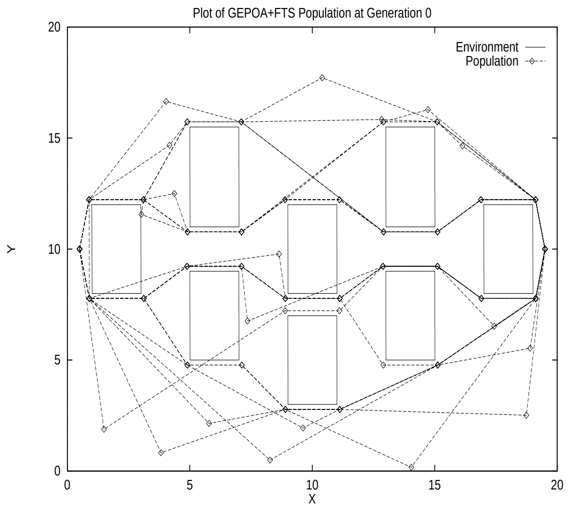

Fig. 8. Population of GEPOA+FTS at Generation 0

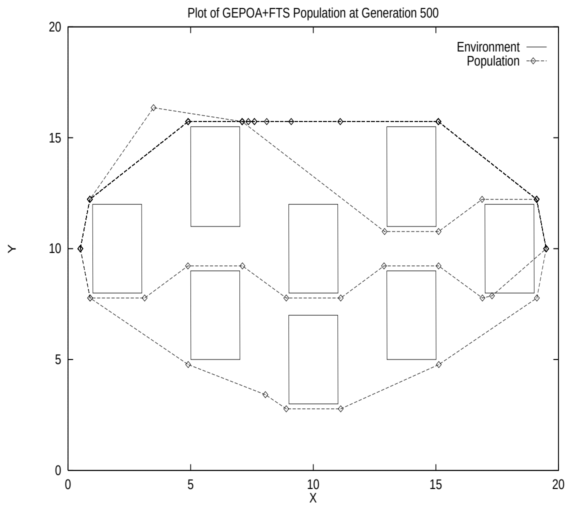

Fig. 10. Population of GEPOA+FTS after 500 Generations

VI. Conclusions

In this paper we presented a FTSA that can be used for multiobjective path planning by almost any evolutionary-based motion planning system. Despite the simple nature of the fuzzy inference system used, the FTSA exhibits complex behavior. Even when the variable parameters of the FTSA are constant, the FTSA seems to be able to allow evolutionary search to converge upon a diversity of optimal and/or near optimal paths. The availability of alternative feasible paths is important in the event that a local navigation system cannot traverse a particular global path. This can happen, for example, when unfavorable conditions are sensed locally, replanning becomes necessary, or task constraints intervene.

VII. Ongoing Work

Our ongoing work will be devoted to developing and experimenting with strategies which will allow GEPOA+FTS to adapt the values of D, S, and A during run-time. This will allow the algorithm greater flexibility as it seeks to strike a balance between selection pressure and diversity.

One can see how the FTSA presented in this paper can be modified to co-evolve feasible and unfeasible paths. Upon closer inspection of the fuzzy rulebase in Figure 2, one can see that distance is the primary attribute, SCS is the secondary attribute, and ACS is the tertiary attribute. The FTSA can be modified in the following way: let the violation distance be the primary attribute, let distance be the secondary attribute and let ACS be the tertiary attribute. Our ongoing work will also be devoted to investigating co-evolutionary FTSAs.

Acknowledgment.

The authors wish to thank the NASA ACE Center (under grant ACE-48146) and NASA Dryden (under grant NAG4-131) for their financial support.

References

[1] Ashiru, I. and Czarnecki, C. (1995). Optimal Motion Planning for Mobile Robots Using Genetic Algorithms, *Proceedings of the 1995 IEEE/IAS International Conference on Industrial Automation and Control.*

[2] Chang, T.-Y., Kuo, S-W., and Hsu, J. Y.-J. (1994). A Two-Phase Navigation System for Mobile Robots in Dynamic Environments, *Proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pp. 297-300.

[3] Dozier, G., Esterline, A., Homaifar, A., and Bikdash, M. (1997). Hybrid Evolutionary Motion Planning via Visibility-Based Repair, *Proceedings of 1997 IEEE International Conference on Evolutionary Computation.*

[4] Eshelman, L. J. and Shaffer, J. D. (1993). Real-Coded Genetic Algorithms and Interval-Schemata, in *Foundations of Genetic Algorithms II*, pp. 187-202, ed. L. Darrell Whitley, Morgan Kaufman Publishers.

[5] Fujimura, K. (1996). Path Planning with Multiple Objectives, *IEEE Robotics & Automation Magazine*, March 1996, pp. 33-37.

[6] Hocaoglu, C. and Sanderson, A. (1996). Planning Multi-Paths Using Speciation in Genetic Algorithms, *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 378-383.

[7] Jang, J.-S. R., Sun, C.-T., and Mizutani, E.. *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Inc., 1997.

[8] Lin H-S, Xiao, J., and Michalewicz, Z. (1994). Evolutionary Algorithm for Path Planning in Mobile Robot Environment, *Proceedings of the 1st IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence.*

[9] Lis, J., and Eiben, A. E. (1997). A Multi-Sexual Genetic Algorithm for Multiobjective Optimization, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 59-64.

[10] Xiao, J., Michalewicz, Z., Zhang, L., Trojanowski, K. (1997). Adaptive Evolutionary Planner/Navigator for Mobile Robots, *IEEE Transactions on Evolutionary Computation*, pp. 18-28, VOL. 1, NO. 1, April 1997.

[11] Shibata, T., and Fukuda, T. (1993). Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic, *Proceedings of the 1993 International Symposium on Intelligent Control.*

[12] Tamaki, H., Kita, H., Kobayashi, S. (1996). Multi-Objective Optimization by Genetic Algorithms: A Review, *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 517-522.

[13] Wu, K. H., Chen, C. H. and Lee, J. D. (1996) Genetic-Based Adaptive Fuzzy Controller for Robot Path Planning, *Proceedings of the 1996 International Conference on Fuzzy Systems*, pp. 1687–1692.

[14] Xiao, J., Michalewicz, Z., and Zhang, L. (1996). Evolutionary Planner/Navigator: Operator Performance and Self-Tuning,*Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 366-371.

[15] Yu, P. L. (1989). Multiple Criteria Decision Making: Five Basic Concepts, *Handbooks in Operations Research and Management Science, Vol 1 (Optimization)*, pp. 663-699, G.L. Nemhauser et al. Eds. Elsevier Science Publishers B.V. (North-Holland).

[16] Zadeh, L. A. (1965). Fuzzy Sets, *Information and Control*, Vol. 12, pp. 338–353.