

# ThunderPelican: An Interactive Food Web Display

Zhangbo Liu\*

Department of Computer Science  
University of British Columbia  
2366 Main Mall  
Vancouver BC Canada V6T 1Z4

## ABSTRACT

Visualizing food web is an interesting and challenging task because of the high complexity of the food web. Creating diagram in existing ecosystem modeling applications such as Ecopath is extremely tedious and time-consuming. In this paper, we introduce the ThunderPelican, an interactive food web display system by which we could easily and efficiently generate interactive food web diagram. An algorithm is proposed to reduce edge crossing. Some other novel designs and ideas to draw food web diagrams, although have not been implemented into this version of our prototype, are also presented. We also ran an informal user evaluation of the ThunderPelican and collected mixed evaluations.

## 1 INTRODUCTION

Food web diagram, which is pervasive in ecology, is a directed graph in which nodes represent species or groups of species and edges represent trophic interactions (eating). Figure 1 shows a simple food web. When there are many species in a model, the food web may become difficult to draw.

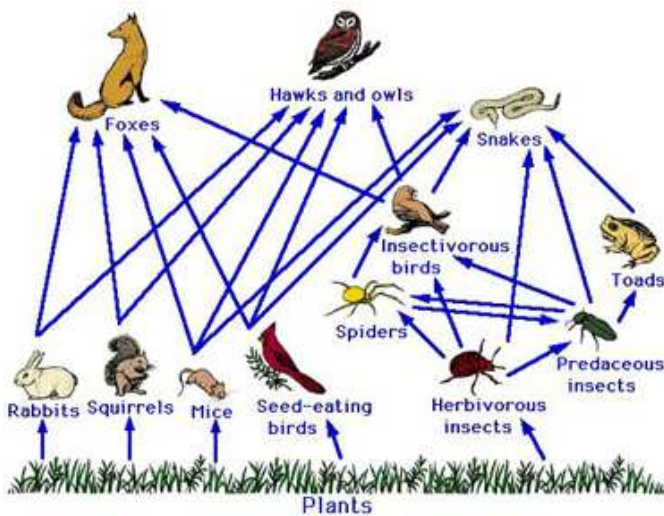


Figure 1: A Simple Food Web.

A number of systems and tools have been developed to help draw the food web diagram and Ecopath (shown in Figure 2) is one of them. Ecopath employs a circuit diagramming model which was started used by an ecologist to draw food web diagrams some decades ago. We found two key issues after interviewing users to

\*e-mail: zephyr@cs.ubc.ca

determine the major problems with the current Ecopath. First, it was a very tough and time-consuming work to create the diagram with current system. Second, the complexity of the information presented for large models makes the diagram difficult to read.

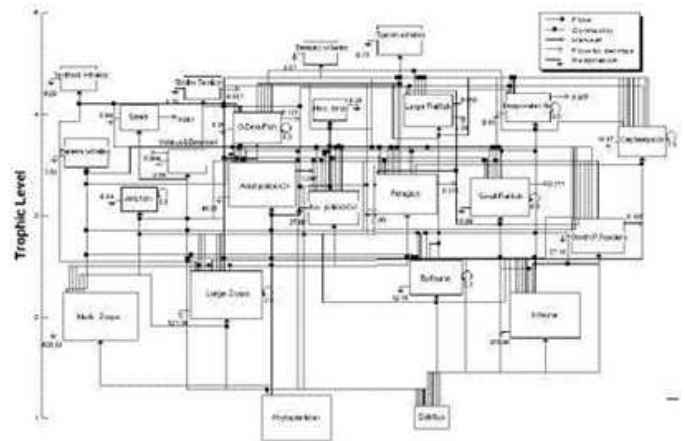


Figure 2: Ecopath Diagram

We propose a series of novel designs and ideas that might improve on both of these issues. Our motivating insight is that most of the biomass in an ecosystem is concentrated at the bottom, i.e., the total weight of all the plankton in the ocean is many orders of magnitude greater than the total weight of all the whales. Moreover, the diversity of life at the bottom far exceeds that at the top. Since food web diagrams are organized hierarchically by trophic level, this should create a concentration of information at the bottom of the graph, leaving substantial white space near the top. Based on this, we proposed circular layouts (See Section 3.1). Unfortunately, this strategy fails in practice because users are generally not interested in species at the bottom of the food chain, so these species are shown as just one or two large groups while higher-level species are broken into many smaller groups, giving the impression that the biomass and diversity are concentrated in the middle.

We expand on the prototype that received the most favorable feedback: the one organized hierarchically by trophic level and then added features to which users ascribed the greatest importance to build the current version of ThunderPelican which received mixed user evaluations.

The rest of this paper is as follows: In Section 2 we briefly describe related work in this area. Following this, in Section 3 we introduce several alternative solutions we proposed to our task then in Section 4 we concentrate on describing current version of ThunderPelican. Section 5 presents the results of user evaluation of ThunderPelican, while Section 6 presents the future work that needs to be done to improve its performance. The final section concludes.

## 2 RELATED WORK

Ecopath with Ecosim (EwE)<sup>1</sup> is a complex ecological modeling software suit that has been developing for more than a decade. It has three main components: Ecopath—a static, mass-balanced snapshot of the system; Ecosim—a time dynamic simulation module for policy exploration; and Ecospace—a spatial and temporal dynamic module primarily designed for exploring impact and placement of protected areas. One of the main functions of Ecopath shows trophic relationships (essentially what eats what and in what amounts, in an ecosystem) using food web diagram. From conversations with its users, we found that they complaint it mainly on two points. First, it takes much time to draw such a diagram. Figure 3 depicts the initial layout of Ecopath. Users have to manually drag each group to some appropriate positions and manually adjust the whole graph to avoid overlapping (See Figure 4). Sometimes it even cost users a couple of days to do that. Second, it provides all information into one graph, which makes the readability of the graph too low to be accepted by novice users. In addition, according to users' experiences, when the number of groups shown in one graph exceeds 20, the graph will become quite difficult to read.

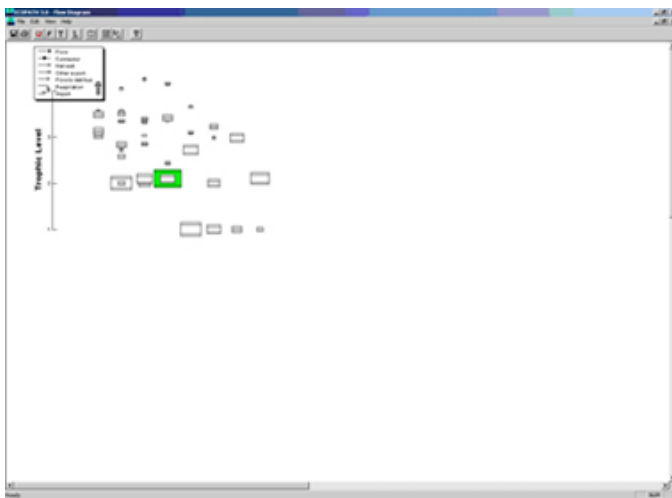


Figure 3: Initial State of Ecopath

A more recent food web diagram is created by 3D MDS REGE Dynamic Visualization<sup>2</sup> (See Figure 5). It is a three-dimensional graph that can be rotated on all three axes via mouse movement. This graph does not show essential information on food web such as group name, trophic level or interaction strength. Biomass is indicated by the size of the group symbol (the circles). Diagrams made in this fashion do not print well and must be annotated by hand to make them meaningful.

## 3 ALTERNATIVE PROPOSED SOLUTIONS

At the beginning of our work, we did the first round prototyping and proposed some solutions. Then we interviewed with some users of Ecopath and got some feedbacks for each proposed solution.

### 3.1 Circular Display

One way to make better use of screen space is to use circular layouts instead of traditional up-down or left-right orientations [2]. This

<sup>1</sup><http://www.ecopath.org>

<sup>2</sup><http://drjoe.biology.ecu.edu/AERS/index.html>

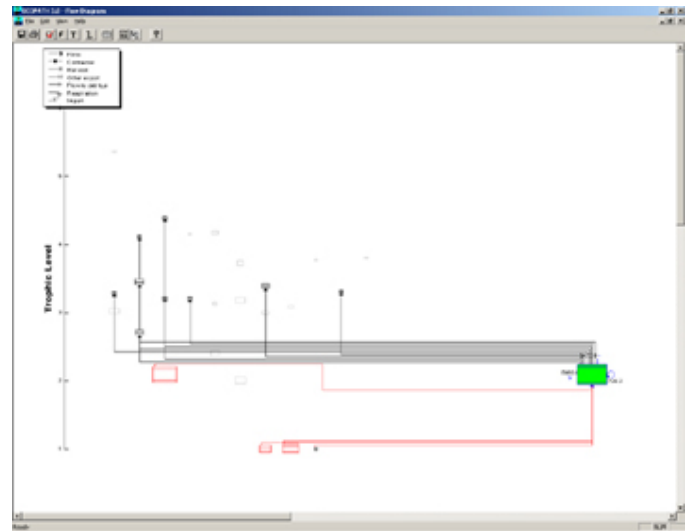


Figure 4: Node Allocation of Ecopath

seems especially relevant to food web diagrams because the creatures on the bottom of the food chain are so much more numerous than those on the top. In order to show more groups within one diagram, we first proposed a circular display with the high trophic level groups in the center, and lower level groups on concentric circles, as shown in Figure 6<sup>3</sup>. Note that our circular graph is not a "radial drawing" [1], since it is not tree rooted and there is no focus node in the center. After implementing the prototype, we quickly realized that the line density in the center was very high in several models. However, that's not the most important issue. Users mainly concerned about two problems though they seemed to understand the idea behind the diagram. First, biologists view an ecosystem as a hierarchical structure. Their mental model has predators at the top and prey at the bottom. Although the circular graph shows the same information, it appears in a format entirely inconsistent with the viewers' conceptualization of the information. Users set firmly in this top-down paradigm strongly resist this type of display. Second, we did not fully appreciate the difference between the structure of an ecosystem and the structure of the model of the ecosystem. Despite the diversity and magnitude of small organisms over large ones, most modelers are unconcerned with the former. This results in models with many groups at high trophic levels and few groups at low levels - the opposite of what is being modeled. In other words, in the ecosystem, almost everything is near the bottom, while in the model, almost everything is in the middle or near the top. This creates serious problems for our circular display. Aside from bashing the general concept, however, one user indicated that circular group symbols would be better than square ones for both aesthetic and functional reasons.

We also added an interactive feature in this and all other prototypes that we thought would be helpful to users. This allows users to highlight all of the interaction lines related to a selected group. In our demo, when users hover mouse on a group, the lines would be highlighted in red, and the thickness of each line was scaled to the amount of the interaction it represented.

In an attempt to make the graph neater, we tried classifying groups into whole-number trophic levels (See Figure 7). By rounding the trophic levels, groups appeared on consistent lines. It seemed to us that made interactions between levels clearer. One of our users informed us that an old argument existed in biology about whether trophic level was a discrete or random variable. Ap-

<sup>3</sup>Here we only encoded one variable in early prototypes.

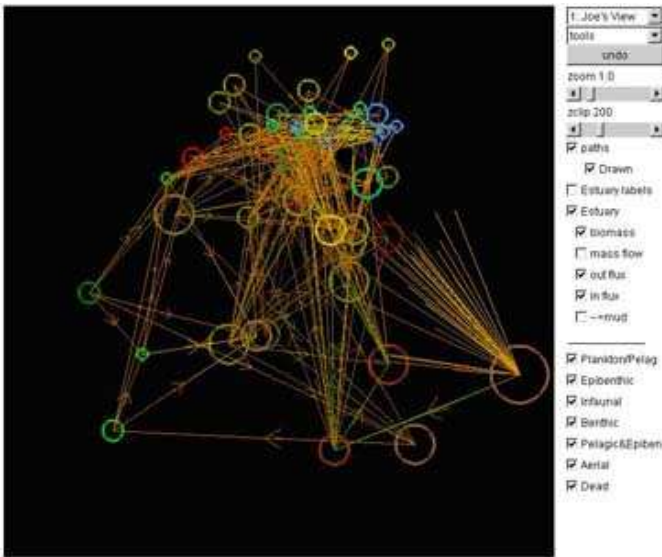


Figure 5: 3D Display

parently, those in favor of discrete lost a long time ago. Besides the problems with the circular display described in the previous section, users immediately rejected this diagram on the basis that rounding trophic levels is inherently misleading.

### 3.2 Pie Display

During the development process, we serendipitously created the pie diagram, so named because it is shaped like a piece of pie, shown in Figure 8. Again, this is similar to the Circular Diagram except the entire graph is compressed into a 90 degree angle. In a sense, this option is a hybrid between the conventional and circle displays. This graph has the same line density issue as the circular graph.

We found the rounded pie diagram clarified interactions between trophic levels, but many of the group names overlapped. User reactions to this diagram were mixed. One user pointed out that no information was really gained in this approach, but nothing was lost either, so it could be useful in some situations. Another user liked how the connections between levels are clarified in the rounded display (Figure 9) despite earlier misgivings about rounding in principle. As with other diagrams, users pointed out that overlapping text and connections confused the diagram.

### 3.3 Conventional Display

This prototype is similar to the current version of the Ecopath Flow Diagram. However, we suspected that the size of the boxes took up too much space, so we switched to smaller symbols wherein biomass was represented by the color of the box instead of the size. Figure 10 shows the layout of this prototype. As we suspected, the diagrams did look significantly less cluttered with the smaller boxes. However, the straight lines become problematic when three groups are on the same line. That may cause some misunderstanding on which groups are connected.

All users preferred this prototype over the others, and over the original Ecopath diagrams. However, they had some reservations which we'll discuss in next section.

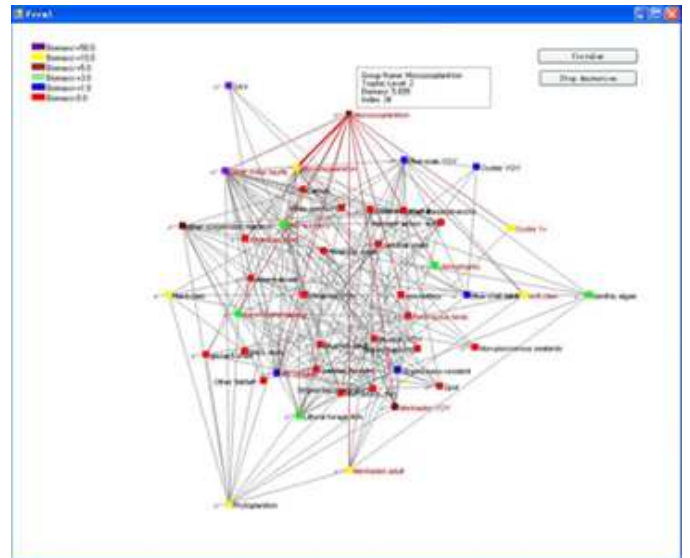


Figure 6: Circular Display

## 4 THUNDERPELICAN

ThunderPelican is built based on the conventional display we proposed in the last section and at the same time we carefully consider the feedback we got from user interview of the first round prototyping. Figure 16 shows an initial layout of ThunderPelican. It can generate the flow diagram automatically for users and provide an algorithm to optimize the layout. It also allows users to custom information encoding methods according to their preference. A number of interactive technologies in Information Visualization were integrated into ThunderPelican.

### 4.1 Information Encoding

According to [4], using graphical elements to encode data can make patterns apparent thus users are able to understand and manipulate the data much easier. In current version of Ecopath, it shows several kinds of information such as biomass, diet, respiration and harvest. However, all information stuck to one graph which makes it difficult for users to find information that they want through visual perception. ThunderPelican mainly provides two kinds of information: biomass and diet because of two reasons: First, information provided in Ecopath is separated in different dimensions and independent to each other; Second, we didn't get other data from the dataset which we are using currently.

#### 4.1.1 Biomass

We propose two ways to encode biomass (which represents the population of one group of fish) both in node size and color. The effect of representing biomass with node size can be seen in Figure 16. While Figure 17 shows how we used color to represent biomass information. We did this because users have different preferences. For an instance, during our first round user interview, one user mentioned that showing biomass with different sizes of nodes might be very helpful, while another user highly preferred to use color because he is more sensitive to colors.

#### 4.1.2 Diet

Also, in order to satisfy our users, we leveraged two methods, color and line width, to represent diet information. From Figure 16 and

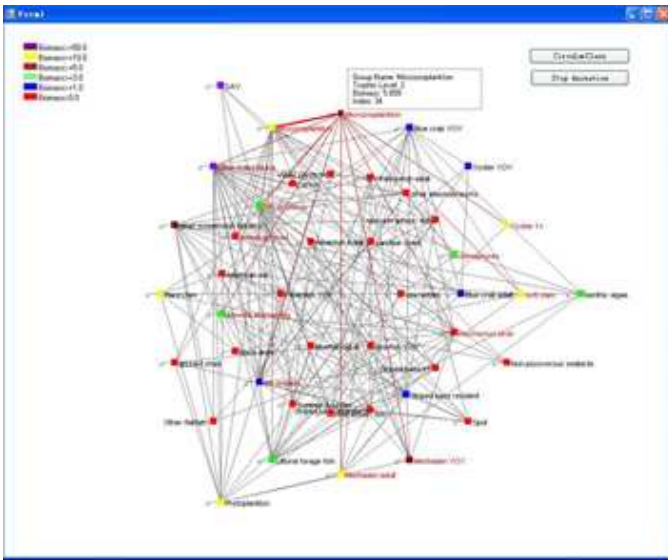


Figure 7: Rounded Circular Display

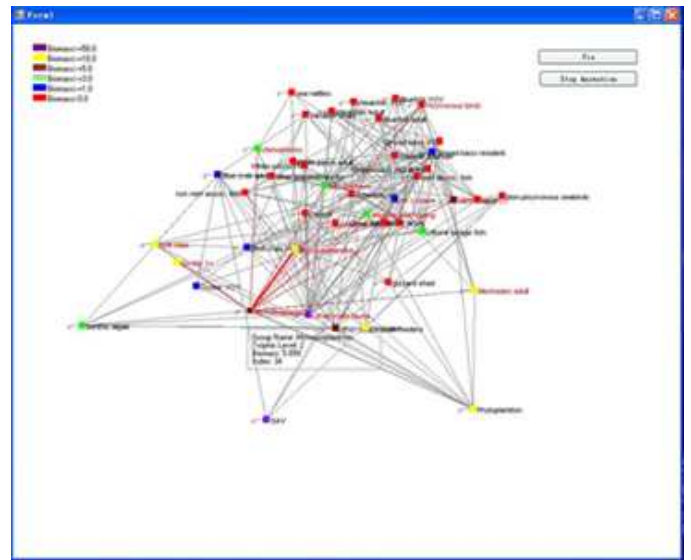


Figure 8: Pie Display

17 we could see the graph layout encoding diet with line width and color respectively. Figure 11 shows the legends of biomass and diet shown with sizes.

Basically, users are allowed to select any of the encoding methods above to custom their own diagram layout and switch to others at any time if necessary. Currently we allow users to select from two encoding methods for the same variable at the same time but we don't encourage this because redundantly showing the same information is unnecessary and distracting. In addition, users are allowed to select none of those encoding choices.

We believe that if users are satisfied with the encoding of biomass and diet in this way, then it will be not difficult to encoding any other potential information into the diagram. Because different information is independent to each other, there is no problem that we show some of them at one time and show others at another time, and then we don't need to import many new encoding methods.

## 4.2 Edge Crossing Reduction

Edge crossing is an important problem in graph drawing. In [6], the authors proposed an interactive lens system to distort and hide overlapped edges, while [5] introduced a simple method to fade those edges that connected children with a parent other than currently focused one in a genealogical graph. However, they are not helpful to our work for the following two reasons: 1) We want a global neater layout instead of single region; 2) A food web is more complicated than a family tree. In addition, minimizing overlapping itself is NP-hard [3]; therefore we have to use a heuristic approach to improve graph clarity.

### 4.2.1 Algorithm

We use Crossing Value as a measure of graph complexity and we propose a mechanism to calculate the Crossing Value of each graph we generated. The crossing value is calculated from 3 cases in Figure 12.

- CASE I: Edge crossing
- CASE II: Edge-Node crossing
- CASE III: Node closing

CASE I is easy to understand, here we explain CASE II & III. CASE II is when a node occludes on an edge, it is misleading because we don't know whether the edge comes out from the node or

not and whether the node has interaction with those two nodes with which the edge connected. III is a worse case based on II because when two nodes are close to each other there might be more than one edge comes out from them that go across each other. Based on analysis above, we proposed the following formula to calculate the Crossing Value:

$$CrossingValue = \sum_{i=1}^3 k_i w_i$$

Here  $k_i$  ( $i=1,2,3$ ) represents number of each CASE;  $w_i$  ( $i=1,2,3$ ) represents weight of each CASE.

After calculating the Crossing Value out, we could try reducing it by move nodes as well as edges of the graph so as to improve the graph clarity. However, in practical experiment, we found that it didn't bring much benefit by singly changing node's position at one time. Then we proposed another method which randomly chose a pair of nodes and switched their x-coordinates (y-coordinates were fixed). This method was comparably better than singly moving one node and then another simply because it changed much of the graph thus we might have more chances to get some local optimal results. The optimizing is a hill climbing process. In an optimizing loop, we first randomly choose two nodes and switch their positions, and then we calculate the current Crossing Value, if the Crossing Value reduced, we confirm this switch, otherwise we discard the switching. Users are allowed to change the times of iteration in the control panel according to their requirement (See Figure 13).

### 4.2.2 Performance

Since we had a good initial state of the graph layout in which nodes were scattered over the graph instead of all stuck together, we didn't worry about the effect of using switching instead moving nodes. We tested the program with at most 1000 iterations. Figure 18 shows a layout after 1000 iterations. Data analysis is shown in Figure 14 and Figure 15:

From the test results we could see that based on the given mechanism, the Crossing Value, which represented the edge crossing, was reduced a lot by running the optimizing program. We could also find that the Crossing Value reduced as we increased the times of iterations. However, from Figure 14 we could see that it has a

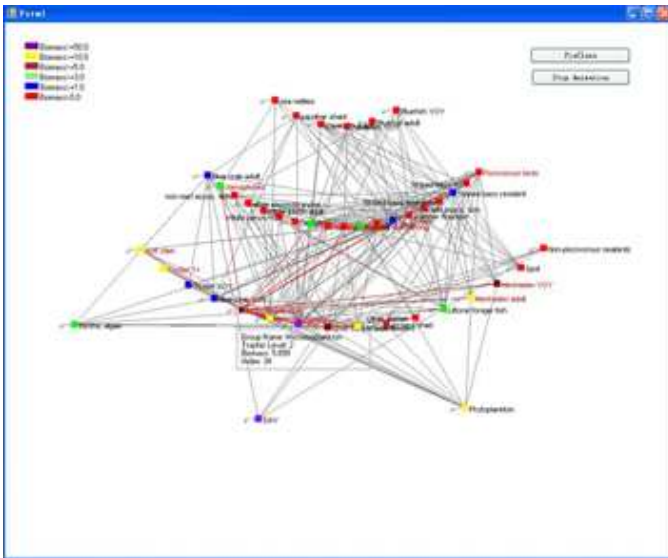


Figure 9: Rounded Pie Display

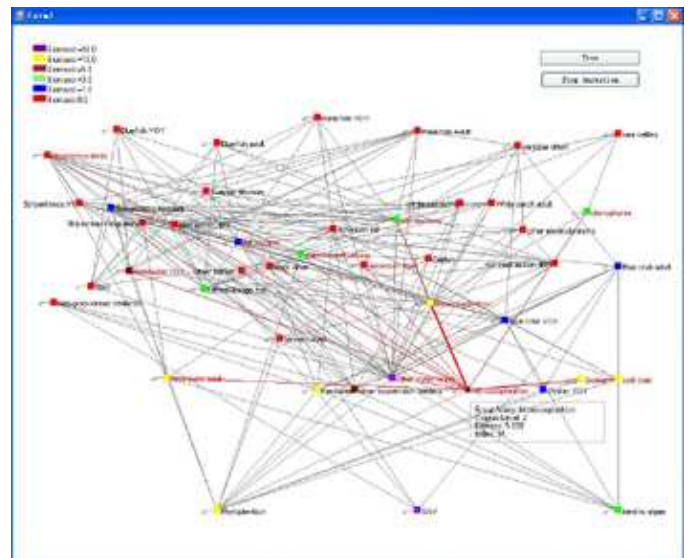


Figure 10: Conventional Display

threshold which is a limit. Beyond the limit, the program will be hard to get any progress. That's reasonable because those nodes and edges were still there even we change their positions, and that's why we find the clutter of the graph still remains! But from numerical analysis, the edge crossing has been reduced.

### 4.3 Direct Manipulation

Instead of using the program to reduce the edge crossing, users are allowed to manually change the position of the node horizontally by dragging and dropping. When users hover a mouse on a specific node, detail information of that node such as group name, biomass, trophic level will be shown on the middle top of the diagram. When users click on the node, all interactions the node has with other nodes will be highlighted so that users can clearly see the diet information of that node (See Figure 19). In addition, we set a combo box from which users could find any node in the diagram by group name. We ordered the list by trophic level for that might be easier for users to locate specific node. All features above are easy to manipulate as to make users to feel that they are touching the data so that their level of engagement to the system might be maintained.

### 4.4 Miscellaneous

We also implement some features that we think might be helpful to users. We allow users to switch the node type of the diagram between circle and rectangle (See Figure 17). Users were allowed to decide whether to show all information we got from the dataset in the graph, just like the Ecopath. There are two styles of lines that users can select: straight and arc. We were also planning to integrate the traditional right-angle style but were running out of time. Users could also eliminate all edges in the graph and only show the edges which connected to the highlighted node so as to make the graph neat clear. These features can be seen in Figure 20. In addition, we allow users to save the current graph to flow diagram file (.fd) and load existing flow diagram files so that they won't need to start from the beginning every time.

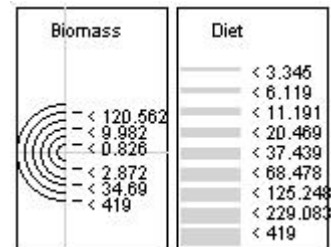


Figure 11: Size Legends

### 4.5 Implementation

ThunderPelican is implemented using VB.net in order to be integrated into current Ecopath later. Approximately 3500 lines of commented code were written to create ThunderPelican; the dataset and color legend module are provided by Ecopath developer team from Fishery Centre at University of British Columbia<sup>4</sup>. Since current Ecopath is a large system with tens of thousands lines of code that we couldn't work directly on it, we started our work all from scratch. During the whole procedure we developed several versions of ThunderPelican (some of them were presented in Section 3) and obtained much precious experience that helped us to build the current version.

## 5 EVALUATION

We ran the second round informal user evaluation after finishing our final version prototyping. Three researchers from Fishery Center at UBC who regularly draw food web diagrams participated in the evaluation.

Our primary hypotheses, stated in general terms, are as follows. We omit the null hypothesis because it would be entirely pretentious to reject the notion that users perceive no difference in efficiency or effectiveness between two dissimilar systems.

H1 ThunderPelican is more effective than current facilities in Ecopath  
H2 ThunderPelican is more efficient than current facilities in Ecopath

<sup>4</sup><http://www.ecopath.org/index.php?name=About&sub=Contacts>

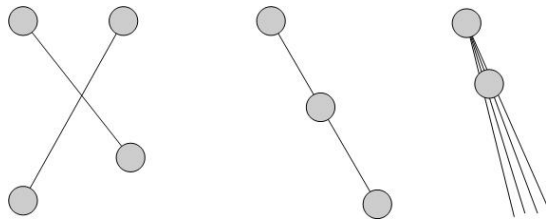


Figure 12: CASE I, II and III



Figure 13: Control Panel

Using the questionnaire we administered, users rated Ecopath and ThunderPelican on 11 dimensions, as shown in Table 1. The questionnaire was constructed from 1 to 9 scales such that higher numbers indicate greater satisfaction. In the Average Difference row, a positive number indicates preference for ThunderPelican; a negative number indicated preference for Ecopath. These difference are not rounded, they just all happened to be whole numbers. Under satisfactory diagrams, User 3's data was not used because that user left out that question.

### 5.1 Strengths

The most important result is that users reported higher satisfaction with diagrams created using the ThunderPelican. These results also indicate that users felt that the ThunderPelican produced more aesthetically pleasing diagrams.

In general, users appreciated the features available, especially variations in line style, although they clearly required additional features for the software to be practical. Although no users stated explicitly that they liked the flexibility of features, different users expressed preference for different options, supporting our belief that the options were necessary to maximize satisfaction.

Users estimated that creating a reasonable diagram based on the 45-node model we provided using Ecopath would take between 4 hours and a couple of days. In contrast, users indicated that they could create a reasonable diagram using ThunderPelican in "a matter of minutes" to "a couple of hours." This variation was not surprising because users have different proficiencies and standards. It seemed that the automatic node placement would greatly decrease the number of clicks needed to complete the diagram. This supports hypothesis H2.

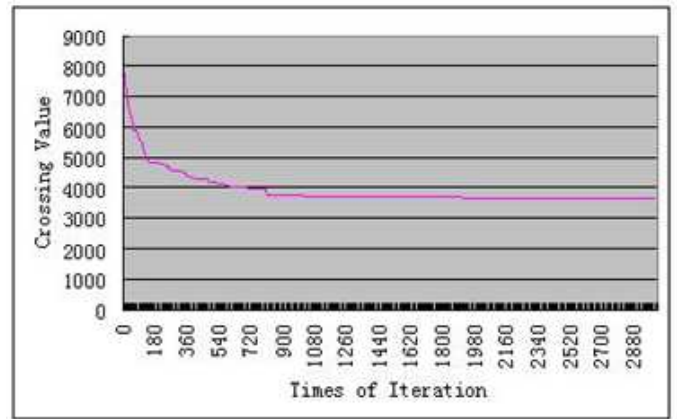


Figure 14: Test Results

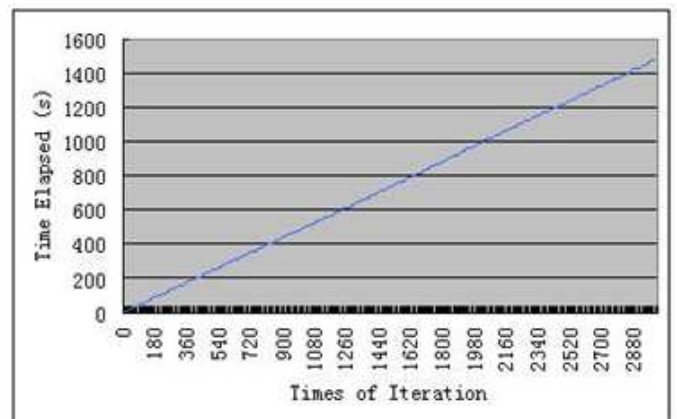


Figure 15: Test Results

### 5.2 Weakness

Users explicitly indicated that the ThunderPelican diagram would be better for presentation and the Ecopath diagram would be better for publications, this directly contradicts the result of the survey. Users agreed that the Ecopath diagram showed more information than the ThunderPelican diagram. In addition, those participants seemed to imply that the Ecopath diagram was generally better than the ThunderPelican, again contradicting the questionnaire results described above. Users described the ThunderPelican diagrams as "unprofessional" because they were used to the circuit diagram in ecosystem which has been existing for decades.

A weakness that both we and users noticed is the overlapping of text label. Since some group have extremely long name and their position vary a lot, it is very complicated to solve this problem during our project. We've tried some approaches but couldn't avoid all overlapping cases.

Based on the above discussion, we report mixed support for hypothesis H1.

Another weakness of ThunderPelican users didn't notice but we know is the color legend. Currently the color legend module we got contains 500 legends which are too long to be shown in limited space. That's why some colors e.g. light blue could not be seen in the legend.

Table 1: Questionnaire Results

User	System	Meets needs	Easy to read	Publications	Presentation	Clear	Aesthetic
1	Ecopath	2	4	4	4	3	3
	ThunderPelican	5	5	7	5	7	7
2	Ecopath	7	8	8	6	8	6
	ThunderPelican	5	6	7	5	7	6
3	Ecopath	6	6	4	7	5	4
	ThunderPelican	6	4	7	4	4	7
Average Difference		0	-1	2	-1	1	2

User	System	Simple	Understandable	Satisfactory diagrams	Flexibility	Features
1	Ecopath	3	4	2	2	3
	ThunderPelican	5	7	7	5	5
2	Ecopath	6	8	6	5	4
	ThunderPelican	5	7	5	5	4
3	Ecopath	5	6	5	7	7
	ThunderPelican	6	6	5	7	7
Average Difference		1	1	2	1	1

### 5.3 Lesson Learned

Over the project, we learned several things:

- Survey prior to the developing process is very important. Like other user-centered designs, visualization systems must satisfy users' requirements, not what developers thought.
- Version control is essential if the members of the team are working on the code simultaneously.
- The contradiction between the results of interview and questionnaire indicated some problems of our questionnaire. It would be helpful if we force choice on each dimension instead of asking users to rate both systems respectively, especially for a small number of users.
- We realized the importance of the circuit diagram information visualization paradigm held by the users. Whether or not this schema is actually the best way of depicting ecosystems, it is clear that the users will resist a new paradigm. That's probably why users described our prototype as "unprofessional".

### 6 FUTURE WORK

Based on the above analysis, we make several recommendations for future work. First, as it is unclear that any advantage accrues from changing to a general graph paradigm, and any deviation from the circuit drawing paradigm seems to meet user resistance, we recommend sticking with the original style while providing users several other options. Second, more information should be added into current version of our prototype to satisfy users different requirements. In addition, the crossing checking mechanism used by the algorithm is crucial to success. In current version of ThunderPelican, the mechanism only contains a measure of overlapping of nodes and edges. To this, we should add several more dimensions of graph clarity, including overlapping of labels, node dispersion and edge length. Other criteria may become apparent during the design process.

### 7 CONCLUSION

We develop the ThunderPelican food web visualization tool to improve current Ecopath system. Many visualization techniques are investigated during the development. We tried to propose a novel graph drawing paradigm to visualization ecosystems. However, users seemed reluctant to use the new layout because they were more familiar with the current version and it seems to be a default standard. Though ThunderPelican may not be used in any commercial application, many of the things we learned are likely to influence the development of the Ecopath software. Significant progress has been made regarding a graph layout algorithm which may be applied in this and other tools. Finally, we identified a number of features that many Ecopath users may appreciate.

### REFERENCES

- [1] Eades P. Tamassia R. Battista, G. Di and I. G. Tollis. Graph drawing: Algorithms for the visualization of graphs. In *Upper Saddle River, N. J: Prentice Hall*, 1999.
- [2] J. Bertin. Semiology of graphics, diagrams, networks, maps. In *University of Wisconsin Press, Madison, WI*, 1983.
- [3] I. Cruz and Tamassia. Graph drawing tutorial (online). In <http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf>, 2005.
- [4] Stephen G. Eick and Graham J. Wills. High interaction graphics. In *European Journal of Operational Research*, volume 81, pages 445–459, 1995.
- [5] Michael J. McGuffin and Ravin Balakrishnan. Interactive visualization of genealogical graphs. In *Proc. InfoVis 2005*, 2005.
- [6] Saul Greenberg Nelson Wong, M. Sheelagh T. Carpendale. Edgelens: An interactive method for managing edge congestion in graphs. In *Proc. InfoVis03*, pages 51–58, 2003.

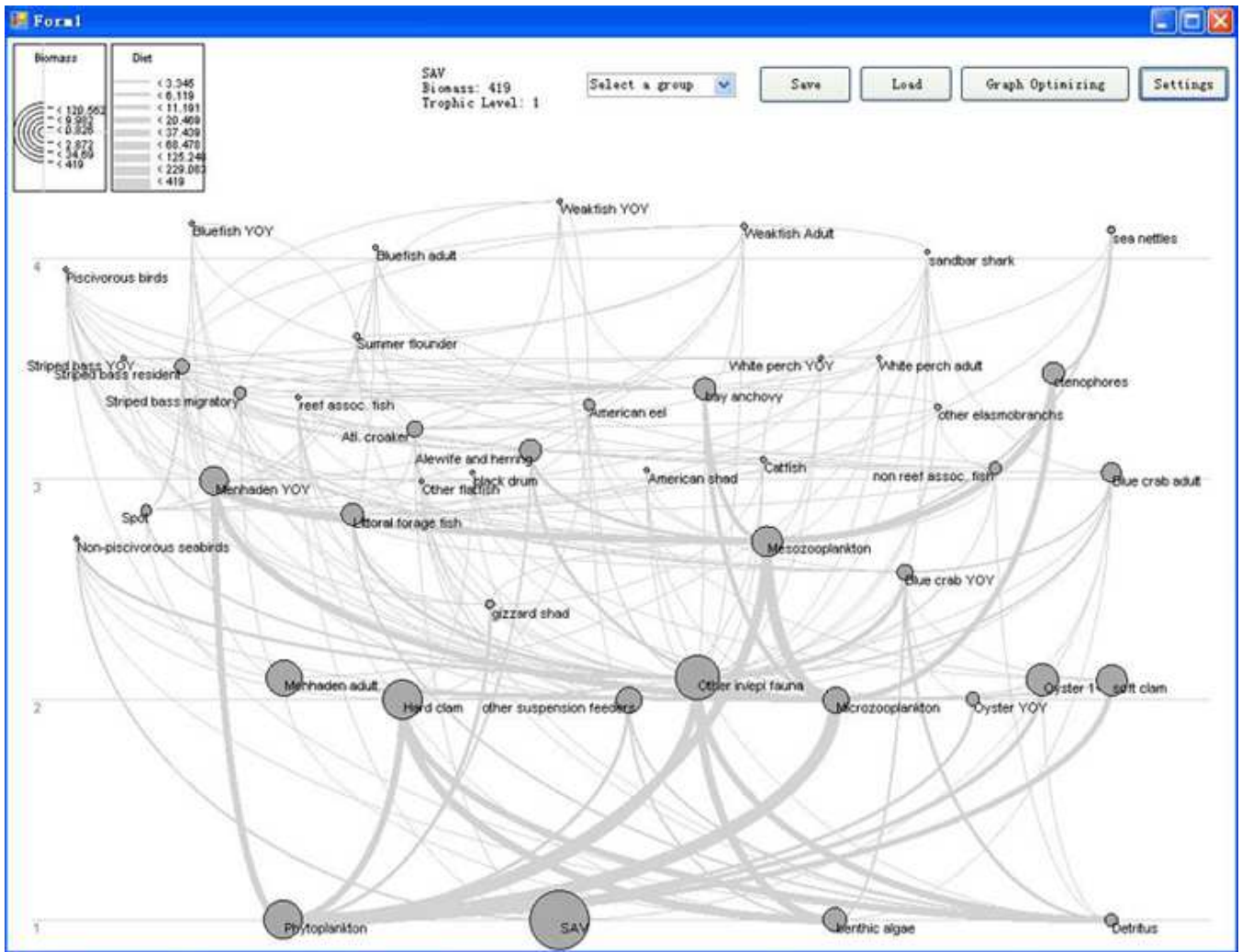


Figure 16: ThunderPelican 1



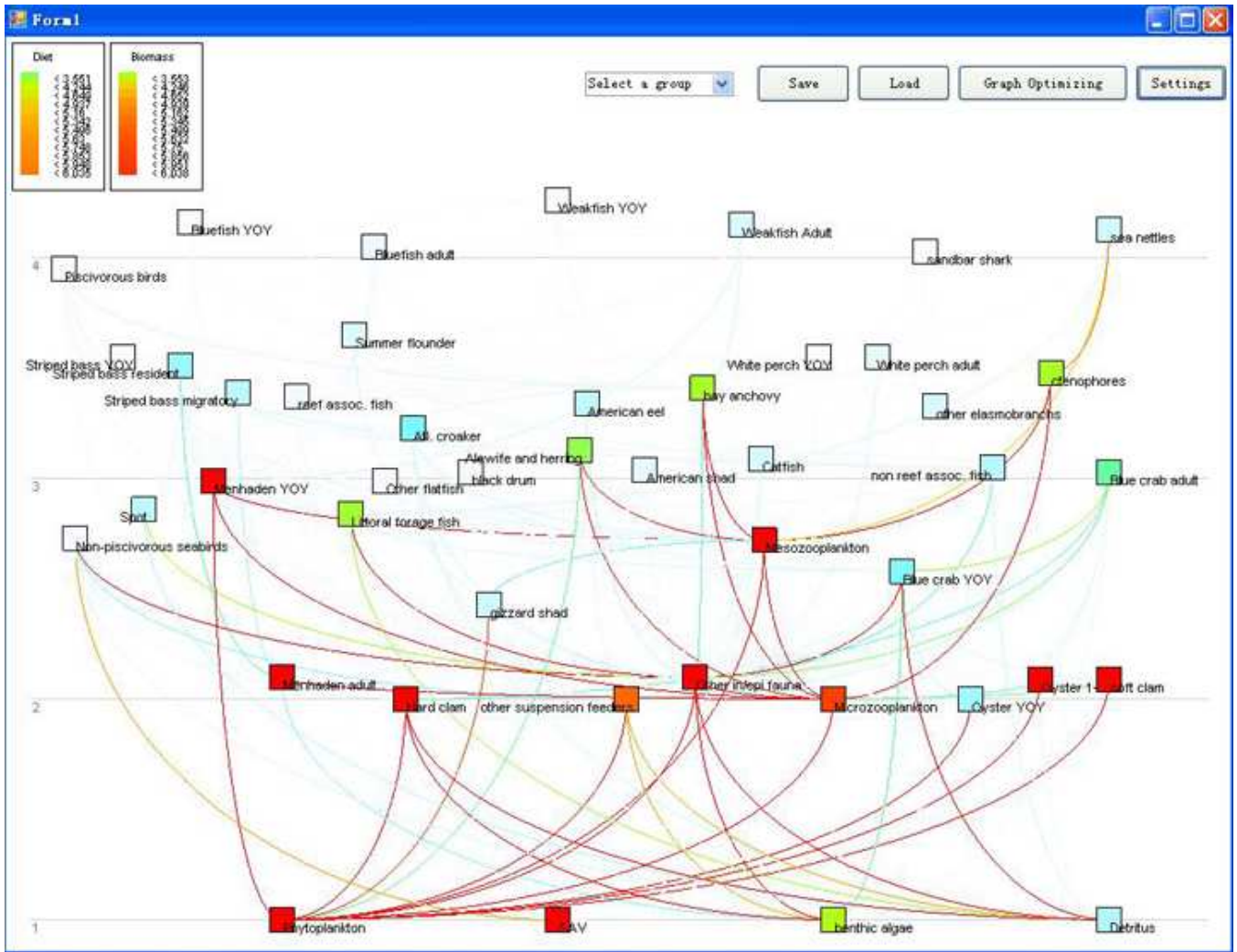


Figure 17: ThunderPelican 2

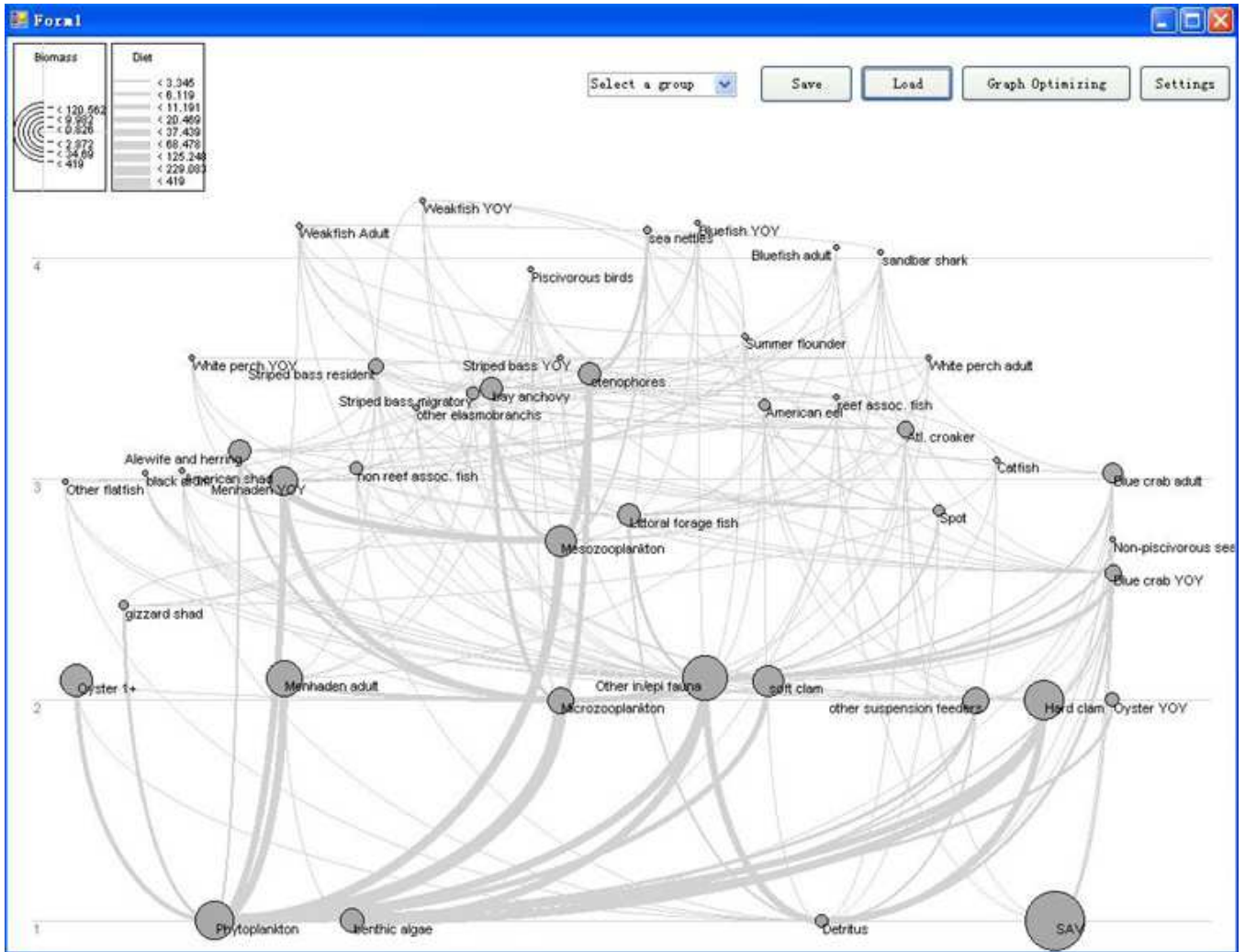


Figure 18: ThunderPelican 3: Optimized Graph

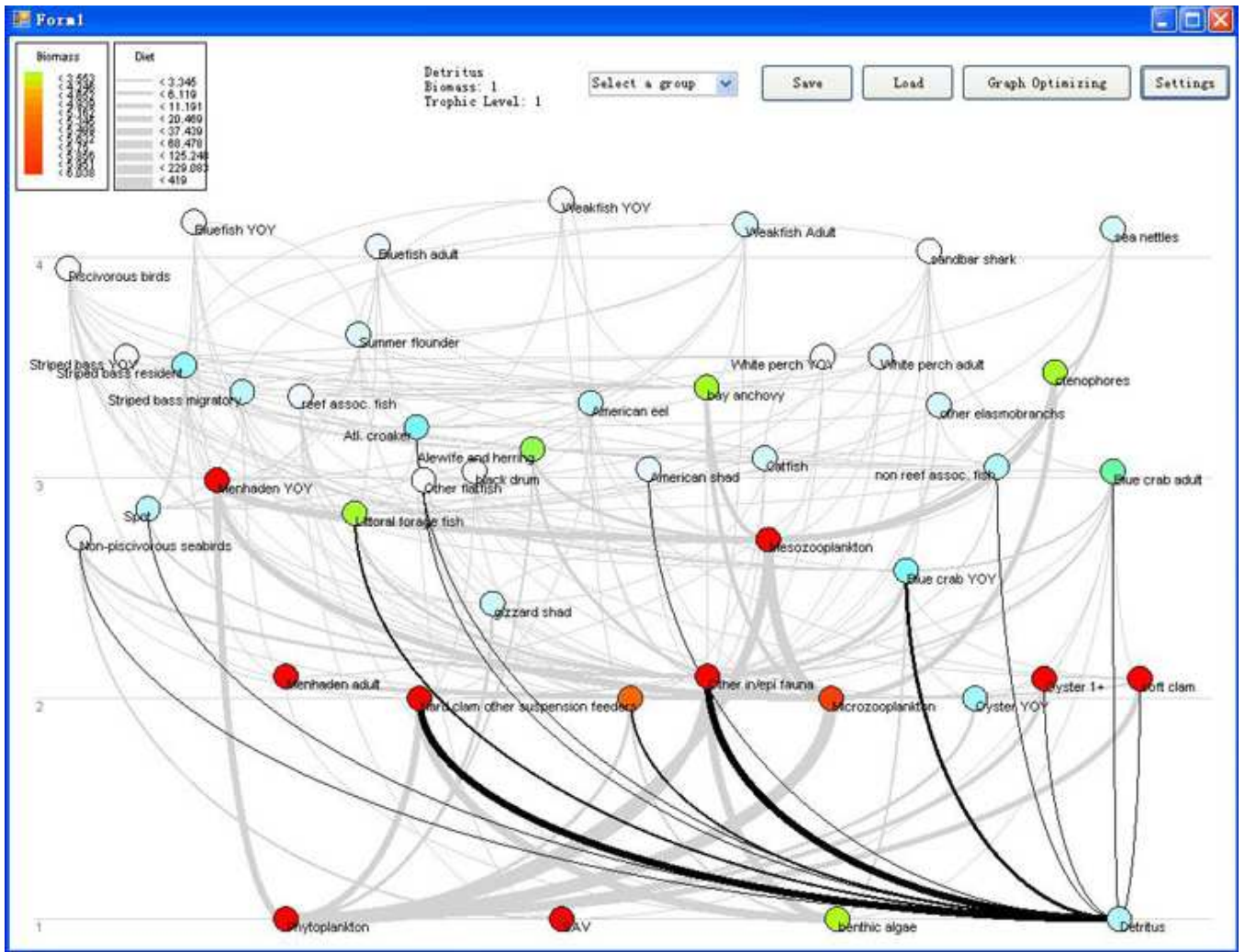


Figure 19: ThunderPelican 4: Highlight

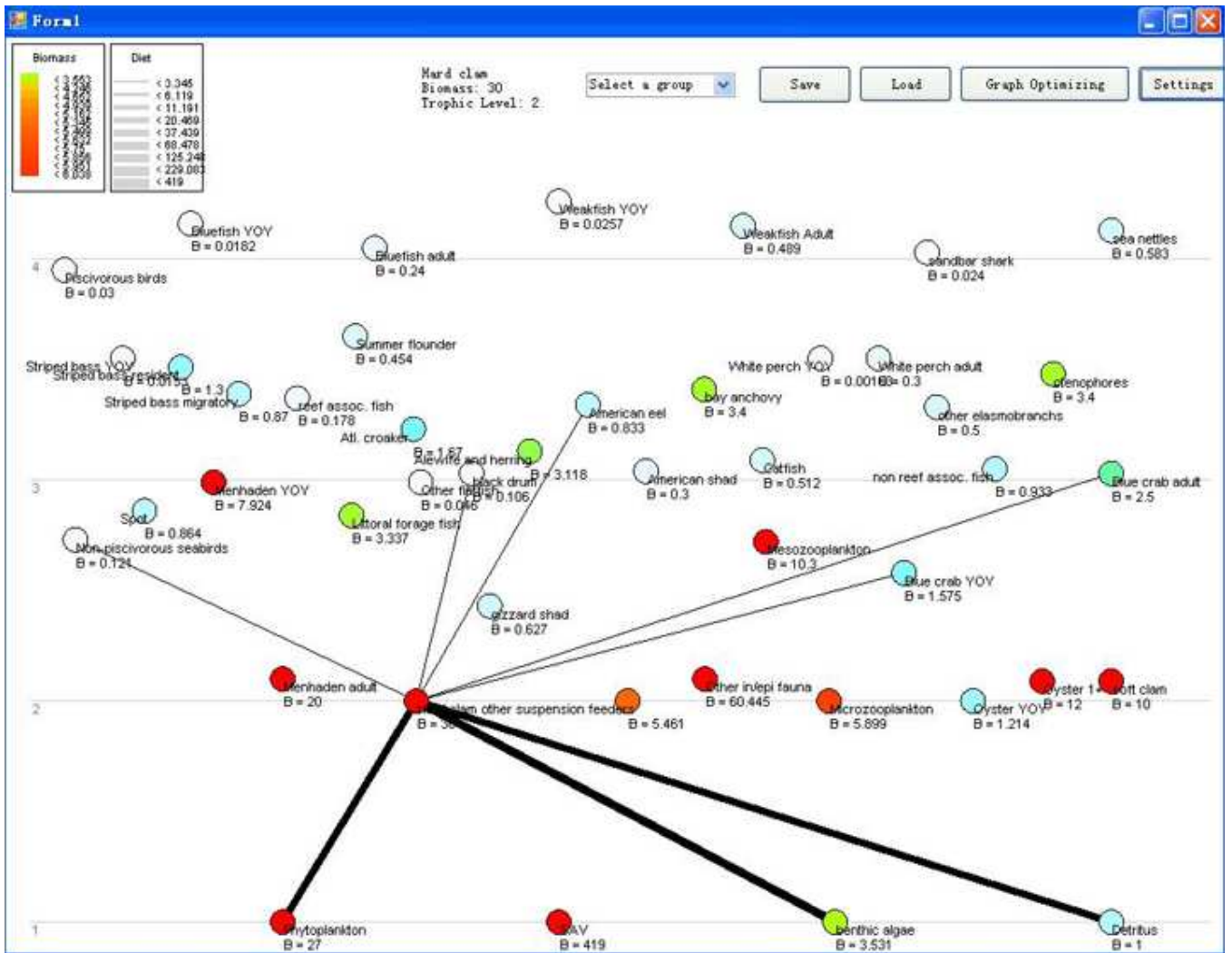


Figure 20: ThunderPelican 5: Hide All Edges