2016 International Conference on Applied Mathematics and Mechanics (ICAMM 2016)
ISBN: 978-1-60595-399-1

Optimal Neural Networks for the Shortest Path Computation

Leon A. Luxemburg

Texas A&M University Galveston PO Box 1675 Galveston, Texas 77553-1675 USA

Corresponding author

Keywords: Neural network, Optimization, Shortest path, Voronoi diagram.

Abstract. One of the key algorithm research areas for unpiloted aerial vehicles team cooperative control is solution of network optimization models. Computational speed is an important success criterion as much of the cooperative work is done in real time. In this paper a neural network (NN) approach to solving the shortest path problem in the context of cooperative control problems will be introduced. We prove the optimality of our approach mathematically. Computational time will be used as a comparison and it will be shown that significant time savings result from the NN approach.

Introduction

Team cooperative control problems consist of generating trajectories for each UAV team member that enables the team to accomplish a mission with a desired level of optimality. Much work has been accomplished recently in the area of optimization of team cooperative control, [1-3].

To be a successful candidate for the solution of cooperative control a solution must satisfy four criteria:

1. Performance or accuracy of solution
2. Stability
3. Minimal computation requirements
4. Flexibility to accept comprehensive team cooperative control formulations

The solution must be a reasonably close approximation to the optimal solution within a few iterations. The solution should be numerically stable so that it will not 'hang' during iterations. Nearly all missions must achieve time/fuel optimality. During the course of a mission the information map may be updated requiring a new optimization. If a new trajectory is generated during the optimization process minimal time should pass before the UAV's are redirected. Optimization will occur at multiple levels within team cooperative control. At the highest level missions will be assigned to various heterogeneous teams. At an intermediate level, tasks will be assigned to team members. At the lowest level waypoints will be generated to compose a trajectory for an individual team member. In this article, a proposed neural network approach to the shortest path problem is explained. Some linear programming (LP) problems will be formulated and solved by the proposed method. The remainder of the paper is organized as follows: 1) proposed NN method, 2) mathematical proofs of optimality, 3) hardware solutions, 4) test problem formulations, and 5) numerical results.

Proposed Shortest Path NN (SPNN) Method

Consider a neural network with n(n+1)/2 nodes, one node for each unordered pair $(i, j), i \leq n, j \leq n$, where n is the number of nodes in the original path. This includes pairs (i, i) as well and node v_{ij} is considered the same as v_{ji}. Each node v_{ij} is interconnected with a node v_{sk} if and only if they have at least one common index, i.e. $s = i \text{ or } s = j \text{ or } k = i \text{ or } k = j$. The outputs of nodes of SPNN can be arranged into a symmetric matrix, where the outputs of the previous iteration are fed as inputs in the next iteration. Once the outputs of the k^{th} iteration are defined, then the ij output of the (k+1)$^{\text{st}}$ iteration is as follows: take the sum of the i^{th} row and j^{th} column of the output of k^{th} iteration and feed it into the transfer function of the ij^{th} node.

The transfer function is the minimum of the inputs: $f_{ij}(x) = \min_i \{x_i : x = (x_i)\}$. Let now P_n be a connected polygon with n vertices such that all arcs have a nonnegative associated cost. We initialize our neural net by setting v_{ij} to be the cost of the arc between the i^{th} and j^{th} vertex if it exists and ∞ otherwise. We set the output of node v_{ii} to zero. The algorithm determining the final output works as follows:

1. For n given points in space assign the distance d_{ij} (cost) between each point x_i *and* x_j *to node* v_{ij} of the neural net SPNN.

2. In each iteration (clock cycle) SPNN calculates the output of each node v_{ij} as the transfer function output of its inputs. The transfer function is simply the minimum of its inputs and the inputs to the node v_{ij} are sums $o_{is} + o_{sj}$ *where* o_{mn} *is the output of the node* v_{mn} in the previous iteration. The first iteration assigns to node v_{ij} the output $o_{ij} = d_{ij}$.

We can now formulate the following:

Proposition 1: With the initialization described above, SPNN(n) converges after $\le \lceil \log_2(n) \rceil$ steps and the output of the ij^{th} node v_{ij} is the length of the shortest path between vertices i and j.

Proof: We can prove inductively that after the k^{th} step the output of v_{ij} is the length of the shortest path between vertices i and j provided it consists of $\le 2^k$ arcs. For k=0 it is obvious. Suppose it is true for k and we want to prove it for k+1. Let T be the shortest path between vertices i and j such that T consists of $\le 2^{k+1}$ arcs. Then T can be presented as the juxtaposition of 2 paths $T = T_{il}T_{lj}$ with T_{il} and T_{lj} each consisting of $\le 2^k$ arcs. By inductive assumption, T_{il} and T_{lj} are shortest paths of the k^{th} iteration, so v_{il} and v_{lj} are outputs of the k^{th} iteration. Now, the length of $T = v_{il} + v_{lj}$ is the output of the $(k+1)^{st}$ iteration. Indeed, in the (k+1)-st iteration the node v_{ij} has an output which is the minimum of sums $v_{il} + v_{lj}$ over l. Therefore the shortest path length T is among these sums which represent lengths of paths of length $\le 2^{k+1}$ between nodes i and j. Clearly, the shortest path length T is the output of node v_{ij}. This proves the inductive assumption. Let $k = \lceil log2(n) \rceil$. Then any path with no repeating nodes will have $\le 2k$ arcs so after the kth iteration the shortest path length will be the output of the node v_{ij} . QED.

A circuit block diagram of the proposed neural network hardware realization is shown on Figure 1.

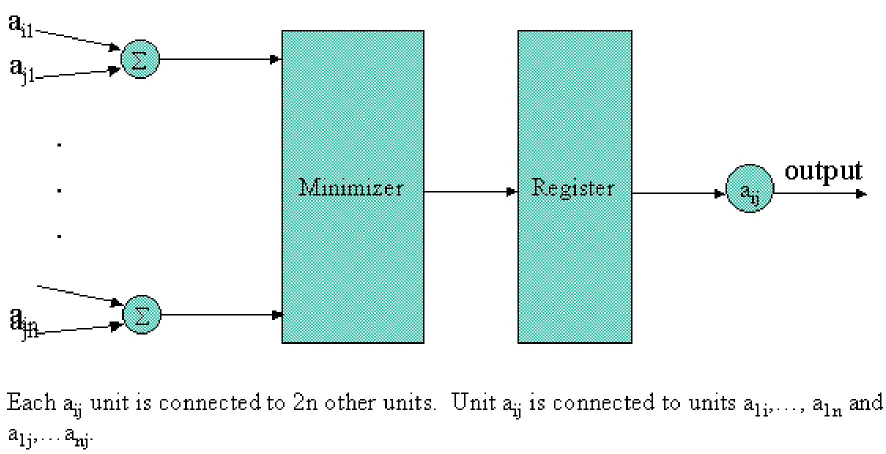

Figure 1. Circuit block diagram for 1 unit of LP network.

Let $x_1, \ldots, x_n$ be points on a plane. Then the Voronoi diagram defined by these points is the set of points which are equidistant from at least two of the points x_i , $1 \le i \le n$. Voronoi diagram is a polygon consisting of vertices and edges.

The regions $V(x_i)$ consist of points which are closer to x_i than to other points x_j. These regions sometimes are called faces of the Voronoi diagram. If points x_i are the points where threats to UAV are located, the UAVs would want to move along the lines most removed from these points, i.e. along the edges of the Voronoi diagram. The problem of finding the shortest path along the graph formed by the edges of this Voronoi diagram is the problem relevant to our Neural Net. We should note that the computational complexity of the problem as it applies to the UAV tasks should be formulated not in terms of the nodes of the Voronoi diagram but in terms of the number of original threat points.

The following proposition establishes the pertinent upper bound on the computations.

Proposition 2. Let n be the number of points on a plane defining the Voronoi diagram. Then SPNN(n) computes the shortest paths in $\lceil \log_2(n)+1 \rceil$ steps.

Proof: Given n points, the number of vertices of the Voronoi diagram constructed from them is $\leq 2n-4$, see [4], page 161.

Therefore, the total number of iterations is $\leq \lceil \log_2(2n-4) \rceil \leq \lceil \log_2(2n) \rceil = \lceil \log_2(n)+1]$. QED

Proposition 3. The shortest path problem requires at least $\Omega(n\log_2 n)$ computational steps in serial computations for each pair of nodes i, j.

Proof: Consider a set P of n+2 points $p_0,\ldots,p_{n+1}$ on a real line such that the smallest is $p_0=0$ and the largest is $p_{n+1}=1$. Assume a fully interconnected graph. Then the shortest path between p_0 and p_{n+1} is a reordering of set P in increasing order from $p_0=0$ to $p_{n+1}=1$. Thus, serial computation complexity of the shortest path cannot be less than that for the sorting problem. However, the sorting algorithm requires at least $\Omega(n\log_2 n)$ steps, [5].

A significant improvement of our method in reducing the computational complexity bound given in Proposition 3 is achieved through massive parallelism of the neural network.

Because SPNN(n) computes the result in $\lceil \log_2(n)+1 \rceil$ steps and each step requires n additions (done in parallel in 1 step) and taking of a minimum of n numbers which requires at least $\lceil \log_2(n) \rceil$ parallel comparisons, the total number of arithmetic operations required is $O(\log_2^2(n))$. An interesting question is to establish if the bound in Proposition 2 can be improved by lowering the estimate of the number of Voronoi nodes. The following two theorems show that this is not so. The proofs will be given in the next section.

First, we need a definition. Let I^2 be a square $\{(x, y)\colon 0 < x < 1, 0 < y < 1\}$ in a Euclidean plane R^2 and let $p(x, y)$ be a uniform pdf,

$p(x, y) = 1\ if\ (x, y) \in I^2$

And $p(x, y) = 0$ otherwise. Then the following Theorems are true:

Theorem 1. Let $x_1, \ldots, x_n$ be n points in I^2 chosen randomly according to the uniform distribution. Then the expected number $E(V_n)$ of Voronoi nodes in the Voronoi diagram defined by points

$x_1, \ldots, x_n$ is $2n - 4 - o(n)$,

or, equivalently,

$$2n - o(n),\ \text{i.e} \lim_{n\to\infty} \frac{E(V_n)}{2n} = 1.$$

This theorem shows that there exists an integer N such that $E(V_n) > n\ for\ n \geq N$ and so on average

$V_n > n$ and $\lceil \log_2(V_n) \rceil = \lceil \log_2(n)+1 \rceil$.

Therefore, the estimate in Proposition 2 can not be reduced by decreasing the upper bound on the number V_n .

Theorem 2. For any $\varepsilon > 0$ and $\delta > 0$ there exists an integer N such that for any $n \geq N$ and for the points $x_1,\ldots,x_n$ chosen randomly from I^2 with uniform probability, the following inequalities are true for the number of Voronoi nodes $V_n = V_n(x_1,\ldots,x_n)$ defined by points $x_1,\ldots,x_n$:

$2n - 4 - n\varepsilon \leq V_n \leq 2n - 4$ with probability $\geq 1-\delta$.

In fact, Theorem 1 follows from Theorem 2. Indeed, for the expected value $E(V_n)$ of V_n from Theorem 2 it follows that

$$2n - 4 \geq E(V_n) \geq (2n - 4 - n\varepsilon)(1-\delta) = 2n - 4 - [n(\varepsilon + 2\delta) - 4(1-\delta)]$$

Since δ and ε are arbitrary, and number s > 0, the expression in brackets is, obviously, $o(n)$.

Theorem 3. The statement of Theorem 2 remains correct if we replace I^2 with any bounded open set U in R^2.

Proof of Theorems 2 and 3.

The proof is based on a series of preliminary statements. First, note that if n is an integer, the probability distribution of n tuples $x_1, \ldots, x_n$, $x_i \in I^2$ is the product of uniform distributions on $\prod_{i=1}^{n} I^2$ and is simply the Lebesgue measure on I^{2n} which is the Cartesian product of n copies of I^2. Therefore, to every n tuple $x_1, \ldots, x_n$ of points in I^2 there is a corresponding point $p \in I^{2n}$ whose i-th coordinate is x_i.

In what follows, all points taken from I^2 will be assumed to be chosen randomly according to the uniform distribution.

Lemma 1. With probability 1 the Voronoi diagram V($x_1, \ldots, x_n$) defined by points $x_1, \ldots, x_n$ has only vertices which are intersections of ≤ 3 edges.

Proof. Let B be the set of points p=p $(x_1, \ldots, x_n) \in I^{2n}$ defining Voronoi diagram V such that some of its vertices are intersections of at least 4 edges. Therefore, there is a vertex $v \in V$ which is equidistant from 4 points, say, $x_1, \ldots, x_4$. Then x_1 lies on the circle in I^2 circumscribed about, x_2, x_3, x_4 . Let D(1,2,3,4) be the set of all points
p=p($x_1, \ldots, x_n$)$\in I^{2n}$,
such that x_1 lies on the circle in I^2 circumscribed about , x_2, x_3, x_4. Then D is a 2n-1 dimensional subset of a (2n-1) dimensional manifold and therefore has measure 0 in I^{2n}. clearly, B is a union of a finite number of sets $D(i, j, k, l)$ and , therefore has measure 0.

Lemma 2. With probability 1, the following formula is true for the number v_0 of vertices of Voronoi diagram defined by points $x_1, \ldots, x_n \in I^2$.

$$v_0 = 2n-2-u \tag{1}$$

where u is the number of points x_i with unbounded Voronoi region. (Even though $x_i \in I^2$, the Voronoi region is in $R^2 \supset I^2$ and can be unbounded).

Proof. We will use Euler's formula
$v_2 - v_1 + v_0 = 1$, where $v_2 = n$ is the number of Voronoi regions and v_1 is the number of edges. (2)

By Lemma 1 with probability 1 each Voronoi vertex belongs to exactly 3 edges and each edge has two vertices unless one end is unbounded and in this case the edge has one vertex. Also, it is easy to see that the number of unbounded edges is equal to the number u of unbounded Voronoi regions. Therefore,

$$3v_0 = 2v_1 - u \text{ implying } v_1 = 3v_0/2 + u/2 \tag{3}$$

Substituting $v_2 = n$ and (3) into (2) we obtain (1).

Lemma 3. Let Q be a bounded polygon on the Euclidean plane R^2 with vertices $v_1, \ldots, v_q$ such that each edge (v_i , v_{i+1}) has length <ε. Let X be a finite set of points in I^2 containing all vertices of Q and let $x \in X$ lie inside of Q and be more than ε/2 away from all vertices of Q. Let V(X) be the Voronoi diagram defined by X. Then the Voronoi region V(x) of point x belongs to the interior of Q and is, therefore, bounded.

Proof. If V(x) does not belong to the interior of Q, then it must intersect Q at one of the edges (v_i , v_{i+1}) at a point z. Then, since the length of (v_i , v_{i+1}) is <ε, z is less than ε/2 away from either v_i or v_{i+1}. However, x is more than ε/2 away from all vertices of Q. Therefore z is closer to either v_i or v_{i+1} than to x and can't belong to the Voronoi region of z.

Let now k be an integer > 1 and let square I^2 be divided into k^2 equal small squares with side lengths 1/ k^2 and sides parallel to the sides of I^2. There are, obviously, 4(k-1) "peripheral" small squares touching the boundary of I^2. This construction and terminology will be assumed for the rest of this section.

Lemma 4. The probability that given m points on I^2, there is at least 1 point in every peripheral small square is at least
$1 - 4(k-1)(1-\frac{1}{k^2})^m$.

Proof. Because of the uniform distribution on I^2 and the fact that every small square has area $1/k^2$ and side $1/k$, the probability of not having any points in a particular small square is $(1-\frac{1}{k^2})^m$. Since there are $4(k-1)$ peripheral squares, the probability of not having points in at least one of them is $\leq 4(k-1)(1-\frac{1}{k^2})^m$.

Therefore, the probability of having a point in each peripheral square is $\geq 1-4(k-1)(1-\frac{1}{k^2})^m$.

Lemma 5. Let $M \subset I^2$ be a square inside I^2 with the same center and sides parallel to those of I^2 and having side length $(1-\frac{6}{k})$. Let m be a fixed integer, $\varepsilon>0$ and $p=(1-\frac{6}{k})^2$ (p is the area of M). Then the probability that out of m randomly chosen points on I^2 at least $m(1-\varepsilon)p$ will be in M is at least $1-\exp(\frac{-\varepsilon^2 mp}{2})$.

Proof. Since the distribution is uniform, the probability of getting one point inside of M is p. Thus, we have a binomial distribution of degree m. We can now use the Chernoff inequality[6] which states that in m binomial trials the probability of getting less than $m(1-\varepsilon)p$ successes (when success is defined as an event that a randomly chosen point $x\in I^2$ belongs to M) is $\leq \exp(\frac{-\varepsilon^2 mp}{2})$. The Lemma follows.

Lemma 6. Define the event $L = L(m,\varepsilon,k)$ which occurs when out of a set X of m randomly chosen points in I^2 at least $m(1-\varepsilon)p$ points will be in M and each of the peripheral squares has at least one point of X inside. Then the probability of L is

$$\geq P(m,\varepsilon,k) = 1-4(k-1)(1-\tfrac{1}{k^2})^m - \exp(\frac{-\varepsilon^2 mp}{2}) \tag{4}$$

Proof. Immediately follows from Lemmas 4 and 5.

Lemma 7. In notations of Lemma 6, in the event $L = L(m,\varepsilon,k)$ with probability $\geq P(m, \varepsilon,k)$ at least $r = m(1-\varepsilon)p$ points out of m have bounded Voronoi region, when Voronoi diagram is defined by the set X.

Proof. Indeed, in the event L, every point lying inside M is at least 2/k distance away from any peripheral square. Also, in this case there is at least one point in every peripheral square. Let us choose one point of X in every peripheral square and connect two points in every pair of adjacent peripheral squares. This will form a polygon Q with sides of lengths $\leq\frac{\sqrt{5}}{k}$ **(**which is the largest distance within two adjacent squares). Since $2/k > \frac{\sqrt{5}}{2k}$**,** by Lemma 3 every point of X lying in M has bounded Voronoi region.

Let k=k(m) be an integer defined so that

$$2k^2\ln(k) \leq m < 2(k+1)^2\ln(k+1) \tag{5}$$

Then, clearly,

$$k\to\infty \text{ if and only if } m\to\infty. \tag{6}$$

Lemma 8. Let X be a set of m points randomly chosen on I^2 then the probability P that out of these m points at least

$r = m(1-\frac{1}{k})(1-\frac{6}{k})^2$

have bounded Voronoi region is $\geq (1-\frac{5}{k})$. Here we assume that Voronoi diagram is defined by set X and k=k(m) is defined by (5).

Proof. Let $\varepsilon=1/k$ then because $p=(1-\frac{6}{k})^2$, from Lemma7 it follows that

$P \geq P(m, 1/k,k)= 1\text{-}4(k\text{-}1)\ (1-\frac{1}{k^2})^m\ \text{-exp}(\frac{-mp}{2k^2})$.

From (5), we obtain

$P(m, 1/k,k) \geq 1\text{-}4(k\text{-}1)\ (1-\frac{1}{k^2})^{k^2 \ln(k^2)}\ \text{-exp}(-\ln(k)\ (1\text{-}\frac{6}{k})^2)$.

Using inequality $(1\text{-}1/n)^n < e^{-1}$ we note that the right hand side of the previous inequality is $\geq 1\text{-}4(k\text{-}1) \exp(-\ln(k^2)) -(1/k)(1\text{-}\frac{6}{k})^2 \geq 1\text{-}4(k\text{-}1)/k^2\text{-}1/k \geq 1\text{-}4/k +4/k^2\text{-}1/k \geq 1\text{-}5/k$

Lemma 9. Let ε>0 and δ>0. Then there exists an integer N>0 such that for every m≥N the probability that out of m randomly chosen points on I^2 at least (1- ε)m have bounded Voronoi regions in the Voronoi diagram defined by these m points is ≥ 1-δ.

Proof. Follows immediately from Lemma 8 and from (6).

*Proof of Theorem 2***.** Let ε>0 and δ>0 then choose N as in Lemma 9. By Lemma 2 with probability 1 we have equality (1). Since with probability ≥ 1-δ we have u ≤m – (1-ε)m=mε, from formula (1) it follows that 2m-2-mε≤ 2m-2 –u ≤V_m. Inequality V_m≤ 2m-4 is always true as we have stated before.

*Proof of Theorem 3.*For anyε>0 and δ>0 there exists a finite set of squares $Q_1,\ldots, Q_r$ in U such that

$$\mu(U)\text{-}\ \mu(\bigcup_{i=1}^{r} Q_i) < \delta,$$

where μ is the Lebesgue measure on R^2. Let X be a finite set in U. Now we apply Theorem 2 to each square Q_i and to Voronoi diagrams V_i defined by sets $X_i=X\cap Q_i$ and to the Voronoi diagram V defined by X. It is easy to see that $u \leq \sum_{i=1}^{r} u_i$, where u_i is the number of points with unbounded Voronoi regions in the diagram defined by points in X_i and u is the number of unbounded Voronoi regions in V. Using Chernoff inequality as in Lemmas 5 and 6 we can show that with probability at least $1\text{-exp}(\frac{-\varepsilon^2 mp}{2})$ there are at least m(1-ε)p points in the set X which lie in $\bigcup_{i=1}^{r} Q_i$, where

$$p= \mu(\bigcup_{i=1}^{r} Q_i)/\mu(U).$$

The rest of the proof is the same as in Theorem 2.

Test Problem Formulations

A typical test problem scenario requires safely maneuvering through a field of known threats in order to arrive at a particular location. The optimization objective is to minimize travel distance while minimizing threat exposure, subject to constraints of coordinated team arrival, total time, and team capability. Minimum paths generally fall along two-dimensional Voronoi diagram links, [4-6]. If a set of lines were drawn between nearest neighbors of a cluster of objects the set of bisectors is the foundation of the Voronoi diagram. The following plot illustrates the concept. The circles are the input points to the diagram. The lines equidistant from the nearest neighbors (circles) compose the Voronoi diagram. A suboptimal approach to generating a minimum time path to a location within the Voronoi diagram then consists of choosing the shortest path. When the threat costs are incorporated with the distance costs, a network may be formed. For the test a four-node network and a seven-node network were formulated as shown in the Figure 2 and 3.

Summary

We have developed a serial implementation of the SPNN algorithm in the matlab environment. We used the matlab command linprog for validation of the algorithm. Although the linprog output is a Dijkstra-type single path output we used it to validate the single path outputs.

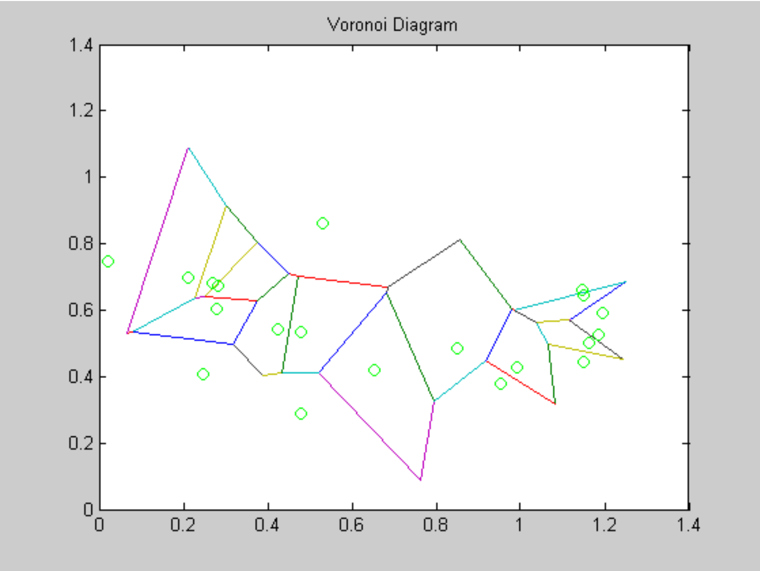

Figure 2. Voronoi diagram.

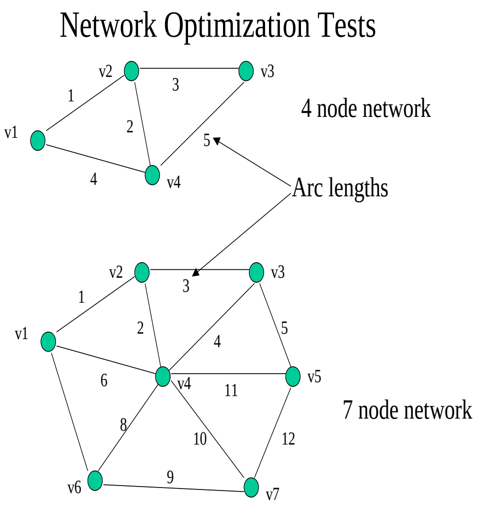

Figure 3. Network optimization tests.

In summary, the SPNN (parallel version) requires $O((\log_2 n)^2)$ steps, while a specialized circuit presented on the diagram above can do the computations in $O(\log_2(n))$ steps (clock cycles) because the summing and minimization is achieved by combinational logic between the clock cycles. The serial version of SPNN requires $O(n\log_2 n)$ computational steps and by our example is shown to be much faster than the LP approach. Note the linprog approach requires at least $O(mn\log_2 n)$ computational steps for completion time where m is the number of paths and n is the number of network nodes. For example, a team of m = 30 must traverse a threat field of 50 -> n ~ 100 (max. number of voronoi faces = (2n-4)[19]. Therefore we may compute the upper limits; linprog would require $O(30*100*\log_2 100) \sim O(3000*7) = O(210{,}000)$, SPNN requires $O((\log_2 100)^2) \sim O(7*7) = O(49)$, and for SPNN specialized circuit $O(\log_2 n) \sim O(7)$. This shows that SPNN gives a fast convergence for parallel computing of the shortest path problem.

Acknowledgement

This research was financially supported by the Office of Naval Research.

References

[1] T. McLain and R. Beard, “Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles,” AIAA-2000-4369.

[2] J. K. Howlett, “Path Planning and Cooperative Assignment,” technical report, Air Force Research Laboratory, Air Vehicles Directorate, Sept., 2001.

[3] P. Chandler, M. Pachter, S. Rasmussen, and C. Schumaker, “Multiple Task Assignment for a UAV Team,” AIAA 2002-4587.

[4] O’Rourke, J., Computation Geometry in C, 2nd Edition, Cambridge University Press, 1998, ISBN 0-521-64976-5.

[5] Preparata, F.P. & Shamos, M.I. (1985), Computational Geometry: An Introduction, Springer – Verlag, New York. Corrected and expanded second printing, 1988.

[6] Chernoff, H. (1952) A measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations, Annals of Mathematical Statistics, 23:493-509.