



Realtime optimization and management system (ROAM): A decision support system for digital agriculture systems

Shiang-Wan Chin ^{a,*}, Gloire Rubambiza ^a, Yifan Zhao ^a, Keyvan Malek ^b, Hakim Weatherspoon ^a

^a Cornell University, Ithaca, NY, USA

^b University of Illinois Urbana-Champaign, Champaign, IL, USA

ARTICLE INFO

Keywords:

Digital agriculture
Decision making under deep uncertainty
Systems optimization
Systems engineering
Internet of things
Sustainability

ABSTRACT

The growing disparity between food supply and demand requires innovative Digital Agriculture (DA) systems to increase farm sustainability and profitability. However, current systems suffer from problems of complexity stemming from the challenge of integrating diverse, often non-interoperable hardware and software components. In order to tackle these complexities to increase farm efficiency and understand the tradeoffs of these new DA innovations we developed Realtime Optimization and Management System (ROAM), which is a decision-support system developed to find a Pareto optimal architectural design to build DA systems. To find the Pareto optimal solution, we employed the Rhodium Multi-Objective Evolutionary Algorithm (MOEA), which systematically evaluates the trade-offs in DA system designs. Based on data from five live deployments at Cornell University, each DA design can be analyzed based on user defined objectives and evaluated under uncertain farming environments with ROAM. Paired with this, we develop a web interface that allows users to define personalized decision spaces and visualize decision tradeoffs. To help validate ROAM, it was deployed to a commercial farm where the user was recommended a DA architecture design method to increase farm efficiency. ROAM allows users to quickly make key decisions in designing their DA systems to increase farm profitability.

1. Introduction

The 2018 Global Agricultural Productivity (GAP) index highlights a growing disparity between food supply and demand, for both developed and developing countries [1]. Conservative estimates predict that agricultural production will need to increase by 25-70% above current levels to meet the demand expected by 2050. As a result, the world is likely to face a large-scale food security crisis [1]. A major challenge to increasing food production is farm efficiency which is challenged by limited rural infrastructure [2].

Digital Agriculture (DA), which is the use of data-driven techniques to increase farm productivity and sustainability, is thought of as a method of addressing the crisis [3]. Research into data-driven agriculture is growing. It envisions a future in which on-farm data collection, processing, and transmission are ubiquitous [4]. Several start-up companies are developing applications for data-driven farms [5], while major agribusiness firms are developing data collection and processing systems [5].

According to Douthwaite et al., DA innovations are complex and require involving farm stakeholders to understand their goals and constraints to successfully deploy [2]. First, current DA solutions are often fragile due to non-interoperable hardware and software [6]. Second, DA solutions often take a generalized approach that is not suitable for the myriad of farmers, each of whom has unique demands and constraints that require personalized solutions; e.g. a specialty grape farm can focus on achieving a specific taste profile while a row crop corn farm can focus on optimizing yield [3]. These challenges often lead to low understanding, slow adoption, and high costs in implementing DA systems [2].

1.1. Digital agriculture systems

According to Nemes, farms are agricultural systems that are typically used to grow a crop to be sold for profit [7]. Farmers are the main stakeholder as they typically own and/or operate the farm [8]. According to the United Nations, a farmer can be defined as a person who

* Corresponding author.

E-mail addresses: sc2983@cornell.edu (S.-W. Chin), gloire@cs.cornell.edu (G. Rubambiza), yz348@cornell.edu (Y. Zhao), k1malek@illinois.edu (K. Malek), hweather@cs.cornell.edu (H. Weatherspoon).

<https://doi.org/10.1016/j.atech.2024.100452>

Received 24 January 2024; Received in revised form 28 March 2024; Accepted 5 April 2024

Available online 20 April 2024

2772-3755/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

cultivates land, raises crops, and/or livestock for sustenance or commercial purposes.

Digital Agriculture (DA) is the use of data to improve farm decision-making which can lead to increased environmental sustainability and farm profitability [3]. DA is composed of sensing, storing, computing, and actuating technologies that leverage on-farm data [7]. Gathering massive amounts of sensor data requires a robust network, but this is a challenge as farms in rural areas often have limited or no on-farm networking or Internet access [8]. A Network-Enabled Farm (NEF) addresses these issues by using new technologies or old technologies repurposed to provide networking capabilities in the middle of a farm such as 4G LTE, Long Range Radio (LoRa), and unlicensed TV White Spaces (TVWS) [9]. A NEF uses these capabilities to sense, transmit, and analyze farm data to produce actionable insights for farm stakeholders, as described in Seamless Visions, Seamlful Realities: Anticipating Rural Infrastructural Fragility in Early Design of Digital Agriculture [6]. A NEF provides the networking infrastructure to enable data-driven DA to optimize farm management. Integrating advanced digital technologies, as explored by Gebresenbet et al. (2023), can further enhance the efficiency and sustainability of these agricultural systems [10].

A NEF is a modular abstraction of software and hardware technologies that are designed to fit the various needs of farmers. The software abstraction is split into three modules: Sensing, Computing, and Actuating. The Sensing module abstracts away sensors that allow different hardware sensors to be connected through software. The Computing module allows for different analytic algorithms to be run to support decision-making. The Actuating module performs some type of action such as releasing irrigation valves. These modules can connect manufacturer-agnostic hardware devices such as computers located at the farmhouse, field sensors, and water valves. With both the software and hardware connected, farmers can visualize aggregate data from normally incompatible farming systems on a web application interface [11]. To gain operational insights, farmers can run analytics on their data to make farm decisions. Lastly, a NEF enables the creation of digital twins of the physical farming system to automate farm processes such as precision irrigation.

Our goal, then, is to address the gap in balancing utility, risk, and uncertainties when building DA systems. Our approach includes integrating systems engineering principles with our experience designing and deploying NEFs. Specifically, Realtime Optimization and Management System (ROAM) employs a methodology that evaluates Pareto optimal decisions for farmers. The result has been the design and implementation of ROAM, a decision-support tool designed to empower farmers with insights into the tradeoffs involved in constructing DA systems. ROAM facilitates holistic analysis, real-time simulation, and uncertainty and risk modeling. In the next sections we compare ROAM to other approaches, discuss the design of ROAM in detail, our results, and the contributions of applying ROAM to a commercial farm in California.

1.2. Related works

In recent years, there has been a growing recognition of the complexities involved in decision-making tradeoffs within agricultural systems and in particular for Pareto optimal approaches. Jones et al. have highlighted the intricacies of these tradeoffs and emphasized the need for leveraging new networking technologies to explore research opportunities in this domain [12]. Historically, agricultural systems research has often relied on non-real-time data and indirect farm data in modeling and simulations, neglecting real-time farmer preferences and decision-making tradeoffs [13]. However, the advent of advanced networking capabilities, such as the NEF system, has paved the way for a new generation of models that integrate real-time farm data, thus offering valuable insights into decision-making processes [12].

The current state of the art in decision support systems (DSS) for agriculture reflects a divide between non-real-time simulation and

modeling and those incorporating uncertainties. Non-real-time models, while valuable in understanding decision-making under risk and utility, have traditionally relied on frameworks like Subjective Expected Utility Theory (SEUT) to assess decision outcomes [14]. SEUT provides a systematic approach to analyze decision alternatives based on expected utility and risk, yet it falls short in capturing real-time decision dynamics and unique decision contexts [15].

On the other hand, modeling with uncertainties has gained traction, especially concerning risks induced by factors like climate change. This segment of research emphasizes the adaptation of management strategies to evolving uncertainties, utilizing bio-economic and bio-decision models to optimize decision processes [14]. Collaborative efforts between climate specialists and agricultural scientists have yielded climate change models that assess the impact of uncertain climate conditions on crop yields, underscoring the importance of interdisciplinary collaborations in addressing complex agricultural challenges [14].

However, despite these advancements, existing research still faces challenges in adequately addressing utility, risk, and uncertainties to recommend Pareto optimal decision-making strategies. The insufficiency of current models in quantitatively capturing real-time decision-making processes has been discussed, with calls for methods to model changing uncertainties effectively and assess decision support systems' efficacy in addressing these challenges [7,12]. Additionally, while some progress has been made in sequential decision-making modeling using ensembles of crop and climate models, this area of research is still nascent [16].

In summary, the literature underscores the need for advanced decision support systems that merge systems engineering with real-time farm networking capabilities, enabling farmers to make informed decisions based on up-to-date data and addressing the evolving challenges in agricultural systems management.

1.3. ROAM

In this paper, we present the Realtime Optimization and Management System (ROAM), a tool that identifies a Pareto optimal set of architectural decisions for farmers to build their own NEF. ROAM determines a Pareto front of optimal DA system architectures a farmer can choose between, usually eliminating the vast majority of potential architectures. Thus, ROAM potentially advances the state of the art in deploying DA systems. It performs up-front analysis necessary to deploy DA systems and eliminates major barriers to the diffusion of DA techniques into real-world farms and increasing farm efficiency.

The design of ROAM is based on formalizing a method to evaluate a DA architecture by encoding user generated evaluation metrics and uncertainties to assess a set of architectural decisions. An architectural decision is the choice between different components of the DA system such as between a soil moisture or light sensor. Then, the ROAM Configuration File is used to create nodes or objects that represent unique architectural configurations of a DA system. An architectural representation is a subset of architectural decisions made to create a NEF system. Next, the nodes are passed into an optimization function to uncover a Pareto optimal set of architectural representations most suitable to a user's need. To abstract away the complexity of the ROAM implementation, a front-end user interface is designed and used to allow for easy input of key features of the user's farm, constraints, and uncertainties. This frontend creates the ROAM Configuration File used for ROAM evaluation. In addition, as output, the frontend displays interactive 3-D data visualizations of the farmer's potential DA system tradespace, which is then used to allow for better understanding of the recommendations of the system. The entire process from beginning to end, from encoding the ROAM Configuration File to the end step of visualization of the analysis is modularized to allow for swapping in and out interchangeable software. For example, different types of optimization models can be used for ROAM.

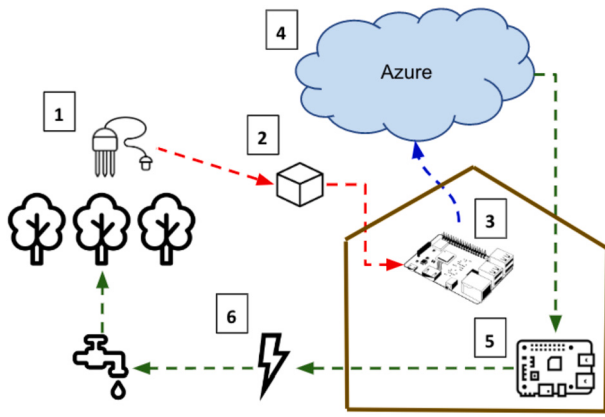


Fig. 1. NEF decision space.

1.4. Contributions

ROAM is a tool composed of various algorithms and optimization methods based on systems engineering principles. We validate the utility of ROAM through our own field trials at Cornell University and through an application on Cheng Xin Garden LLC (CXG), a commercial California-based viticulture farm. For the CXG deployment, ROAM identified a Pareto optimal set of architectures and ultimately assisted the user in selecting one architecture that increased the CXG farm efficiency while accounting for constraints and uncertainties. To summarize our work makes the following research contributions:

- (1) A framework, model, and software system tool for understanding decision making when building network-enabled farms
- (2) Design and implementation of ROAM
- (3) Validation of the utility of ROAM through a commercial farm deployment

2. Materials and methods

2.1. Data collection

We implemented and deployed several NEF instances, including an apple orchard, corn and hemp greenhouse, dairy cow farm, and vineyard [17]. These instances of NEF deployments utilized research farms associated with Cornell University and were implemented between August 2020 to July 2023. We have collected millions of sensor readings for our crop NEFs and tens of gigabytes of data from our livestock NEF at the time of writing. These deployments highlight both the flexibility of the NEF concept, as well as the importance of tailoring each deployment to fit the needs of each individual farm. Fig. 1 illustrates a common digital agriculture system framework, emphasizing the novelty that lies in the mass customization possible for specific components like sensors and networking. This customization aspect is fundamental to ROAM, which tailors solutions to unique farm needs. A NEF uses cutting-edge networking approaches and technologies such as TV White-space (TVWS), LoRa, and sensors such as in situ plant water sensors [17] (See Fig. 1). The specific hardware and software of these deployments are described in Comosum: An Extensible, Reconfigurable, and Fault-Tolerant IoT Platform for Digital Agriculture [17]. Fig. 1 shows a data-driven irrigation graphic of how a NEF connects the Sensing Module through a (1) sensor, (2) sensor box, and (3) subedge or edge computation device to the (4) Computing Module through a cloud software service to the (5) Actuating Module with a raspberry pi [18] and (6) actuation function.

In addition, we conducted a stakeholder analysis by interviewing 11 farmers in California, Washington, and New York. In these interviews, we identified cost, performance, and risk as objectives farmers

use to evaluate new technology investments. The farmers we interviewed expressed sensitivity to decisions that affected these objectives and through our analysis we understood variations across different possible NEF architectures using principles in system engineering.

To validate ROAM on a commercial farm, it was used on Cheng Xin Garden LLC (CXG). CXG is a 120-acre wine grape commercial viticulture farm in Bakersfield, California. Due to limitations in California's water supply caused by frequent droughts and forest fires, CXG was seeking to increase their farm efficiency, but was constrained by the lack of knowledge on available technologies and the ability to envision the results from adopting a NEF. As part of the process, CXG used ROAM to consider different decisions necessary to create a NEF that could meet their needs. The process required in-depth user interviews where the results were then encoded as input to ROAM, and used to evaluate the output. The goal was to present a set of Pareto optimal architectures that could increase CXG's farm operational efficiency while accounting for constraints and uncertainties.

After four weeks of interviews with the farm owner, we holistically understood the current situation, needs, and challenges of CXG's farming practices and created a configuration for their viticulture farm. Through this process, CXG shared farm data that they had collected over the course of six years. For example, CXG had data that showed how a manual water tensiometer saved the farm 20% of water usage and their cost of water was \$100 per day in California, which can vary from \$50-\$200 per day [19]. With this data, we created a tradespace configuration file for CXG to be inputted into ROAM.

2.2. Decision tradeoffs

To effectively analyze the collected data, we use approaches and techniques from Systems Engineering (SE). The tools in SE include a range of quantitative and qualitative techniques that deepen our understanding of trade-offs [20]. These SE techniques enable the evaluation and generation of a Pareto optimal set of decisions when navigating complex systems [3]. Pareto optimal is a state in which resources within a system are optimized such that improvement in one dimension would lead to deterioration in another [21].

Our first SE approach to analyze NEF systems uses systems thinking [22]. It is both holistic and reductionist and aimed at comprehending and analyzing complex systems, along with the relationships among their parts [22]. For example, in agriculture, it can be helpful to understand the elaborate interactions among different components of an agricultural system, such as the interconnections between crops, soil, water, and weather [15]. By viewing the system as a whole initially and then honing in on specific parts, it becomes possible to identify the main leverage points to boost the system's performance and resilience, thereby cultivating more effective and sustainable solutions.

2.3. ROAM architecture

ROAM is an open-source software system. It includes a client-side browser-based interactive application and a server-side back-end service. ROAM is designed and developed in a back-end and front-end setup due to the need for computational resources and data storage in the back-end, as well as the need for a user-friendly interface to lower technology barriers for our users. The server-side back-end is developed with Python as the core programming language and hosts most functionalities, including optimization, analytics, and data storage. We selected the Python Flask framework to develop the client-side web application with Javascript as a core programming language. Both the back-end service and the front-end application integrates functionalities from multiple external libraries and custom modules.

The system consists of four main modules: the Decision, Rhodium, Uncertainty, and Graphical User Interface (GUI) modules as seen in Fig. 2. The Decision module defines and maintains the tradespace architecture from the Tradespace Configuration File and it hosts the

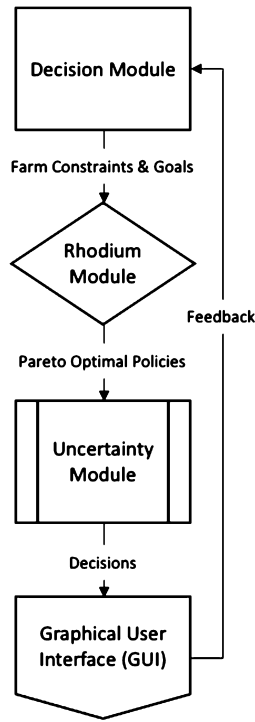


Fig. 2. System modules.

Tradespace Enumeration and Optimization algorithms. The Uncertainty module defines the uncertainty variables and models uncertain farming environments using historical data. The Rhodium module hosts functions responsible for the extension and orchestration of the integrated third-party Many-Objective Robust Decision Making (MORDM) libraries and provides key analysis of the tradespace. The GUI hosts the front-end interface and handles user data acquisition and visualization.

In the following sections we will describe how ROAM computationally models and analyzes decision tradeoffs to recommend a set of Pareto optimal NEFs.

2.3.1. Tradespace model

To model and evaluate NEF designs, we draw from the study of systems architecture within SE for developing configurable complex systems and evaluating how well they satisfy stakeholder needs [23]. To decompose a complex system, we formulate a systems architecting optimization problem that represents a complex architecture as a set of decisions using an encoding scheme [23]. Generally, optimization problems that result from decisions in systems architecture are combinatorial. To treat programmed decisions analytically we segment the decisions into six canonical decision classes: standard form, assigning, partitioning, permuting, downselecting, and connecting [23]. These patterns are interlinked and have some overlap, so we can think of the six classes as combinations of standard form and down-selecting decisions.

The standard form (SF) decisions are decisions in which a user can only select one option from a set of alternatives. When making multiple SF decisions, the number of possible combinations of decisions is given by

$$\prod_{i=1}^N m_i \quad (1)$$

where m_i is the number of alternatives for an i decision and N is the number of decisions to be made [23]. In contrast, down-selecting (DS) decisions are where a user can choose more than one alternative. The number of possible choices is given by

$$2^N \quad (2)$$

where N is the number of alternatives. The next step of creating the tradespace model is to create decisions to define the architecture space and subsequently to create objectives to evaluate the architectures. For our analyses, a NEF focuses both on pragmatic deployments of software and hardware components, so in any decision space we need to consider multiple types of decisions. Table 1 is an example of a set of decisions, their descriptions and importance, and the canonical class used to create and evaluate a NEF.

2.3.2. Problem formulation

The objective formulations and their subsequent values were informed by robust data sources, including journal publications [24][25], insights from 11 farmer interviews, and practical experience from five NEF deployments. This comprehensive data set encompasses a range of crucial factors, such as annual yield increments, production objectives, device pricing, component costs, and maintenance expenses. This approach ensures that our evaluation is not only grounded in empirical data but also reflects the real-world intricacies of agricultural systems.

Once the tradespace has been constructed, it is essential to define evaluation objectives for each architecture, as guided by literature and stakeholder analysis [26][23]. In our case, cost, performance, and risk were identified as key objectives, reflecting the priorities and sensitivities of the farmers we interviewed. This methodology, supported by principles in system architecture, allows for a nuanced understanding of variations across different architectures. The creation of value functions for each decision, akin to Crawley and Selva's "transfer function", facilitates a comprehensive evaluation, taking into account the complexity and dynamic nature of agricultural systems.

2.3.3. Formulation of objectives

To formulate the Pareto optimal investment operating policy for a given farmer we create a function composed of three objectives. A policy represents a specific set of decisions made from available choices, each affecting the farm's operational outcome. For example, selecting a water sensor and pairing it with a data storage mechanism, like a Raspberry Pi, forms part of a policy. A complete policy is thus an amalgamation of such decisions across different categories. For instance, in Table 3 a policy is composed of six decisions. From work done by Cohon and Marks, and Reed et al. we can define our multi-objective problem with a vector, $F(d)$, as demonstrated by the following equation [27][28].

$$F(d) = (F_{cost}, F_{risk}, F_{perf}) \quad (3)$$

$$\forall d \in \Omega \quad (4)$$

subject to user defined:

$$F_{cost}, F_{risk}, F_{perf} \quad (5)$$

Here d is a vector of decision variables in the tradespace Ω . These decisions can be expressed as real numbers utilizing value functions. Each $F(d)$ operating policy is evaluated based on its cost, risk, and performance which can be constrained by user input. For example, because we want a NEF system to be low cost, we can constrain cost to be less than or equal to \$2000 and it would be denoted as $F_{cost} \leq 2000$. In terms of optimization, the performance objective is maximized while the cost and risk objectives are minimized. Each objective will be explained in the following sections.

Cost objective. The first goal of the system is to minimize cost as denoted by the equation:

$$F_{cost} = H + M + S + I \quad (6)$$

The cost objective includes the cost of Hardware (H), subsequent Farm Maintenance Cost (M), Software (S), and Installation (I). We input the

Table 1

Description of canonical decisions and their importance for the architecture.

#	Decision Name	Why it is important	Importance	Justification
1	Product Information	The type of Product Information to be collected is an important decision that will also impact scalability. Animals will likely require a higher-frequency monitoring as opposed to plants.	Very High	This is a downselecting decision as we are able to decide on multiple alternatives from the initial set. Decisions range from resources that require the lowest-frequency monitoring to animals requiring the highest-frequency monitoring.
2	IoT Devices	IoT devices are a crucial decision that must be weighed between cost and functionality. The devices that are too costly will not be feasible for farmers to implement, while those that are not functional will not be able to collect robust enough data.	High	This is a standard form decision because we think that a system with more than one manufacturer would not be scalable enough to accommodate a host of users.
3	User Interface	The type of user interface is an important component that can affect performance and user attraction. The different user interfaces can provide different functions and it is where the customer can directly interact with our system, so we think it's a high priority.	Medium	We can provide multiple types of user interfaces for our users at the same time, such as a message, website, and application. These options are not exclusive to each other.
4	Systems Architecture	The possibility of scaling is important for our system as different system architectures might rule out a growing user base in the future. Similarly, scalable architectures are likely to require more initial effort to set up the system and will only pay off with a large user base.	Medium	This decision is SF since it is formulated as picking one range from a set of options.
5	Data Type	The type of Data Storage is an important decision as it determines the security measures we intend to implement. A blockchain-based data storage would be the most secure decision which will impose constraints on the scalability of the possible user base.	Low	This is a downselection decision as we could think of a hybrid system that uses a cloud-based database and a blockchain backend in concert with each other. A CSV based backend would have the smallest amount of dependencies but would likely lack scalability and performance.
6	Data Collection	One important process in our system is Data Collection from the user side. There are multiple ways we can do them, each method can strongly affect our system architecture and performance. For example, if we choose manual input, then we need to consider a model for human labor. The options are flexible since the method of collecting data does not block our system performance.	Low	Since our system has multiple components for data collecting, such as measuring temperature, tracking product information. Some of them can be automatic, while some of them have to be manual. We can have manual, automatic, or semi-automatic.
7	Data Storage	The data storage size is used to limit our capacity for storing our product information, user account information, and some intermediate data. The scale of our storage size determines our project scale and server stress.	Low	We consider this decision as SF since the options are exclusive to each other, we can only choose one from them.
8	Notification System	This is a process that is crucial for the functionality of the system. In order for the stakeholders in the network to receive value, they must be able to interface with the system.	Low	We can see this as a down selecting decision as a subset of alternatives would be possible such as Email and real-time display simultaneously.
9	File Exchange Type	File exchange types that are streamlined will allow the system to run more efficiently. If they are not, then the processing time will increase.	Very Low	This is a standard form decision as a system with more than one file format would be very fragile with respect to ensuring data consistency.
10	Machine Learning	Machine learning model allows us to make predictions on yield, risk, weather, etc.	High	This is a standard form decision since it takes too long to make a prediction; at present, we can try only one option.

actual brand names along with the associated costs of the hardware or software component, recognizing that prices can vary widely across different types and brands. This approach ensures a precise cost estimation tailored to specific farm requirements and budgetary constraints. For example, we input the cost of a Davis Instruments soil moisture sensor to be \$85 [29]. The hardware and installation costs are vital to minimize the total costs of implementing an NEF. Farmers typically have a limited upfront budget for investments and face many costly decisions when investing in new technologies [30]. For example, given a particular user's budget, the cost of sensors may make deploying full sensor networks unfeasible in some contexts [31]. Thus, if the cost of adding sensors is too high from a particular user's perspective, they will not be implemented on farms where capital and cash reserves are a constraint. In regards to maintenance cost, unreliable sensors that need constant repair would increase the M, resulting in large labor costs for the farmer. We factor in the time needed to calibrate sensors, fix devices, clean equipment, and change batteries based on experience from deployments of sensors onto a farm [32]. If the costs to keep the systems running outweigh the benefit of optimizing the farm, it will be ineffective at helping farmers. Lastly, software costs are increasingly

important as corporations pivot to Software as a Service (SaaS) models where cost per computation is the norm. As a result, for larger farms with an abundance of sensors, computation costs, and software services will be much more expensive. It is also important to note that the type of farm, region, and climate also influence which sensors and decisions are the most suitable. For example, a soil moisture sensor is less suitable in environments where the temperature regularly drops below freezing point and the ground freezes. It is important to note that efforts were made to create a holistic cost objective, but in complex living systems such as a farm, there are many unforeseen costs.

Risk objective. The second system goal is to minimize risk,

$$F_{risk} = S + N \quad (7)$$

This equation quantifies the interruption risk of the Sensor Devices (S) and Networking (N) of an NEF design. In a deployed NEF, there are two reasons why data from sensors might be incorrect or missing. First, the sensor hardware itself can malfunction due to climate, environmental, or implementation factors. These malfunctions can lead to both gaps in

Table 2
Uncertainties problem formulation.

Uncertainty Variable	Notation	Lower Bound	Upper Bound
Climate Complexity	C	0	3
Precipitation	R	0	3*Expected Precipitation (user input)

data collection and incorrect data collection, both of which can lead to inaccurate decision support and potentially necessitate costly repairs. These risks are captured by S in the above formula. On the other hand, if the network is unreliable, even if the sensors are collecting data properly, it cannot be transmitted to edge and cloud computers. This risk is captured by N in the above formula.

Understanding S and N are important for the quality of insights a NEF can generate. As a result, if there is a great deal of interruption risk, it can be linked to a bad quality NEF architecture. In ROAM, we define interruption risk as the probability of failure in the S to send and N to transmit data packets. As a result, the range of risk is between 0% to 100%. For example, in section 3 we used input from the farmer and our deployments to quantify the risk that S and N are not able to operate. While ROAM can use default quantities for these risks determined by averaging the risks experienced by farmers in our user interview studies, we allow farmers to alternatively provide their own quantities based on their personal evaluation based on the local conditions at their farm. As systems become ever more complex with many dependencies the risk objective will be all the more important.

Performance objective. The third system goal is to maximize performance,

$$F_{perf} = Y + W + E + L \quad (8)$$

The equation above represents the utility of the system's service to users, an objective directly tied to creating value or increasing profitability for users. The performance objective is developed as a combination of Yield Increase (Y), Water Cost Savings (W), Electricity Cost Saving (E), and Labor Cost Savings (L), representing four ways in which an NEF deployment can add value for farmers. One of the primary ways in which an NEF can improve farms is by generating insights that allow farmers to grow more high-quality crops per acre of farmland. For example, a NEF can identify underperforming parts of the field and suggest how to improve them and increase yield. In addition, an NEF can improve water costs by suggesting optimal watering amounts based on sensor data such as soil moisture levels [33]. NEFs also have different electricity costs depending on the specific technologies used; for example, solar power may be cheaper than disposable batteries in the long run. Finally, NEFs can remove the need for human labor in some cases. For example, one of the farmers we interviewed during our user research described needing to hire a worker to walk the field that cost him \$128 everyday to measure soil moisture in every hectare of the farm, labor which would not be necessary for a NEF with a sensor network to measure soil moisture. For instance, if a yield increase of \$1000 and a decrease in water use by \$500 is achieved, the performance would be an increase of profit by \$1500. There is no lower bound or upper bound to performance as it can be negative and the values depend on user input. For example, in section 3 the data utilized for this metric is based on both our three years of deploying DA systems and the farmer's input. Performance was often thought about as the most important objective for our farmers in evaluating new technology investments.

2.3.4. Uncertainties

Once we establish the objectives and value functions for evaluating architectures in the tradespace, we must define the uncertainties and their effects on the various architectures within the tradespace. To improve farmers' competitiveness and extract insights from farming for decision-making, the system must be evaluated under the deeply uncertain farming environment reflecting reality. More formally, uncertainty

in the tradespace model characterizes the behavior of an uncertain factor affecting a farm as a variable [28]. The reason for having these uncertainties is to measure an architecture's performance in a variety of uncertain environment instances, which provides a more realistic evaluation of the architecture and aids the decision-making process. The following sections focus on the uncertainty variables constructed in ROAM.

Climate. The farm climate is a complex nonlinear system, where different levels of short-term climate-induced complexity may affect the performance of the farm. Climate Complexity (CC) can lead to risks of sensor malfunction and suboptimal performance of hardware devices as they operate while exposed to outdoor farming environments. For example, solar power sources can face risks of interruption in extreme weather events such as large storms. Utilizing information theory techniques, the CC uncertainty variable aims to represent an approximate proxy to analyze and predict the level of regional short-term climate variability in a given farm area. CC uncertainty is modeled using an entropy-based measurement that is referred to as SampEn. It provides a nonlinear approach for analyzing and predicting the entropy or complexity of climatic time series [34]. It is a probability measure that quantifies the likelihood that sequences of consecutive data match one another within a tolerance r and remain similar when the length of the sequences is increased by one sample. In this way, we quantify the regularity and the unpredictability of fluctuations in weather to factor into our model. In order to calculate individual farm level SampEn we use data from the Global Climate Models (GCMs) dataset [35]. The data is then processed based on the algorithm introduced in the paper Approximate Entropy and Sample Entropy: A Comprehensive Tutorial [36]. According to the SampEn calculations of climate complexity of regional meteorological data found by Shuangcheng et al. in the paper Measurement of Climate Complexity, the authors found from using random climate data that SampEn approached 0 and with fully homogeneous data that it approached 3 [34]. As a result, we use the SampEn range from 0 to 3 with a uniform distribution to model climate complexity as shown in Table 2.

Precipitation. Precipitation has been directly linked to impacting the yield of agricultural products [37]. According to Hunho et al., it is seen that increased precipitation leads to a longer growing season and higher yields which in turn becomes higher profits for the farmer. On the other hand, in a study published in the journal Global Change Biology, precipitation was detrimental to certain crop yields [38]. In the study, corn yields were reduced by as much as 34 percent during years with excessive precipitation [38]. It was estimated that between 1989 and 2016, intense rain events caused \$10 billion in agricultural loss [38].

The effects of climate change have a large impact on precipitation [37]. It was cited as a reason for the increased and unpredictable precipitation [37]. Precipitation is highly regional, so climate change is a great cause of concern for precipitation in the future as farmers will need to plan for excessive or shortages in precipitation which will affect the profitability of the farm. To model precipitation we utilize a normal distribution of historical annual precipitation and calculate the mean and standard deviation for the region of the farm area being studied. To anticipate how precipitation affects the performance of the farm, we built linear regression models based on publicly available United States datasets that correlate historical precipitation measurements with historical crop yield to represent the effect of precipitation on crop yield. To set the range we use the empirical rule which states 99.7 percent of

values lie above and below three Standard Deviations (SD) of the mean [39]. To run the model users input the geocoordinate of the farm of interest.

How uncertainties affect decision making. Understanding the impact of climatic uncertainties on digital agriculture systems is crucial for tailoring solutions to the specific environmental conditions of a farm. Factors such as CC and precipitation patterns can significantly affect the performance and reliability of DA technologies. In ROAM, these uncertainties are integrated into the decision-making process to ensure that the recommended solutions are resilient and effective under various climatic conditions. For instance, we will explore different states of the world (SOW) by assuming varied amounts of precipitation, systematically varying this key parameter over 1,000 iterations to simulate its potential impacts on agricultural outcomes. This approach allows us to examine the resilience and adaptability of policy decisions under a wide range of climatic scenarios. By integrating a detailed analysis of precipitation and CC variability, we aim to provide a nuanced understanding of how each decision fares in different SOWs. Further details on this methodology and its implications for policy evaluation will be discussed in section 2.3.5. The ROAM architecture takes into account climate-induced risks and uncertainties by evaluating how different components of the system, such as sensors and networking devices, perform under different climatic scenarios. For instance, sensor selection in ROAM is influenced by the local climate, as some sensors may be more susceptible to malfunction or degradation in harsh weather conditions. Similarly, networking technologies are assessed for their ability to maintain stable communication in the face of climate-induced disruptions. When running ROAM for a farmer, we utilize the user's latitude and longitude to identify historical weather data from their farm area to model climatic uncertainties. By simulating various climate scenarios, ROAM helps identify architectures that are not only cost-effective and performance-optimized but also resilient to uncertain climatic conditions. This approach increases the reliability that DA systems designed using ROAM are adaptable to the specific climatic realities of each farm.

Many Objective Robust Decision Making (MORDM) is a model that does optimizations under uncertainty. In creating ROAM, we leverage Rhodium to apply MORDM to ROAM to find the Pareto optimal NEF under uncertain environments (subsubsection 2.3.4) [40][41]. Rhodium is an open-source Python library developed by researchers at Cornell University [41]. Robust Decision Making (RDM) is an analytic framework that helps identify potential robust strategies for a particular problem, characterize the vulnerabilities of such strategies, and evaluate trade-offs among them [28]. MORDM is an extension of RDM to account for problems with multiple competing performance objectives, enabling the exploration of performance tradeoffs with respect to robustness [27]. We use the Multi-Objective Evolutionary Algorithm (MOEA) provided by Rhodium to optimize the Pareto set of 'policies' calculated in the Decision Module under a representative or average instance of the uncertain environment also known as State-of-the-World or SOW. Each representative instance is taken by examining a distribution and utilizing the average. The Pareto efficient policies are further explored using the uncertainty analysis functions provided by Rhodium. Finally, the sensitivity analysis provided by the SALib python library [27] is used to analyze and categorize the effect of different uncertain elements in the farming environment.

2.3.5. MORDM

Optimization. To find the Pareto optimal policy for a NEF we built a function that calculates the tradespace Pareto front. The function called 'calcPareto', is run in the Decision Module. In order for us to consider uncertainties, we use a Rhodium optimization function to find the Pareto optimal set of policies based on the performance, cost, and risk objectives (subsubsection 2.3.3). The optimization function in the Rhodium Module is a MOEA that utilizes the Non-dominated Sorting Genetic Algorithm (NSGA-II) algorithm provided by the Rhodium li-

brary [41]. Together these two optimization methods serve different purposes in exploring the Tradespace. The 'calcPareto' function in the Decision Module enumerates all possible policies solely based on the static decision configurations defined by the Tradespace Configuration File, which finds the initial optimal set of policies on paper based on prior knowledge about the decisions. The optimization function in the Rhodium Module iteratively adjusts the controlled parameters or combination of decisions while searching for the optimal set of policies under the mean SOW. This function then finds the set of policies most optimal under the characterized uncertainty model.

Scenario discovery. The scenario discovery function is used to explore and analyze the influence of uncertainties in the Pareto optimal set of 'policies' that are found by the Rhodium optimization module [41]. To run this function, first, a set of uncertainty variables are defined using the parameters and distributions on the Uncertainty model (subsubsection 2.3.4). Then, a Rhodium internal function, 'sample_lhs', is called to generate a standard 1000 SOWs through a Latin Hypercube Sample – a technique used to reflect the true underlying distribution on the uncertain parameters [41]. Each SOW consists of a combination of uncertainty variables and represents an instance of the uncertain environment. Then, the policy evaluation function is executed to evaluate each policy in the Pareto optimal set on the 1000 SOWs. The results produced from scenario discovery can be used to visualize and explore different characteristics of various Pareto policies, such that policies demonstrate tradeoffs in objectives when evaluating against uncertainties. The analysis of these tradeoffs can provide us with insights into how different system architectures may be a better fit for certain scenarios (e.g. excessive precipitation) that cause a policy to fail and be vulnerable. These tradeoffs will be further explored and conclusions can be drawn through sensitivity analysis.

Sensitivity analysis. The Rhodium library's internal implementation extended from Python's SALib is used to perform global and regional Sensitivity Analysis (SA) on modeled uncertainties which are performed to prioritize the factors (parameters) most significantly affecting the output and fix those that are not [41]. This functionality is enabled by the browser-end interface; here users can specify an objective and policy of their interest to investigate, and then the SA function performs global SA using commonly used methods. First, the Method of Morris is used to analyze which decisions are most influential to the output objectives and the effect of uncertainty variables in isolation [42]. Second, the Sobol method is used to calculate second-order and total-order indices for capturing the interactional effects between uncertainties [43]. The function can also perform one-at-a-time (OAT) or regional SA to explore each parameter in detail. In OAT SA, we fix all parameters at their default value except one [44]. For this one parameter, we then sample across its entire range and observe how the objective of interest changes.

Uncertainty input. A user is able to input their own farm uncertainties. The first step is to define the farm uncertainties and how they affect an architectural policy. If it is unknown than default values predefined within the function 'farm_approach' are used. For example, with greater precipitation, yield may increase and watering costs may decrease. The function 'setupModel' is used to allow for user input through the web interface of what the average uncertainty value will be for their farm. Once these uncertainty parameters are set, we can use the 'optimizeModel' function to run 10,000 function evaluation calls of NSGAII to calculate the optimal policy in the uncertain state of the world.

3. Results

In our study, we assisted Cheng Xin Garden (CXG) in streamlining their decision-making process by applying ROAM to navigate through

Table 3
Configuration for Cheng Xin Garden LLC.

#	Decision Name	Description	Alternative 1	Alternative 2	Alternative 3	Class	Importance
1	Water Sensor	The methods to collect water stress data	Manual Sampling	Glass Tube	Digital Sensor	SF	1
2	Humidity Sensor	The methods to collect humidity and temperature	Manual Sampling	Digital Sensor	N/A	SF	1
3	Microcontroller	The devices put in the agriculture field	FarmBeats	CR6 Datalogger	Arduino	SF	.75
4	Data Storage	The type of storage for product information and user data	Raspberry Pi	Cloud	N/A	SF	1
5	Analytics	The model for prediction or applications for analytics of water stress	Model Predictive Control (machine learning model)	On/off Control (closed loop)	Scheduling (open loop)	SF	1
6	Irrigation Controller	How to water the plants	B-Hyve Smart Hose Watering Timer	Rachio	Raspberry Pi	SF	1

Trade Space Exploration

For The Software-Defined Farm

Tradespace Exploration is a decision-support tool developed to find the optimal architectural designs for The Software Defined Farm using a Robust Decision Making framework.

This tool has the capability of identifying potential robust strategies for the architectural design, analyzing the vulnerability of each strategy, and evaluate their performance in deep uncertainties.

Parameters:

Crop Type:

Farm Area (acres):

User Amount:

SampEn:

Precipitation:

Functions:

Log:

```
*****
[ 1 ] Generating TradeSpace. Please wait for outputs...
```

Fig. 3. Generate tradespace.

a vast array of choices to become a Network-Enabled Farm (NEF). The Tradespace Configuration File (TCF) was configured to encapsulate CXG's distinct farm requirements and constraints, leading to the identification of 324 sets of potential decisions or policies. Our approach, utilizing advanced analytical techniques, progressively narrowed these possibilities to 18 Pareto optimal policies, considering the unique factors of CXG's farm environment. Through the integration of uncertainty analysis and the integration of the farm owner's preferences, we further refined these choices to two and ultimately recommended the most effective policy. This tailored solution was achieved by leveraging a combination of scenario discovery, optimization algorithms, and a deep understanding of CXG's operational context, thus demonstrating the utility of ROAM.

The following result sections will discuss our work with CXG. First, the tradespace was enumerated as shown in Fig. 4. From this enumeration, we found the Pareto front as shown in Fig. 5. Here we highlight distinct policy points within the Tradespace: those reflecting a risk-averse approach with lower performance as measured in profit as defined in 2.3.3 and cost the (risk < 0.3) and a singular, higher risk point (risk = 0.6) indicative of a risk-tolerant strategy. Despite their differences, all points are Pareto optimal, with the higher risk point correlating with increased performance, while the lower risk points manifest in reduced performance. The introduction of uncertainties in Fig. 6 further refines these optimal points, underscoring the impact of environmental factors on policy efficacy. Subsequently, we conduct a

what-if analysis through scenario discovery, examining the performance of each Pareto optimal policy under 1000 varied situations, as shown in Fig. 7. This comprehensive approach, as demonstrated in our collaboration with CXG, was pivotal for understanding decision-making dynamics in the CXG complex farming scenarios.

3.1. Generate tradespace

To test the utility of ROAM on CXG, we first inputted all of CXG's decision points to create the TCF that represented the needs and constraints of their farm environment. The TCF was created from a JSON skeleton provided by ROAM, which consists of a list of decision structures as detailed in section 2. Table 3 shows the various decision points we identified and encoded in the TCF.

After the configuration was imported, Generate Tradespace initiated the Tradespace Exploration workflow by generating the Tradespace Network and enumerating all possible policies that can be constructed based on the given configuration. In the unconstrained architecture space, there were six Standard Form (SF) decisions with three alternatives each and two SF decisions with two alternatives. As a result, ROAM found 324 possible decisions in CXG's tradespace, as shown in Fig. 3.

Calculate Pareto Front then computed the Pareto optimal set of policies in the architectural space without considering uncertain factors in the farming environment. Using the ROAM optimization al-

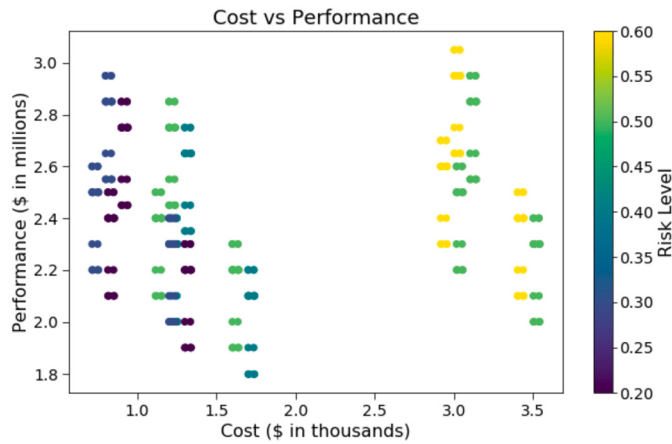


Fig. 4. Visualization of enumerating the trade space with performance being the increase in farm profitability.

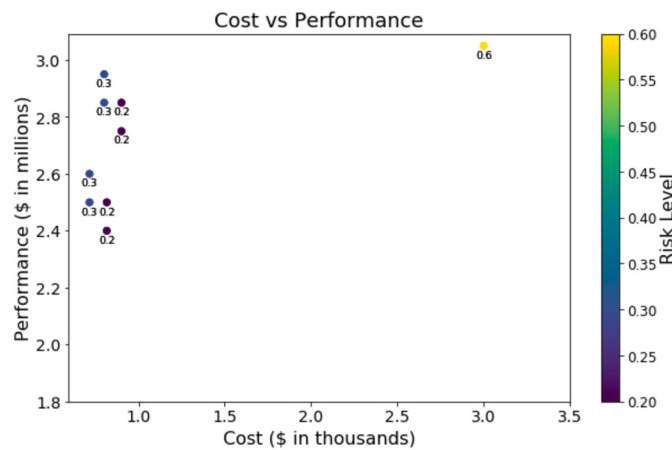


Fig. 5. Visualization of the trade space Pareto front.

algorithm, 18 Pareto optimal policies were found in the tradespace for CXG's configuration. Once narrowing the tradespace to 18 policies we perform further analysis in MORDM where we mined for the Pareto optimal set of policies to analyze decision tradeoffs. Fig. 4 shows the entire tradespace on a two-dimensional plot, where the x-axis is the cost denoted in dollars, the y-axis is performance denoted by profit in dollars, and the z-axis depicted through color coding is risk denoted by a percent as discussed in 2.3.3. Fig. 5 shows the policies on the Pareto front of the tradespace that were generated for CXG.

3.2. Analysis in MORDM

Many-Object Robust Decision Making (MORDM) was then used to decipher the policies' performance, cost, and risk of CXG under simulated uncertain environments. To initiate analysis using MORDM, CXG inputted additional constraints and specifications on the tradespace. This allowed CXG to narrow the scope of analysis by specifying the key parameters of their farm region, including the type of crop grown, the area of the farm, the estimated climate complexity, and the average precipitation level. CXG was then also able to specify a set of constraints on the objectives to define the ideal tradeoffs for their farm. Finally, imposing objective constraints was used to classify the policies of interest and identify the key uncertainties for later analysis.

With the defined parameters, the software used a many-objective optimization algorithm to calculate the optimal set of policies among the tradespace Pareto front, at the mean State-of-the-World (SOWs). The new set of policies found by the algorithm are the optimal policies after accounting for the effects of uncertainties, modeled through stakeholder

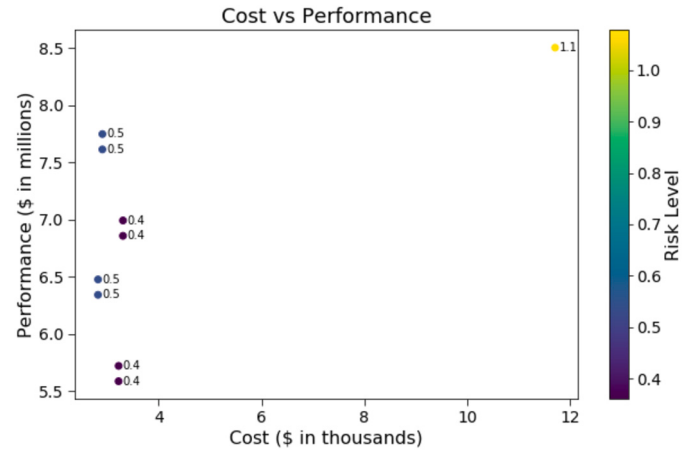


Fig. 6. Optimal set of policies after factoring in uncertainties.

analysis and research detailed in subsubsection 2.3.4. Although hard to see as the 18 points in Fig. 5 are overlapping with each other due to similar values, in Fig. 6 after factoring the mean SOWs there are nine remaining optimal policies found among the tradespace Pareto front.

With the Pareto optimal set of policies under the mean SOWs identified, the software performed a what-if analysis of each Pareto policy through Scenario Discovery under more robust uncertainties. A set of 1000 SOWs were generated based on the distribution for each of CXG's uncertainty variables. This analysis is depicted in Fig. 7, where Fig. 7a presents a 3D plot and Fig. 7b showcases a 2D projection of the same data. Notably, in Fig. 7b, it is observed that Policy 2 exhibits a higher cost and better performance under the what-if analysis compared to the other eight policies. In addition, Fig. 7 shows the continuous range of the performance objective, in contrast to the discrete cost objective. The insights from this what-if analysis were instrumental in narrowing down the set of Pareto optimal policies, ultimately focusing on Policies 5 and 6, as depicted in Fig. 8. With Scenario Discovery, we were able to illustrate distinctions in outcomes under what-if analysis among the nine equivalently optimal set of policies to support further decision making.

3.3. Final result

Building on our Tradespace Exploration, we conducted further analysis to refine our selection from the initial nine options by taking into account user preferences. The owner of CXG farm prioritized performance enhancements over system cost and exhibited a high tolerance for risk. This preference guided us to focus on Policies 5 and 6, as illustrated in Fig. 8. Both policies offered a similar balance between risk and cost, yet differed slightly in performance. Scenario Discovery revealed that Policy 5 aligns more closely with the decision maker's preferences compared to Policy 6, likely due to differences in sensitivity to labor costs and farm area. Therefore, the components of Policy 5, which include Water Sensor: 'Glass tube', Humidity Sensor: 'None', Microcontroller: 'Farmbeats', Data Storage: 'Cloud', Analytics: 'Model Predictive Control', and Irrigation Control: 'Raspberry Pi', with a performance of \$2.85 million, a cost of \$801.5, and a risk of 0.3, were recommended as the most Pareto optimal setup for the NEF system at CXG.

Finally, when compared to CXG's previous deployment, which was not a network-enabled farm (e.g. sensors had to be read manually), ROAM's Pareto optimal Policy 5 would lead to a 37% reduction in costs, a 33% improvement in performance, and an 11% decrease in risk.

4. Discussion

With ROAM, farmers can understand what a Pareto optimal set of choices for a farm of interest might be. The idea of creating a NEF system is daunting due to the number of choices that must be made. In

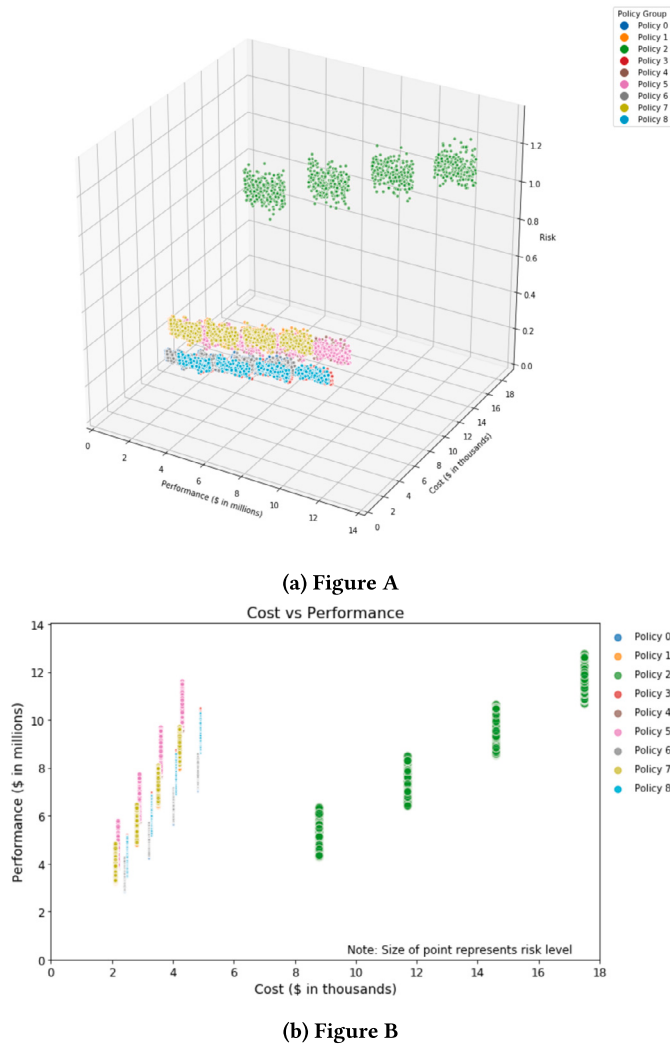


Fig. 7. Visualization of optimal decisions in Scenario Discovery 2D Plot.

section 3, the farm owner had over 324 policies to consider. ROAM simplified the process and allowed the user to understand the trade-offs when examining design decisions and to filter choices based on their needs. ROAM presents Pareto optimal farm architectures based on performance, cost, and risk while factoring in climatic uncertainties. Further, ROAM is extensible, as the code is written in an object-oriented manner, and allows interchanging new parameters and analytical optimization models. ROAM users can conceptualize what a data-driven farm management system might look like based on their specific goals and farming environment.

To allow for ease of use for our target users such as farm-owners, scientific researchers, industry professionals, and decision makers, we have developed a browser-end interface to host the workflow of ROAM. Users can use ROAM to generate interactive visualizations for communication and demonstrations with colleagues. Farm-owners and farm stakeholders specifically utilize a configuration file and input parameter features to customize and explore the decision space for their farms. ROAM's current implementation optimizes for cost, performance, and risk. For additional optimization goals, an extension of the software and further data analysis can be implemented.

ROAM allows the intricate dynamics of DA decision making to be explored, particularly focusing on the nuanced tradeoffs inherent in building a Pareto optimal NEF. Drawing from empirical studies, our five live deployments at Cornell University, and validation of ROAM through deployment to CXG, we have explored the granular aspects

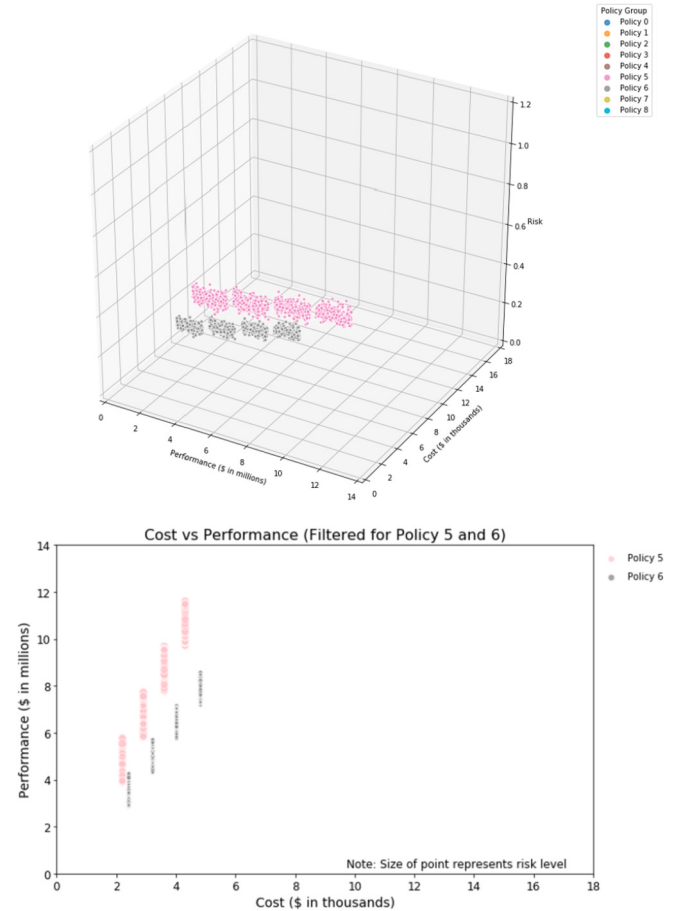


Fig. 8. Policy 5 and 6 in Scenario Discovery.

of building Pareto optimal DA systems. We recognize the importance of integrating user input and modeling uncertainty to tailor robust DA systems to individual needs. Our methodology underscores the necessity of a DSS that not only enhances the profitability of farming operations but also navigates the complexities of utility, risk, and uncertainties inherent in agriculture.

While our approach with ROAM has shown promising potential, it is important to acknowledge its limitations. Firstly, the efficacy of ROAM hinges on the availability and accuracy of farm data, underscoring the need for precise and up-to-date information for Pareto optimal results. Secondly, although ROAM is designed as a versatile tool adaptable to various agricultural contexts, its utility is maximized only with detailed user input specific to their farm's operational goals. Third, ROAM's predictive capabilities, particularly in forecasting future climatic conditions, are grounded in historical data analysis. This approach, while informative, might not always capture the unpredictable nature of agricultural changes. We acknowledge that past trends may not always be reliable predictors of the future. Lastly, our initial validation focused on wine grape production at CXG farm, revealing specific insights applicable to this crop. However, agricultural systems are diverse, and future work will extend ROAM's application to a broader range of crops, enhancing its utility and adaptability across different agricultural contexts. Despite these limitations, ROAM allows for a systematic understanding of the decision-making processes in building DA systems.

5. Conclusion

We presented the Realtime Optimization and Management System (ROAM). It is designed to identify the Pareto optimal set of tradeoffs for a Digital Agriculture (DA) based farm, which is important because DA is seen as an approach to address the Global Agricultural Productivity

(GAP) shortfall [1]. Specifically, DA enables data driven farm management, which requires on farm networking. A Network-Enabled Farm (NEF) uses new networking on a farm to enable DA. Based on deploying five NEFs, 11 farmer interviews, and testing on a farm in California, ROAM is able to present Pareto optimal NEF architectures for a given farm area of interest. ROAM presents general recommendations as to how to best implement a NEF based off of data inputted by the user and climatic data.

CRedit authorship contribution statement

Shiang-Wan Chin: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Gloire Rubambiza:** Conceptualization. **Yifan Zhao:** Data curation. **Keyvan Malek:** Validation. **Hakim Weatherspoon:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Software and data availability

Software is available open source and can be found at https://github.com/ShiangC/Cornell_SDF. Data will be made available on request.

Acknowledgements

This work was supported by the US National Science Foundation (NSF) under grants #1955125, NSF STC under grant #DBI-2019674, a Microsoft Investigator Fellowship, and the Cornell Institute for Digital Agriculture (CIDA). Shiang-Wan Chin was supported by the Foundation for Food and Agriculture Research (FFAR) and a Microsoft Cornell Summer Research Fellowship. The authors would like to thank Ranveer Chandra, Sachille Atapattu, Daniel Amir, Patrick Kastner, and Xinpi Du and the anonymous reviewers for their constructive feedback and insightful discussions during the manuscript's preparation. The authors appreciate the support from Microsoft partners, including Tusher Chakraborty, Elizabeth Bruce, and Stacey Wood. The networking infrastructure for the project would not have been possible without the help of numerous Information Technology professionals (especially Scott Yoest), building managers (especially Scott Albrecht), orchard managers, and carpenters at Cornell University. All opinions expressed are those of the researchers, not the NSF, Microsoft, or CIDA.

References

- [1] A. Steensland, M. Zeigler, Global Agricultural Productivity Report, Global Harvest Initiative, 2018, pp. 1–2, https://globalagriculturalproductivity.org/wp-content/uploads/2019/01/GHI_2018-GAP-Report_FINAL-10.03.pdf.
- [2] J.S. Wallace, Increasing agricultural water use efficiency to meet future food production, *Agric. Ecosyst. Environ.* (2000), [https://doi.org/10.1016/S0167-8809\(00\)00220-6](https://doi.org/10.1016/S0167-8809(00)00220-6).
- [3] J. Antle, B. Basso, Digital agriculture to design sustainable agricultural systems, *Nat. Sustain.* (2020), <https://doi.org/10.1038/s41893-020-0510-0>.
- [4] J. MacDonald, R. Hoppe, America's diverse family farms, *Econ. Inf. Bull.* (2016) 16, <https://www.ers.usda.gov/webdocs/publications/81408/eib-164.pdf?v=835.1>.
- [5] M. Schipanski, L. Atwood, D. Mortensen, M. Hunter, R. Smith, Agriculture in 2050: recalibrating targets for sustainable intensification, *Manag. Sci.* (2017) 386–391, https://globalagriculturalproductivity.org/wp-content/uploads/2019/01/GHI_2018-GAP-Report_FINAL-10.03.pdf.
- [6] P. Sengers, H. Weatherspoon, G. Rubambiza, Paradoxes in producing the future of farm work: anticipating social impact through the lens of early adopters, in: CHI, 2021, <https://ceur-ws.org/Vol-2905/paper13.pdf>.
- [7] N. Nemes, Comparative analysis of organic and non-organic farming systems: a critical assessment of farm profitability, <https://www.fao.org/documents/card/en/c/28b2b959-80f1-5244-bd52-0a98806d37ea/>, 2009.
- [8] R.E. Howell, J. Waud, R.E. Dunlap, C.E. Beus, What is sustainable agriculture?, *J. Sustain. Agric.* (2008) 5–41, https://doi.org/10.1300/J064v03n01_03.
- [9] K. Schroeder, J. Lampietti, G. Elabed, What's Cooking: Digital Transformation of the Agrifood System, Agriculture and Food Series, World Bank, 2021, <https://documents1.worldbank.org/curated/en/417641615957226621/pdf/Whats-Cooking-Digital-Transformation-of-the-Agrifood-System.pdf>.
- [10] D. Patterson, H. Persson, B. Fischer, N. Mandaluniz, G. Chirici, A. Zacepins, V. Komasilovs, T. Pitulac, A. Nasirahmadi, G. Gebresenbet, T. Bosona, A concept for application of integrated digital technologies to enhance future smart agricultural systems, 2023.
- [11] N. Tsolakis, E. Aivazidou, Transitioning towards human–robot synergy in agriculture: a systems thinking perspective, *Syst. Res. Behav. Sci.* (2022), <https://doi.org/10.1002/sres.2887>.
- [12] T. Kassanuk, S.S. Jha, T. Ghosh, M.C. Thakar, S.T. Jagtap, K. Phasinam, Towards application of various machine learning techniques in agriculture, *Mater. Today Proc.* (2022), <https://doi.org/10.1016/j.matpr.2021.06.236>.
- [13] G.S. Parnell, T.E. Trainor, Using stakeholder analysis to define the problem in systems engineering, in: INCOSE International Symposium, 2007.
- [14] S. Hallett, R. Sakrabani, A. Thompson, L. Papadimitriou, J. Knox, D. Agostino, M. Borg, Multi-stakeholder analysis to improve agricultural water management policy and practice in Malta, 2020.
- [15] H. Meinke, The role of modeling and systems thinking in contemporary agriculture, *Sustain. Food Suppl. Chains* (2019) 39–47, <https://doi.org/10.1016/B978-0-12-813411-5.00003-X>.
- [16] F. Cancian, Risk and uncertainty in agricultural decision making, in: *Agricultural Decision Making: Anthropological Contributions to Rural Development*, 1980, https://books.google.com/books?hl=en&lr=&id=viHgBAAQAQAJ&oi=fnd&pg=PA161&dq=Risk+and+Uncertainty+in+Agricultural+Decision+Making&ots=7NYMSmLd78&sig=IN_DW4WbGt7q1gKAL1nfZlw1y1#v=onepage&q=Risk+20and%20Uncertainty%20in%20Agricultural%20Decision%20Making&f=false.
- [17] S. Chin, S. Atapattu, M. Rehman, M. Jose, H. Weatherspoon, G. Rubambiza, Cosmosum: An Extensible, Reconfigurable, and Fault-Tolerant IoT Platform for Digital Agriculture, in: USENIX, 2023, <https://www.usenix.org/system/files/atc23-rubambiza.pdf>.
- [18] Raspberry Pi, Raspberry pi 3 model b, <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>, 2022.
- [19] Aquaoso, California agricultural water price by water district, <https://aquaoso.com/water-trends/california-agricultural-water-prices/>, 2021.
- [20] T. Yaqub, J. Katupitiya, R. Eaton, Systems engineering approach to agricultural automation: new developments, in: *1st Annual IEEE Systems Conference*, 2007.
- [21] S. Avraamidou, S.G. Baratsas, E.N. Pistikopoulos, A systems engineering framework for the optimization of food supply chains under circular economy considerations, *Sci. Total Environ.* (2021), <https://doi.org/10.1016/j.scitotenv.2021.148726>.
- [22] C. Lobdell, D. Cabrera, L. Colosi, Systems thinking, *Eval. Program Plann.* 21 (2008) 299–310, <https://doi.org/10.1016/j.evalprogplan.2007.12.001>.
- [23] B. Cameron, E. Crawley, D. Selva, Patterns in system architecture decisions, *Syst. Eng.* 19 (6) (2016) 477–497, <https://doi.org/10.1002/sys.21370>.
- [24] Hewlett Packard Enterprise, Precision agriculture yields higher profits, lower risks, in: 3BL, 2019, <https://www.3blmedia.com/news/precision-agriculture-yields-higher-profits-lower-risks>.
- [25] B. Tekindogdogan, Systems architecture design pattern catalog for developing digital twins, *Sensors* 20 (18) (2020) 5103, <https://doi.org/10.3390/s20185103>.
- [26] R. Singh, P. Reed, K. Keller, V. Ward, Confronting tipping points: can multi-objective evolutionary algorithms discover pollution control tradeoffs given environmental thresholds?, *Environ. Model. Softw.* 73 (2015) 27–43, <https://doi.org/10.1016/j.envsoft.2015.07.020>.
- [27] D. Marks, J. Cohon, A review and evaluation of multiobjective programming techniques, *Water Resour. Res.* 11 (1975) 208–220, https://ui.adsabs.harvard.edu/link_gateway/1975WRR....11..208C/doi:10.1029/WR011i002p00208.
- [28] S. Nataraj, P.M. Reed, J.R. Kasprzyk, R.J. Lempert, Many objective robust decision making for complex environmental systems undergoing change, *Environ. Model. Softw.* 42 (2013) 55–71, <https://doi.org/10.1016/j.envsoft.2012.12.007>.
- [29] Davis Instruments, Soil moisture sensor, vantage pro2™ and enviromonitor, <https://www.davisinstruments.com/collections/add-on-sensors/products/soil-moisture-sensor-vantage-pro-2-and-enviromonitor>, 2024.
- [30] C.J. Rutten, A.G.J.M. Oude Lansink, H. Hogeveen, W. Steeneveld, Delaying investments in sensor technology: the rationality of dairy farmers' investment decisions illustrated within the framework of real options theory, *J. Dairy Sci.* (2018), <https://doi.org/10.3168/jds.2017-13358>.
- [31] K. Sasloglou, H. Goh, T. Wu, B. Stephen, M. Gilroy, C. Tachtatzis, I. Glover, C. Michie, I. Andonovic, K. Kwong, Adaptation of wireless sensor network for farming industries, in: *2009 Sixth International Conference on Networked Sensing Systems (INSS)*, 2009.
- [32] S. Thessler, J. Koskiahio, A. Hannukkala, H. Huitu, T. Huttula, J. Havento, M. Jarvenpaa, N. Kotamaki, Wireless in-situ sensor network for agriculture and water monitoring on a river basin scale in southern Finland: evaluation from a data user's perspective, *Sensors* 9 (4) (2009) 2862–2883, <https://doi.org/10.3390/s90402862>.

- [33] I. Jaskulska, D. Jaskulska, M. Szczepanczyk, L. Galewski, Analysis of the need for soil moisture, salinity and temperature sensing in agriculture: a case study in Poland, *Sci. Rep.* (2021), <https://www.nature.com/articles/s41598-021-96182-1>.
- [34] Z. Qiaofu, W. Shaohong, D. Erfu, L. Shuangcheng, Measurement of climate complexity using sample entropy, *Int. J. Climatol.* (2006) 2131–2139, <https://doi.org/10.1002/joc.1357>.
- [35] J.T. Abatzoglou, Development of gridded surface meteorological data for ecological applications and modelling, *Int. J. Climatol.* (2011), <https://doi.org/10.1002/joc.3413>.
- [36] A. Marshak, A. Delgado-Bonal, Approximate entropy and sample entropy: a comprehensive tutorial, *Entropy* (2019), <https://doi.org/10.3390/e21060541>.
- [37] J. Ngaira, H. Ogindo, N. Masayi, J. Hunho, The changing rainfall pattern and the associated impacts on subsistence agriculture in Laikipia East district, Kenya, *J. Geogr. Reg. Plan.* (2012), <https://doi.org/10.5897/JGRP12.018>.
- [38] K. Guan, G. Schnitkey, E. DeLucia, B. Peng, Y. Li, Excessive rainfall leads to maize yield loss of a comparable magnitude to extreme drought in the United States, *Glob. Change Biol.* (2019), <https://doi.org/10.1111/gcb.14628>.
- [39] R.C. Serlin, L.A. Marasculio, *Statistical Methods for the Social and Behavioral Sciences*, American Psychological Association, 1988, https://books.google.com/books/about/Statistical_Methods_for_the_Social_and_B.html?id=-MBLvgAACAAJ.
- [40] J. Herman, P.M. Reed, K. Keller, D. Hadka, An open source framework for many-objective robust decision making, *Environ. Model. Softw.* 74 (2015) 114–129, <https://doi.org/10.1016/j.envsoft.2015.07.014>.
- [41] H. David, R. Patrick, H. Antonia, G. David, Rhodium: python library for many-objective robust decision making and exploratory modeling, *J. Open Res. Softw.* (2020), <https://doi.org/10.5334/jors.293>.
- [42] J. Cariboni, A. Saltelli, F. Campolongo, An effective screening design for sensitivity analysis of large models, *Environ. Model. Softw.* (2007), <https://doi.org/10.1016/j.envsoft.2006.10.004>.
- [43] I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Math. Comput. Simul.* (2001), [https://doi.org/10.1016/S0378-4754\(00\)00270-6](https://doi.org/10.1016/S0378-4754(00)00270-6).
- [44] A. Saltelli, *Global Sensitivity Analysis. The Primer*, <https://doi.org/10.1002/9780470725184.ch6>, 2008.