Introduction to Decision Support based on Cake Cutting problem

Michal Mazurek

February 26, 2017

Overview

1. What is Decision Support?
   - Specific areas of Decision Support
   - What do we need to construct decision support algorithm?
2. Cake cutting
   - Historical background
   - Formal problem definition
   - Proportionality for $n = 2$: Cut and Choose
   - Proportionality for $n = 2$: Banach-Knaster
   - Proportionality for any n: Dubins-Spanier and Even-Paz
   - Envy-freeness for $n = 3$: Selfridge-Conway
   - Where are we now?
3. References

What is Decision Support?

The goal of Decision Support

The main aim of decision support is proposing algorithms that simplify a process of making decisions i.e. choosing a new car, camera, etc. In other words, we look for a method which solves a specific decision problem and let us achieve a goal.

Decision problem

A situation where there is a necessity to choose one of at least two possible variants of actions. A decision maker has to answer one of the following questions:

- How to choose the best variant? (Choice problem)
- How to classify variants into decision classes? (Classification problem)
- How to order variants from the best to the worst? (Ordering problem)

Specific areas of Decision Support

What do we need to construct decision support algorithm?

Preference information

The information that is given by decision maker in order support solving a problem.

Preference model

Preference model allows to aggregate evaluations on each criterion of specific variant. It is built by preference information given by decision maker. We usually distinguish three types of preference model:

- function,
- relational system,
- set of decision rules.

Criterion

Criterion is a real-valued function reflecting a worth of variants from a particular point of view. Family of criteria should be consistent.

Cake cutting

Cake-cutting is a metaphor for a wide range of real-world problems that involve dividing some continuous object, whether its cake or, say, a tract of land, among people who value its features differently. The ideal method, which solves the problem, should:

Historical background

Formal problem definition

Proportionality for n = 2: Cut and Choose

Number of players	Proportionality	Envy-freeness	Complexity
2	yes	yes	2

Proportionality for $n = 2$: Banach-Knaster

Number of players	Proportionality	Envy-freeness	Complexity
2	yes	yes	2

Proportionality for any n: Dubins-Spanier and Even-Paz

Number of players	Proportionality	Envy-freeness	Complexity
2	yes	yes	2

Number of players	Proportionality	Envy-freeness	Complexity
2	yes	yes	2

Envy-freeness for n = 3: Selfridge-Conway

Number of players	Proportionality	Envy-freeness	Complexity
2	yes	yes	2

Where are we now?

References

Scott Meyers (2002)
Effective C++: 50 Specific Ways to Improve Your Programs and Design

C++ reference
http://en.cppreference.com/

The End