AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum
10.2514/6.2022-0925

REPRESENTING AND ANALYZING SEQUENTIAL SATELLITE MISSION DESIGN DECISIONS THROUGH ANISOMORPHIC TREES AND DIRECTED GRAPHS

Ada-Rhodes Short
Texas A&M University
College Station, Texas

Prachi Dutta
Texas A&M University
College Station, Texas

Ben Gorr
Texas A&M University
College Station, Texas

Luke Bedrosian
Texas A&M University
College Station, Texas

Daniel Selva
Texas A&M University
College Station, Texas

ABSTRACT

In recent years, decision trees have been an uncommon approach to design space exploration, and meta-heuristics and machine learning have been the primary approach. Although these approaches have proven effective, decision trees have the unique property that allows us to explicitly traverse the design space sequentially, in a human-understandable form. The problem is the order of the decisions in the tree is not pre-defined but chosen by the user, leading to anisomorphic trees and inconsistent performance during design space exploration. In the past, we viewed this as a drawback of decision trees that designers needed to overcome. However, in this paper, we study anisomorphic decision trees representing a satellite mission design problem to glean insights into design decision-making in general. In this case study, we look at an earth observation satellite mission design formulated as an instrument-to-orbit assignment problem and quantify the effects of design decision tree characteristics. This work has applications in design space exploration and design automation when working on complex design problems by improving our understanding of how to represent and search the space. Additionally, we define the relationship between all anisomorphic trees for a problem and a novel structure that we call the Design Space Directed Graph, contributing to the general understanding of design decisions and their mathematical representations.

Keywords: Design Space Exploration, Decision Trees, Anisomorphic, Graph Theory, Graph Representations, Design Space Directed Graph,

NOMENCLATURE

$T_{A,B,C}$	Design decision tree with ordered decisions A, B, and C
$DSDG_{x,y,z}$	Design Space Directed Graph of size x, y, z. $x \leq y \leq z$
Nd	The number of design decisions that must be made to make a complete design
Nc	The sum of all decision options in the design space
$t(T_{A,B,C}) = t(DSDG_{x,y,z})$	The set of terminal nodes in tree $T_{A,B,C}$ and $DSDG_{x,y,z}$ representing all complete designs
$Nt = \|t\|$	The number of terminal nodes or complete designs in a design tree or DSDG
$i(T_{A,B,C}) \neq i(DSDG_{x,y,z})$	The set of intermediate nodes in tree $T_{A,B,C}$ or $DSDG_{x,y,z}$ representing incomplete designs
Ni	The number of intermediate nodes or incomplete designs in a design tree of DSDG
$CE(DSDG_{x,y,z})$	The set of continuation edges in $DSDG_{x,y,z}$; all bounded $T_{A,B,C}$ edges are a subset of CE
$ME(DSDG_{x,y,z})$	The set of modification edges in $DSDG_{x,y,z}$ which is the complement of the set CE (meaning they are mutually exclusive)
$b(Decision), b(T_{A,B,C})$	The branching factor of an individual decision and the average branching factor of an entire tree, respectively

1. INTRODUCTION

Searching design spaces for high-quality solutions is at the core of effective design. Common approaches to Design Space Exploration (DSE) include genetic algorithms, simulated annealing, and various other optimization approaches **[1]–[4]**.

Unfortunately, searching decision trees is not currently a common approach to the problem. There are some excellent reasons for this. The most notable is that decision trees can become intractably large after just a handful of choices, and simple genetic algorithms can easily outperform tree searches **[5], [6]**. Additionally, because we can place decisions in different orders, a single design space can be represented by $\boldsymbol{n}!$ anisomorphic decision trees (anisomorphic, meaning that each tree is unique) where $\boldsymbol{n}$ is the number of individual decisions (assuming we can make each decision only once).

So what value is there in studying design decision trees? First and foremost, design decision trees capture the order of decisions the way humans navigate decision problems. A clear example of this is when you order a sandwich from Subway **[7]**, they ask you to pick bread, meat, cheese, veggies, and condiments in that order. However, if they one day asked you to choose your condiments first and then work backward, you would likely find the experience disorienting **[8]**.

In addition to the characteristic of being ordered, in previous work, we have identified *multimodality*, *opaqueness*, *unboundedness*, and *increasing internal locations* as common design decision characteristics that can be represented by decision trees [5], [6]. *Multimodality* describes the objective function used to evaluate the design space. A multimodal design space has multiple local minima and or maxima, meaning that at any time, the quality of a design could increase or decrease as the result of the next decision.

Closely related to multimodality is opaqueness. *Opaqueness* means we cannot evaluate a design until we have made all required design decisions and the design is considered *complete*. One cause of opaqueness can be that we can not apply the evaluation method to the incomplete design. For example, you would not want to evaluate the quality of a chair that is not yet fully assembled by sitting on it. Alternatively, it could be infeasible to evaluate incomplete designs due to the time and cost to evaluate the increased number of combinations.

Unboundedness means that you can always technically make additional decisions. For example, you could tell a beleaguered sandwich artist to keep adding more and more pepperoni to your sandwich. In this paper, we introduce two variants on the property of unboundedness, *replacement,* and *addition*. *Replacement* means that new decisions made after a design is complete replace old decisions, allowing designers to change their minds. *Addition* means that when designers make new decisions, they add something to the design, as exemplified by the excess of pepperonis. These characteristics have a unique relationship to the topology of the trees that we elaborate on below. Another common characteristic related to addition is *increasing internal locations,* which means that the number of available decisions increases every time a designer makes a decision. For example, when you are building something from Lego bricks, each Lego brick added creates a new location for you to place additional bricks. However, this characteristic can be ignored by constraining the problem to a limited catalog of options.

We can represent design spaces possessing some or all of these characteristics using a novel representation that we call a Design Space Directed Graph (DSDG) where each node represents a set of choices composing an incomplete or complete design and each edge represents the decision to change the current design as demonstrated in FIGURE 1.

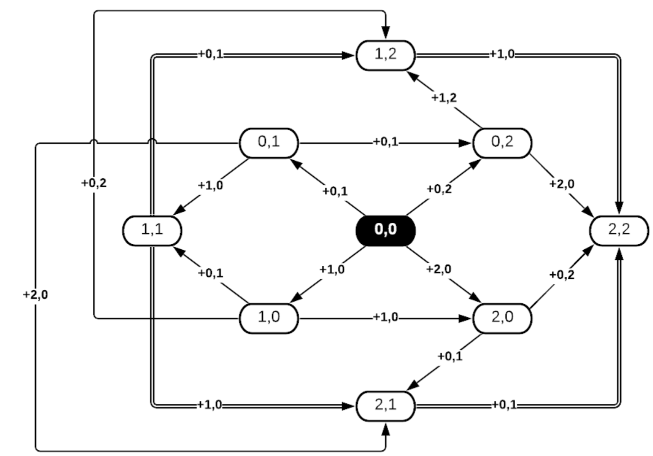


FIGURE 1: A DESIGN SPACE DIRECTED GRAPH FOR A DESIGN PROBLEM WITH TWO DECISIONS WITH TWO OPTIONS EACH. APPENDIX A CONTAINS A LARGER VERSION OF THIS FIGURE.

In this example we have two decisions that must be made A and B, both of which have two options. In the Null Node (shown in black) neither decision has been made so they are both given a value of 0, resulting in $A, B = 0{,}0$. The edges are labelled with the difference between nodes. So, the edge between $0{,}0$ and $1{,}0$ is labelled as $+1{,}0$. The DSDG is represented by a directed graph notation to keep track of changes, however traversals against the direction are allowed. In that case, the negative of the change would occur. So, moving between $1{,}0$ and $0{,}0$ would represent a $-1{,}0$ move. You can also think of this as each edge representing two traditional directed edges representing opposite changes.

Anisomorphic design decision trees are subgraphs of the DSDG as shown in FIGURE 2 where each anisomorph has a different path through the DSDG.

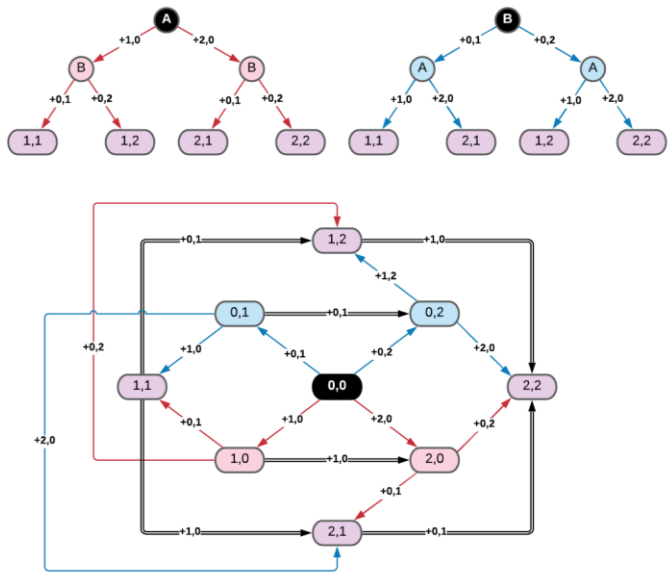


FIGURE 2: TOP LEFT: DECISIONS TREE IN ORDER A, B. TOP RIGHT: DECISION TREE IN ORDER B, A. BOTTOM: THE DSDG OF THE DESIGN SPACE WITH THE EDGES OF TREE A, B SHOWN IN RED AND THE EDGES OF TREE B, A SHOWN IN BLUE. APPENDIX B CONTAINS A LARGER VERSION OF THIS FIGURE.

In this example, we see that in both trees, the initial decision is the Null Node (shown in black). To demonstrate their relationship to the DSDG, we have color-coded the edges. The edges of the Tree containing A first and B second, alternatively written as $T_{A,B}$ (top left), are shown in red and the edges of the Tree with B first and A second, $T_{B,A}$ (top right), are shown in blue. You can see that both trees exist as subgraphs of the DSDG and contain the same Null Node and Terminal Nodes. The terminal nodes (shown in purple) represent completed designs. In this example of two design decisions with two options each (which we can write as $DSDG_{2,2}$), the union of the two trees represent the complete set of nodes in the $DSDG_{2,2}$ and therefore account for all the anisomorphic decision trees that can represent the design space. However, four edges are present in $DSDG_{2,2}$ that are not present in $T_{A,B}$ or $T_{B,A}$. To make these edges stand out, we have drawn them with a double line. The double-lined edges do not represent decisions that continue the design by adding new choices but instead represent modifications to previous choices. We call these edges *Modification Edges,* and the edges in the trees are called *Continuation Edges*. Modification edges change a previous design decision, and the continuation edges make a decision where one has not previously been made.

This paper focuses on the characteristics of multimodality, opaqueness, and unbounded replacement and examines their effects on DSE. We do this through a satellite mission design case study informed by a preliminary examination of a toy problem. In the toy problem, we explore how the characteristics affect tree traversal algorithms' performance. We then perform a case study in which we apply insights gained from the toy problem to improve the efficiency of a design space exploration of an Earth Observation satellite system design problem formulated as an instrument-to-orbit assignment problem. We hypothesize that by better understanding the effects of the design decision tree characteristics, we can search the design decision trees more efficiently using common tree traversal algorithms for DSE.

1.1 Contribution

This paper has three major contributions. The first is that it introduces the DSDG and describes its relationship to design decision trees. The DSDG is a novel graph-based structure for representing design spaces with potential applications in machine learning, design space exploration, and human observation that we plan to explore in future work. The second major impact is that it expands on our previous work in characterizing general design decision-making by introducing unbounded replacement and addition characteristics. While the general properties behind these characteristics predate this work, we give them names and demonstrate their relationship to the novel DSDG. Finally, the third impact of this work is that it explores the influences of the design decision characteristics on DSE through design decision trees, improving general knowledge of DSE, and informing strategies for improved tree search performance when specifically dealing with design decisions.

1.2 Background

This work builds on previous research and domain knowledge in design automation and satellite mission design. In this section, we discuss previous work and important concepts.

1.2.1 Design Space Exploration

Design Space Exploration (DSE) enables system designers to explore various design alternatives before implementing the design. For the system designers to find the optimal or most preferred design, they must explore many potential design candidates [9], [10]. The issue with DSE arises from the sheer size of the design space, which grows exponentially with the (geometric) average number of options per decision. Another problem is that the objectives that measure solution quality may be competing, so designs that perform well at one objective may perform poorly on another. Finally, the system designer uses visual and data analytics to analyze the datasets and draw conclusions about the design space to identify the features that make a design 'good' or 'bad.'

1.2.3 Satellite Instrument-to-Orbit Assignment

This paper looks at an instrument-assignment problem to design a satellite mission for earth observation. For pre-Phase A studies of satellite missions, instrument and orbit selection are critical decisions. Instrument selection determines the observational capabilities of the system. Pairing an instrument suite with an orbit is equally important – different instruments will provide the best data in certain orbits, and two instruments on the same spacecraft may have conflicting ideal orbits. Since instruments and orbits are closely coupled, in this work we consider their joint design space. Instrument-assignment has been previously explored in [11]–[13].

2. METHODS

In order to carry out this study, we developed a MATLAB Application called Tree-Top using MATLAB App Designer [14], [15]. Using Tree-Top we performed a case study informed by preliminary exploration of a toy problem. In the toy problem we used designing a sandwich to look at the effects of multimodality and opaqueness on common tree traversal algorithms' ability to perform DSE. Additionally, we leverage the properties of the DSDG to explore the effects of unboundedness. In the satellite instrument-assignment case study, we take the insights gained from the toy problem and apply them to a real-world satellite mission design problem and test to see if we can have a positive impact on DSE performance.

2.1 The Sandwich Problem

The sandwich design toy problem was created to explore anisomorphic design decision trees and DSDGs and their characteristics through an example that could be understood by all designers regardless of their domain expertise. The design space consists of five decisions, $Nd = 5$, with a total of 22 options, $Nc = 22$. The decisions are Bread, Meat, Cheese, Vegetable, and Condiments. Table 1 shows the available choices for each decision.

Table 1: Sandwich Problem Options

Meat	Cheese	Vegetable	Bread	Condiments
Ham	Cheddar	Lettuce	White	Mayo
Turkey	Swiss	Tomato	Wheat	Mustard
Salami	Provolone	Peppers	Rye	Russian Dressing
Corned Beef	None Cheese	Onions		None Condiments
None Meat		Sauerkraut		
		None Veggies		

The sandwich problem is represented by $DSDG_{3,4,4,5,6}$ or by 120 different anisomorphic design decision trees. Each anisomorphic tree has a unique notation. For example, a design decision tree in the order Bread, Meat, Cheese, Vegetables, and Condiments can be denoted by $T_{B,M,Ch,V,Co}$. The total number of unique completed sandwiches for this problem is 1440.

For all experiments performed in this case study the primary metric of performance is the mean number of nodes evaluated with our objective function before the best possible design is found. This means that a perfect score would be 1, meaning that the best design was found on the first evaluation.

2.1.1 Multimodality

To study the effects of multimodality on DSE using a design decision tree, we created two different objective functions for the sandwich problem. The first objective function is a weighted lookup table to evaluate each sandwich through tabulation, similar to how you would apply a Pugh Chart [16] with the datum set at the second-worst option from the list of choices. The weights for each category were Meat=5, Cheese=4, Vegetable=3, Bread=2, and Condiments=1. The complete table can be found in APPENDIX C. This form of evaluation is not multimodal because at every decision the final score can be improved by selecting the best ranked individual choice. A Sandwich Synergy Matrix was created to make a multimodal form of evaluation. In this matrix, we cross-reference each choice with every other choice and sum their combined scores to obtain a quantifiable measure of sandwich quality. The mean or ranked preference of individual choices is kept the same to keep things consistent between the two methods. APPENDIX D contains the full sandwich synergy matrix.

Using these two separate modes of evaluation, we carried out multiple experiments. We performed a Best First Search (BFS), Full Factorial Enumeration (FFE), and two variations of a Monte Carlo Tree Search (MCTS) on six trees: $T_{M,Ch,V,B,Co}$ and $T_{Co,B,V,Ch,M}$ which represented the choices put in order and reverse order of weighting, $T_{B,Ch,Co,M,V}$ and $T_{V,M,Co,Ch,B}$ which are the trees with the number of choices in ascending and descending order, and $T_{B,M,Ch,V,Co}$ and $T_{Co,V,Ch,M,B}$ representing the order of decisions presented at most chain sandwich restaurants [7] and therefore potentially relevant to underlying design grammars [17].

The BFS evaluates the current leaf nodes in the tree and expands on the node with the best score for continued exploration. The FFE evaluates every node in an arbitrary order. The two variations of MCTS used were an MCTS with full expansion, meaning that on each iteration, the algorithm generated a complete sandwich and then evaluated the design and backpropagated the results, and an MCTS with partial expansion, meaning that on each iteration the algorithm makes an individual decision and then back propagates the results.

2.1.2 Opaqueness

In order to study the effect of opaqueness on tree search performance, we compare the results of algorithms that do not evaluate incomplete designs and compare them to equivalent algorithms that do evaluate incomplete designs. These pairings are MCTS with full expansion vs. partial expansion and a BFS vs. FFE. BFS and FFE are comparable in this case because FFE is the evaluation of all leaf nodes representing completed designs without being able to evaluate incomplete designs.

2.1.3 Unbounded Replacement

To study the effects of replacement, we are taking a different approach. Using the DSDG we can prove that mathematically treating a problem as unbounded using a Design Decision Tree is always redundant and will increase search time and computational cost. The proof for this is simple. The number of terminal nodes of all anisomorphic design decision trees that create a complete design, Nt, is the same. In FIGURE 1, in the DSDG we can see that the only nodes adjacent to the terminal nodes, t, are incomplete designs, i, reached by traversing backwards against the direction of a continuation edge, CE, or other terminal nodes that can be reached by traversing a modification edge, ME. There is no value to adding duplicate incomplete designs to the tree, so we would only want to include modification decisions. A straightforward way to represent this in the tree would be to add a decision to the tree containing each modification edge, as shown in Figure 3. APPENDIX E includes a larger version of this figure.

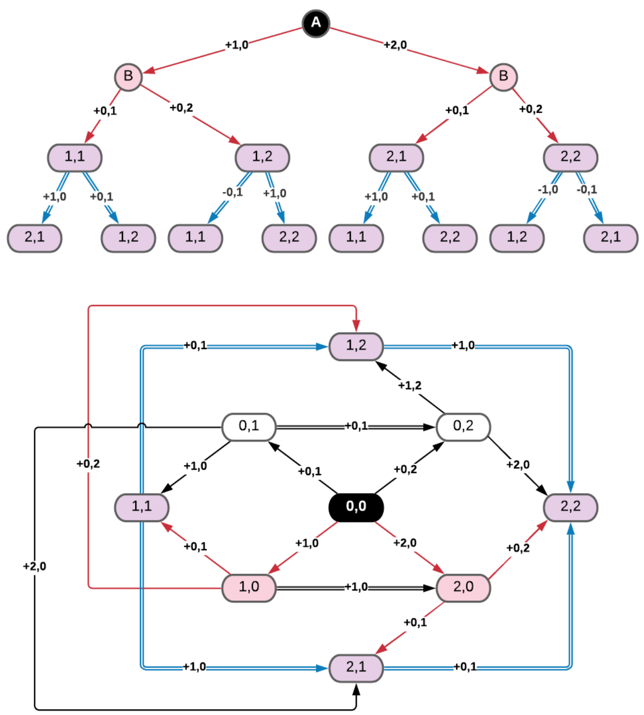


Figure 3: Top: Design Decision Tree A, B with continuation edges shown in red and modification edges shown in blue with double lines. Bottom: DSDG 2, 2 with edges color-coded to match the design decision tree

The branching factor of this replacement decision would be equivalent to the number of design decisions minus one multiplied by the number of choices minus the number of choices already made (the number of decisions already made is equivalent to the number of decisions for a complete design). Equation 1 and 2 shows this relationship as a single equation in two forms.

$$b(Mod) = (Nd - 1)(Nc - Nd) \quad (1)$$
$$= Nd \times Nc - Nd^2 + Nd - Nc \quad (2)$$

For the sandwich problem $Nd = 5$ and $Nc = 22$ giving us a total modification branching factor of 68 for the modification decision. This is large compared to the other decisions, so our expectation is that including modification will drastically reduce performance.

2.2 The Satellite Instrument-Assignment Case Study

The instrument-assignment problem is a real-world DSE problem that we have examined in previous work [11], [12], [18]. The problem is inherently multimodal and comparatively computationally expensive as, in some cases, the evaluation could take as long as two minutes per design.

While the instrument-assignment problem exists in many different forms, here we consider a version of the problem with five potential instruments (BIOMASS, VIIRS, CMIS, SMAP radiometer, SMAP radar), six potential orbit classes (equatorial, tropical, polar, SSO-DD, SSO-AM, SSO-PM), and four altitudes (400 km, 500 km, 600 km, and 700 km). A complete design consists of an orbit class, an altitude, and any subset of the five instruments. The total number of possible complete designs is 768.

Each complete design is evaluated using complex objective functions for cost and scientific value that are multimodal. These fitness functions include, for example, calculation of revisit time based on instrument field of view and orbit properties.

Additionally, a full design must be completed before evaluation can occur – without an orbit to go with an instrument, coverage metrics cannot be evaluated. Therefore, we find that the problem is opaque.

To check the validity of our findings from the first case study in application on a real-world problem, we compared what we would expect to be the best design decision tree DSE to the worst design decision tree DSE. Our metric for comparison is the number of evaluations needed to find the best design. Based on our explorations with the sandwich toy problem we predict that ascending order of branching factor (VIIRS, CMIS, SMAP Radiometer, SMAP Radar, BIOMASS, Altitude, and Orbit) will perform better than descending order (Orbit Altitude, BIOMASS, SMAP Radar, SMAP Radiometer, CMIS, VIIRS) and that including modification will lead to reduced DSE performance. Table 2 summarizes these hypotheses. We discuss the results of the toy problem exploration more below.

Table 2: DSE Performance Predictions

Hypothesis		
V,C,Rio,Rar,B,A,O	>	O,A,V,C,Rio,Rar,B
V,C,Rio,Rar,B,A,O	>	V,C,Rio,Rar,B,A,O,Mod

3. RESULTS AND DISCUSSION

This section will explain our findings for the two case studies based on various characteristics of decision trees in detail. We evaluated the effects of multimodality on full MCTS, partial MCTS, Best First, and expected FFE performance.

3.1 The Sandwich Problem

In the sandwich toy problem, we performed three experiments to compare the effects of multimodality, opaqueness, and replacement. We found that treating the design problem as opaque had the most significant effect overall.

Modification in combination with multimodality also had a significant effect, but the effect of modification was reduced or eliminated when the objective function was non-multimodal. We reported the number of evaluations performed before finding the best design for deterministic methods (best first and FFE). We reported the mean and standard deviation of the number of evaluations for the stochastic method before finding the best design. For the non-deterministic experiments, we ran ten trials each. The following sections detail our results.

3.1.1 Multimodality

We summarized the results of the multimodality experiments in Table 3, with the multimodal cases on the left and non-multimodal on the right.

For the Full Expansion MCTS, there was no general trend related to multimodality. This is potentially because the algorithm was only looking at completed designs, so changes in quality between individual decisions were minor. However, for the partial MCTS, there was a general trend of the multimodal cases outperforming the non-multimodal cases ($p = 0.043$). This result was unexpected because common sense would dictate that searching a multimodal space should be more difficult. We plan to repeat this experiment in future work for validation.

For the Best First search, the non-multimodal case significantly outperformed the multimodal case across the board ($p < .00001$). We expected this behavior because, for a non-multimodal space, a best-first search should find the best design when it reaches its first completed design. The variability in the results came from the order the decisions were presented, leading to a different number of nodes needing to be evaluated before a completed design was reached. We expected that descending order of decision weight was preferable in both the multimodal and non-multimodal cases, but the results suggested the opposite was true. However, our expectation that ascending order of the branching factor was preferable was supported by the results. Additionally, the typical restaurant sequence outperformed its opposite.

We initially planned to perform an experiment for Full Factorial Enumeration (FFE) but realized that arbitrary order of the choices would bias the results. So instead, we calculated the expected number of completed designs that would need to be evaluated before the best design was found. This is equal to 720.5 in the case of the sandwich problem and is the same regardless of multimodality.

There was no general conclusion from these experiments that we could draw about multimodality, and future studies are needed.

Table 3: Multimodality Results

MULTIMODALITY						
		Multimodal			Nonmultimodal	
	TREE	Mean	std		Mean	std
Full MCTS	M,Ch,V,B,Co	**655**	**358**	<	480	395
Full MCTS	Co,B,V,Ch,M	811	357	<	**378**	**288**
Full MCTS	B,Ch,Co,M,V	**480**	**338**	>	864	343
Full MCTS	V,M,Co,Ch,B	823	537	<	**191**	**127**
Full MCTS	B,M,Ch,V,Co	**531**	**423**	=	**526**	**343**
Full MCTS	Co,V,Ch,M,B	703	380	<	601	423
Partial MCTS	M,Ch,V,B,Co	2425	1590	>	**2992**	**2464**
Partial MCTS	Co,B,V,Ch,M	**2044**	**1608**	>	3419	1690
Partial MCTS	B,Ch,Co,M,V	**2163**	**1081**	>	3316	1734
Partial MCTS	V,M,Co,Ch,B	2659	1945	>	**3180**	**2123**
Partial MCTS	B,M,Ch,V,Co	**894.3**	**784**	>	2626	1880
Partial MCTS	Co,V,Ch,M,B	2271	797	<	**1425**	**1043**
Best First	M,Ch,V,B,Co	18353		<	60	
Best First	Co,B,V,Ch,M	**9798**		<	**54**	
Best First	B,Ch,Co,M,V	**11902**		<	**50**	
Best First	V,M,Co,Ch,B	19369		<	64	
Best First	B,M,Ch,V,Co	**11513**		<	**54**	
Best First	Co,V,Ch,M,B	12756		<	60	
Expected FFE	M,Ch,V,B,Co	720.5		=	720.5	
Expected FFE	Co,B,V,Ch,M	720.5		=	720.5	
Expected FFE	B,Ch,Co,M,V	720.5		=	720.5	
Expected FFE	V,M,Co,Ch,B	720.5		=	720.5	
Expected FFE	B,M,Ch,V,Co	720.5		=	720.5	
Expected FFE	Co,V,Ch,M,B	720.5		=	720.5	

3.1.2 Opaqueness

Table 4 summarizes the results of the opaqueness experiments. The left column presents the opaque methods with their non-opaque correspondent in the right column. We repeated the experiments with both multimodal and non-multimodal evaluation.

For MCTS, opaqueness would force us to use full expansion, and non-opaqueness would allow partial expansion. We found that opaqueness was significantly better regardless of multimodality ($p < .00001$). This makes sense because the non-opaque method has to run more evaluations between completed designs.

Both Best First Search and FFE require evaluating all nodes in a subset of the tree before identifying the best design. In the case of FFE it evaluates every terminal node, so again we expect it to find the best design after 720.5 nodes have been evaluated. We can use FFE in opaque cases because we only look at completed designs. Best First search is comparable to this because it looks at the best available option by evaluating incomplete designs following the sequence of decisions. In the non-multimodal case, the non-opaque Best First method is better because it finds the best design once it reaches a terminal node. However, in the case of multimodality the Best First approach was always worse because the number of nodes that needed to be evaluated was significantly higher.

In general we can conclude from these results that an opaque approach is always better unless we have non-multimodal data and can apply a Best First search.

Table 4: Opaqueness Results, bolded rows performed better than their reversed order counterpart.

OPAQUENESS						
		Opaque Full MCTS			Non-Opaque Partial MCTS	
	TREE	Mean	std		Mean	std
Multimodal	M,Ch,V,B,Co	**655**	**358**	>	2425	1590
	Co,B,V,Ch,M	811	357	>	**2044**	**1608**
	B,Ch,Co,M,V	**480**	**338**	>	**2163**	**1081**
	V,M,Co,Ch,B	823	537	>	2659	1945
	B,M,Ch,V,Co	**531**	**423**	>	**894**	**784**
	Co,V,Ch,M,B	703	380	>	2271	797
Nonmultimodal	M,Ch,V,B,Co	480	395	>	**2992**	**2464**
	Co,B,V,Ch,M	**378**	**288**	>	3419	1690
	B,Ch,Co,M,V	864	343	>	3316	1734
	V,M,Co,Ch,B	**191**	**127**	>	**3180**	**2123**
	B,M,Ch,V,Co	**526**	**343**	>	2626	1880
	Co,V,Ch,M,B	601	423	>	**1425**	**1043**
		Expected FFE			Best First	
Multimodal	M,Ch,V,B,Co	720.5		>	18353	
	Co,B,V,Ch,M	720.5		>	**9798**	
	B,Ch,Co,M,V	720.5		>	**11902**	
	V,M,Co,Ch,B	720.5		>	19369	
	B,M,Ch,V,Co	720.5		>	**11513**	
	Co,V,Ch,M,B	720.5		>	12756	
Nonmultimodal	M,Ch,V,B,Co	720.5		<	60	
	Co,B,V,Ch,M	720.5		<	**54**	
	B,Ch,Co,M,V	720.5		<	**50**	
	V,M,Co,Ch,B	720.5		<	64	
	B,M,Ch,V,Co	720.5		<	**54**	
	Co,V,Ch,M,B	720.5		<	60	

3.1.3 Replacement

Table 5 presents the results of the replacement experiments. The left column presented the cases without modification decisions and the right column with modification. To study the effects of replacement, we modified each decision tree to include a modification decision at the end of the tree, which enables the modification of any prior decision in the tree.

For the non-multimodal cases, there was no difference between the trees with modification and the trees with no modification, which aligns with our expectations because a Best First search can find the best decision before a modification is needed.

For multimodal trees, we find that the algorithm performs as expected in five of the six tree orderings, with the trees without a modification option outperforming the modification trees. However, in the case of ascending choice, $T_{B,Ch,Co,M,V}$, the algorithm performed better on the tree with the modification. We think it's because the algorithm was able to change the first choice, which led to finding the best sandwich after reaching the first completed sandwich and then making a single modification to that sandwich. This demonstrates the potential appeal of modification because it will sometimes allow a designer to find the best-completed design earlier. However, in almost all cases the number of evaluations needed was increased by an order of magnitude by adding the modification decision. Additionally, almost all multimodal cases performed worse than the expected FFE or even max possible FFE, which is 1440 nodes.

We can conclude that including modification in a tree will not benefit DSE performance.

Table 5: Modification Experiment Results

MODIFICATION				
		Best First Search		
		No Modification		Modification
Nonmultimodal	M,Ch,V,B,Co	60	=	60
	Co,B,V,Ch,M	54	=	54
	B,Ch,Co,M,V	50	=	50
	V,M,Co,Ch,B	64	=	64
	B,M,Ch,V,Co	54	=	54
	Co,V,Ch,M,B	60	=	60
Multimodal	M,Ch,V,B,Co	18353	>	**60243**
	Co,B,V,Ch,M	**9798**	>	>100,000
	B,Ch,Co,M,V	**11902**	<	**84**
	V,M,Co,Ch,B	19369	>	>100000
	B,M,Ch,V,Co	**11513**	>	>100000
	Co,V,Ch,M,B	12756	>	>100000

3.2. The Satellite Instrument-Assignment Case Study

In the satellite instrument-assignment case study, we tested three hypotheses informed by the sandwich design toy problem results to validate our observations using a real-world problem. We treated the problem as opaque and multimodal. Table 6 summarizes our results.

The first hypothesis was that placing the decisions in increasing order of branching factor would improve performance which we found to be supported by our experiments so far ($p = 0.012$).

Our second hypothesis was that including modification would reduce performance which we observed in our results, but the effects were marginal ($p = 0.15$).

Our final hypothesis was that the negative impact of modification would be less than the negative impact of the worst expected sequence. Again we observed this to be supported by the evidence, but the result was not clearly significant ($p = 0.067$).

Table 6: Satellite Instrument-Assignment Result

Experiment			p-Value
V,C,Rio,Rar,B,A,O	>	O,A,V,C,Rio,Rar,B	**0.012**
V,C,Rio,Rar,B,A,O	>	V,C,Rio,Rar,B,A,O,Mod	0.15
V,C,Rio,Rar,B,A,O,Mod	>	O,A,V,C,Rio,Rar,B	0.067

4 CONCLUSION

In this paper, we presented a satellite mission design case study informed by exploration of a toy problem to examine the effects of the design decision characteristics of multimodality, opaqueness, and replacement on Design Space Exploration (DSE) using anisomorphic design decision trees.

Through the toy problem of designing a sandwich to meet the first author's personal preference, we found that when a design space is multimodal, sequencing the decisions in the design decision tree in ascending order of branching factor improved performance. Additionally, we found that we can expect that allowing for modification decisions will negatively impact DSE in all cases.

In the satellite design case study, we performed experiments to validate our observations by applying them to a real-world design problem of satellite instrument-assignment. We observed that these conclusions appeared to be valid, but the effects of modification were marginal.

In addition to the computational experiment, this paper also presents the novel concept of a Design Space Directed Graph (DSDG) that relates all anisomorphic design decision trees together.

4.1. Future Work

We plan to explore further the effects of design decision characteristics on DSE when using graphs to represent decisions. We hope to perform additional experiments to validate this paper's conclusion and see if these effects occur when we perform a search of the DSDG directly or if they are exclusive to the anisomorphic decision trees, which are subgraphs of the DSDG. Additionally, we hope to use the DSDG as a structure to observe human design decision makers' choices and enable new design automation and design assistants by leveraging its unique structure.

REFERENCES

[1] L. Nardi, D. Koeplinger, and K. Olukotun, "Practical design space exploration," in *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2019, pp. 347–358.

[2] M. Palesi and T. Givargis, "Multi-objective design space exploration using genetic algorithms," in *Proceedings of the tenth international symposium on Hardware/software codesign*, 2002, pp. 67–72.

[3] E. Kang, E. Jackson, and W. Schulte, "An approach for effective design space exploration," in *Monterey Workshop*, 2010, pp. 33–54.

[4] H.-Y. Liu and L. P. Carloni, "On learning-based methods for design-space exploration with high-level synthesis," in *Proceedings of the 50th annual design automation conference*, 2013, pp. 1–7.

[5] A.-R. Short, B. L. DuPont, and M. I. Campbell, "A Comparison of Tree Search Methods for Graph Topology Design Problems," in *International Conference on-Design Computing and Cognition*, 2018, pp. 75–94.

[6] A.-R. Short, "Autonomous Decision Making Facing Uncertainty, Risk, and Complexity," 2018.

[7] "Sub Sandwiches - Breakfast, Sandwiches, Salads & More | SUBWAY® | SUBWAY.com - United States (English)," *SUBWAY*. https://www.subway.com//en-US/ (accessed Mar. 09, 2021).

[8] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," in *Environmental Impact Assessment, Technology Assessment, and Risk Analysis*, Springer, 1985, pp. 107–129. Accessed: May 02, 2017. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-70634-9_6

[9] X. Zhang, T. Simpson, M. Frecker, and G. Lesieutre, "Supporting knowledge exploration and discovery in multi-dimensional data with interactive multiscale visualisation," *J. Eng. Des.*, vol. 23, no. 1, pp. 23–47, 2012.

[10] A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, "Multi-attribute tradespace exploration as front end for effective space system design," *J. Spacecr. Rockets*, vol. 41, no. 1, pp. 20–28, 2004.

[11] D. Selva Valero, "Rule-based system architecting of Earth observation satellite systems," PhD Thesis, Massachusetts Institute of Technology, 2012.

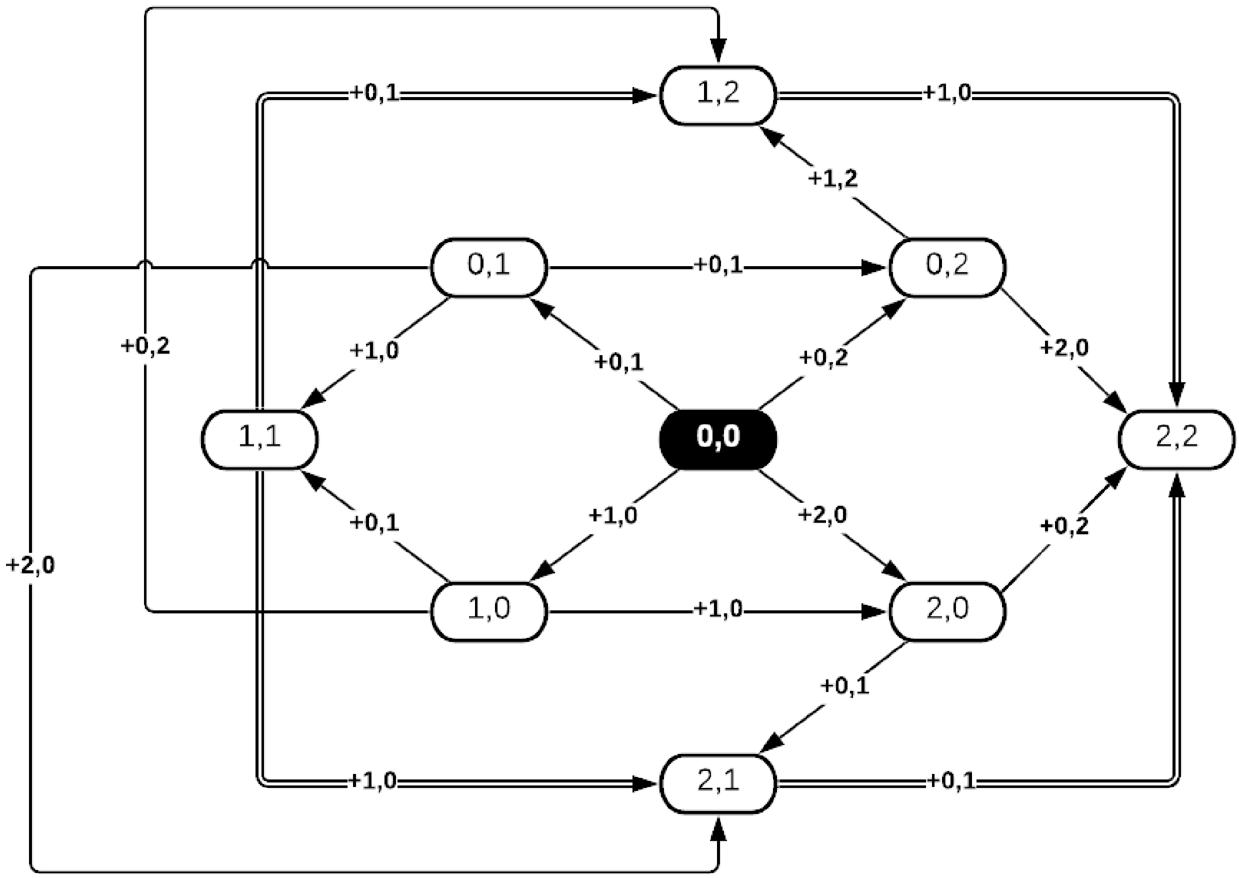
[12] D. Selva and E. F. Crawley, "VASSAR: Value assessment of system architectures using rules," in *2013 IEEE Aerospace Conference*, 2013, pp. 1–21.

[13] D. Selva, B. G. Cameron, and E. F. Crawley, "Rule-based system architecting of earth observing systems: earth science decadal survey," *J. Spacecr. Rockets*, vol. 51, no. 5, pp. 1505–1521, 2014.

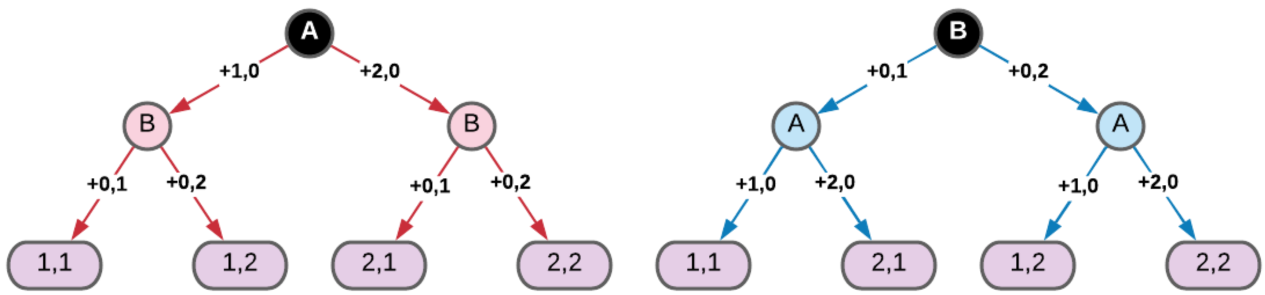
[14] "MATLAB - The Language of Technical Computing," Dec. 09, 2015.

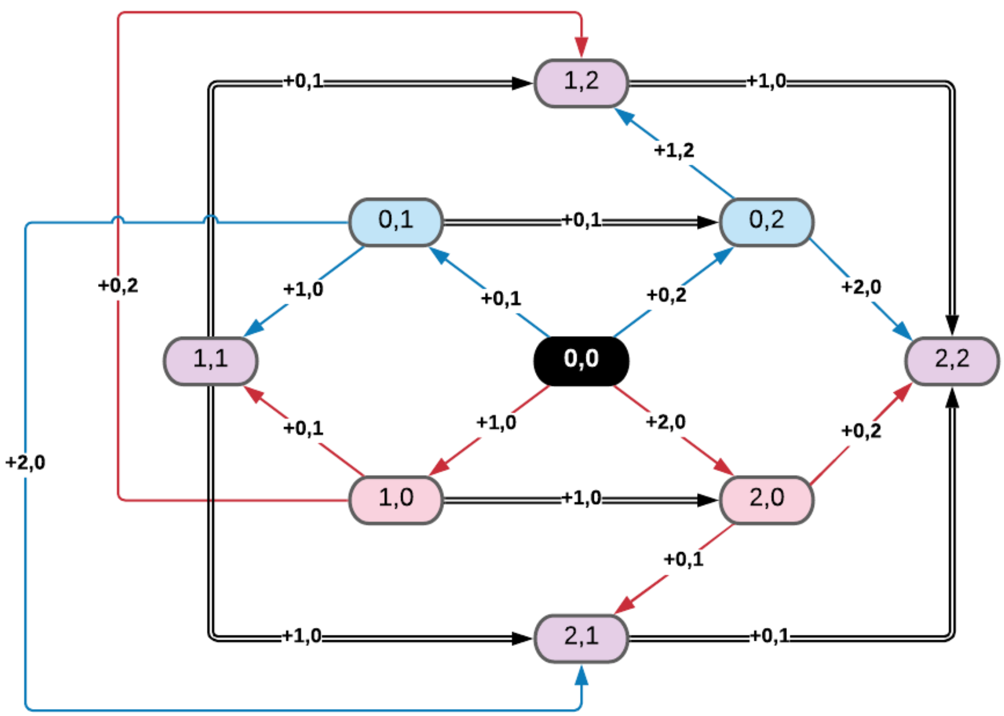
http://www.mathworks.com/products/matlab/ (accessed Dec. 09, 2015).
[15] "MATLAB App Designer - MATLAB." https://www.mathworks.com/products/matlab/app-designer.html (accessed Mar. 09, 2021).
[16] S. Pugh, "Concept selection: a method that works," in *Proceedings of the International conference on Engineering Design*, 1981, pp. 497–506.
[17] T. W. Knight, "Designing with grammars," 1991.
[18] D. Selva and P. Dutta, "Is There a Future for Geo-Based Weather Monitoring? the Coverage-Cost Argument," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 273–276.

APPENDIX A


1,2
+0,1
+1,0
+1,2
0,1
+0,1
0,2
+0,2
+1,0
+0,1
+0,2
+2,0
1,1
0,0
2,2
+0,1
+1,0
+2,0
+0,2
+2,0
1,0
+1,0
2,0
+0,1
+1,0
2,1
+0,1

APPENDIX B


A
+1,0
+2,0
B
B
+0,1
+0,2
+0,1
+0,2
1,1
1,2
2,1
2,2
B
+0,1
+0,2
A
A
+1,0
+2,0
+1,0
+2,0
1,1
2,1
1,2
2,2


1,2
+0,1
+1,0
+1,2
0,1
+0,1
0,2
+0,2
+1,0
+0,1
+0,2
+2,0
1,1
0,0
2,2
+2,0
+0,1
+1,0
+2,0
+0,2
1,0
+1,0
2,0
+0,1
2,1
+1,0
+0,1

APPENDIX C

MEAT			
WEIGHT	OPTIONS	SCORE	SCORE x WEIGHT
5	Ham	1	5
	Turkey	3	15
	Salami	2	10
	Corned Beef	0	0
	None Meat	-1	-5

CHEESE			
WEIGHT	OPTIONS	SCORE	RANK x WEIGHT
4	Cheddar	1	4
	Swiss	2	8
	Provolone	0	0
	None Cheese	-1	-4

VEGETABLE			
WEIGHT	OPTIONS	SCORE	RANK x WEIGHT
3	Lettuce	2	6
	Tomato	4	12
	Peppers	3	9
	Onions	1	3
	Sourkraut	0	0
	None Veggies	-1	-3

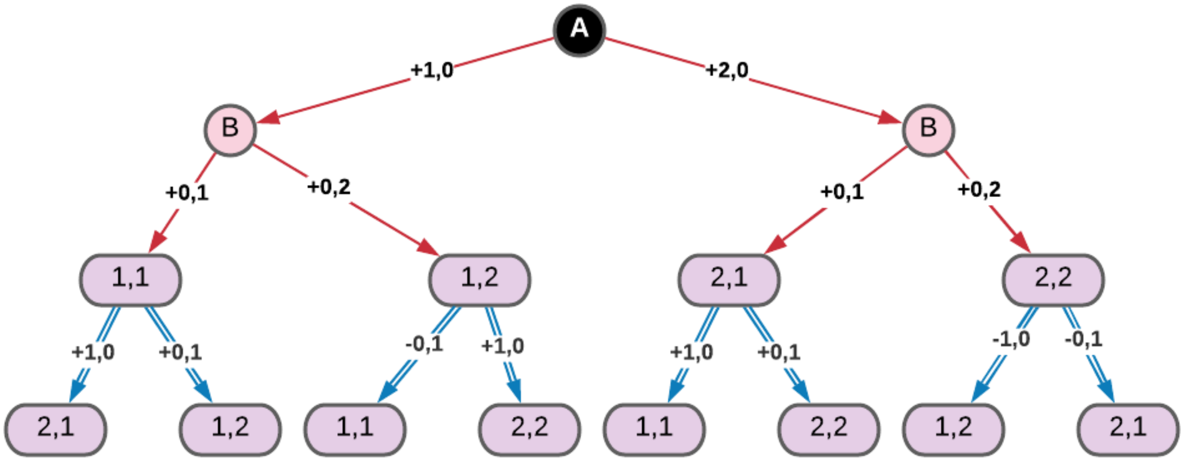
BREAD			
WEIGHT	OPTIONS	SCORE	RANK x WEIGHT
2	White	0	0
	Wheat	1	2
	Rye	-1	-2

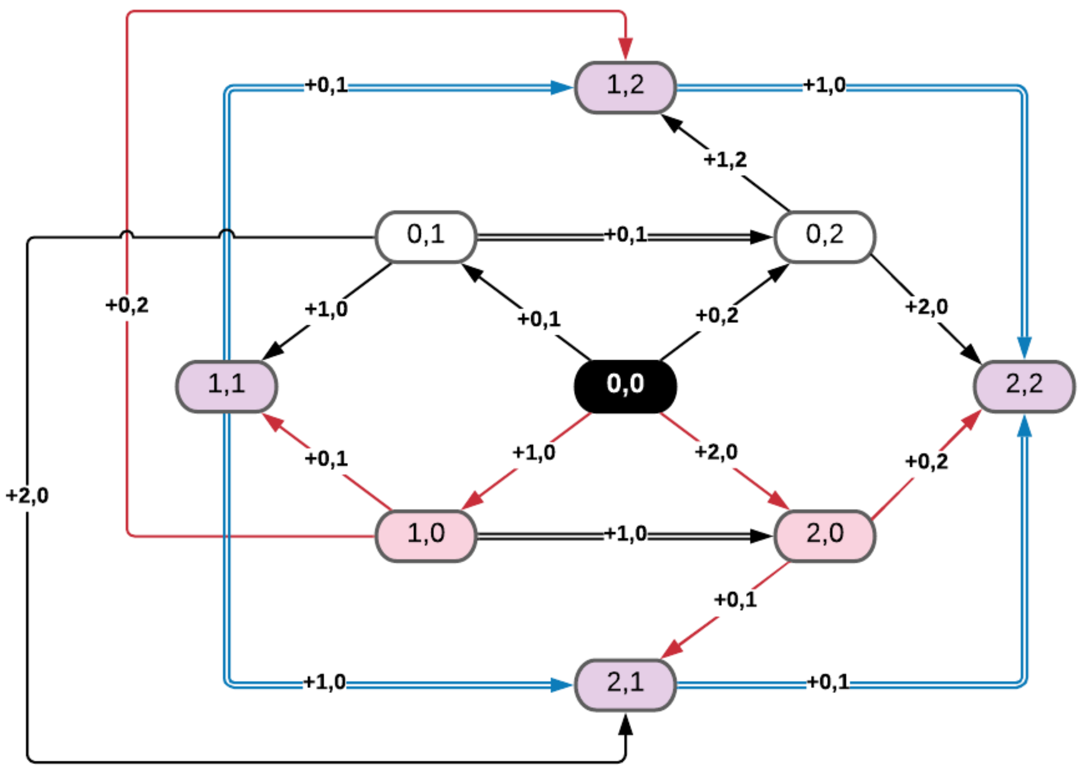
CONDIMENTS			
WEIGHT	OPTIONS	SCORE	RANK x WEIGHT
1	Mayo	1	1
	Mustard	2	2
	Russian Dress	0	0
	None Condim	-1	-1

APPENDIX D

	MEAT					CHEESE				VEGETABLE						BREAD			CONDIMENT			
	Ham	Turkey	Salami	*Corned Beef*	None Meat	Cheddar	*Swiss*	Provolone	None Cheese	Lettuce	Tomato	Peppers	Onions	*Sourkraut*	None Veggies	White	Wheat	*Rye*	Mayo	Mustard	*Russian Dressing*	None Condiments
Ham	10	20	15	-5	0	9	13	5	1	11	17	14	8	-4	2	5	7	-3.5	6	7	-3	4
Turkey		30	25	-5	10	19	23	15	11	21	27	24	18	-4	12	15	17	-3.5	16	17	-3	14
Salami			20	-5	5	14	18	10	6	16	22	19	13	-4	7	10	12	-3.5	11	12	-3	9
Corned Beef				***45***	-5	-4.5	***21.5***	-4.5	-4.5	-4	*-4*	-4	-4	***88***	-4	-3.5	-3.5	***38***	-3	-3	***48***	-3
None Meat					-10	-1	3	-5	-9	1	7	4	-2	-4	-8	-5	-3	-3.5	-4	-3	-3	-6
Cheddar						8	12	4	0	10	16	13	7	-3.5	1	4	6	-3	5	6	-2.5	3
Swiss							***24***	8	4	14	20	17	11	***18.5***	5	8	10	***9***	9	10	***15.5***	7
Provolone								0	-4	6	12	9	3	-3.5	-3	0	2	-3	1	2	-2.5	-1
None Cheese									-8	2	8	5	-1	-3.5	-7	-4	-2	-3	-3	-2	-2.5	-5
Lettuce										12	18	15	9	-3	3	6	8	-2.5	7	8	-2	5
Tomato											24	21	15	-3	9	12	14	-2.5	13	14	-2	11
Peppers												18	12	-3	6	9	11	-2.5	10	11	-2	8
Onions													6	-3	0	3	5	-2.5	4	5	-2	2
Sourkraut														***42***	-3	-2.5	-2.5	***34.5***	-2	-2	***45***	-2
None Veggies															-6	-3	-1	-2.5	-2	-1	-2	-4
White																0	2	-2	1	2	-1.5	-1
Wheat																	4	-2	3	4	-1.5	1
Rye																		***-2***	-1.5	-1.5	***3.5***	-1.5
Mayo																			2	3	-1	0
Mustard																				4	-1	1
Russian Dressing																					***5***	-1
None Condiments																						-2

APPENDIX E


A
+1,0
+2,0
B
B
+0,1
+0,2
+0,1
+0,2
1,1
1,2
2,1
2,2
+1,0
+0,1
-0,1
+1,0
+1,0
+0,1
-1,0
-0,1
2,1
1,2
1,1
2,2
1,1
2,2
1,2
2,1


+0,1
1,2
+1,0
+1,2
0,1
+0,1
0,2
+0,2
+1,0
+0,1
+0,2
+2,0
1,1
0,0
2,2
+0,1
+1,0
+2,0
+0,2
+2,0
1,0
+1,0
2,0
+0,1
+1,0
2,1
+0,1