Fuzzy Logic Clustering of Multiple Traveling Salesman Problem for Self-Crossover Based Genetic Algorithm

Conference Paper · January 2012

DOI: 10.2514/6.2012-487

2 authors:

Nicholas Ernest
Thales USA

Kelly Cohen
University of Cincinnati

50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee

AIAA 2012-0487

Fuzzy Logic Clustering of Multiple Traveling Salesman Problem for Self-Crossover Based Genetic Algorithm

Nicholas Ernest[1] and Dr. Kelly Cohen[2]
School of Aerospace Systems, University of Cincinnati, Cincinnati, OH,45221

The traveling salesman problem (TSP) is a much studied scenario in combinatorial optimization and it serves as a benchmark for many approximate approaches comparing accuracy and computational speed. In this effort, a heuristic method in the form of a fuzzy logic system (FLS) was implanted within the Self-CROssover GEnetic algorithm (SCROOGE) in order to improve performance for the multiple traveling salesman problem (MTSP). This FLS is part of the "UNburdening through CLustering Efficiently" (UNCLE) system. UNCLE SCROOGE takes a MTSP, breaks it down into individual different TSP problems with optimized clusters, and then produces accurate results within a much more reasonable timeframe.

Nomenclature

TSP	=	Traveling Salesman Problem
MTSP	=	Multiple TSP
GA	=	Genetic Algorithm
FCM	=	Fuzzy C-Means clustering
FLS	=	Fuzzy Logic System
MF	=	Membership Function

I. Introduction

THE multiple traveling salesman problem introduces a great deal more complexity to the TSP. However, this type of problem is more applicable to real world scenarios, as many major tasks are completed by more than one operator. However, prior results show that purely adapting the SCROOGE (Self-CROssover Optimized GEnetic) algorithm from the previously published article[1] to the complexities of a 100 city, 5 salesmen MTSP does not perform as well as in the same city layout TSP case. This example city layout was gathered from a sample map on TSPLib, KroA100. While SCROOGE was able to produce a satisfactory solution within 8,424 seconds, these long run-times inhibit the algorithm from being implemented to many scenarios. Despite the lengthy run-times, a seemingly "near-optimal" result was determined from the best of 5, long runs as shown below in Figure 1. For the comparisons of the methods in this study, this route is considered the optimal solution since no prior work could be found for reference.

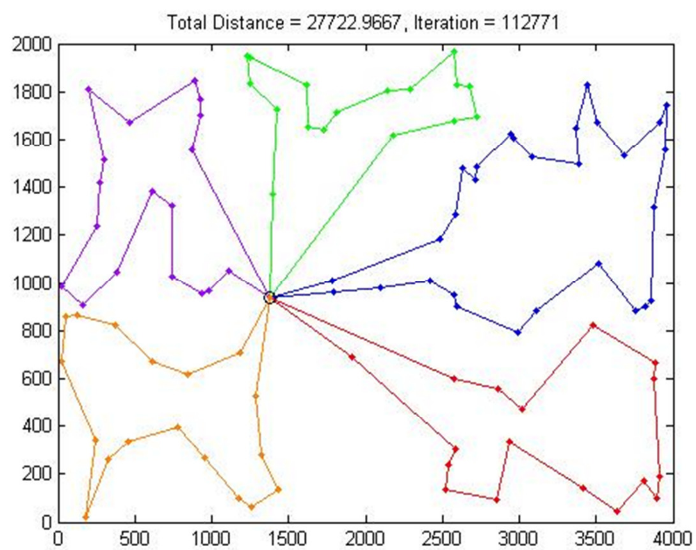

Figure 1: Best result from SCROOGE for 100 city, 5 salesmen MTSP

[1] Graduate Student, School of Aerospace Systems, University of Cincinnati, AIAA Student Member
[2] Associate Professor, School of Aerospace Systems, University of Cincinnati, AIAA Associate Fellow

The traveling salesman problem is a non-deterministic polynomial-time hard mathematical and computational scenario that has a wide array of applications as described by Korte and Vygen[2]. The main objective of the TSP is to determine the shortest route for an agent to visit a set number of targets once, and only once, before returning home. As such, the TSP lends itself to formulate the assignment of a UAV to multiple tasks as depicted by Rasmussen and Shima[3]. However, some additional work is necessary before SCROOGE can be utilized to model multiple UAV's, much less a UAV swarm. The objective of this study was to determine an optimal method at reducing this run-time.

II. UNCLE system

Throughout the optimization of SCROOGE to the MTSP, it was clear that some alternative form of intelligent control is necessary to reach desirable speeds. Fuzzy logic has been shown to be a powerful clustering tool, and the generic goal of the UNCLE system was to break down a MTSP into multiple different TSP's. Increases in speed can be accomplished very easily through this method; however accuracy was a concern.

The UNCLE system was created by the means of a four step iterative design process. First, FCM clustering was utilized after the coordinates of each city were converted to polar. The Θ values (with origin at the starting city) are the only coordinate of interest here, due to the returning nature of the TSP. FCM produces the 5 Θ values that are the centers of each of the clusters. These clusters were left untouched and then ran through SCROOGE. A significant decrease in run-time was noticed, but a decrease in accuracy also followed.

Next was the inclusion of a FLS in the UNCLE system. The same FCM results for determining the centers are utilized, however here they become the centers, or peaks, of the 5 MF's. These Θ values for each city are the inputs, and the output is a clustering number (1 through 5). The MF's are all triangular, except for those near the -180 and 180 degree bounds, which are trapezoidal due to wrapping Θ around the full 360 degrees. The fixed widths of these MF's were optimized off of a "common sense"guess of Θ ranges based on the size of the cluster. Five rules are utilized in this FLS, where each MF applies to a particular cluster number. It is important to note that this produced an increase in accuracy over the first method. This meant that though FCM utilizes fuzzy logic to determine clustering, the FLS allows more expertise and accuracy to be introduced into the system. However, further improvements in accuracy were desired.

The next version of the UNCLE system utilized the same FCM centers, and the same FLS except for an additional function "MembRange", which optimizes the Θ value widths of the MF's. This is accomplished by varying all possible values for these ranges, and calculating the total city density per Θ. The combination of Θ ranges that produce the maximum combined city density is chosen for optimizing the FLS, which results in MF's as shown in Figure 2.

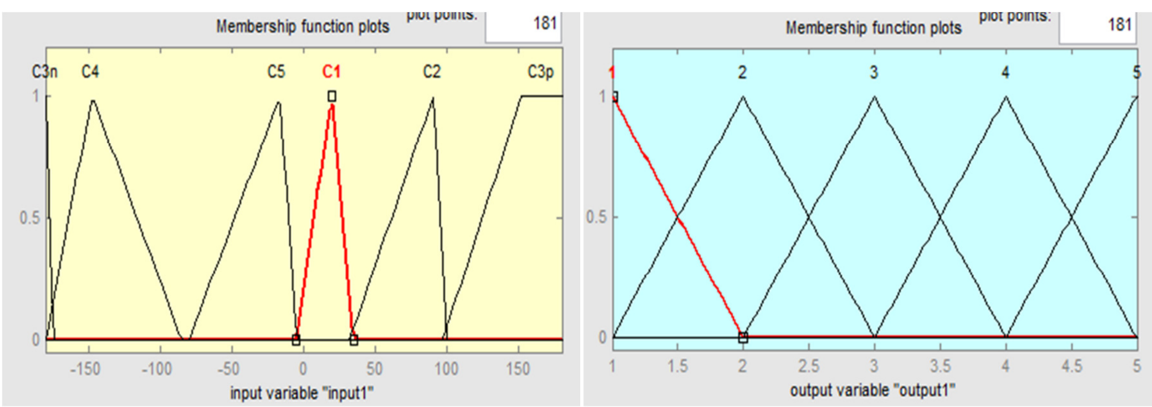

Figure 2: Input and outpud MF's for FLS

While steady improvements were noticed throughout the optimization process, investigating into relieving the duties of the pre-processing scripts that initialize the FLS brought about the final UNCLE system. Since this was found to be where most of the computational cost lay, the investigation lead to analyzing the program with a much simpler pre-processing routine. Relying more on the raw capabilities of the FLS, the "MembRange" function was eliminated entirely, with MF's now just being centered at the results from FCM as before and with widths spread out to a set overshoot past the neighboring cluster's center. This simpler definition was bound to bring about a drastic reduction in run speed.

III. Results

For this study, each method was ran 100 times, and then compared against the previously found optimal solution. As can be seen in Figure 3, the UNCLE system brings significant improvements to SCROOGE, as the previous optimal solution was a distance of 27,722, found in 8,424 seconds. The computer utilized in this study has a 2.2 GHz processor and Matlab version 7.12.

	Total Dist	Time (sec)	Dist % Diff	Time % Diff
Raw FCM	32227.23	23.06	16.25	-36430.79
Fixed MF Width	30221.50	24.82	9.02	-33840.37
Density-based MF Width	27759.51	21.32	0.14	-39412.20
UNCLE SCROOGE	27077.72	3.39	-2.32	-248395.58

UNCLE Stats	
Average	27077.72
Max	27499.73
Min	26919.79
STD	106.51

Figure 3: Numerical Results

Simply utilizing raw FCM produces a fairly acceptable result in terms of accuracy, and a drastic decrease in run-time. However the inclusion of an actual FLS allows SCROOGE to reach a 99% optimal solution very quickly. There was concern that with the additional complexities of the FLS and the "MembRange" functions may bring an increase in accuracy, but would hinder speed. The increasing complexity of the files seen in design iterations 1-3 greatens computational time less than the improved clustering decreases it, netting an overall time improvement as compared to the raw FCM method. The very minor decrease in accuracy compared to the known near-optimal was aptly compensated by the fact that SCROOGE could develop a solution in under 30 seconds. By easing up on the definition of each membership function in the final version of the UNCLE system, run times were drastically reduced again. As another benefit of this method, a more accurate solution than the assumed "near-optimal" was found.

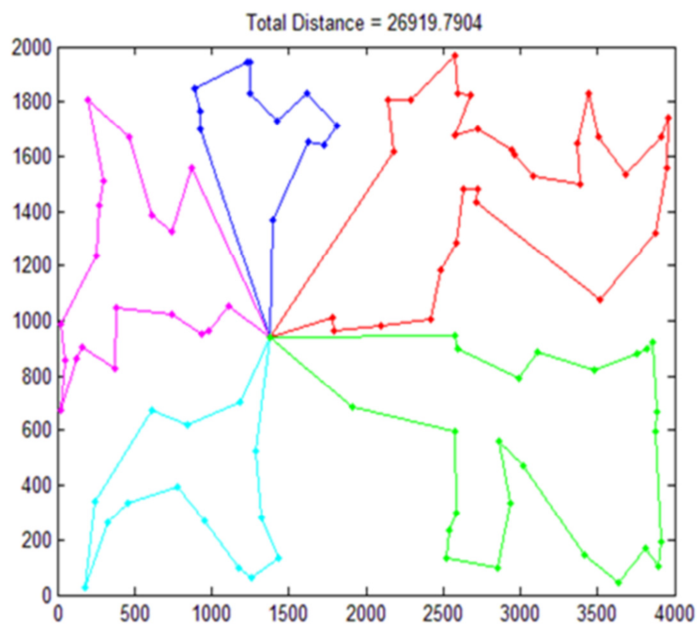

Figure 4: Best solution of UNCLE SCROOGE to 5 salesmen MTSP KroA100

Utilizing heuristics in this manner does bring with it the possibility of differing solutions with each run. UNCLE SCROOGE's statistical data for the course of the 100 runs, as seen in Figure 3, show a worst solution being only marginally less accurate than the previously assumed "near-optimal" solution, and a standard deviation of only 109. While the best solution is approximately one and a half standard deviations away from the mean, the least accurate solution is slightly less than four standard deviations away. This gives a clear picture of the data gathered, showing the robustness of the system.

Better performance in terms of computational cost and accuracy are not the only benefits of UNCLE. It has also liberated the program from a major constraint; the origin city no longer has to be located somewhere inside the exterior of the city map. Additionally, almost completely removing pre-processing needs for the FLS has allowed UNCLE SCROOGE to scale even better with larger city quantities.

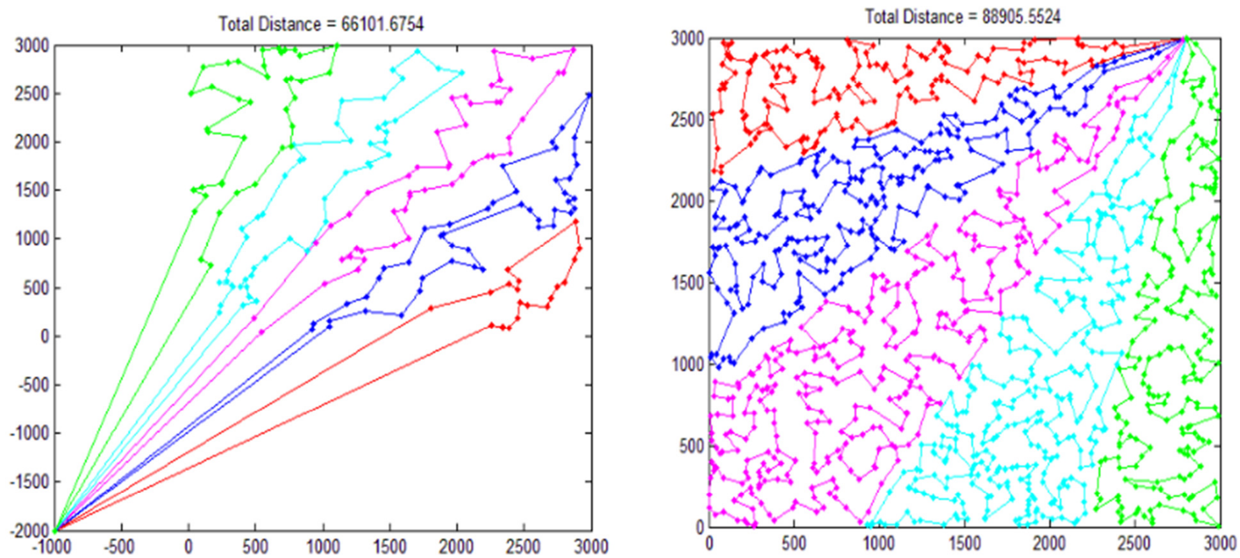

Figure 5: Additional examples showing capabilities of UNCLE SCROOGE

IV. Conclusions and Future Work

Through this study, the UNCLE system was developed and optimized to allow SCROOGE to solve large scale MTSP's efficiently and accurately. These results for this particular case show great promise in this method, as it brings the solution back into a satisfactory run-time realm and slightly boosts accuracy. Future work outside the scope of this paper has began including polygonal target areas rather than point locations, as well as minimum turn radii. This brings the problem scenario closer to something that is applicable to a UAV swarm. The next steps include introducing more complex aircraft dynamics, wind effects (which can be quite drastic for the case of a micro-UAV swarm), and dynamic targets or target areas.

V. References

[1]Ernest, N., and Cohen, K., 2011, *"Self-Crossover Based Genetic Algorithm for Performance Augmentation of the Traveling Salesman Problem",* AIAA, Infotech@Aerospace 2011, St. Louis, Missouri.

[2]Korte, B., and Vygen, J., 2008, *Combinatorial Optimization – Theory and Application*, 4th Edition, Algorithms and Combinatorics, Vol. 21, Springer-Verlag Berlin Heidelberg.

[3]Rasmussen, S.J., and Shima, T., 2008, "Tree Search Algorithm for Assigning Cooperating UAVs to Multiple Tasks", International Journal of Robust and Nonlinear Control, Vol. 18, pp135-153.