Interactive Visualization of Continuous Node Features in Graphs

Stefan Haun, Marcus Nitsche, Andreas Nürnberger
Motivation

- Ordinary Graph:
  - Solid approach to model relations between items
  - But: Discrete Structure

  How to model continuous data?

- Problem:
  How to map continuous property to discrete graph?
Content

- Motivation
- Embedding Continuous Features
- Non-Discrete Graph Structure
- Visualization
- Application Example
- Conclusion
Embedding Continuous Features (1)

- Mapping by **Meta-Data**

- Continuous information attached to vertices
- Ordinary graph with embedded meta-data

- However:
  - Continuous domain not part of graph structure
  - Labeling edges not possible
Embedding Continuous Features (2)

- Mapping by **Meta-Nodes**
- Discrete nodes as instances from continuous domain
- Only represent the value, but are distinct nodes on their own
- Can be treated as ordinary graph elements
- Implicitly connected by the continuous domain
Embedding Continuous Features (3)

- Advantages
  - Structure is still an ordinary graph
  - Properties of continuous domain taken into account:
    - Distance
    - Ordering

- However:
  - Equal/similar values must be represented by same node
Non-Discrete Graph Structure

- Like ordinary graph
  \[ G = (V, E) \]
- Vertex set \( V \): union of \( r \) multiple, arbitrary domains
  \[ V = V_1 \cup V_2 \cup \ldots \cup V_r \]
- Edge set \( E \): union of discrete and implied relations
  \[ E = E_{\text{Discrete}} \cup E_{\text{Implicit}} \]

**Non-Discrete** Graph **Structure** (NoDeS)
Visualization (1)

- Use Properties from continuous domain for layouting
  - Align according to ordering
  - Derive distances
Visualization (2)

- Context switching by using bipartite properties
- Interpret specific domain as context
- Hide or center contexts
Application Example (1)

- Agent-driven event scheduling
  - Represent schedules as graph
  - New appointments can be determined automatically

- Model
  - Appointment in continuous domain
  - Participants in discrete domain
Application Example (2)

Step 1: Calendar I, Calendar II, Calendar III

Step 2: Calendar I, Calendar II, Calendar III

Step 3: Calendar I, Calendar II, Calendar III

Step 4: Identified free time, No free time, Depending on priority level

Calendar I, II and III overlaid
Application Example (3)

Workflow:
1. Coordinator initiates
2. Participants provide suggestions
3. Coordinator supposes appointment
4. Participants agree or process restarts
Conclusion

- Conceptual design for graph structure supporting continuous node features (NoDeS)
- Combination of discrete and continuous domains in link structures
- Context switching by partitions

Application examples:
- Agent-driven event scheduling
- Context-based graph exploration
Thanks for listening. :)