Item Combination In Destination Recommender Systems

- Abidemi Shobayo-Eniola
- Technische Universität München
- Supervisor: Prof. Dr.-Ing. Jörg Ott
- Advisor: Dr. Wolfgang Wörndl

- 28 April 2022

Destination Recommendation

Item Combination in Destination Recommender Systems

Problem

- Efficiently combine regions
- Recommend best combination of regions
- Recommend stay duration
- Computationally hard problem

Goal

- Research and develop solutions
- Improve previous work
- Implement algorithms
- Evaluate from user's perspective

Recommender Systems

Recommender systems

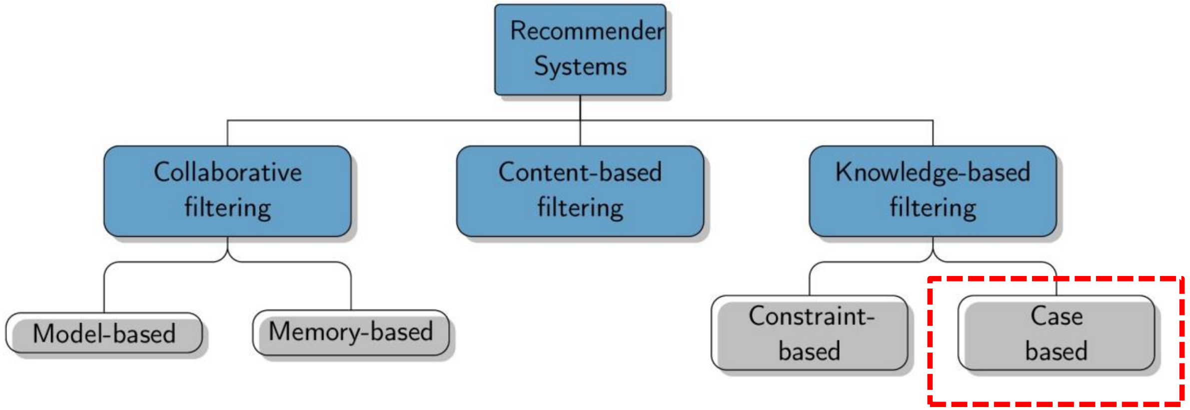
- Produce individualized recommendations
- Aggregate user’s preference

Knowledge-Based filtering

- Depends on knowledge models of the object domain

Case-Based

- Utilize experiences and expertise of agents



Destination Recommendation Problems are Orienteering Problems

TTDP

- Aka Tourist Trip Design Problem

Candidate Points of Interests (POIs)

- Attributes
- Score

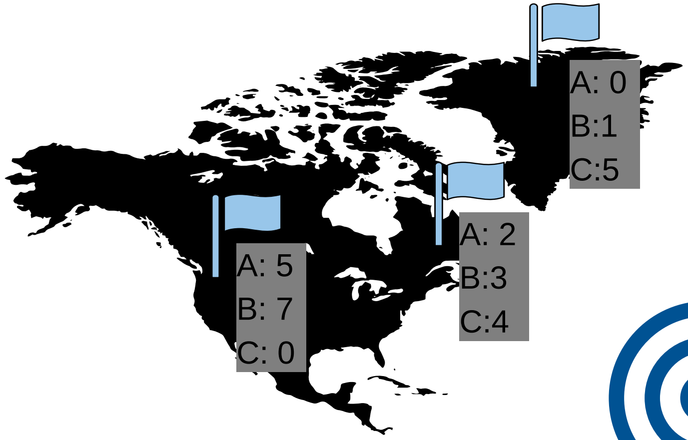
Goal

- Maximize collected score
- Respect constraints

Knapsack and Traveling Salesman problem

- Computationally hard problems

Problem Model



Many-Objective Orienteering Problem

Optimization problem with more than one objectives

Item with different scores for different attributes

Objectives:

Single objective is a possible travel region

Maximize score from chosen region

Optimize score from region combination

Constraints

Duration constraint

Budget constraint

Must satisfy user preference

Optimize duration of stay

Regions must fit together

Evolutionary Algorithms

Individuals

- Chromosomes
- Properties are genes
- Gene altering via mutations operator
- Offspring via recombination operator

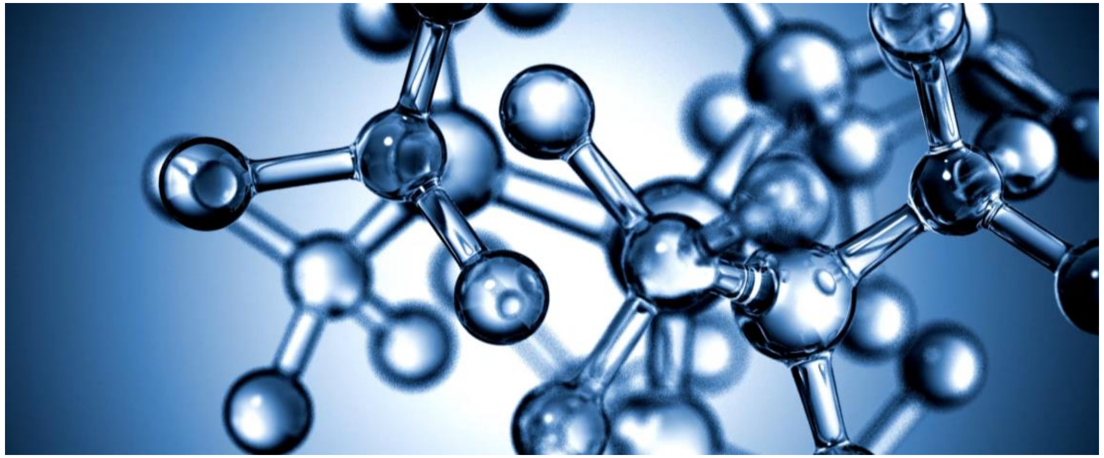
Classical approaches

- Single objective optimizer run many times
- Good distribution not guaranteed
- Absence of parallel search

Emo Methodologies

- Converge to the Pareto-optimal front
- Maintain as diverse a distribution as possible
- Finds multiple solutions simultaneously

Meta-heuristics based on Darwin's theory of evolution



NSGA - III

Non-Dominated sorting genetic algorithm

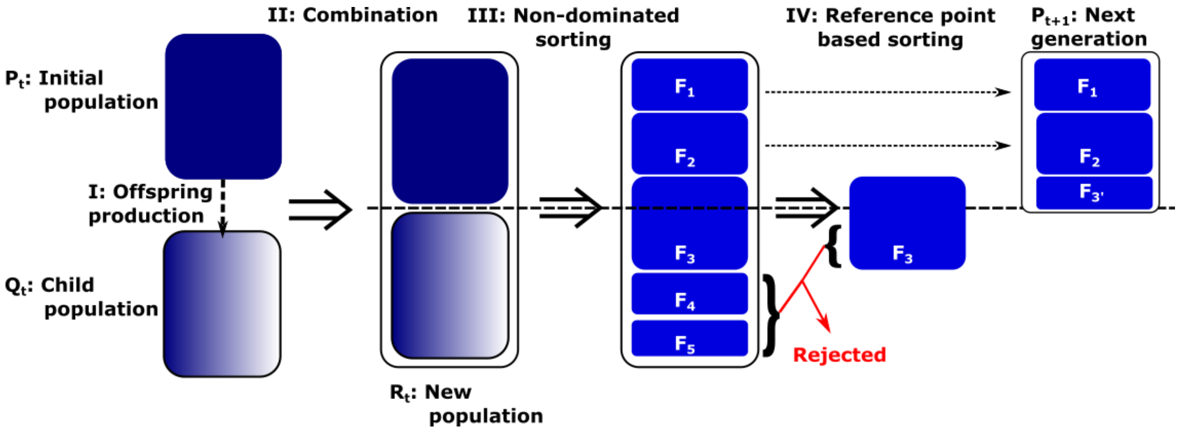
- Better suited for many-objectives problems
- Emphasize non-dominated solutions for convergence
- Emphasize solutions closest to a pre-defined reference point for diversity

NSGA-III without constraints

- Rank members into levels
- Uses individuals' fitness values

Constraint-Domination principle

- A solution *i* constraint-dominates another solution *j* if any is true:
  - *i* is feasible and *j* is not
  - *i* and *j* are both infeasible, but *i* has a small overall constraint violation
  - *i* and *j* are feasible and *i* dominates *j*



Implementation

Combinations of various initialization and constraint handling techniques produced five algorithm variants of NSGA-III. Six different adaptations of NSGA-III implemented

DEAP

Distributed evolutionary algorithms in Python

For rapid prototyping and testing of ideas

Flexible

NSGA - III

Unconstrained version

Return individuals from first non-dominated front

Data model

Hierarchical data model used in previous work

World is root region, continents or regions as parents; sub-regions are children

Individuals

Possible region combination encoded as binary arrays

Accompanying information

Flip bit mutation operator

Implementation – Initialization Techniques

Random Initialization

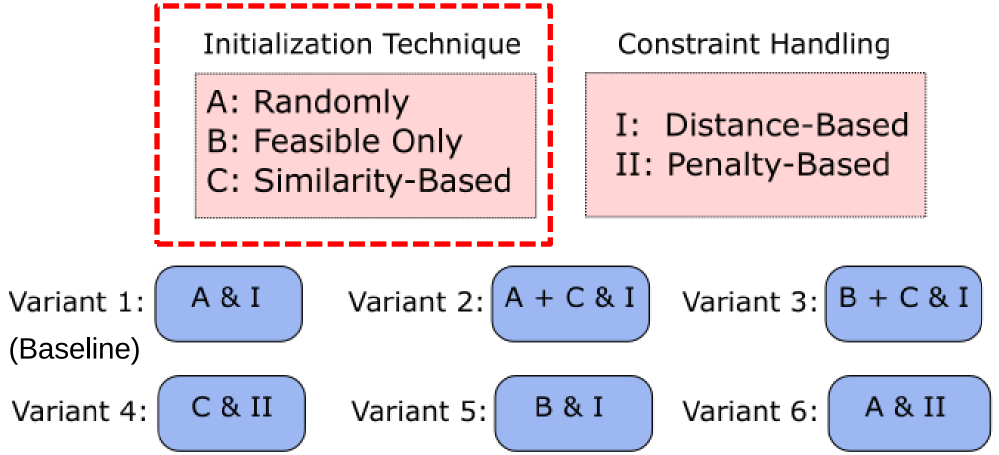
- Randomly generate bits for individuals to form starting population
- 0.81% feasible individuals

Similarity-Based

- Similarity metric from previous work
- Pre-rank regions according to similarity value
- Remove regions below rank
- 0.5% feasible individuals

Feasible Only

- Generate starting population with individuals satisfying all constraints



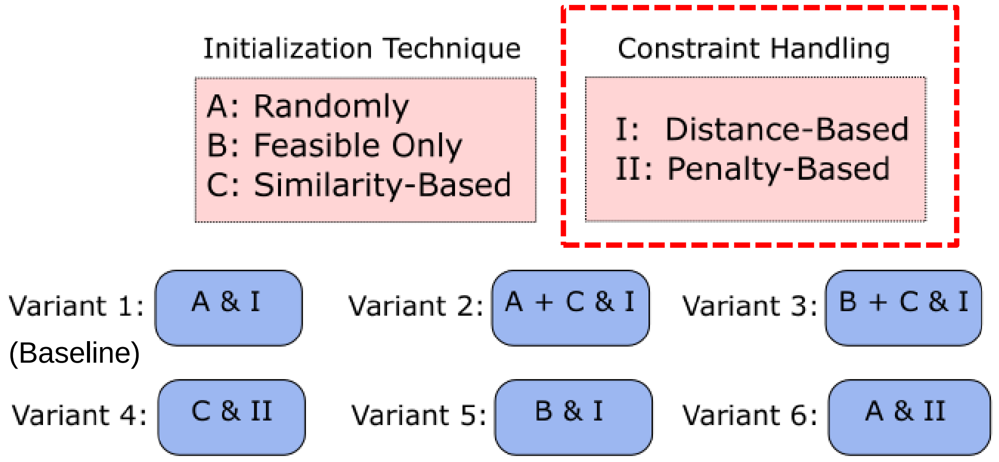
Implementation – Constraint Handling Techniques

Distance-Based

- Penalize individuals based on distance to feasible area
- Distance a function of single constraints
- Varying level of penalty for region in combination

Penalty-Based

- Penalize individuals as a whole
- Penalty a function constraint violation degree
- Constraint violation degree a function of single constraints



Evaluation I

Setup

- Three sample queries with preferences, budget, duration in weeks, preferred months
- Different user preferences
- Ten trials per algorithm variant and sample query
- Variant 1 as baseline

Online evaluation

- Goal: Eliminate low performing variants
- Best score-oriented evaluation of region combination per sample query and variant
- Computational time
- Diversity of suggestions from other variants
- Number of regions in combination

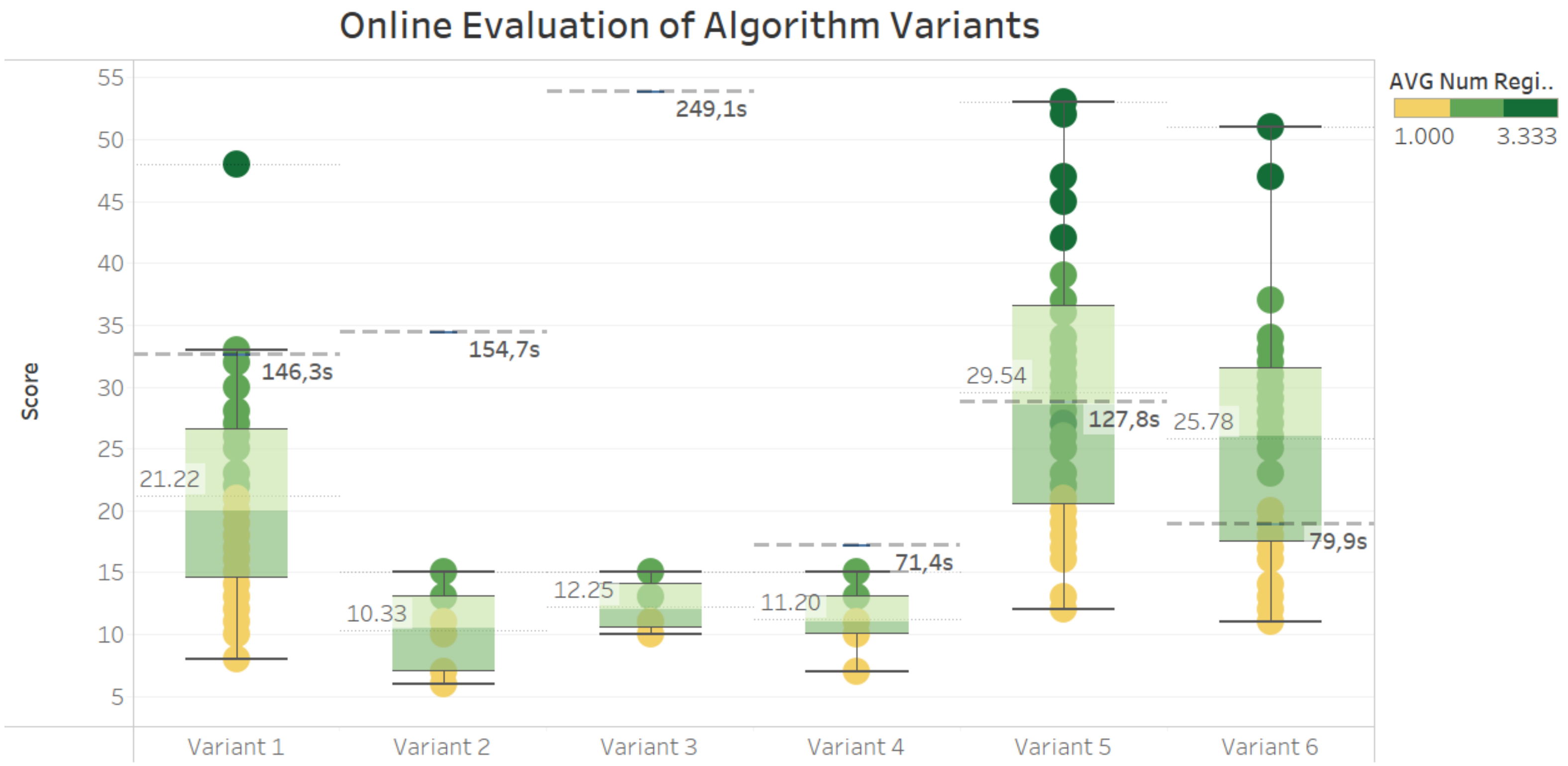
Results – Online Evaluation

Initialization Technique
A: Randomly
B: Feasible Only
C: Similarity-Based

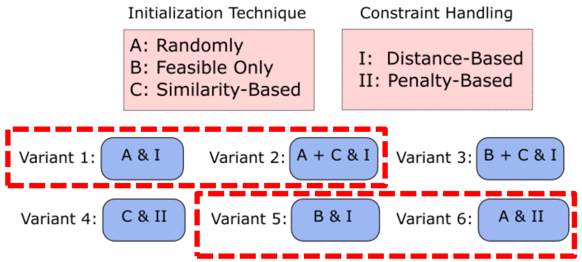
Constraint Handling
I: Distance-Based
II: Penalty-Based

Variant 1: A & I
Variant 2: A + C & I
Variant 3: B + C & I
Variant 4: C & II
Variant 5: B & I
Variant 6: A & II

Variant 1, 5, & 6: best total scores; higher number of regions in a combination
Variant 4, 5, & 6: better computational times; Variant 3 and 4 eliminated; Variant 1, 2, 5, & 6 chosen



Evaluation II



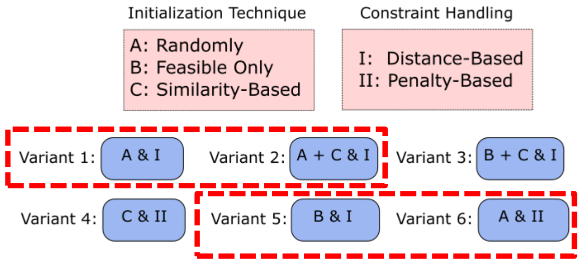
Setup

- Three sample queries with preferences, budget, duration in weeks, preferred months
- Different user preferences
- Ten trials per algorithm variant and sample query

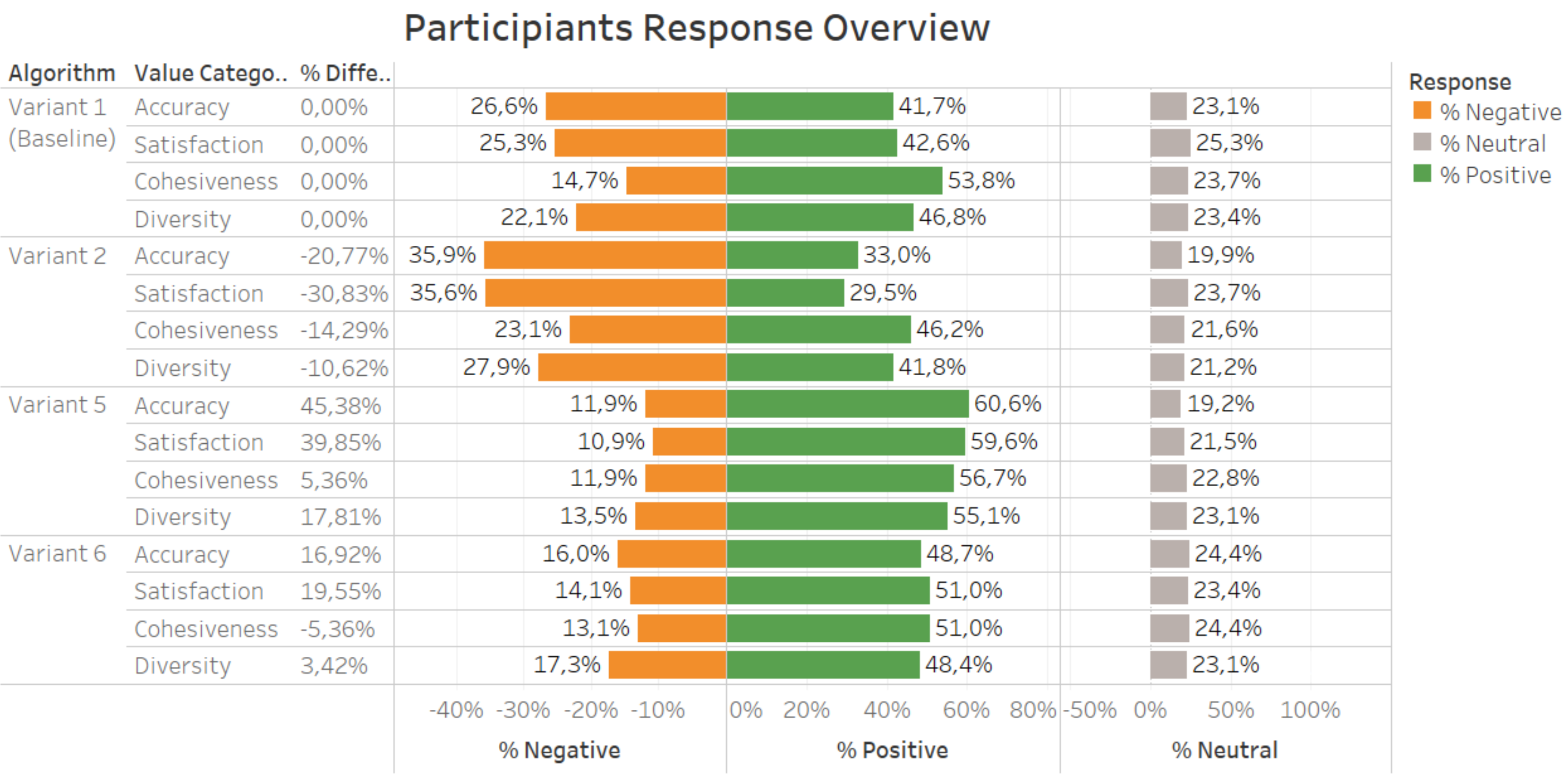
User survey

- Goal: Evaluate region combinations from each variant per sample query
- 104 participants
- Background information
- Rating accuracy, cohesiveness, satisfaction, and diversity on 5-point Likert scale
- Region combinations from variant passing online evaluation

Results – User Survey

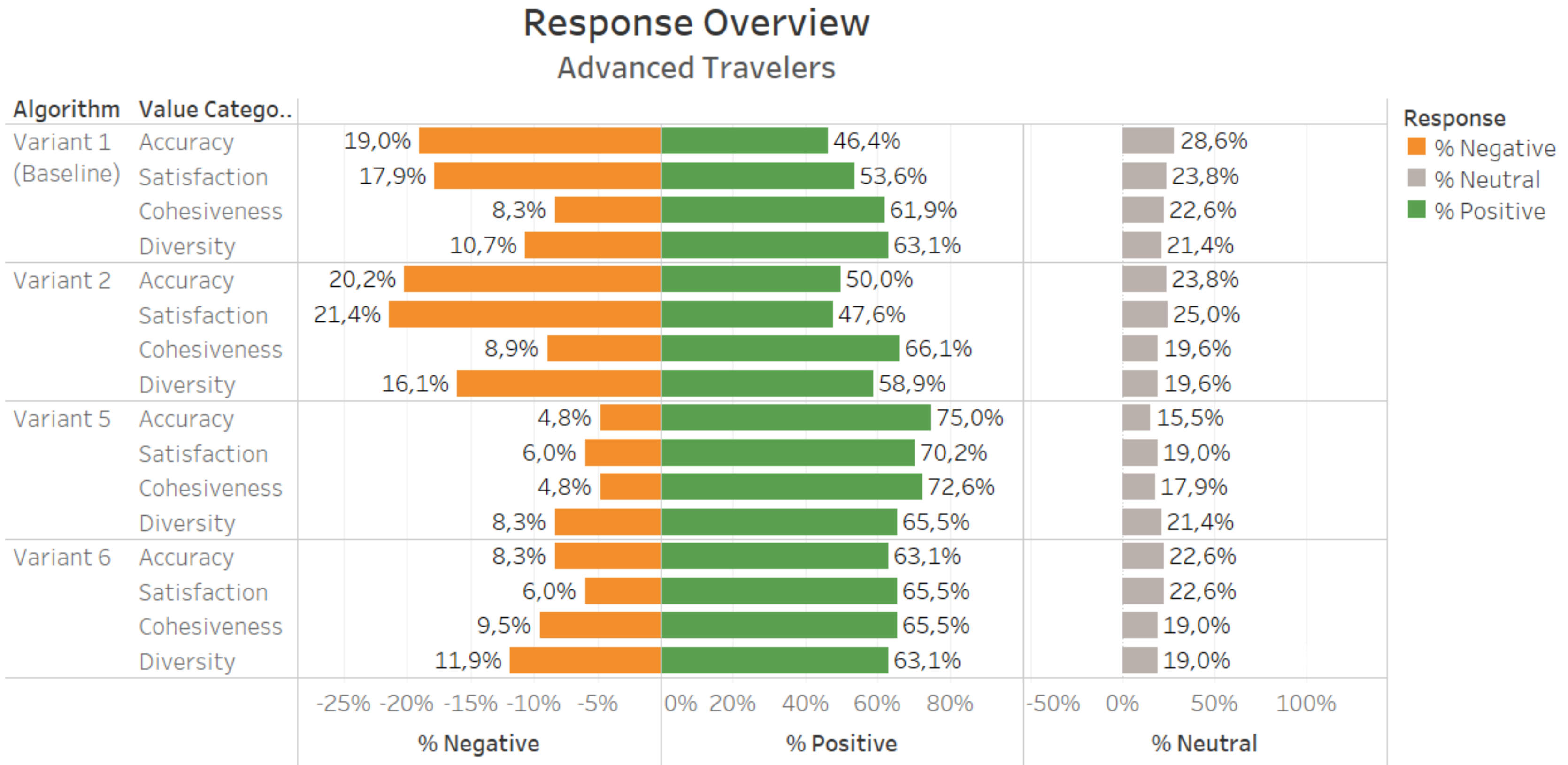


Variant 1, 5, and 6 were best rated; Variant 5 had the best accuracy, satisfaction, cohesiveness, and diversity rating compared to baseline.



Results – User Survey

Travelers with advanced travel knowledge rated the algorithm variants better; Kruskall Wallis test showed that there is indeed a statistically significant difference (p <= 0.01) between travelers with advanced knowledge and average or basic knowledge.



Limitations

NSGA-III

- Currently returning region combination in first non-dominated front only
- Only one region combination in first non-dominated front

Computational time

- Very slow
- Only suitable for asynchronous recommender systems

Region combinations

- No routing between regions
- No budget suggestions

Conclusion

Implications

- NSGA-III can provide region combinations
- Return individuals from different non-domination levels in rank order
- Additional variant using feasible start population and penalty-based constraint handling
- User evaluation of algorithm better with expert travelers

Contributions

- Formulation of item combination in destination recommender systems as a non-time dependent many-objective orienteering problem
- Evolutionary algorithm for generating region combinations
- Above average user ratings with respect to accuracy, cohesiveness, diversity, and overall satisfaction with suggestions

Future work

- Further investigation of NSGA-III (time can be improved)
- Extensive user survey with advanced travelers
- Transformation of database (ISO codes for regions; Longitude and Latitude coordinates in database)