

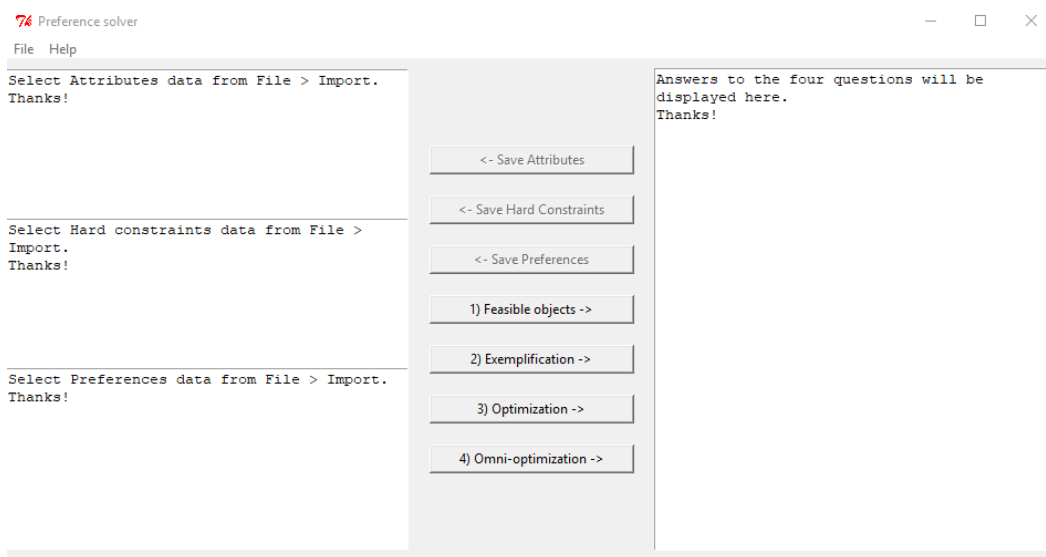
Contents

Test Scenario 1:	2
Show feasible models when 2 models exist, exemplification with strict preference, optimization and omni-optimization.	2
Test Scenario 2:	5
Show feasible models when 2 models exist, exemplification with equal preference, optimization and omni-optimization.	5
Test Scenario: 3	7
Show feasible models when zero models exist with the specified hard constraints.	7
Test Scenario: 4	11
Show feasible models when only 1 feasible model exists with the specified hard constraints.	11
Test Scenario: 5	13
Show feasible models when more than 2 feasible models exist with the specified hard constraints...	13

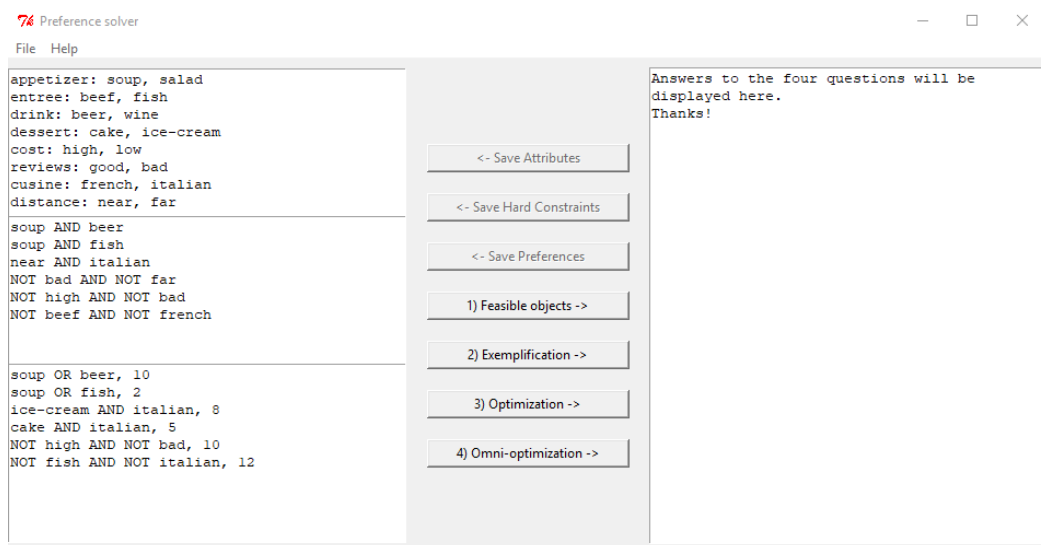
Test Scenario 1:

Show feasible models when 2 models exist, exemplification with strict preference, optimization and omni-optimization.

Initial screen on execution



Import attributes, hard constraints and preferences



Click Feasible objects

74 Preference solver

File Help

appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup AND beer
soup AND fish
near AND italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12

<- Save Attributes

<- Save Hard Constraints

<- Save Preferences

1) Feasible objects ->

2) Exemplification ->

3) Optimization ->

4) Omni-optimization ->

There are 2 models that satisfy the hard constraints.
Models are:
[['soup', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near']]

Click Exemplification

74 Preference solver

File Help

appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup AND beer
soup AND fish
near AND italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12

<- Save Attributes

<- Save Hard Constraints

<- Save Preferences

1) Feasible objects ->

2) Exemplification ->

3) Optimization ->

4) Omni-optimization ->

Two models [['soup', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near']] were randomly picked from the feasible objects.
['soup', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near'] is strictly preferred over ['soup', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near']

Click Optimization

The screenshot shows the 'Preference solver' window with the 'Click Optimization' process. The left pane contains the following text:

```
appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup AND beer
soup AND fish
near AND italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12
```

The middle pane contains the following buttons:

- <- Save Attributes
- <- Save Hard Constraints
- <- Save Preferences
- 1) Feasible objects ->
- 2) Exemplification ->
- 3) Optimization ->
- 4) Omni-optimization ->

The right pane displays the result of the optimization:

```
[[ 'soup', 'fish', 'beer', 'ice-cream',
  'low', 'good', 'italian', 'near' ]] is an
optimal model.
```

Click Omni-optimization

The screenshot shows the 'Preference solver' window with the 'Click Omni-optimization' process. The left pane contains the same text as the previous screenshot:

```
appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup AND beer
soup AND fish
near AND italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12
```

The middle pane contains the same buttons as the previous screenshot:

- <- Save Attributes
- <- Save Hard Constraints
- <- Save Preferences
- 1) Feasible objects ->
- 2) Exemplification ->
- 3) Optimization ->
- 4) Omni-optimization ->

The right pane displays the result of the optimization:

```
There is only 1 optimal model : [[ 'soup',
  'fish', 'beer', 'ice-cream', 'low', 'good',
  'italian', 'near' ]]
```

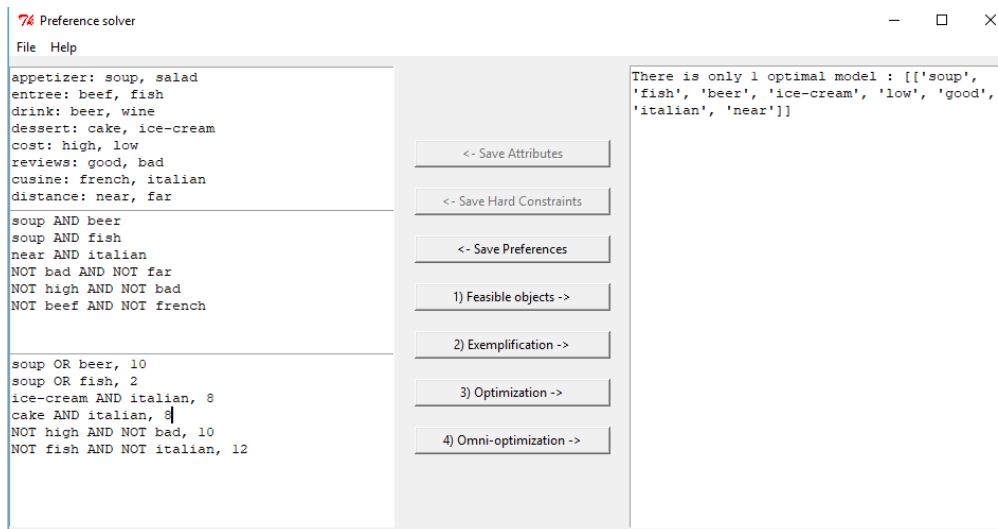
Test Scenario 2:

Show feasible models when 2 models exist, exemplification with equal preference, optimization and omni-optimization.

Screen at the end of test scenario 1



Edit penalty points for “cake AND italian” to 8 and click “Save Preferences”

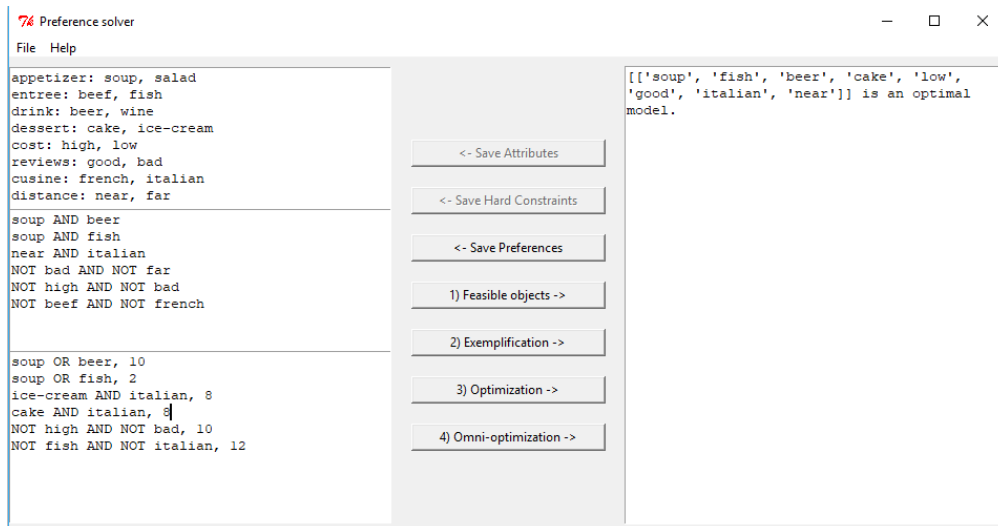


Since only penalty points are changed, there will be no change to feasible models.

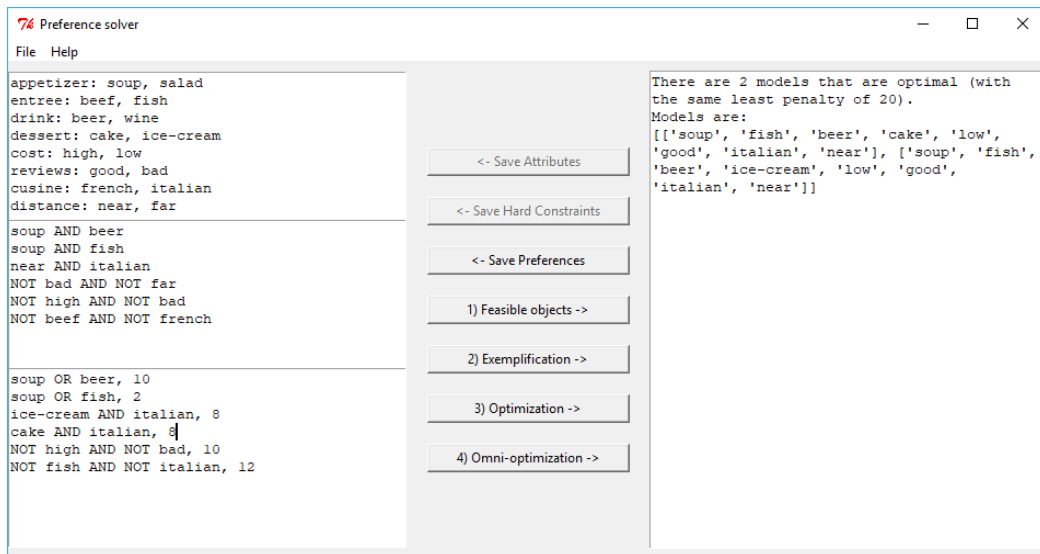
Click Exemplification



Click Optimization



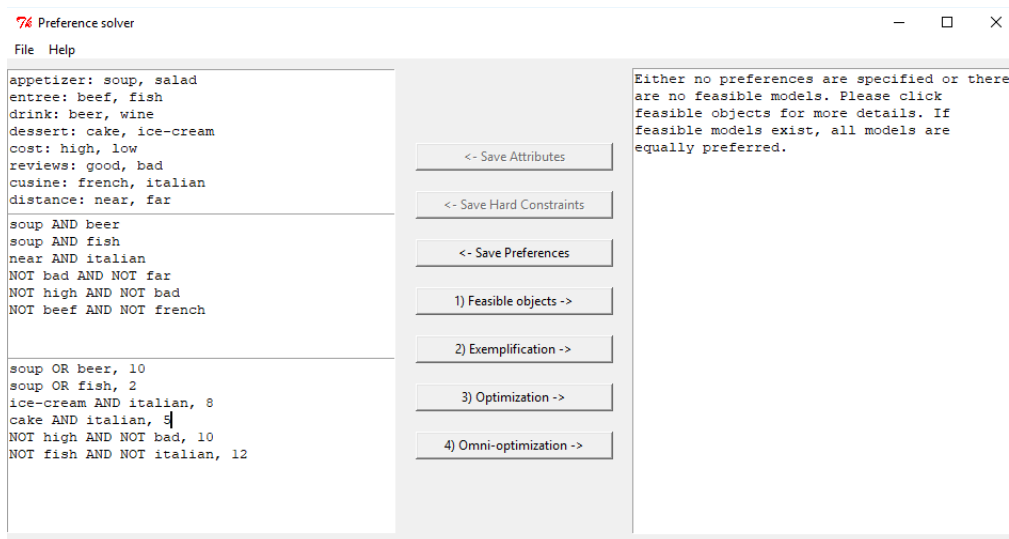
Click Omni-optimization



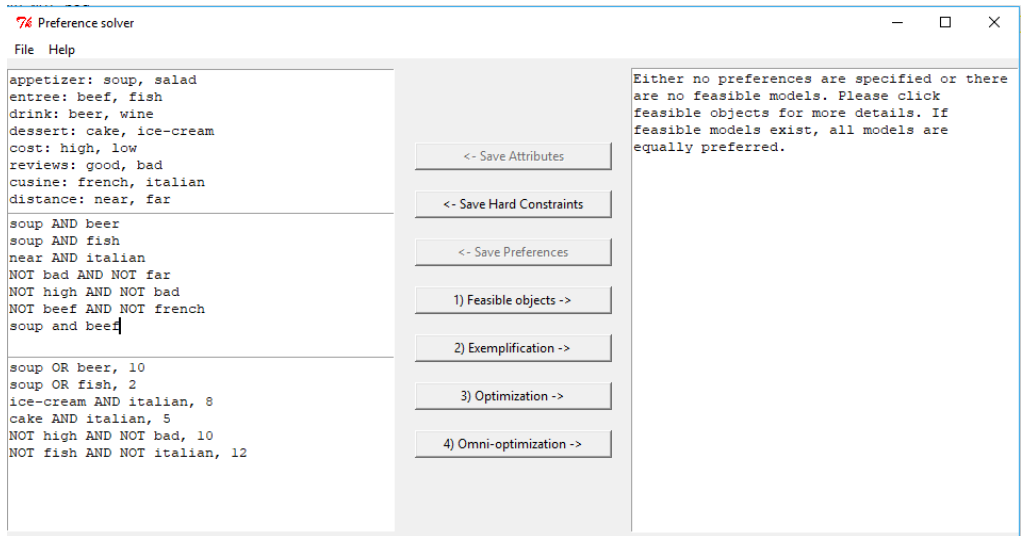
Test Scenario: 3

Show feasible models when zero models exist with the specified hard constraints.

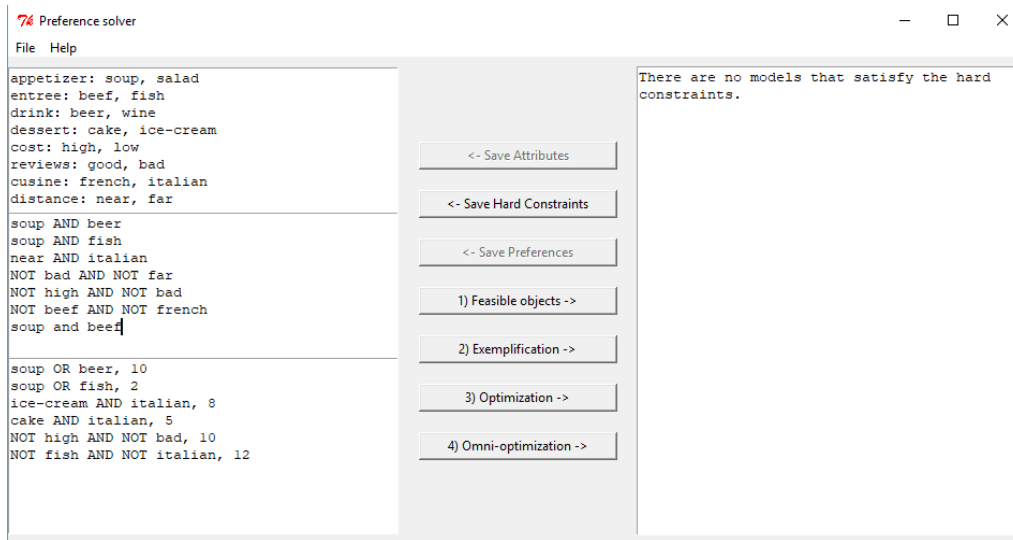
Copy and paste all previous preferences from Test scenario 1 and click “Save Preferences”



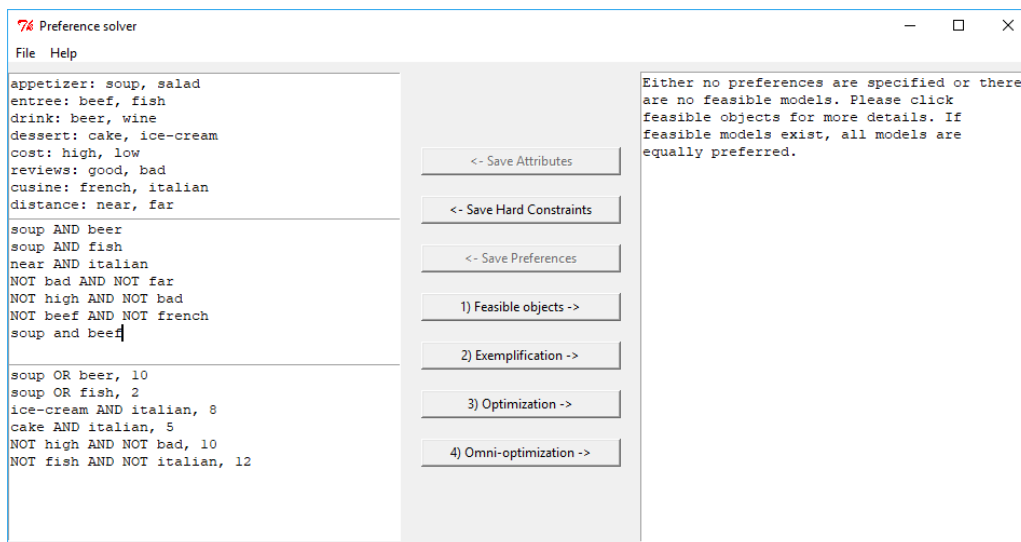
Include a new hard constraint “soup and beef” and save



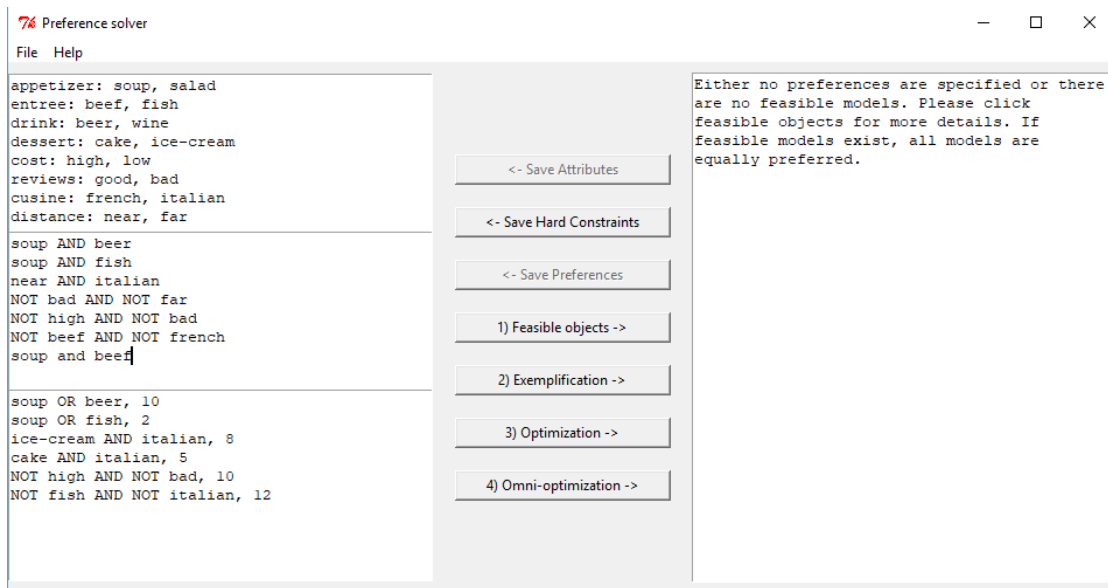
Click Feasible objects



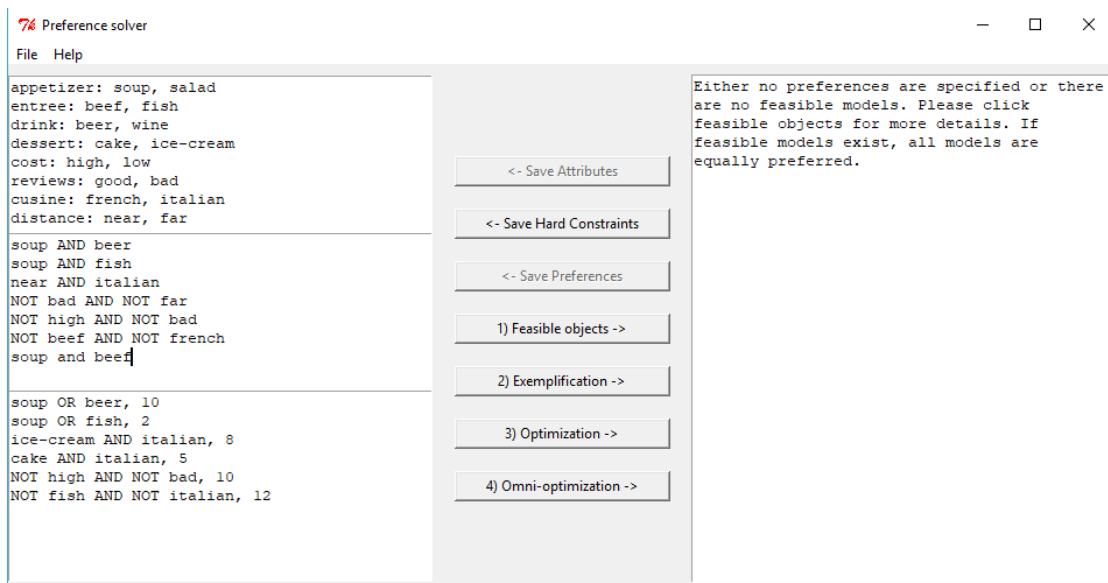
Click Exemplification



Click Optimization



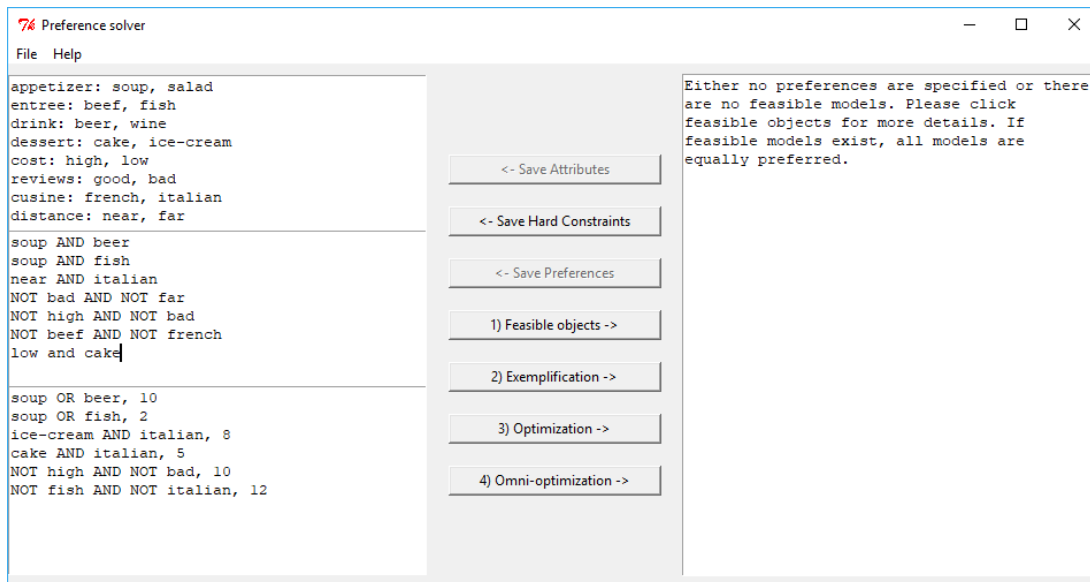
Click Omni-optimization



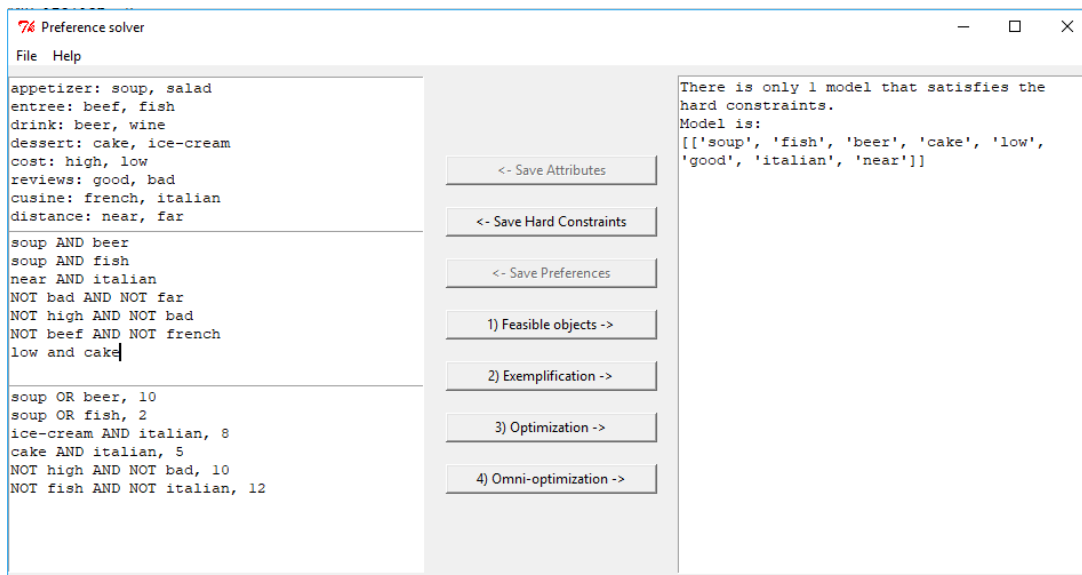
Test Scenario: 4

Show feasible models when only 1 feasible model exists with the specified hard constraints.

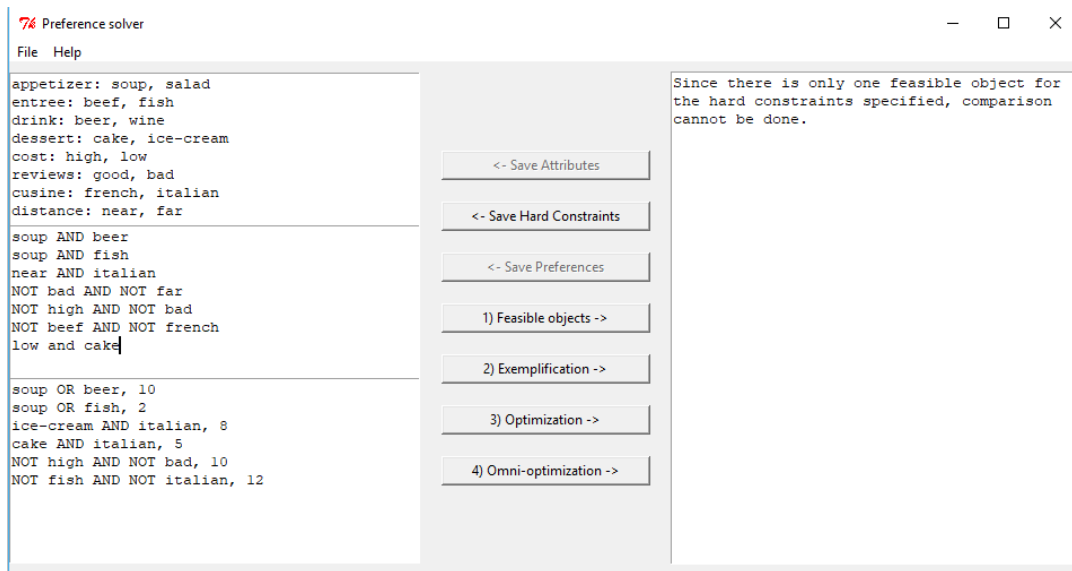
Replace “soup AND beef” hard constraint to “low and cake” and save



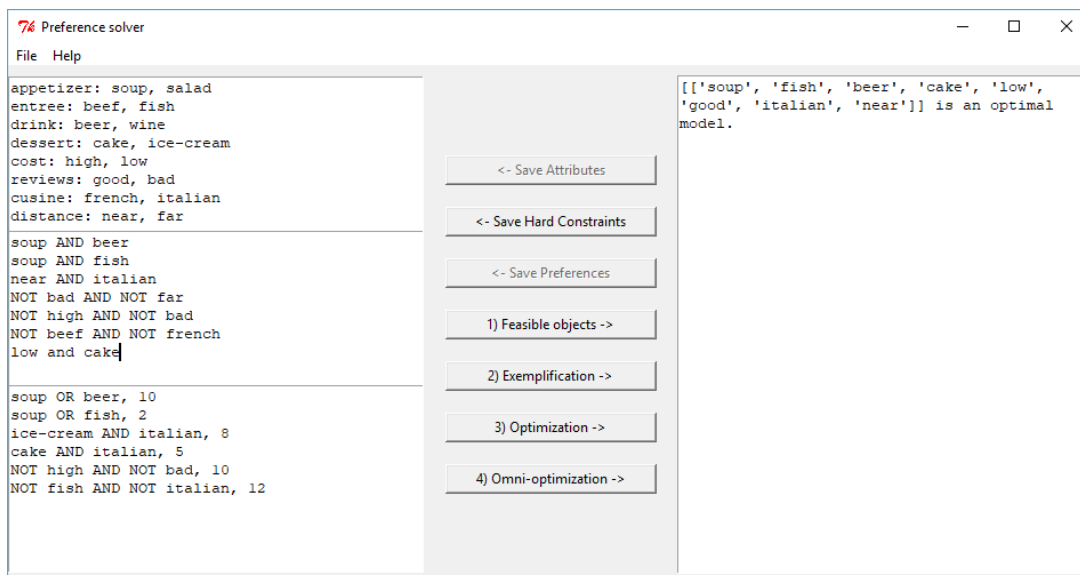
Click Feasible objects



Click Exemplification



Click Optimization



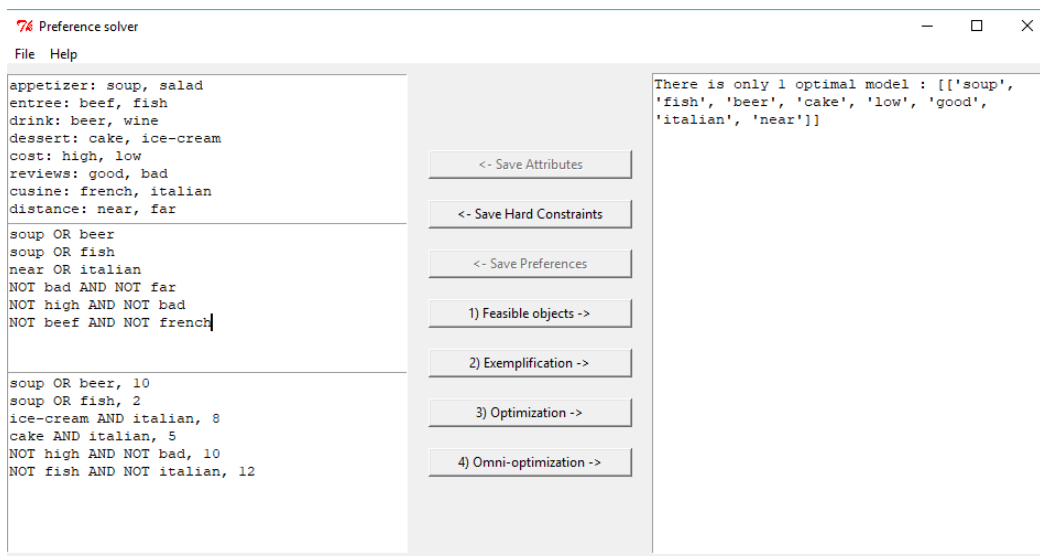
Click Omni-optimization



Test Scenario: 5

Show feasible models when more than 2 feasible models exist with the specified hard constraints.

Update hard constraints as per test instance description in test scenario 5 and save.



Click Feasible objects

76 Preference solver

File Help

appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup OR beer
soup OR fish
near OR italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12

<- Save Attributes

<- Save Hard Constraints

<- Save Preferences

1) Feasible objects ->

2) Exemplification ->

3) Optimization ->

4) Omni-optimization ->

There are 6 models that satisfy the hard constraints.
Models are:
[['soup', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'wine', 'ice-cream', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'wine', 'cake', 'low', 'good', 'italian', 'near'], ['salad', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near'], ['salad', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near']]

Click Exemplification

76 Preference solver

File Help

appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup OR beer
soup OR fish
near OR italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12

<- Save Attributes

<- Save Hard Constraints

<- Save Preferences

1) Feasible objects ->

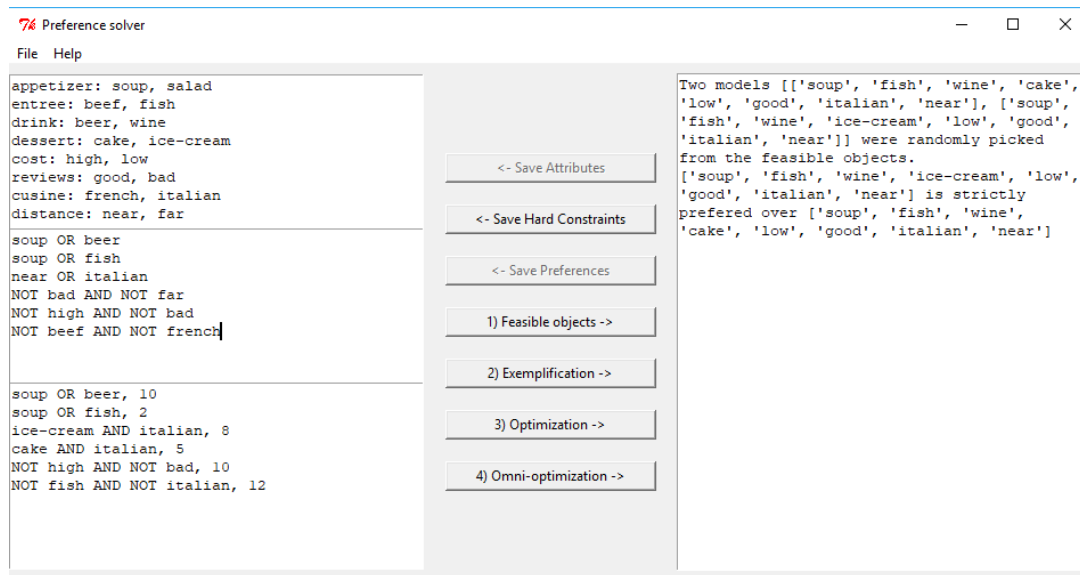
2) Exemplification ->

3) Optimization ->

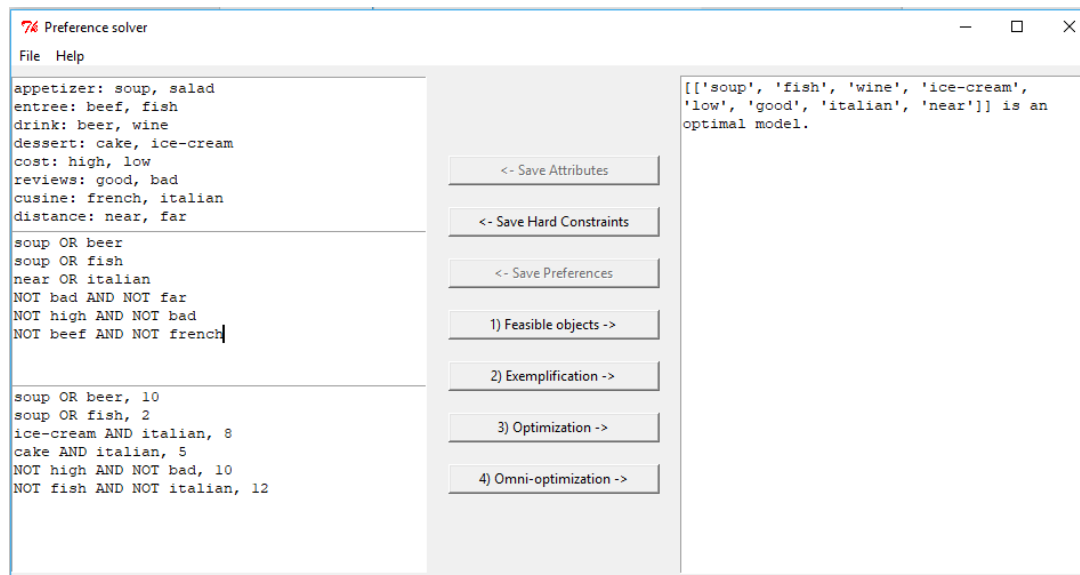
4) Omni-optimization ->

Two models [['soup', 'fish', 'wine', 'cake', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'beer', 'cake', 'low', 'good', 'italian', 'near']] were randomly picked from the feasible objects.
Both these items are equally preferred.

Click Exemplification again for a second time and notice changes in the random result



Click Optimization



Click Omni-optimization

74 Preference solver

File Help

appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
cost: high, low
reviews: good, bad
cuisine: french, italian
distance: near, far

soup OR beer
soup OR fish
near OR italian
NOT bad AND NOT far
NOT high AND NOT bad
NOT beef AND NOT french

soup OR beer, 10
soup OR fish, 2
ice-cream AND italian, 8
cake AND italian, 5
NOT high AND NOT bad, 10
NOT fish AND NOT italian, 12

<- Save Attributes

<- Save Hard Constraints

<- Save Preferences

1) Feasible objects ->

2) Exemplification ->

3) Optimization ->

4) Omni-optimization ->

There are 3 models that are optimal (with the same least penalty of 17).
Models are:
[['soup', 'fish', 'wine', 'ice-cream', 'low', 'good', 'italian', 'near'], ['soup', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near'], ['salad', 'fish', 'beer', 'ice-cream', 'low', 'good', 'italian', 'near']]