AIAA 1st Intelligent Systems Technical Conference
20 - 22 September 2004, Chicago, Illinois
AIAA 2004-6230

# PARALLEL EVOLUTIONARY ALGORITHMS FOR UAV PATH PLANNING

Dong Jia
Post-Doctoral Research Associate, Electrical and Computer Engineering
*Carnegie Mellon University, Pittsburgh, PA 15213*

Juris Vagners
Professor Emeritus, Aeronautics and Astronautics
*University of Washington, Seattle, WA 98195*

## Abstract

Evolutionary computation (EC) techniques have been successfully applied to compute near-optimal paths for unmanned aerial vehicles (UAVs). Premature convergence prevents evolutionary-based algorithms from reaching global optimal solutions. This often leads to unsatisfactory routes that are sub-optimal to optimal path planning problems. To overcome this problem, this paper presents a framework of parallel evolutionary algorithms for UAV path planning, in which several populations evolve simultaneously and compete with each other. The parallel evolution technique provides more exploration capability to planners and significantly reduces the probability that planners are trapped in local optimal solutions.

## 1. Introduction

In recent years, *evolutionary computation* (EC) techniques have been successfully applied to real-time task and path planning problems for unmanned aerial vehicle (UAV) systems, including single-vehicle systems [2,5,9] and multi-vehicle systems [2,3,8,12]. The EC-based techniques are attractive for solving large-scale complex optimal planning problems because gradient information about objective functions and constraints is not needed during search for optimal solutions. Gradients usually do not exist for all feasible solutions in the search space. Another major advantage of EC-based techniques is that they can eventually give the global optimal solutions for large-scale complex planning problems c.f. [8]. In practice, it is not always the case when the available computation resources and/or computation time are limited. Once the evolution converges to a local optimal plan, it often takes a long time for planners to escape from that local optimal solution. This is called *premature convergence*. In dynamic environments, premature convergence slows planners' responses to environmental changes because the optimal plans before changes are often locally optimal in new environments.

Parallel evolution is one approach to overcome premature convergence [1,4,6,7]. In parallel evolution, several populations evolve simultaneously. After several generations, each population updates its individuals by individuals in the planner or other sources. This process is repeated until a good solution is generated. In [4], populations with different encoding schemes evolve concurrently. After a number of generations, each population combines its own individuals with the best individuals from all other populations. In [7], populations are encoded in the same way, but evolve independently with different values of evolution parameters. In [1], after several generations, each population merges with its adjacent populations to form a bigger population. The number of populations is reduced by 1. Eventually, there is only one huge population. In [6], each population only maintains individuals with fitness in a certain range and an individual will be moved to another population once its fitness goes outside that range.

In this paper, we present a parallel evolutionary algorithm for real-time UAV path planning applications where several populations with the same encoding scheme evolve simultaneously but independently. When comparable fitness levels are achieved, either in part or for the entire set, the populations are compared based on population fitness. The best population continues to evolve and the others are reinitialized. The re-initialization can be based on a random seed or deterministic input, e.g. from a human operator of the UAVs or from other path planner algorithms such as A* search. By exploring more regions in the search space, this parallel evolutionary algorithm provides planners more chances to reach global or near global optimal solutions.

This paper is organized as follows. Section 2 describes the path planning problem for UAV systems and reviews the evolutionary computation techniques used in this paper. Section 3 presents and analyzes the parallel evolutionary path planning framework. In Section 4, an example demonstrates the effectiveness of the proposed parallel evolutionary framework.

## 2. Path Planning for UAVs and Evolutionary Computation

In this paper, we consider the path planning problem for a UAV that starts from a initial position $x_O$, visits several sites of interest for certain tasks, avoids unsafe regions and goes to a goal position $x_G$. The path needs to satisfy certain physical constraints, e.g., maximal and minimal speeds, maximal acceleration and minimal turn radius. We formulate this path planning problem as the following optimization problem:

$$\begin{aligned}
&\min_{x_{[t_O,t_G]}} J = E\left\{\alpha_F F\left(x_{[t_O,t_G]}\right) + \alpha_D D\left(x_{[t_O,t_G]}\right) - \alpha_R R\left(x_{[t_O,t_G]}\right)\right\} \\
&\qquad v = \dot{x} \\
&\qquad a = \dot{v} \\
&\qquad f(x,v,a) = 0 \\
&\qquad g(x,v,a) \le 0 \\
&\qquad x(t_O) = x_O \qquad\qquad (1) \\
&\qquad v(t_O) = v_O \\
&\qquad x(t_G) = x_G \\
&\qquad v(t_G) = v_G
\end{aligned}$$

In this optimization, the objective function $J$ consists of three parts: the cost of UAV fuel consumption, $F$, the penalty of loss of the UAV, $D$, and the reward for finishing tasks, $R$. $\alpha_F$, $\alpha_D$ and $\alpha_R$ are weights for these three terms respectively. $x_{[t_O,t_G]}$ denotes the values of the UAV position, $x$, between time $t_O$ and $t_G$. $v$ and $a$ are the velocity and acceleration of the UAV, respectively. The optimization contains a set of equality constraints and inequality constraints. The equality constraints define the UAV dynamics and the initial and end constraints of the path. The inequality constraint $g$ includes constraints that reflect the UAV capacities, e.g. maximal and minimal speed, maximal acceleration and minimal turn radius.

The optimization minimizes the expected value of the weighted sum of $F$, $D$ and $R$ because their values are random, not deterministic. The cost of UAV fuel consumption $F$ is not deterministic because the fuel consumption depends not only on geometric characteristics

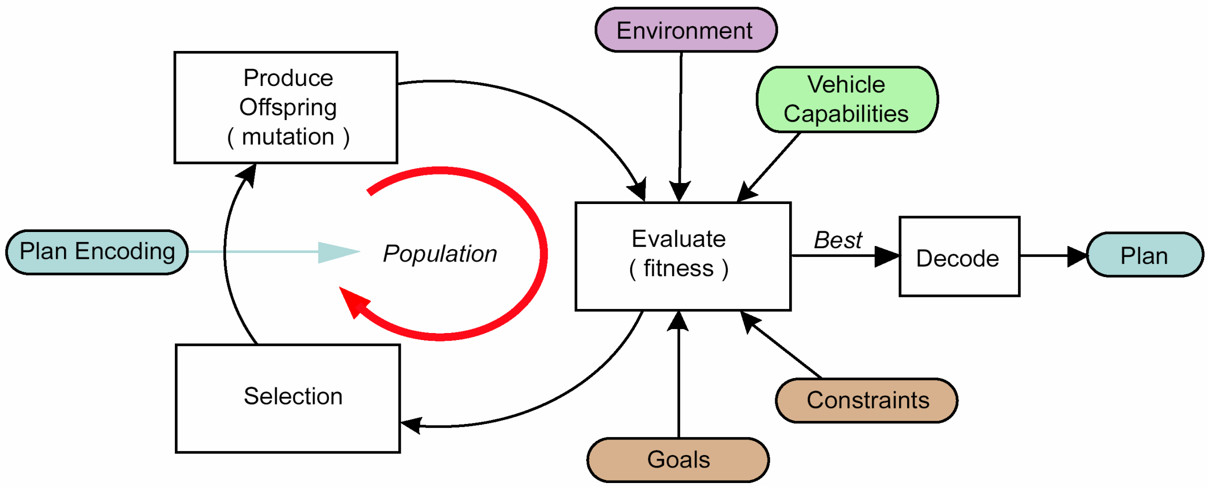


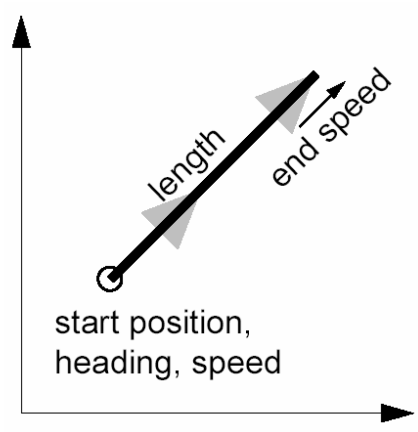

**Figure 1. Evolution process for path planning**

of a path but also on the weather condition along the path, which is not deterministic. For many applications c.f. [10,11] one must use a probabilistic weather model of the region wherein the UAVs operate. Once the UAV appears in an unsafe region, e.g. icing areas or areas with defense weapons, it could be destroyed with certain probability depending on the duration in that region. The value of penalty $D$ along the path is random since the loss of the UAV is random. Similarly, when the UAV passes through a site of interest, the corresponding task is finished with some probability depending on capacities of onboard payloads and the duration over that site. So, the reward $R$ is random. For details about computation of the expected value of the sum of $F$, $D$ and $R$ and computation of penalty $D$, see [8].

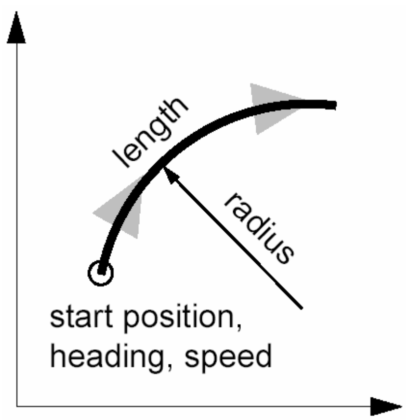
Solving the optimization problem (1) described above exceeds the capability of traditional optimization techniques. The remainder of this section gives a brief review of evolutionary path planning techniques we used in this paper. For details, see [8].

Figure 1 illustrates the evolutionary computation used in this paper. It starts from creating the first generation of the population, usually by encoding a set of randomly selected feasible solutions. During each iteration loop, the planner first evaluates fitness of each individual based on information about environment, vehicle capacities, goals and other constraints. Then, a set of individuals are selected as parents for the next generation according to their fitness. The last step is to generate offspring individuals by cloning a single parent with a mutation or combing two parents by crossover. The iteration is repeated until it converges or the number of iterations reaches a pre-set value. The individual with the best fitness in the last generation is decoded as the optimal plan.

A path for the UAV is encoded as a sequence of segments from the start point $x_O$ to the goal position $x_G$. Each element is either a straight line or an arc, illustrated in Figure 2. The acceleration is constant along a straight line and the speed is constant along an arc. If two segments are joined, the heading and the speed at the starting point of the second segment are enforced to be the same as those at the ending point of the first segment.

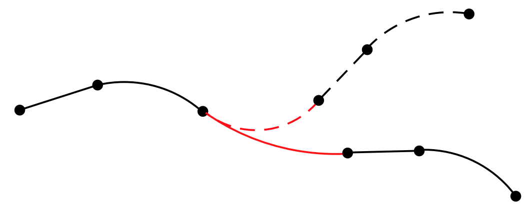



**(a) A straight line**

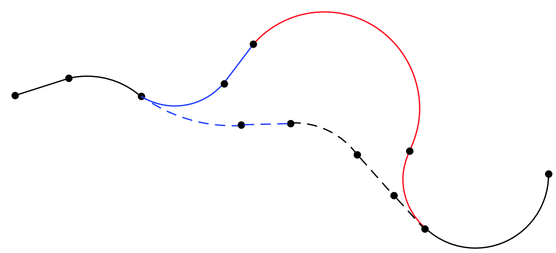



**(b) An arc**

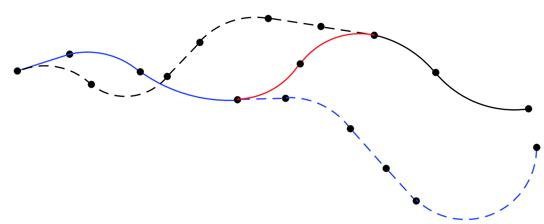
**Figure 2. Path segment types for constructing a path**



**(a) 1-point mutation**



**(b) 2-point mutation**



**(c) Crossover**

**Figure 3. Mutation mechanisms for evolution**

Three mutation mechanisms shown in Figure 3 are used: one-point mutation, two-point mutation and crossover. By one-point mutation, a randomly selected segment is changed randomly and all segments are re-propagated with same length and/or radius as the corresponding previous segments. By two-point mutation, two points are randomly selected and a set of new segments are created to connect them. By crossover, the initial segments of the new path come from one parent and end segments are selected from another parent. A set of segments are created to connect segments from two parents. If needed, a set of segments will be set up to connect the end point of the new path to the goal position. If a new path does not satisfy capacity constraints, it will be dropped. For details about connecting two segments, see [8].

## 3. Parallel Evolutionary Path Planning Algorithm

The evolutionary computation scheme described in the above section also has the premature convergence problem as other evolutionary computations. To solve this problem, we designed a parallel evolutionary path planning framework illustrated in Figure 4.

This parallel evolution process of a group of populations has a similar structure as the evolution process of a population. The process starts from initialization of all populations. The first step in iteration is to evolve all populations. The initialization and evolution of a single

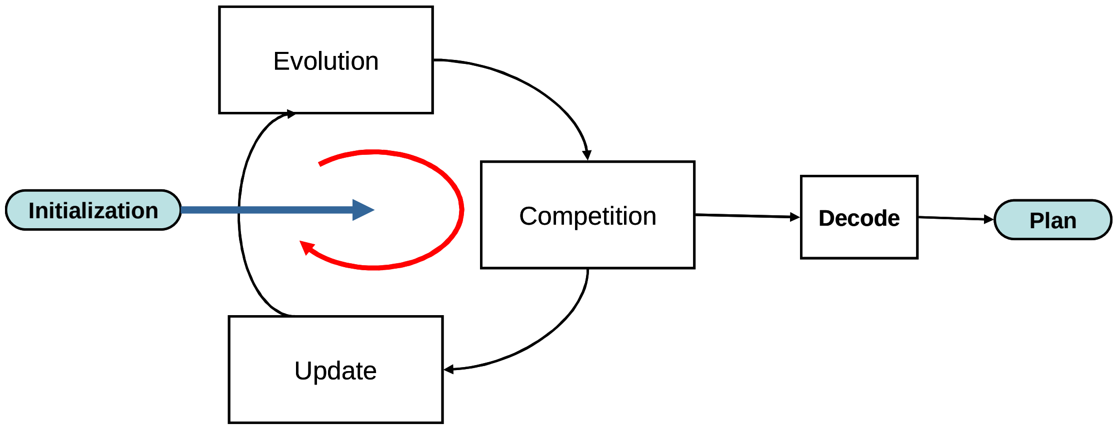



**Figure 4. Parallel evolution of multiple populations**

population use the techniques presented in Section 2. When all populations are well evolved, they compete with each other based on population fitness. The winner continues evolving and all other population are re-initialized. After several iterations, the planner decodes the best individual in the best population to output the optimal path. Within this framework, we could have different parallel evolutionary algorithms by setting different parameters, such as different encoding schemes for different populations [4], different evolution parameter values for different populations [7], different update schemes for different populations [1,6]. In this paper, we use the same encoding schemes and set the same evolution values for all populations. We look at different update schemes.

In our scheme, the populations in a planner are divided into three different types: one principal population, one or several randomized populations and some specialized populations. The principal population and randomized populations are initialized randomly, while the specialized populations are initialized based on results from other search algorithms, e.g. A* and D*, or from human operator inputs. After competition, the principal population is updated based on individuals in all populations. Similar to the initialization process, the randomized populations are re-initialized randomly and the specialized populations are re-initialized by the results from the corresponding search algorithms used in initialization. To compare different populations, the population fitness is set to be the fitness of the best individual. Although the average fitness reflects the evolution of the whole population, the best individual shows the potential optimal results it can achieve since the best individual will be decoded as the optimal path. To update the principal population, we consider the following schemes:

(1) Replacing the individuals in the principal population by the individuals in the best population.

(2) Taking a selected number of the best individuals from each population and constructing the principal population from these individuals.

(3) Putting individuals from all populations together and selecting the best individuals to construct the new principal population.

Among these three strategies, the first one is the easiest to implement, but it ignores potential improvements available from individuals in other populations. The second strategy takes into account all populations by combining individuals from them. If the number of populations is large and all populations are quite different, it becomes similar to randomly re-initializing the principal population. The third strategy combines the advantages of the first two strategies. If the best population beats other populations significantly, the third strategy then is the same as the first one. If all populations are of similar fitness, the third strategy works like the second strategy. If several populations with similar fitness are much better than others, the third strategy combines these populations by taking individuals from them.

Evolutionary computation is a random search strategy. The evolution process can be modeled as a Markov Chain. Given the encoding method, evolution parameters and search space, there is one corresponding Markov model. According to the theory for Markov models, if, from any point in the search space, there exists a mutation or a sequence of mutations to drive the search to the global optimal solution, the computation can eventually converge to the global optimal solution. After a certain number of iterations, the probability of reaching local optimal solutions, including the global optimal solution is significantly larger than that for other non-optimal solutions, but it takes many more iterations, usually more than allowed by the available computation time constraints, to have the probability of reaching the global optimal solution significantly larger than that of reaching other local optimal solutions. This leads to premature convergence.

To analyze the effectiveness of the parallel evolution strategy, we consider a simple case where the planner has one principal population and $N$ randomized populations. We assume that the planner compares all populations after $M$ generations or when improvements of each population fitness are smaller than a small number $\varepsilon$. If there are $K$ local optimal solutions and one global optimal solution, from the theory of Markov models, given a distribution of initial individuals in a population, this population gives the global optimal solution with probability $p_0$ and the $k^{th}$ local optimal solution with probability $p_k$, $k = 1,2,\cdots,K$. Suppose the local optimal solutions are ordered in the way such that the $i^{th}$ local optimum is better than the $j^{th}$ local optimum for $1 \le i < j \le K$. After updating, the best individual in the principal population is the global optimum with probability $1-(1-p_0)^{N+1}$ and the $k^{th}$ local optimum with probability $\left(1-\sum_{i=1}^{k-1} p_i\right)^{N+1}\left[1-\left(1-p_k \Big/ \sum_{j=k}^{K} p_j\right)^{N+1}\right]$. After $Q$ competitions, the planner outputs the global optimal solution with probability $1-(1-p_0)^{QN+1}$ and the $k^{th}$ local optimal solution with probability $\left(1-\sum_{i=1}^{k-1} p_i\right)^{QN+1}\left[1-\left(1-p_k \Big/ \sum_{j=k}^{K} p_j\right)^{QN+1}\right]$. To increase the probability of reaching the global optimum, we can increase either the number of populations or the number of competitions. To guarantee that the planner outputs the global optimal solution with probability no less than $1-\delta$, $N$ and $Q$ need to satisfy the following condition:

$$QN \ge \frac{\ln \delta}{\ln(1-p_0)} - 1 . \qquad (2)$$

# 4. Experimental Studies

Interpretation of complex UAV path planning simulation results is often very difficult, particularly since EC algorithms often give non-intuitive results. In this section, we present a simple example for which the local and global optimal solutions are easily visualized. As shown in Figure 4, the UAV (green triangle) goes to destroy two targets (black circles) and to come back safely to the goal position (small black point). The red circles illustrate the unsafe areas. The green curve gives the global optimal path for the UAV. In the following discussion, the term *global optimal path* refers to paths along which the UAV attacks the closest target first and then the other target and the UAV only flies in the middle region. For this path planning problem, local optimal solutions can be categorized into four types, shown as Figure 5. We note that type I local plans are similar to the global optimal plan except for the attack order.

We performed 30 experiments for the planner with a single population and for the planner with one principal population and one randomized population. In these experiments, each population contains 40 individuals and 20 of them are parents for new generations. In the experiments for the planner with a single population, the populations evolve 8000 generations before it gives optimal solutions. In the experiments for the planner with two populations, each population evolves 2000 generations before comparison. The evolver gives solutions after 2 comparisons. This section presents the experiment results by the first strategy for updating the principal population. After competition, the individuals in the principal population are replaced by the individuals in the randomized population if it is better than the principal population. The results by the other two strategies are similar to this, because there is only one randomized population and no specialized population. For the second strategy, even if we pick the best 20 individuals from both populations to construct the principal population, only the best 20 individuals, usually from the best population, are selected as parents. By the third strategy, the best 40 individuals from the two populations are used to build the new principal population. Typically, most of them come from the best individual.

Figures 6 and 7 summarize these experiments. The value of the fitness function is within the ranges [40, 42], [41, 43], [43, 45], [45, 47] and [47, 49] for global optimal plans and type I, II, III, and IV local optimal plans, respectively. The regions of the fitness function for type I local optimal plans overlap with that for global optimal plans because they are similar to each other. Comparing Figure 6(a) and Figure 6(b), we see that the parallel evolver with two populations over beats the one with a single population with respect to occurrences of local optima. Figure 7 gives the distribution of solutions in different types of optima. From Figure 7(a), 16.67% of solutions are global optimal and the type I, II, III and IV local minimal solutions are 50%, 13.33%, 13.33% and 6.67%, respectively. According to the analysis in Section 3, by the parallel evolutionary plan, the expected occurrence of the global optimum is 42.13% and the expected occurrences of the type I, II, III and IV local minima are 54.17%, 2.9%, 0.77% and 0.03%, respectively. The experimental result shown in Figure 7 (b) is consistent with this theoretic result. The experimental occurrence of the global optimum is 46.67% and the experimental occurrences of the type I, II, III and IV local optima are 50%, 3.33%, 0 and 0, respectively. The experiment results are close to the theoretical analysis in Section 3.

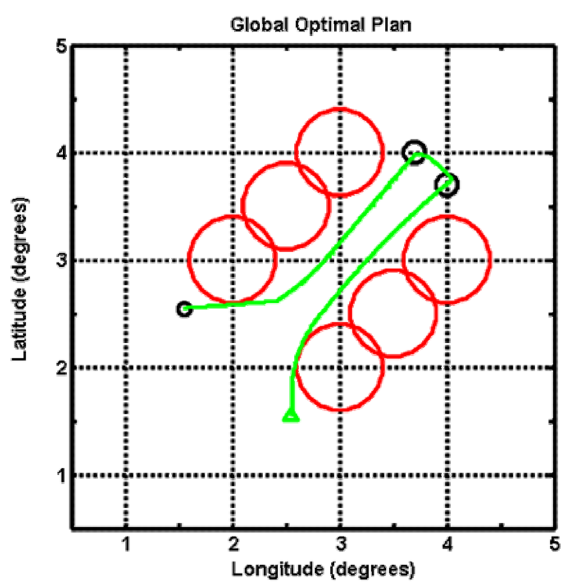
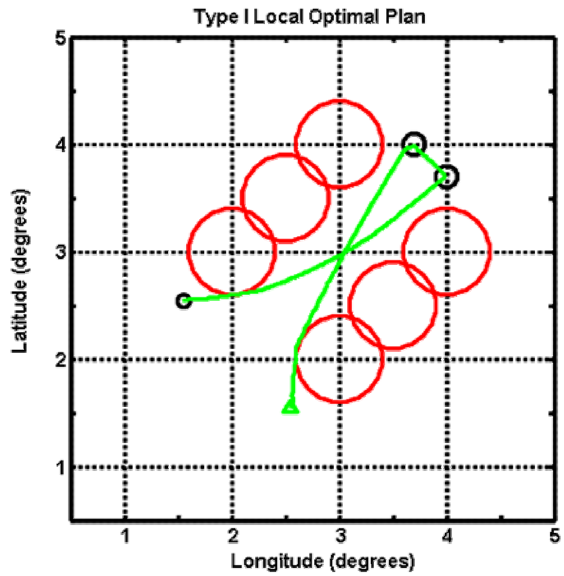
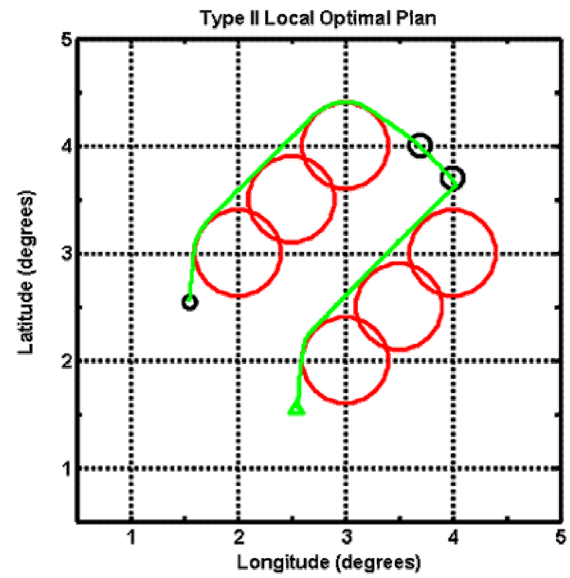


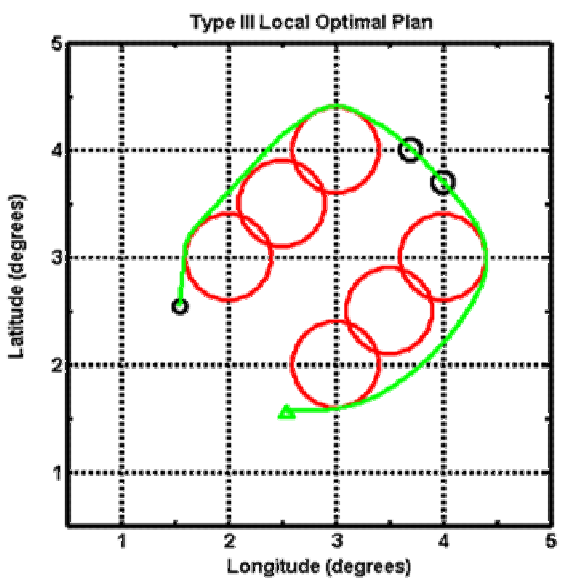

**Figure 4. Example planning problem**



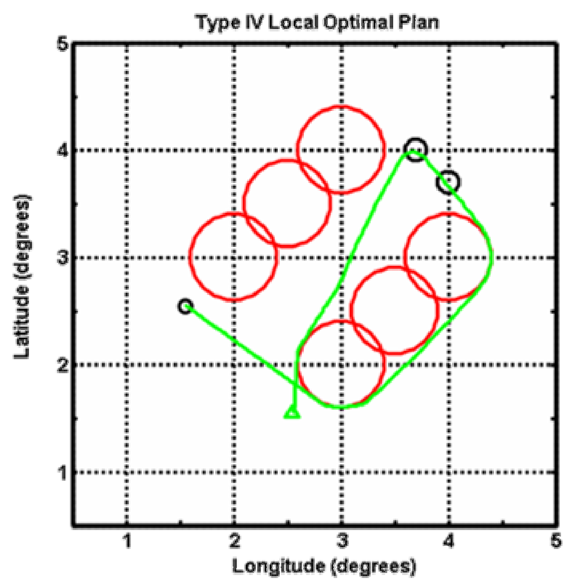

**(a) Type I local optimal solution**




**(b) Type II local optimal solution**

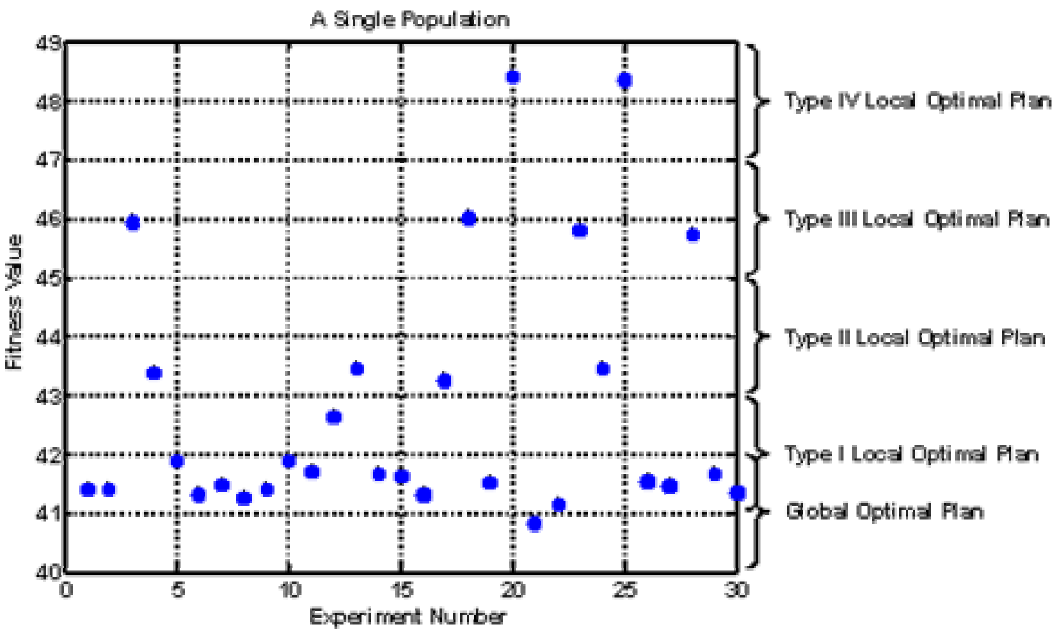



**(c) Type III local optimal solution**

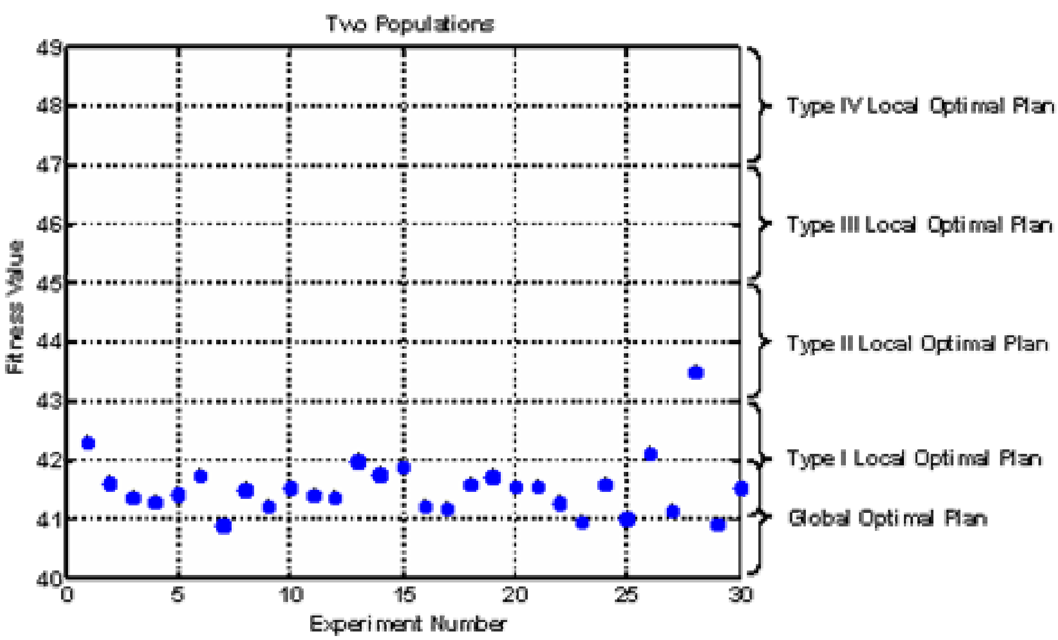



**(d) Type IV local optimal solution**

**Figure 5. Local solutions to the example planning problem.**

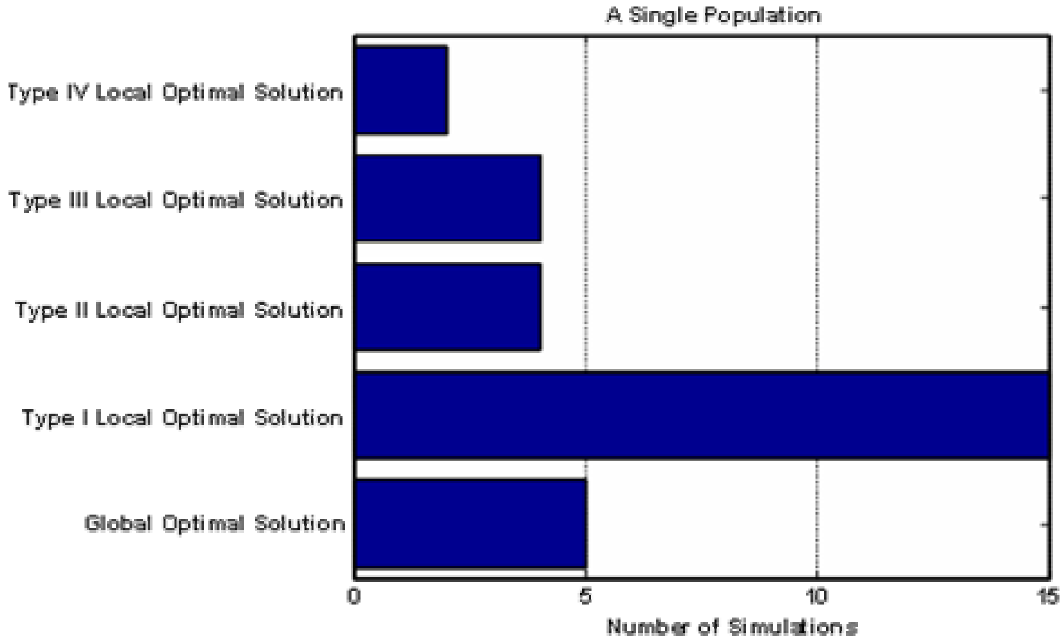



**(a) Fitness values of solutions by the planner with a single population**

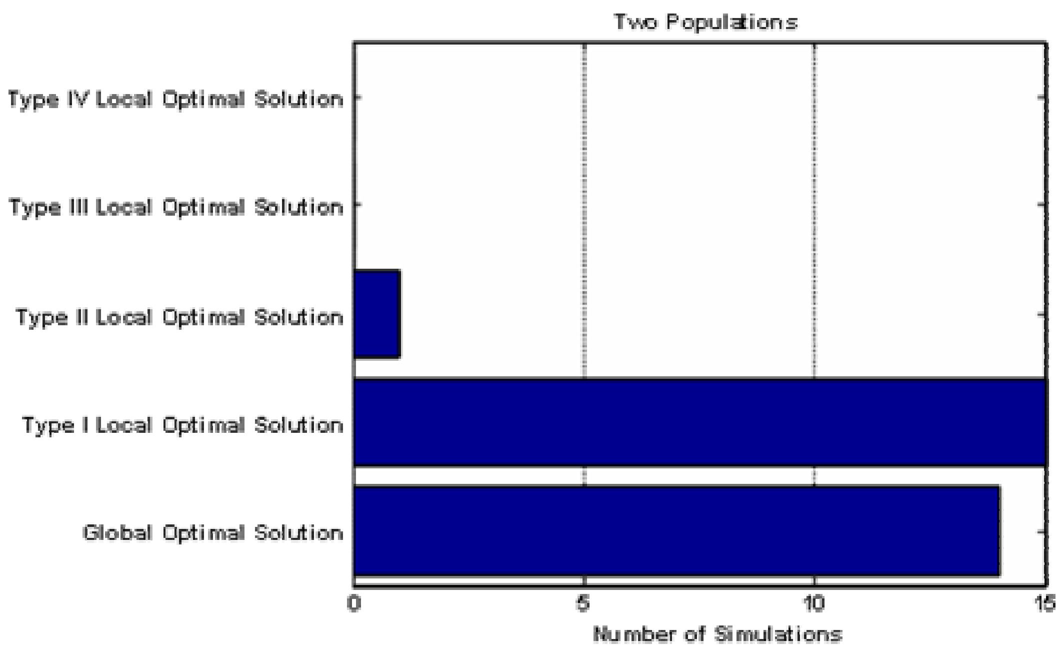



**(b) Fitness values of solutions by the planner with two populations**

**Figure 6. Fitness values of solutions.**




**(a) Distribution of solutions by the planner with a single population.**




**(b) Distribution of solutions by the planner with two populations**

**Figure 7. Distribution of solutions.**

# 5. Conclusion

This paper presents a parallel evolutionary computation scheme for UAV path planning to overcome premature convergence. Analysis shows that the parallel scheme significantly increases the probability of the planner to output the global optimal path with limited computation resources. A small example, one vehicle, two targets and six unsafe regions with four local optimal solutions, demonstrates the performance of the parallel scheme. Given the performance requirement, we give the conditions for design parameters: number of populations and number of competitions (Equation 2). If memory is limited, we need to increase the number of competition, i.e., computation time. For online dynamic optimization, to achieve quick computation, we need to put more memory on board.

For the example presented in Section 4, all three strategies for updating the principal populations perform similarly because the number of populations is very small. For large scale problems, two populations might not be enough. For the planner with more than two populations, the three strategies can work differently. In the future, we will apply the parallel evolutionary planning scheme to larger problems and investigate the performance of different strategies.

This paper studies the parallel evolutionary path planning in a centralized fashion, i.e. all computations in a single computer. In distributed path planning, all vehicles have onboard computation payload and each vehicle plans path for itself. In a coordinated system, the vehicles will exchange their plans. We will investigate how to utilize information from other vehicles in their local parallel evolutionary planners.

## Acknowledgement

This work was supported in part by the DARPA Mixed Initiative Control of Automa-teams Contract N66001-01-C-8074. The authors would like to thank Dr. Anawat Pongpunwattana and Juan Carlos Rubio for helpful discussions on this work.

## References

[1] Affenzeller, M. and Wagner, S.. SASEGASA: An Evolutionary Algorithm for Retarding Premature Convergence by Self-Adaptive Selection Pressure Steering, *Computational Methods in Neural Modeling*, Lecture Notes in Computer Science 2686, pp. 438-445, Springer Verlag, Berlin, 2003.

[2] Capozzi, B.J., *Evolution-based Path Planning and Management for Autonomous Vehicles*, Ph.D. Thesis, University of Washington, 2001.

[3] Fogel, D. and Fogel, L., Optimal routing of multiple autonomous underwater vehicles through evolutionary programming, *Proceedings of the 1990 Symposium on Autonomous Underwater Vehicle Technology*, Washington, DC, 1990, pp. 44-47.

[4] Fonlupt, C., Preux, P. and Robilliard, D., Preventing premature convergence via cooperating genetic algorithms, in *Proceedings Mandel*'97, Brno, Czech Republic, June 1997

[5] Hocaoglu, C. and Sanderson, A.C., Planning multiple paths with evolutionary speciation, *IEEE Transactions on Evolutionary Computation*, Vol. 5, No. 3, June 2001, pp. 169-191.

[6] Hu, J.J. and Goodman, E.D., The hierarchical fair competition model for parallel evolutionary algorithms, *Proceedings of 2002 IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation*, Honolulu, HI. Pp. 49-54, 2002

[7] Miki, M., Hiroyasu, T. and Hatanaka, K., Parallel genetic algorithms with distributed-environment multiple population scheme, *Proceedings of 3rd World Congress of Structural and Multidisciplinary Optimization* (WCSMO), Vol. 1, pp. 186-191, 1999

[8] Pongpunwattana, A.., *Distributed Real-Time Mission Planning for Autonomous Vehicles using Evolutionary Computation*, Ph.D. Thesis, University of Washington, 2004

[9] Rathbun, D. and Capozzi, B.J., Evolutionary approaches to path planning through uncertain environments, *AIAA 1st*

*Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference and Workshop*, Portsmouth, VA, May 2002.

[10] Rubio, J.C., *Long Range Evolution-based Path Planning for UAVs Through Realistic Weather Environments*, M.S. Thesis, University of Washington, 2004

[11] Rubio, J.C., Vagners, J., and Rysdyk, R., Adaptive Path Planning for Autonomous UAV Oceanic Search Missions, *AIAA 3rd "Unmanned Unlimited" Conference*, 20-23 September, 2004, Chicago, IL

[12] Xiao, J. et al., Adaptive evolutionary planner navigator for mobile robots, *IEEE Transactions on Evolutionary Computation*, Vol. 1, April 1997, pp. 18-28.