6-1-2008

Divide and Conquer Evolutionary TSP Solution for Vehicle Path Planning

Ryan J. Meuth

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Recommended Citation

R. J. Meuth and D. C. Wunsch, "Divide and Conquer Evolutionary TSP Solution for Vehicle Path Planning," *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008*, Institute of Electrical and Electronics Engineers (IEEE), Jun 2008.
The definitive version is available at https://doi.org/10.1109/CEC.2008.4630868

Divide and Conquer Evolutionary TSP Solution for Vehicle Path Planning

Ryan J. Meuth, and Donald C. Wunsch II

Abstract— The problem of robotic area coverage is applicable to many domains, such as search, agriculture, cleaning, and machine tooling. The robotic area coverage task is concerned with moving a vehicle with an effector, or sensor, through the task space such that the sensor passes over every point in the space. For covering complex areas, back and forth paths are inadequate. This paper presents a real-time path planning architecture consisting of layers of a clustering method to divide and conquer the problem combined with a two-layered, global and local optimization method. This architecture is able to optimize the execution of a series of waypoints for a restricted mobility vehicle, a fixed wing airplane.

I. INTRODUCTION

For large continuous tasks, such as area coverage, it is desirable to decompose the problem space to reduce the complexity of the problem. Several methods have been devised to accomplish this, including grid division methods of various geometries, and Voronoi divisions [1]. These methods work well for uniform environments and simple sensing apparatus', but they often do not account for the intersection of the agent's sensor with the environment. Introducing environmental effects and sensor characteristics increases the complexity of area coverage

A vehicle assigned to search area coverage must plan an optimal path through the allocated search space. This optimal path is considered as the shortest route through the set of points that allows the vehicle's sensing apparatus to visit all points in the search space. The path between these points can be optimized using many different algorithms, including graph search, and TSP solution methods. At this level, the vehicle dynamics, vehicle sensing characteristics, and environmental effects can all be included to affect the optimal solution. With the inclusion of vehicle dynamics, the true value of a path is not only dependant on the distance between its points, but also the vehicle capabilities and characteristics of the vehicle's control system.

The traveling salesman problem is focused on finding the shortest tour through a fixed set of points. This has been deeply researched in literature, as efficient solutions have wide ranging applications in domains such as vehicle routing and job-scheduling. As an NP-hard problem, it is extremely difficult to solve large instances of the TSP exactly, so heuristic methods must be used. One of the most powerful heuristic algorithms is the Lin-Kernighan method, which is an iterative method that begins with a random tour, and performs pair-wise optimization by steps to arrive at local minima, and then repeats with a new random tour, always saving the best tour encountered [2]. While not an exact method, it has been shown that the method can find near optimal solutions in a reasonable amount of time. Supplemented by an intersection removal step, as found in the chained LK method, the performance and quality of the LK algorithm is increased. Unfortunately, the time complexity of this algorithm is still unsuitable for real-time operation under problem sets larger than a few thousand points [3].

It has been shown that the time complexity of the LK method can be somewhat linearized at the cost of optimality by first clustering the point set, using the LK method to find a sub-tour, then merging the cluster sub-tours into a final tour [4-6]. The operation of the algorithm becomes very fast, in exchange for a reasonable drop in tour quality.

The LK method has also been supplemented by genetic algorithms that perform a global search, while the LK method performs a local search. By evolving initial tours instead of randomly generating them, the global search becomes more directed, and is able to converge on higher quality tours faster. This topic of research is still young, and much work remains [7, 8].

Including the vehicle dynamics adds computational complexity to any algorithm, so a new TSP solution method, Clustered Evolutionary Lin-Kernighan or CELK, is presented as a method to bring TSP solution into the real-time domain.

Given a region to be searched and an agent with associated characteristics and capabilities, the task of planning a path through the region is extremely complex if all points in the region are to be considered as part of a possible coverage path for an agent. In light of this, many methods have been developed for dividing the search space based on the agent's sensing characteristics to provide a set of points that, if all points in the set are included in the agent's final path,

guarantee that complete coverage is achieved [9, 10]. This set of points can then be used as an instance of the Traveling Salesmen Problem to find a tour through all points, thus covering the search space.

II. The Clustered Evolutionary LK Algorithm

Once a complete coverage point set has been constructed, the problem becomes one of finding the shortest tour through the set of points, which is an instance of the Traveling Salesman Problem, or TSP.

A three-tiered method has been developed, combining clustering, genetic algorithms, and the LK method to find good quality tours with a near linear time complexity. Given a point set, the set is clustered, by way of the Adaptive Resonance Theory method, into a maximum of **M** clusters, where **M** is chosen to be the number of points in the set divided by a fixed scalar, in order to maintain an average cluster size[11]. Sub-tours are then planned within each cluster, using a genetic algorithm-based LK method, called Evolutionary Lin-Kernighan, or ELK. This algorithm is used to recoup some of the tour quality lost through clustering, and analysis has shown that there is also a speed benefit to using the algorithm. Once sub-tours are found, they are merged back into a final tour using the method described in [4].

Utilizing the clustering method to maintain an average cluster size keeps the time complexity of each instance of the ELK method constant, giving the overall algorithm a linear time complexity, while maintaining good tour quality. Also, it allows for many levels of parallelism, as each sub-tour can be computed individually, and each trial tour within the sub-tour optimization can also be optimized independently.

In order to improve the ratio of tour quality to computational expense, an Evolutionary Lin-Kernighan method was developed. This method uses a genetic algorithm to generate trial tours, which are then optimized to local minima using the LK method. This scheme is not a new idea, and the development of useful genetic operators under similar architectures has been explored [12].

The ELK algorithm begins by generating **P** random individuals, where each individual is a trial tour. Each individual is then evaluated, which consists of applying the LK algorithm with intersection removal until a local minimum is achieved, and the total tour length is assigned to the individual's fitness. Roulette-wheel parent selection is then used to generate **P** children, where the probability of individuals being selected as parents is proportional to their rank in the population. Reproduction is achieved through Cost Preserving Crossover and a Mutation Operator. The cost preserving operator has been shown to have good performance at low computational expense [reference for recombination operators].

A fixed number of 'Strangers' or random individuals are introduced to the population at each generation. The new individuals are evaluated, and Roulette-wheel survival is used to keep **P** individuals with an individual's probability of survival being proportional to their rank in the population. This process is repeated until an evaluation limit is exceeded. Mutation is achieved by swapping edges with probability P_{gMut}.

III. Algorithm Results

The tour quality of the ELK algorithm was compared to that of the LK algorithm alone, over 100 randomly generated points, averaged over 30 runs, each run consisting of 1000 evaluations. As can be seen in Fig. 1, the amount of tour improvement per evaluation is significantly greater using the ELK algorithm.

Additionally, trial tours generated by the ELK algorithm were evaluated significantly faster, as LK needed to make fewer changes due to pre-optimized sections of their tours being inherited from the previous generation. To study this effect, the two algorithms were compared by executing them for 1000 path evaluations on randomly generated tours of varying length. An average was taken over 11 runs for each tour length. The number of optimization iterations per path was recorded, as well as execution times for each run. Fig. 2 shows the average execution times of the methods on various tour sizes. Fig. 3 shows the average number of optimization iterations performed on each tour size under the two algorithms. Fig. 4 shows the average optimizations performed on each trial path under the two algorithms. These Fig.s show that the ELK method has a significant quality and performance advantage over the LK method alone. Fig. 1 shows that the ELK method produces higher quality paths in fewer iterations. Fig. 2 shows the LK and ELK algorithms operating on Tours of increasing length, showing that the ELK algorithm has increasing benefits with path size. Fig. 3 shows the average optimization iterations necessary to converge to a local minimum for increasing tour sizes under the LK and ELK algorithms. Note that the LK method increases iterations with tour size, while the ELK method holds relatively constant.

Fig.s 4 and 5 show the number of optimization iterations that were executed on each path evaluated. For the LK algorithm, this is relatively constant, as each path is randomly generated. The ELK algorithm starts high at values corresponding to the LK method, and then reduces sharply, slowly decreasing as a global search is performed.
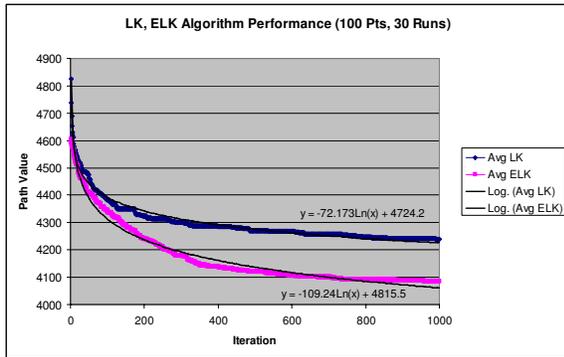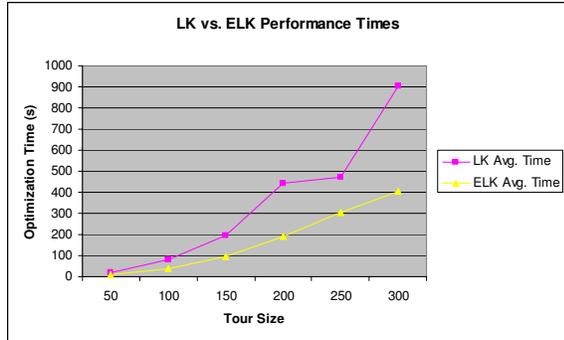
LK, ELK Algorithm Performance (100 Pts, 30 Runs)

$y = -72.173\text{Ln}(x) + 4724.2$

$y = -109.24\text{Ln}(x) + 4815.5$

Fig. 1 - LK vs. ELK Tour Quality.

LK vs. ELK Performance Times

Fig. 2 - Performance Times of LK vs. ELK.
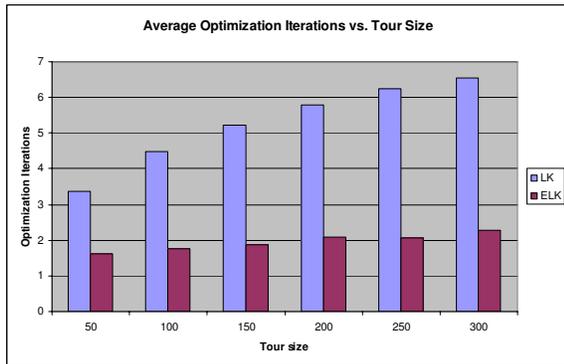
Average Optimization Iterations vs. Tour Size

Fig. 3 - LK vs. ELK Average Optimization Iterations.

LK Optimization Iterations Per Path
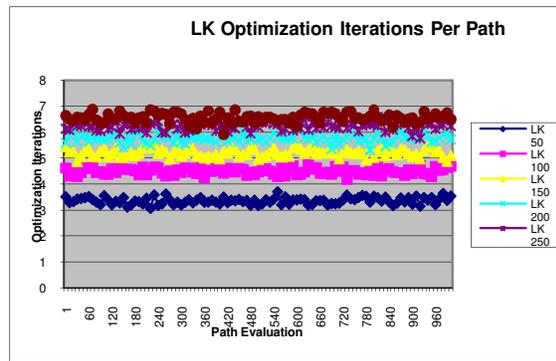
Fig. 4 - LK Optimization Iterations.

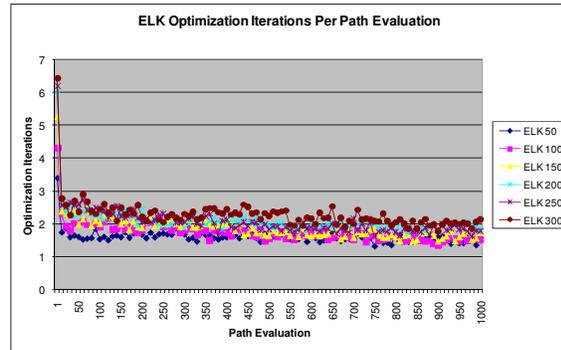**ELK Optimization Iterations Per Path Evaluation**

Fig. 5 - ELK Optimization Iterations.

Clustering the point sets and performing optimization on sub-tours, then merging the sub-tours into a final tour has been shown to linearize the run-time complexity of the ELK method, as shown in Fig. 6.
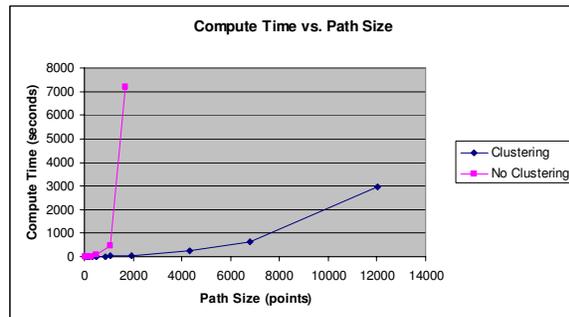
Compute Time vs. Path Size

Fig. 6 - Time Performance of Clustered ELK and ELK on Various Path Sizes.

IV. TSP FOR PHYSICAL VEHICLES

When planning for real-world vehicles, it may be desirable to substitute the Euclidean distance objective function for a cost function that models the mobility of the vehicle. In this way the total behavior of the vehicle can be optimized.

Since the LK method is the underlying algorithm, only metrics should be used as the objective function [2]. As such, we must show that any valid objective function has the properties of symmetry, positivity, reflexivity and triangle inequality.

For heterogeneous vehicles, a simple heuristic metric is proposed, estimating the time to cross the edge, given in equation 1. The guidance speed is a positive scalar representing the average % of the vehicle's maximum velocity that the vehicle travels under guidance. The agility of the vehicle is also a positive scalar, representing a measure of the ability of the vehicles' guidance system to execute turns without incurring extra time, as compared to a straight path. The use of this heuristic allows the optimization of the path based on the vehicle's kinematics.

$$t(e_i) = \frac{d(a_{ei}, b_{ei})}{c_g v_{max}} + c_a \left(\frac{c_g v_{max}}{a} \right) \left(1 - \cos(\theta_{a_{ei}, b_{ei}, c_{ei}}) \right)$$

Equation 1. – Physical vehicle heuristic.

Given a possible edge e_i, consisting of three points, a_{ei}, b_{ei}, and c_{ei}.
Where
- $d(a,b)$ is the Euclidean distance between points a, b.
- v_{max} is the maximum vehicle speed.
- a is the vehicles acceleration capability
- $\theta_{a,b,c}$ is the angle between points a, b and c.
- c_g is a scalar representing the guidance speed.
- c_a is a scalar representing the vehicle agility.

This heuristic has the property that the degenerate case of the heuristic simplifies to Euclidean distance. Also, symmetry, positivity and reflexivity properties are trivial to demonstrate. However, the property of triangle inequality should be shown.

The distance component dominates the physical vehicle heuristic equation and since the angle between edges only adds time, the triangle inequality holds.
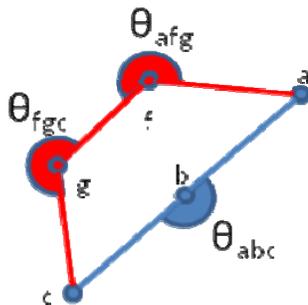

Fig. 7 – The triangle inequality holds for 3-edges.

To adapt the vehicle heuristic to the LK optimization method, the set of possible edges becomes all combinations of three points in the problem set. In this way the algorithm can be directly applied to this instance of the TSP.

V. PHYSICAL VEHICLE HEURISTIC RESULTS

Fig. 8 shows the optimized path on a triangular grid of points using only Euclidean distance. Fig. 9 shows the optimized path on the same grid using the physical vehicle cost heuristic detailed in equation 1. Note the resulting straight path sections, which minimizes the coverage time for those point sequences through the conservation of momentum.
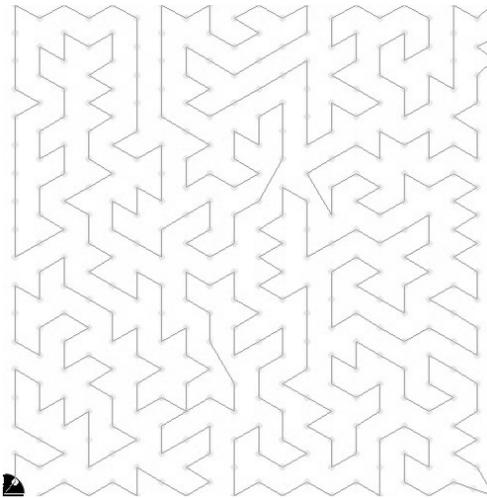

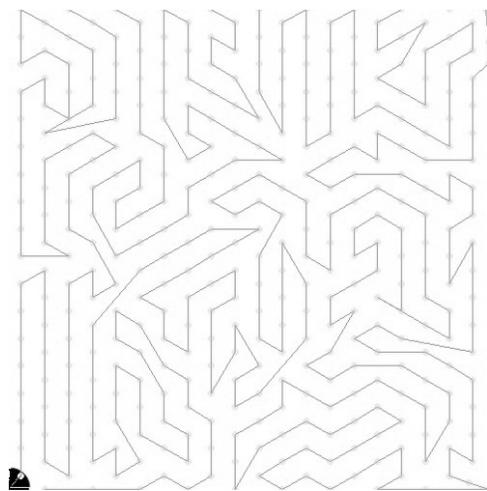Fig. 8 - Optimized path using Euclidean Distance.

Fig. 9 - Optimized Path using a Physical Vehicle Heuristic.

To evaluate the effectiveness of the vehicle mobility heuristic, a simulation was developed to model the behavior of a fixed wing air vehicle. The mobility of such vehicles

are well known and easy to model. The path planning algorithm was tested over 30 runs each over the same environment using the vehicle heuristic and Euclidean distance as the cost. The results are shown in the table below.

Run	PVH	Euclidean
	Time	Time
1	3104	3381
2	3158	3323
3	3293	3408
4	3179	3388
5	3209	3390
6	3167	3396
7	3223	3492
8	3106	3402
9	3265	3362
10	3271	3445
Mean	3197.5	3398.7

Table 1. Time to complete for two cost structures over 10 runs, physical vehicle heuristic and Euclidean distance.

F-Test Two-Sample for Variances		
	PVH	*Euclid.*
Mean	3207.889	3400.667
Variance	3761.861	2273.25
Observations	9	9
df	8	8
F	1.654838	
P(F<=f) one-tail	0.246041	
F Critical one-tail	3.438101	

Table 2. F-Test showing Equal Variance at α=0.05

The results show that optimization utilizing the mobility characteristics is able to significantly improve the time of completion for a mobility-restricted vehicle. The environmental and mobility characteristics of the vehicle greatly determine the time to complete, so there is great potential to optimize the path using this cost method.

VI. CONCLUSION

For scenarios where a mobility limited vehicle must plan an efficient path through a large, complex environment in real-time, traditional path planning methods are inadequate. A three component path planning architecture, combining divide and conquer, local and global optimization methods. It has been shown that this architecture has the advantage of both speed and accuracy, being able to plan an efficient path for a mobility restricted vehicle in real-time.

t-Test: Two-Sample Assuming Equal Variances		
	PVH	*Euclid.*
Mean	3207.889	3400.667
Variance	3761.861	2273.25
Observations	9	9
Pooled Variance	3017.556	
Hypothesized Mean Difference	0	
df	16	
t Stat	-7.4445	
P(T<=t) one-tail	6.96E-07	
t Critical one-tail	1.745884	
P(T<=t) two-tail	1.39E-06	
t Critical two-tail	2.119905	

Table 3. t-Test assuming equal variances showing that the samples are significantly different at α=0.05

VII. REFERENCES

[1] K. Nagatani and H. Choset, "Toward robust sensor based exploration by constructing reduced generalized Voronoi graph," in *Intelligent Robots and Systems*, 1999, pp. 1687 - 1692.

[2] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research,* vol. 21, pp. 498-516, March-April 1973 1973.

[3] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for Large Traveling Salesman Problems," 2000.

[4] S. Mulder and D. C. Wunsch, "Million City Traveling Salesman Problem Solution by Divide and Conquer Clustering with Adaptive Resonance Neural Networks," *Neural Networks,* July 2003.

[5] D. C. Wunsch and S. Mulder, "Using Adaptive Resonance Theory and Local optimization to divide and conquer large scale traveling salesman problems," in *International Joint Conference on Neural Networks*, 2003, pp. 1408-1411.

[6] D. C. Wunsch and S. Mulder, "Evolutionary Algorithms, Markov Decision Processes, Adaptive Critic Designs, and Clustering: Commonalities, Hybridization and Performance," in *International Conference on Intelligent Sensing and Information Processing*, 2004, pp. 477-482.

[7] H.-K. Tsai, J.-M. Yang, and C.-Y. Kao, "Solving traveling salesman problems by combining global and local search mechanisms," in *Conference on Evolutionary Computation*, 2002, pp. 12-17.

[8] H.-K. Tsai, J.-M. Yang, and C.-Y. Kao, "An evolutionary algorithm for large traveling salesman

problems," *IEEE Transactions on Systems, Man and Cybernetics Part B,* vol. 34, pp. 1718-1729, August 2004.

[9]   E. Gonzalez, A. Suarez, C. Moreno, and F. Artigue, "Complementary regions: a surface filling algorithm," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996, pp. 909-914 vol.1.

[10]   R. N. De Carvalho, H. A. Vidal, P. Vieira, and M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on*, 1997, pp. 677-682 vol.2.

[11]   G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks,* pp. 493-504.

[12]   P. Merz and B. Freisleben, "Genetic Local Search for the TSP: new results," in *IEEE Conference on Evoluationary Computation*, 1997, pp. 159-164.