A Distributed Approach for Robotic Coverage Path Planning Under Steep Slope Terrain Conditions

Conference Paper · December 2022

DOI: 10.1109/SSCI51031.2022.10022252

5 authors, including:

Chetan Badgujar
Kansas State University
16 PUBLICATIONS **32** CITATIONS
SEE PROFILE

Daniel Flippo
Kansas State University
30 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Stephen Welch
Kansas State University
129 PUBLICATIONS **3,719** CITATIONS
SEE PROFILE

A Distributed Approach for Robotic Coverage Path Planning Under Steep Slope Terrain Conditions

Dania Martinez-Figueora
Dept. of Electrical & Computer Engg.
Kansas State University
Manhattan, KS 66505, USA
daniam@ksu.edu

Sanjoy Das*
Dept. of Electrical & Computer Engg.
Kansas State University
Manhattan, KS 66505, USA
sdas@ksu.edu
0000-0003-0699-1848
*Corresponding Author

Chetan Badgujar
Dept. of Biological & Agricultural Engg.
Kansas State University
Manhattan, KS 66505, USA
chetan19@ksu.edu

Daniel Flippo
Dept. of Biological & Agricultural Engg.
Kansas State University
Manhattan, KS 66505, USA
dkflippo@ksu.edu

Stephen J. Welch
Division of Agronomy
Kansas State University
Manhattan, KS 66506, USA
welchsm@ksu.edu

2022 IEEE Symposium Series on Computational Intelligence (SSCI) | 978-1-6654-8768-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SSCI51031.2022.10022252

***Abstract*—This article proposes a novel algorithm to determine the optimal coverage path of a mobile robot for uniform seed dispersion in any tract of agricultural land. The robot is required to operate under steep terrain conditions that are too risky for conventional, human operated equipment. Using data from an field experiment with the real robot, machine learning based function approximators are trained to estimate the minimum energy paths between adjacent points. Exemplar-based clustering is used to identify a suitable subset of way-points that ensure full coverage of a tract of land. An optimal cyclic tour is computed from the way points using a new asymmetric TSP algorithm proposed for this specific application. The clustering and the cyclic path planning algorithms can be implemented entirely through local message passing in a field-deployed sensor network. Simulations with synthetic as well as real topographic data establish the overall effectiveness of the proposed method.**

Keywords—Affinity propagation, agricultural robot, exemplar clustering, message passing, min-sum algorithm, mobile robot, traveling salesperson problem, path planning.

I. Introduction

Although Kansas is a predominantly agricultural state in the US, its northeastern region is characterized by the presence of steep escarpments of up to 400 feet [1]. Land with such steep inclines cannot be utilized for crop production as it is hazardous for humans to operate conventional farm equipment. Under such terrain conditions, cultivation can only be accomplished by using robots to perform various cropping tasks.

This research considers the task of uniform dispersal of seeds in highly uneven land using a mobile robot designed to carry a large quantity of seeds for uniform seed dispersal. At first, performance characteristics of an actual robot are obtained through field experiments. A regression model is trained with this dataset to estimate the robot's performance related metrics under unknown terrain conditions.

This research was supported by the US Department of Agriculture, under National Robotics Initiative Grant No. KS4513081.

A real tract of land is considered as the platform for this research. The regression model is used to predict the energy needed as well as the time taken by the robot to traverse a straight line from one point to another one in its immediate vicinity. The points are assumed to lie in a Riemannian manifold so that distal pairs of points are connected via geodesics.

Other methods for robot path planning (cf. [2], [3]) have been proposed. Unfortunately, they are not suitable for the present application. The recent approach in [4] considers a similar problem. However, it uses energy consumption as a constraint instead of a cost function for minimization. Moreover, it is applicable only in scenarios such as aerial robots/UAVs, where all pairwise distances are symmetric. This approach involves computing the Voronoi tessellation of a set of way points. Next, the spatial locations of these points are incremented in steps until all constraints are satisfied. The algorithm simulates the relaxation of a physical system of multiple, interconnected (and stretchable) springs. As these methods are inherently global, the overall approach in [4] cannot be implemented in a fully distributed computing environment.

Coverage path planning for the robot is carried out in three steps. At first, regression models are developed using experimental data, to estimate robot's performance for arbitrary terrain conditions. Next, an exemplar based clustering algorithm is applied to identify a set of suitable way points from the available high resolution topographic data. At the third step, a minimum energy cyclic tour is obtained by connecting the way points using a novel combinatorial algorithm for the asymmetric traveling salesperson (TSP) problem. A cyclic path is required to allow the robot to return to its starting point. As the final step, a high resolution coverage path plan is extracted by connecting the optimal sequence of waypoints using an asymmetric shortest paths algorithm, to obtain the optimal cyclic tour that fully covers the land tract. The algorithms for clustering and path optimization can be implemented entirely by means of

distributed message passing. The proposed approach can readily be extended for other agricultural purposes, such as fertilizing crops, spraying insecticides, or watering crops in terrains with steep slopes.

The rest of this article is organized as follows. Section II addressed how minimum work distances between pairs of points are obtained. An overview of the message passing approach is provided in Section III. Section IV describes how message-passing clustering is used to identify suitable way points. In Section V, a message-passing algorithm to determine the optimal cyclic tour is proposed. Results are provided in the Section VI, which also concludes this research.

II. Geodesic Distance Estimation

A. Regression Model

Experimental data related to the robot's performance under varying conditions is used in this step. The dataset consisted of the following fields: slope θ (° or degrees) of the terrain, the robot's speed v (meters/minute), its drawbar pull F (Newtons), travel reduction ratio γ, and power number n_p. Regression models [5] were trained to estimate γ and n_p as functions of θ, v, and F. The models serve as function approximators of the following,

$$\gamma \approx f_{TRR}(\theta, v, F). \quad (1)$$

$$n_p \approx f_{PN}(\theta, v, F). \quad (2)$$

In the above expression $f_{TRR}: \mathbb{R}^3 \to \mathbb{R}$ and $f_{PN}: \mathbb{R}^3 \to \mathbb{R}$ is a nonlinear mapping. Hereafter, it is assumed that the approximations in (1) and (2) are equalities.

Suppose the robot is required to traverse a very short straight line segment of length d, with a fixed slope of θ. Its optimal traversal speed v^* to minimize energy consumption, as well as the corresponding drawbar pull F^* can be obtained using convex optimization as,

$$[v^*, F^*] = \underset{v,F}{\operatorname{argmin}}\, f_{TRR}(\theta, v, F). \quad (3)$$

The power number n_p^* at the optimal point can be retrieved using the expression,

$$n_p^* = f_{PN}(\theta, v^*, F^*). \quad (4)$$

The time required by the robot to optimally traverse the distance d is given as,

$$t^*(d, \theta) = \frac{d}{v^*}. \quad (5)$$

If w is the robot's weight, the power required is $n_p^* \times w \times v^*$. Thus, the optimal energy required to traverse the straight line segment is obtained in a straightforward manner as,

$$e^*(d, \theta) = n_p^* \times w \times v^* \times t^*. \quad (6)$$

B. Digraph Weights

The elevation data is in the form of a 2 dimensional lattice $\mathcal{L}$ of points at regularly spaced intervals, and the elevations. Each point $i \in \mathcal{L}$ in is assumed to be connected to a set of 20 points in its vicinity, which are also in $\mathcal{L}$, as shown in Fig. 1. Diagonal connections are avoided in order to prevent the mobile robot from taking sharp turns.

Let $\mathcal{L}_i \subset \mathcal{L}$ represent the set of all such points in $\mathcal{L}$ that are connected directly to i. All points in the lattice are treated as the vertices of a digraph whose weights are determined in the manner described below. Let the horizontal and vertical coordinates of a point $i \in \mathcal{L}$ be $\mathbf{x}_i = (x_i, y_i)$ and its elevation be z_i. Consider any other point in the lattice such that $j \in \mathcal{L}_i$. The distance and slope from i to j are obtained as,

$$\begin{cases} d_{i,j} = \left(\left\| \mathbf{x}_j - \mathbf{x}_i \right\|^2 + \left| z_j - z_i \right|^2 \right)^{\frac{1}{2}} \\ \theta_{i,j} = \tan^{-1} \dfrac{z_j - z_i}{\left\| \mathbf{x}_j - \mathbf{x}_i \right\|} \end{cases}. \quad (7)$$

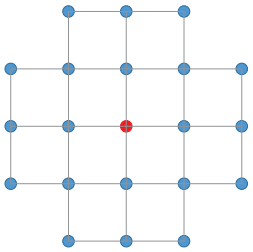

Fig. 1. Direct Connections. A point i (red) in the lattice that is directly connected to other points (blue) in $\mathcal{L}_i$.

The quantities v^*, F^* and n_p^* are determined from (3) and (4) by letting $d = d_{i,j}$ and $\theta = \theta_{i,j}$. The time $t_{i,j}$ required by the robot to cover the edge $i \to j$ is equal to t^*, which is obtained from (5). The weight w_{ij} of the edge $i \to j$ is set to be equal to the optimal energy, which is obtained using (6). It must be noted that $w_{i,j} \neq w_{j,i}$ (since $\theta_{j,i} = -\theta_{i,j}$).

For any other point $j \notin \mathcal{L}_i$ in the lattice, the energy required to go from i to j is assumed to be the geodesic distance [7]. The shortest path $i \rightsquigarrow j$, is a reasonable approximation of the geodesic. It can be obtained using the well-known Dijkstra's algorithm. Let $i \rightsquigarrow j \equiv k_0 \to k_1 \to \cdots \to k_{n-1} \to k_n$, where $k_0 = i, k_n = j$, and $k_1, \ldots, k_{n-1}$ are intermediate points such that $k_{m-1} \to k_m$ exists, i.e. $k_m \in \mathcal{L}_{k_{m-1}}$.

The optimal energy for the robot to go from point $i \in \mathcal{L}$ to any other point $j \in \mathcal{L}$ is used as the weight $w_{i,j}$ of the directed edge (i, j) of a digraph. From the previous discussion, this weight can be expressed as,

$$w_{i,j} \triangleq \begin{cases} e^*(d_{i,j}, \theta_{i,j}), & j \in \mathcal{L}_i \\ \sum_{m=1}^{n} w_{k_{m-1},k_m}, & \text{otherwise} \end{cases}. \quad (8)$$

III. Distributed Message Passing

A. Factor Graph Model

A factor graph model (FGM) is an unweighted digraph consisting of two types of nodes [8], (*i*) variable nodes $v_i, i \in \mathcal{V}$, and (*ii*) factor nodes, $f_k, k \in \mathcal{F}$. Each edge of the FGM links variable nodes with factor nodes. Let $\mathcal{E} \subset \mathcal{V} \times \mathcal{F}$ be the set of all edges. Not all variable nodes and factor nodes are connected. In the proposed approach described in this article, the FGMs are sparse, and the cardinalities $|\mathcal{E}|$ and $|\mathcal{F}|$ bear polynomial/quadratic relationships to $|\mathcal{V}|$.

Let $\mathcal{F}_i \in \mathcal{F}$ denote the set of factor nodes that are connected by edges to any node i in $\mathcal{V}$,

$$\mathcal{F}_i \triangleq \{k \in \mathcal{F} | (i,k) \in \mathcal{E}\}. \tag{9}$$

Likewise, the set of variable nodes $\mathcal{V}_k$ are nodes in $\mathcal{V}$ that are corrected to each factor node $k \in \mathcal{F}$,

$$\mathcal{V}_k \triangleq \{i \in \mathcal{V} | (i,k) \in \mathcal{E}\}. \tag{10}$$

The equivalence $i \in \mathcal{V}_k \Leftrightarrow k \in \mathcal{F}_i$ always holds in an FGM.

B. Min-Sum Message Passing

A variable node i in this research can only a acquire binary value $v_i \in \{0,1\}$. The messages $\mu_{i\to k}$ sent from i to each factor node $k \in \mathcal{F}_i$ is the sum of all incoming messages that i receives from all other factor nodes k' in $\mathcal{F}_i$ (but excluding k). Denoting such an incoming message to i from k as $\mu_{k\to i}$, we have,

$$\mu_{i\to k}(v_i) \triangleq \sum_{k' \in \mathcal{F}_i \setminus k} \mu_{k'\to i}(v_i). \tag{11}$$

A factor node k incorporates a function $f_k: \{0,1\}^{|\mathcal{V}_k|} \to \mathbb{R}$ of its arguments $v_i, i \in \mathcal{V}_k$. It performs a summation all its incoming messages $\mu_{i\to k}(v_i)$. Next, it computes its outgoing message $\mu_{k\to i}$ to each $i \in \mathcal{V}_k$, as the minimum over all v_i values of the product of the summation and $f_k(\cdot)$,

$$\mu_{k\to i}(v_i) \triangleq \min_{[v_i]_{i\in\mathcal{V}_k}} \sum_{i' \in \mathcal{V}_k \setminus i} f_k\big([v_i]_{i\in\mathcal{V}_k}\big). \tag{12}$$

The functions $f_k(\cdot)$ have been used effectively to impose various constraints upon the values v_i in this article. These constraints ensure the feasibility of the final solution.

Taking all possible values of each v_i into account, in the worst case scenario, the messages sent by the nodes should be binary tensors whose tensor dimensions grow exponentially in $|\mathcal{V}|$. However for the purpose of this research, the actual messages passed are always scalars,

$$\begin{cases} \mu_{i\to k} = \mu_{i\to k}(1) - \mu_{i\to k}(0) \\ \mu_{k\to i} = \mu_{k\to i}(1) - \mu_{k\to i}(0) \end{cases}. \tag{13}$$

The message passing described above is carried out iteratively. In an acyclic FGM, the values v_i converge towards the minimizer of the following objective function [8],

$$\Omega([v_i]_{i\in\mathcal{V}}) \triangleq \sum_k f_k\big([v_i]_{i\in\mathcal{V}_k}\big). \tag{14}$$

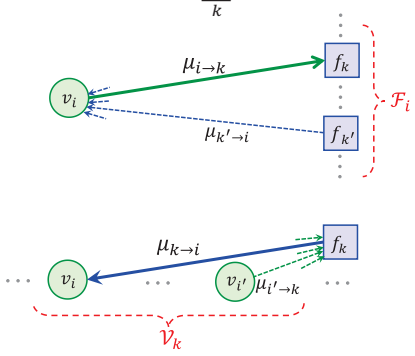

Fig. 2. Factor Graph Model. Variable nodes (green circles) and factor nodes (blue squares) and messages (arrows) are shown.

The proposed approach is implemented in two steps. At first a subset of points from $\mathcal{L}$ is selected using exemplar clustering. These are called the way points. Next, a cyclic tour of the way points is obtained, which defines the robot's path. Both steps are implemented using appropriate message passing algorithms. Although the underlying FGMs contain cycles, it has been shown that message passing can still produce acceptable solutions that are very nearly optimal [9].

IV. Way Point Selection

A. Coverage

From any fixed location, the robot can deliver seeds up to a maximum distance ρ, called its coverage radius (see Fig. 3). Thus, if the robot's path includes any point k with the coordinates $\mathbf{x}_k = (x_k, y_k)$, it covers the neighborhood defined as provided below,

$$\mathcal{N}_k = \{i \in \mathcal{L} | \|\mathbf{x}_k - \mathbf{x}_i\| \le \rho\}. \tag{15}$$

This observation can be leveraged effectively to reduce the total number of points in the original lattice into a smaller set of way points using a message passing algorithm [10].

Given a set of points $\mathcal{L}$ with distance $d_{i,k}$ between every pair of points $i, k \in \mathcal{L}$, exemplar-based clustering is applied to obtain a subset of exemplars $\mathcal{C} \subset \mathcal{L}$, such that each point $i \in \mathcal{L}\setminus\mathcal{C}$ has a corresponding unique exemplar $k \in \mathcal{N}$ such that,

$$\mathcal{C} = \underset{\mathcal{C}\subset\mathcal{L}}{\operatorname{argmin}} \sum_{i\in\mathcal{L}\setminus\mathcal{C}} \sum_{k\in\mathcal{C}} d_{i,k}. \tag{16}$$

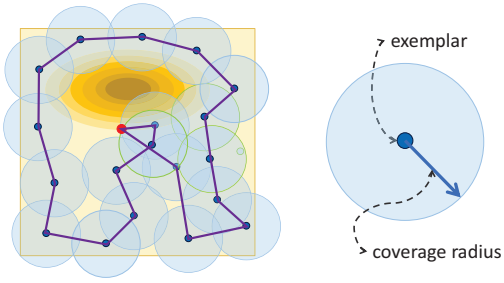

Fig. 3. Coverage. Area covered by connecting a set of exemplars (way points) with a fixed coverage radius (left), coverage by an exemplar (right). Note that additional area is covered as the robot moves along any line segment joining two exemplars.

B. Exemplar Clustering by Message Passing

The clustering algorithm uses a matrix of all pairwise distances $d_{i,k}$ between points $i, k \in \mathcal{L}$, that are equal to the negative distance between them (i.e. $s_{i,k} = -d_{i,k}$). Self-distances may be assigned any non-positive value $d_{i,i}$.

The FGM contains $O(|\mathcal{L}|^2)$ variable nodes. For simplicity, in this section it is assumed that there are exactly $|\mathcal{L}|^2$ nodes. Accordingly, there is a $|\mathcal{L}| \times |\mathcal{L}|$ matrix $\mathbf{V} = [v_{i,k}]_{i,k\in\mathcal{L}}$ of binary variables. If $i \neq k$ then $v_{i,k}$ is assigned a value according to the following expression,

$$v_{i,k\neq i} = \begin{cases} 1, & k \text{ is exemplar of } i \\ 0, & \text{otherwise} \end{cases}. \tag{17}$$

On the other hand, when $i = k$, the value of $v_{k,k}$ is,

$$v_{k,k} = \begin{cases} 1, & k \text{ is exemplar} \\ 0, & \text{otherwise} \end{cases}. \quad (18)$$

The FGM maintains a set of factor nodes to incorporate the distances,

$$D_{i,k}(v_{i,k}) = \begin{cases} d_{i,k}, & v_{i,k} = 1 \\ 0, & v_{i,k} = 0 \end{cases}. \quad (19)$$

There are $|\mathcal{L}|^2$ such factor nodes.

Additionally a valid clustering must satisfy two kinds of constraints. The first kind ensures that each point i picks one and only one exemplar k. Accordingly, each row of $\mathbf{V}$ must have exactly a single unit element while all others are zeros. These constraints are imposed by the following factor nodes,

$$I_i(v_{i,1}, \dots, v_{i,|\mathcal{L}|}) = \begin{cases} \infty, & \sum_k v_{i,k} \neq 1 \\ 0, & \text{otherwise} \end{cases}. \quad (20)$$

Furthermore, any point i may select k as its exemplar only if k is an exemplar itself. In other words, $v_{i,k}$ can be assigned unity only if $v_{k,k}$ in the same column k is also unity. The factor nodes to impose these set of constraints are,

$$E_k(v_{1,k}, \dots, v_{|\mathcal{L}|,k}) = \begin{cases} \infty, & \begin{cases} c_{k,k} = 0, & \text{and} \\ \exists i \neq k \text{ s.t. } c_{n,k} = 1 \end{cases} \\ 0, & \text{otherwise} \end{cases}. \quad (21)$$

There are $|\mathcal{L}|$ constraints of each kind, bringing the number of factor nodes in the FGM to $|\mathcal{L}|^2 + 2|\mathcal{L}|$. In reality, the size of the FGM can be reduced by incorporating variable nodes in the FGM iff $d_{i,k} \leq \rho$, and using only the relevant factor nodes. This yields a sparse binary matrix, $\mathbf{V}$ with only $O(|\mathcal{L}|)$ non-zero elements.

C. Way Points

From the original lattice $\mathcal{L}$, a subset $\mathcal{C}$ of points defined below is regarded as the final set of exemplars for the next stage of the proposed approach,

$$\mathcal{C} = \{k \in \mathcal{L} | v_{k,k} = 1\}. \quad (22)$$

The next step is to obtain a crude robot path by computing a cyclic path in $\mathcal{C} \times \mathcal{C}$. This is addressed in the subsequent section.

V. Robot Path Optimization

A. Cyclic Tour in Digraph

In order for the robot to cover the entire lattice $\mathcal{L}$, it must visit each way point in $\mathcal{C}$. Since the robot must return to original location after performing its task, its path must be a cyclic sequence of way points, that can be represented as an ordered permutation of way points in $\mathcal{C}$.

Such a sequence may alternately be specified as the set of edges $\mathcal{T} \subset \mathcal{C} \times \mathcal{C}$ with cardinality $|\mathcal{T}| = |\mathcal{C}|$, as long as the Held-Karp (H-K) validity conditions [11] are satisfied, which has been extended for use with asymmetric weights. In this manner, the need to handle sequential information is obviated, making it ideal for a binary message passing algorithm. The set $\mathcal{T}$ of edges in the cyclic sequence is frequently referred to as a (cyclic) tour in the context of TSP.

Noting that $w_{i,j}$ as defined earlier in (8) is the energy required to go from way point i to way point j, where $i, j \in \mathcal{C}$, the total energy required by the cyclic tour $\mathcal{T}$ is given as,

$$\Omega(\mathcal{T}) = \sum_{(i,j) \in \mathcal{T}} w_{i,j}. \quad (23)$$

A tour $\mathcal{T}$ that minimizes $\Omega(\mathcal{T})$ in (23) must be found. This task can be accomplished by means of any asymmetric TSP algorithm (since $w_{i,j} \neq w_{j,i}$). A message passing method to find solutions for symmetric TSP problems has been proposed in [10]. As part of the present research, this method has been extended for asymmetric TSP problems. However, before addressing the algorithm in subsection V.*C.*, we outline the H-K conditions below (V.*B.*).

B. H-K Validity Conditions

We restrict our attention to a proper subset $\mathcal{E} \subset \mathcal{C} \times \mathcal{C}$ of edges, so that $\mathcal{T} \subset \mathcal{E} \subset \mathcal{C} \times \mathcal{C}$. The edge-neighborhood of any exemplar node $i \in \mathcal{C}$, is denoted as ∂i, is the subset of edges in $\mathcal{E}$ that connect i,

$$\partial i \triangleq \{e | e \in \mathcal{E}\}. \quad (24)$$

Similarly, for every subset $\mathcal{S} \subset \mathcal{C}$ of nodes, its edge-neighborhood $\partial\mathcal{S} \subset \mathcal{E}$ is defined as the subset of edges with only one node in $\mathcal{S}$ and the other in $\mathcal{N} \backslash \mathcal{S}$,

$$\partial\mathcal{S} \triangleq \{(i,j) | (i,j) \in \mathcal{E}, i \in \mathcal{S}, j \in \mathcal{C} \backslash \mathcal{S}\}. \quad (25)$$

For example (Fig. 4), the edge-neighborhood of $\mathcal{S} = \{k, l, n, q\}$ is $\partial\mathcal{S} = \{(j,k), (k,m), (m,n), (h,q), (q,o), (p,q)\}$. Note that the nodes in $\mathcal{S}$ need not form a fully connected sub-graph. This is possible when $|\mathcal{E}| < |\mathcal{C}|^2$.

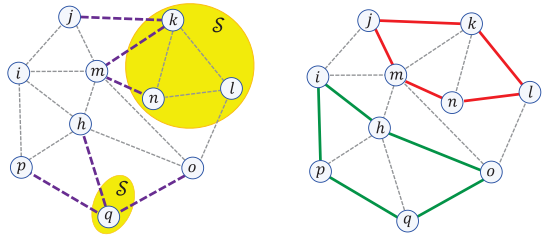

Fig. 4. Examples. Edge-neighborhood (left) and subtours (right).

There are two types of H-K conditions for symmetric weights, the degree conditions, and the subtour conditions. These validity conditions, which must be satisfied by any valid tour, are introduced below.

(*i*) Symmetric degree conditions: $\mathcal{T}$ must contain exactly two edges from the neighborhood ∂i of each node i,

$$\forall i \in \mathcal{C}, |\partial i \cap \{j | (i,j) \in \mathcal{T}\}| = 2. \quad (26)$$

(*ii*) Subtour conditions: The tour $\mathcal{T}$ must not contain any cyclic subtour with nodes in $\mathcal{S} \subset \mathcal{C}$. This is possible iff the following condition is satisfied for every subset $\mathcal{S} \subset \mathcal{C}$,

$$\forall \mathcal{S}, |\mathcal{T} \cap \partial\mathcal{S}| \geq 2. \quad (27)$$

Suppose a cyclic subtour $\mathcal{T}'$ indeed exists in $\mathcal{T}$. Let $\mathcal{S}$ be the set of nodes whose edges are in the cyclic subtour, so that $\mathcal{S} = \{i \in \mathcal{C} | (i,j) \in \mathcal{T}'\}$. Assuming that none of the degree

constraints are violated, there will be no edge in $\mathcal{T}$ that connects $\mathcal{S}$ with $\mathcal{C}\backslash\mathcal{S}$. Under such circumstances, $|\mathcal{T} \cap \partial\mathcal{S}| = 0$. Fig. 4 illustrates what happens when a cyclic subtour (red lines) exists. Its nodes are in $\mathcal{S} = \{(j,k),(k,l),(l,n),(n.m),(j,m)1\}$. There is another such cyclic subtour (green lines) in the same figure.

C. Extension for Asymmetric Weights

A min-sum method of quadratic complexity has been proposed in [12] for symmetric TSP problems. The H-K conditions are imposed through factor nodes in the FGM.

This research considers asymmetric weights where the energy required to go from way point i to way point j is different than the energy required to go from j to i (i.e. $w_{i,j} \neq w_{j,i}$). This implies that the neighborhood ∂i of each node i can be divided into two subsets, one containing the incoming edges and the other one the outgoing edges. Let the symbols ∂i^{in} and ∂i^{out} denote the incoming and outcoming edge-neighborhoods of each node i, so that,

$$\begin{cases} \partial i^{\text{in}} \triangleq \{(i,j)|(i,j) \in \mathcal{E}\} \\ \partial i^{\text{out}} \triangleq \{(i,j)|(i,j) \in \mathcal{E}\} \end{cases}. \tag{28}$$

The new validation condition can be defined as follows.

(*iii*) Asymmetric degree conditions: For an asymmetric tour $\mathcal{T}$ to be valid, it must contain exactly one edge from each of these subsets.

$$\forall i \in \mathcal{C}, \begin{cases} \left|\partial i^{\text{in}} \cap \{j|(i,j) \in \mathcal{T}\}\right| = 1 \\ |\partial i^{\text{out}} \cap \{j|(j,i) \in \mathcal{T}\}| = 1 \end{cases}. \tag{29}$$

This new validity condition shown in (29) replaces that of the original H-K condition in (26).

D. Tour Optimization by Message Passing

There are up to $|\mathcal{C}|(|\mathcal{C}|-1) \approx |\mathcal{C}|^2$ variable nodes in the FGM. The corresponding binary variables are,

$$v_e = \begin{cases} 1, & e \in \mathcal{T} \\ 0, & \text{otherwise} \end{cases}. \tag{30}$$

The extended H-K degree conditions are imposed through the following set of factor nodes,

$$D_i(v_{e\in\partial i}) = \begin{cases} 0, & \begin{cases} \sum_{e\in\partial i^{\text{in}}} c_e = 1 \\ \sum_{e\in\partial i^{\text{out}}} c_e = 1 \end{cases} \\ \infty, & \text{otherwise} \end{cases}. \tag{31}$$

There is a factor node that associated with the function $D_i(\cdot)$ for every point $i \in \mathcal{C}$. Thus, unless curtailed by some means, there can be up to a total of $|\mathcal{C}|$ such nodes in the FGM.

The subtour conditions are imposed by means of another sets of factor nodes. For every $\mathcal{S} \subset \mathcal{C}$,

$$S_{\mathcal{S}}(v_{e\in\partial\mathcal{S}}) = \begin{cases} 0, & \sum_{e\in\partial\mathcal{S}} c_e \geq 2 \\ \infty, & \text{otherwise} \end{cases}. \tag{32}$$

In this way the factor nodes associated with the function $D_i(\cdot)$ for every point $i \in \mathcal{C}$ can be up to a total of $2|\mathcal{C}|$. The messages passed from variable nodes should be modified properly. Subtours nodes need no such modification.

Hypothetically speaking, the total number of such factor nodes in the FGM can be up to $2|\mathcal{C}|!$. Fortunately, practical considerations and heuristics can be leveraged effectively to reduce this size to be within tractable limits. In fact, it can be dramatically lowered to be only of quadratic order as discussed in [11]. In this research, this can be accomplished by excluding edges $(i,j) \in \mathcal{C}$ that connect pairs of distantly located way points that are highly unlikely to be incorporated in any optimal tour $\mathcal{T}$. An upper bound of $(2+\alpha)\rho$ for the distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ can be used as a heuristic, so that,

$$\mathcal{E} = \{(i,j) \in \mathcal{C}|\|\mathbf{x}_j - \mathbf{x}_i\| \leq (2+\alpha)\rho\}. \tag{31}$$

The quantity $\alpha \in [0, \alpha_{\max}]$ in (23) is a suitable algorithmic constant, with $\alpha \ll 1$ being quite effective. In this manner the tour optimization can be of polynomial time complexity.

E. Robot Path

The optimal tour $\mathcal{T}$ is retrieved from the above message passing algorithm in a straightforward manner,

$$\mathcal{T} = \{e \in \mathcal{E}|v_e = 1\}. \tag{32}$$

However, it must be noted that edges in $\mathcal{T}$ merely connects way points in $\mathcal{C}$. Every edge $(i,j) \in \mathcal{T}$ is a geodesic approximation. The intermediate points in the path $i \rightsquigarrow j$ can be obtained from the Dijkstra's shortest path algorithm as discussed in Section II. The total energy required by the robot is the sum of the weights of the edges in $\mathcal{T}$,

$$E^* = \sum_{(i,j)\in\mathcal{T}} w_{i,j}. \tag{33}$$

VI. Results & Conclusion

All algorithms described in this article were implemented in the MATLAB® environment.

A. Regression Models

The traction data used to develop the function approximators were based on field studies (see Fig. 5). The ranges of each field in the traction data are provided in Table I. For this study, samples with power number $n_p > 20$ were discarded as they were beyond the robot's capacity. The selected data included a total of 1288 samples, which were randomly divided into 1032 (80%) training samples, 128 (10%) test samples, and the remaining 128 (10%) were the validation samples.

Fig. 5. Robot Traction Data Collection. This data was used to train function approximators of the robot performance.

Regression models were developed to implement the functions $f_{TRR}(\cdot)$ and $f_{PN}(\cdot)$ in (1) and (2). The data used to train the models was obtained from [5]. An earlier set of experiments with neural networks revealed that although they could be successfully trained in terms of performance metrics, the function gradients $\nabla f_{TRR}(\cdot)$ and $\nabla f_{PN}(\cdot)$ were numerically found to be unacceptably high for some input values. Moreover, an analysis of the Hessian $\nabla^2 f_{TRR}(\cdot)$, indicated the presence of local minima, thereby rendering neural networks unsuitable for numerical optimization. Regularization was ineffective in addressing this issue.

Consequently, an alternate approach using support vector machines [13] was adopted. Two separate support vector machines were trained in dual form to estimate ρ and n_p as functions of θ, v, and F as in (1) and (2). Both models incorporated Gaussian kernels. The cost functions were formulated as ε-losses. LASSO weight regularization terms were added to these losses. The dual formulation of the overall optimization problem was directly solved using quadratic programming with ε-constrains. Support vector regression was suitable for function minimization in (3), which was implemented using trust region optimization.

TABLE 1. ROBOT PERFORMANCE DATA

Field	Slope	Speed	Drawbar pull	Power number	Travel reduction ratio
Symbol unit	θ deg	v m/min	F N	n_p	γ
min	0	1	0	0.77	0.02
max	±18	5	1500	20	1.00

B. Path Planning Algorithm

Although the proposed message passing algorithms are intended for physical realization in a sensor network and to obtain optimal paths for the agricultural robot shown in Fig. 5, the results in this article are limited to simulations studies. The message passing steps in [12] can be simplified using an earlier version that was originally suggested in [14]. The results reported here apply this version, which is called affinity propagation in the existing machine learning research literature.

C. Analysis with Random Points

A total of N random points were generated. The points were distributed uniformly within the square region $[1,21]^2$. The weights were also randomly generated.

Due to the earlier reason (VI.*B*), all execution times reported refer to the sequential execution of each increment. In other words, message passing between pair of nodes would be carried out without any concurrency. The algorithmic constant α used in (31) was set to $\alpha = 18$ so that the worst case performance can be analyzed. This analysis helped the authors of this article to estimate the worst case limitations of the planned hardware implementation. The coverage radius was fixed at $\rho = 8$. Fig. 6 shows the growth in the worst case sequential execution time with increasing N. Of significance was the observation the empirical growth is of polynomial order (instead of $O(Ne^N)$). This was a (highly) desired outcome for planned future research.

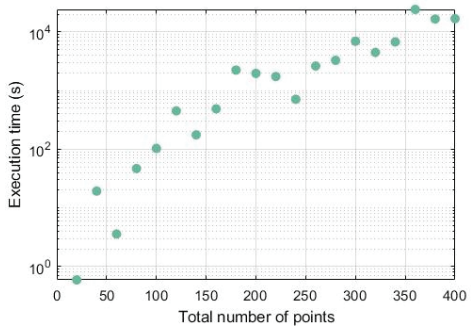

Fig. 6. Sequential Times. The worst case times for sequential execution of the algorithm as a function of N.

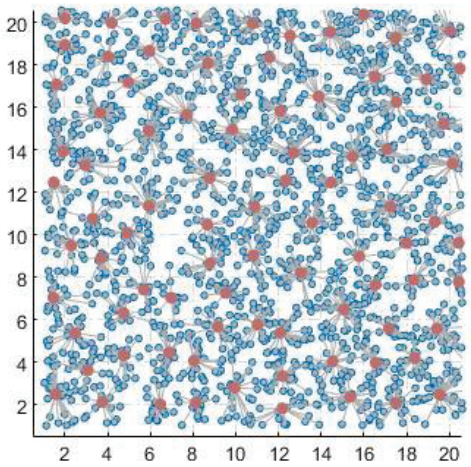

Fig. 7. Exemplar Clustering. The $N = 2000$ points (blue circles) and $|\mathcal{C}| = 67$ exemplars/way points (red circles).

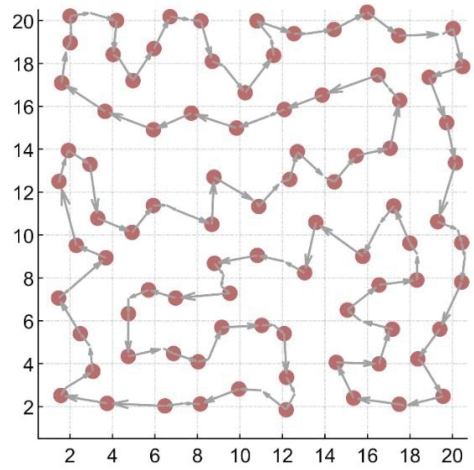

Fig. 8. Optimal Route. The circular path connecting $|\mathcal{C}| = 67$ exemplars/way points.

Fig. 7 shows the outcome of affinity propagation with $N = 2000$. A total of $|\mathcal{C}| = 67$ exemplars were obtained from the clustering. The optimal path planning step was applied using the exemplars as way points. The optimal tour is shown in Fig. 8.

D. CICO Park

Elevation data from CICO park, a public park located in the city of Manhattan, Kansas, USA, was used as a platform for the proposed approach. The data was in the form of a 170×110 lattice $\mathcal{L}$ of elevations sampled at regularly spaced intervals. The total area was 800 m $\times$ 500 m. Due to the robot's low speed, it was assumed that direction changes to not need more energy. However, sharp turns were avoided as previously discussed in Section II. Simulation results for different coverage radii are provided in Table II.

TABLE 1I. ROBOT ENERGY USAGE IN CICO PARK

Variable	Symbol	Values			
Coverage radius (m)	ρ	25	30	35	40
Number of way points	$\|\mathcal{C}\|$	301	210	202	64
Energy used (N-m)	E^*	2.30×10^7	1.77×10^7	1.28×10^7	8.54×10^6

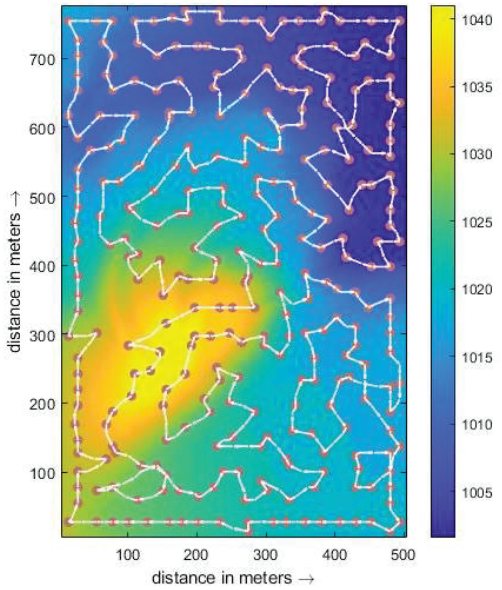

Fig. 9. Optimal Robot Path in CICO Park. Vertical bar (right) shows elevation color scheme. There were 301 way points (red circles). Note that presence of occasional intersections in the optimal path, which is due to large asymmetric weights, a rare occurrence in the function approximators' outputs, but independently verified to be shortest path segments of the robot's cyclic path.

Fig. 9. shows the minimum energy path of the robot for a coverage radius of $\rho = 25$ m. The clustering algorithm was used to obtain a total of $|\mathcal{C}| = 301$ way points. The way points along the optimal tour were connected with the Dijkstra algorithm. A total of 1,000 subtour constraints were generated. Using a fixed relaxation rate of $\lambda = 0.2$ to update the values of the FGM's nodes, the number of iterations required by the proposed algorithm was restricted to 1600.

E. Future Research

Subsequent research (planned by the authors' research team) are along multiple directions. These are outlined below.

Theoretical bounds would be established for the values of α (in terms of the spacing between adjacent rows/columns in the lattice, as well as coverage radius ρ) to use the least number of factor nodes in the FGM, and consequently also the computational complexity of the optimal tour algorithm, beyond what was suggested in [12] with guaranteed full coverage of the lattice.

The computational complexities of the steps of the proposed coverage path planning method will be established with concurrent message passing between sensors in a locally connected network. This would be accompanied by analytic expressions for their expected values.

The following three assumptions will be made: (*i*) the lattice is a closed, bounded, dense (and possibly convex) subset of $\mathbb{R}^2$, (*ii*) the geodesic distances are exact, and (*iii*) the message passing algorithm can find the optimal TSP tour. Under these assumptions, a formal proof of optimality would be established. Specifically, it will be mathematically shown that any infinitesimal perturbation of an arbitrary way point would result in either (*i*) the loss of full coverage, or (*ii*) an infinitesimal increase in the energy needed by the robot.

The research team currently has recently developed other ground robots that are identical to the original one in Fig. 5. Thus, the coverage path planning approach would be extended to find multiple, disjoint optimal paths for full coverage. A new constraint would be incorporated to ensure that the robots do not use more energy than the battery capacity.

The proposed coverage path planning algorithm will be tested in the real world. Field experiments would be carried out by deploying the robot shown in Fig. 5 in the field. Measurements would be collected from field trials. This data would serve to further validate the theoretical methods.

Moreover, techniques to fine-tune the robot's path in real time, and as the robot streams new data *en route*, will also be investigated. Such enhancements can be incorporated into the distributed message-passing algorithm.

At the present time, the authors' are experimenting with multi-agent reinforcement learning (cf. [15]) to refine the multirobot team's planned path in real time.

Lastly, an additional element would be inserted into the coverage path planner. This would involve a UAV, which would be used to replenish the ground robots with seeds. Thus, the overall seed weight carried by the ground robots would be lowered. This would reduce the net energy requirement. Further into the future, multiple UAVs would be considered.

The ultimate goal of this project is to create a heterogeneous team of ground robots and UAVs, to seamlessly carry out agricultural tasks in an automated manner. This research would extend food production to terrains with steep slopes. The authors hope that in the coming decades, these efforts would have a positive impact on the global food supply.

REFERENCES

[1] D. Smith, *Physical geography of Kansas*, Emporia State University, 2012.

[2] E. Galceran, and M. Carreras, "A survey on coverage path planning for robotics," in: *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258-1276, 2013.

[3] L. C. Santos, F. N. Santos, E. J. Solteiro Pires, A. Valente, P. Costa, and S. Magalhães, "Path planning for ground robots in agriculture: a short review," in: *Proceedings*, *IEEE International Conference on Autonomous Robot Systems and Competitions*, pp. 61-66, 2020.

[4] K. R. Jensen-Nau, T. Hermans, and K. K. Leang, "Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring," in: *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1453-1468, 2021.

[5] C Badgujar, D. Flippo, E, Brokesh, S. M. Welch, "Experimental investigation on traction, mobility and energy usage of a tracked autonomous ground vehicle on a sloped soil bin," in: *Journal of the ASABE*, vol. 65. no. 4, 2022.

[6] C. Badjujar, S. M. Welch, and D. Flippo, "Artificial neural network to predict traction performance of autonomous ground vehicle on a sloped soil bin and uncertainty analysis," in: *Computers and Electronics in Agriculture*, vol. 196, 2022.

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," in: *Science*, vol. 290, pp. 2319-2323, 2000.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," in: *IEEE Transactions on Information Theory,* vol. 47, no. 2, pp. 498-519, 2001.

[9] J. M. Mooij, and H. J. Kappen, "Sufficient conditions for convergence of the sum–product algorithm," in: *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4422-4437, 2007.

[10] I. E. Givoni, and B. J. Frey, "A binary variable model for affinity propagation," in: *Neural Computation*, vol. 21, no. 6, pp. 1589-1600, 2009.

[11] M. Held, and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," in: *Operations Research*, vol. 18, no. 6, 1970.

[12] S. Ravanbakhsh, R. Rabbany, and R. Greiner, "Augmentative message passing for traveling salesman problem and graph partitioning," in: *Advances in Neural Information Processing Systems*, vol. 27, pp. 289-297, 2014.

[13] F. Zhang, and L. J. O'Donnell, "Support vector regression," in: *Machine Learning*, Academic Press, pp. 123-140, 2020

[14] B. J. Frey, and D. Dueck, "Clustering by passing messages between data points." in: *Science*, vol. 315, no. 5814, pp.972-976.

[15] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in: *Handbook of Reinforcement Learning and Control* (Eds: Vamvoudakis, Wan, Lewis, and Cansever), Springer, pp. 321-384, 2021.