

Лабораторная работа 1 Решение многокритериальных задач выбора решений методами свертки	ФИО студента	Титов А.К.
	Группа	ИВТ – 360
	Дата отчета	
	Оценка	
	Подпись преподавателя	

Постановка задачи

Разработать программу для помощи в выборе.

Задание 4. Выбрать материнскую карту для сервера ЛВС:

1. Количество слотов для процессоров
2. Количество слотов для расширения памяти
3. Максимальная память
4. Количество слотов для карт расширения
5. Цена

Ход выполнения работы

Для начала определим веса критериев. Я сделал это исходя из определения ЛВС, представленного ниже. Локальная вычислительная сеть (ЛВС, локальная сеть; англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).

	1	2	3	4	5	$\sum r$	$\sum r / \sum c$
1	1	4	2	5	1/5	12,2	0,229
2	1/4	1	½	2	1/7	3,89	0,073
3	1/2	1/2	1	5	1/6	7,16	0,134
4	1/5	1/2	1/5	1	1/9	2,01	0,037
5	5	7	6	9	1	28	0,526
$\sum c_i$	6,95	13	9,7	22	1,62	1	$\sum \sum c_i = 53,27$

Матрица вышла сбалансированной (сумма всех членов последнего столбца равна 1).

Значения из последнего столбца мы будем использовать в качестве весов.

Для реализации задумки был написан универсальный скрипт, который позволяет обрабатывать различные наборы свойств (число и набор компонентов могут различаться).

Из – за этого сложно без «обходных путей» определить какой из критериев будет негативным (как цена). Наличие негативных критериев сразу «отмечает» аддитивную свертку и один из подвидов мультипликативной.

Выбранный вариант свертки: $I = \sum a_i^{w_i}$, где негативного критерия (цены) был отрицательный показатель степени.

Для получения данных был реализован скрипт на языке Ruby, который осуществлял парсинг сайта citilink.ru и сохранял результат в формате JSON.

Второй скрипт считывает данные из файла, выполняет свертку и записывает результат в файл.

Результат

input.json (представлена лишь малая часть в связи с большим числом вариантов)

```
{
  "weights": {
    "Количество сокетов": 0.229,
    "Слотов памяти DDR3": 0.073,
    "Максимальный объем оперативной памяти": 0.134,
    "Слоты расширения": 0.037,
    "Цена": 0.526
  },
  "choices": {
    "Серверная материнская плата SUPERMICRO MBD-X10SLL-S-B": {
      "Количество сокетов": 1,
      "Слотов памяти DDR3": 2,
      "Максимальный объем оперативной памяти": 16,
      "Слоты расширения": 2,
      "Цена": 13760
    },
    "Серверная материнская плата INTEL DBS1200V3RPS": {
      "Количество сокетов": 1,
      "Слотов памяти DDR3": 4,
      "Максимальный объем оперативной памяти": 32,
      "Слоты расширения": 4,
      "Цена": 15940
    },
    "Серверная материнская плата SUPERMICRO X9SCL-F": {
      "Количество сокетов": 1,
      "Слотов памяти DDR3": 4,
      "Максимальный объем оперативной памяти": 32,
      "Слоты расширения": 3,
      "Цена": 16110
    }
  },
}
```

output.json (представлена лишь малая часть в связи с большим числом вариантов)

- 1) Серверная материнская плата SUPERMICRO MBD-X9DRI-F-B : 1.3381561533027035
- 2) Серверная материнская плата SUPERMICRO MBD-X9DRI-F-O : 1.3237000963444638
- 3) Серверная материнская плата SUPERMICRO MBD-X9DBU-3F-O : 1.315198392532369
- 4) Серверная материнская плата SUPERMICRO MBD-X9DRL-IF-B : 1.284415439307752
- 5) Серверная материнская плата SUPERMICRO MBD-X9DRI-LN4F+-O : 1.2829916431131136
- 6) Серверная материнская плата SUPERMICRO MBD-X9DRH-IF-O : 1.282908672422974
- 7) Серверная материнская плата SUPERMICRO MBD-X9DRL-IF-O : 1.2352402461749303
- 8) Серверная материнская плата INTEL DBS2600CP2 : 1.2172720680606768
- 9) Серверная материнская плата SUPERMICRO MBD-X9SRL-F-B : 1.215419784108854
- 10) Серверная материнская плата SUPERMICRO MBD-X9SRL-F-O : 1.1925859916802017
- 11) Серверная материнская плата SUPERMICRO MBD-X9SRA-B без аксессуаров : 1.1681464565774062
- 12) Серверная материнская плата SUPERMICRO MBD-X9SRE-3F-O : 1.1542218307214627
- 13) Серверная материнская плата SUPERMICRO MBD-X9SRA-O : 1.1504460601225788
- 14) Серверная материнская плата SUPERMICRO MBD-X9SRI-3F-O : 1.1420494631801434
- 15) Серверная материнская плата INTEL DBS1200V3RPS : 1.141321087054441
- 16) Серверная материнская плата SUPERMICRO X9SCL-F : 1.1229533333384816
- 17) Серверная материнская плата INTEL DBS1200BTSR : 1.1136990067009638
- 18) Серверная материнская плата SUPERMICRO MBD-X9SCL-F-O : 1.1050439118758164
- 19) Серверная материнская плата SUPERMICRO MBD-X9SRI-F-B без аксессуаров : 1.0976455599992008
- 20) Серверная материнская плата SUPERMICRO MBD-X9SCM-F-B : 1.0965507419056433
- 21) Серверная материнская плата SUPERMICRO MBD-X9SRI-F-O : 1.0825368381368052

Код программы (свертка)

```
require 'json'
# This program make convolution
# input in input.json (see example below)
# output in output.json
# By Titov Alex

# Format of json is
# {
#   "weights":
#     {
#       "property_1" : property_weight1,
#       "property_2" : property_weight2,
#       "property_n" : property_weightn,
#     },
#   "choices":
#     {
#       "name1" :
#         {
#           "property_1" : value_1,
#           "property_2" : value_2,
#           "property_n" : value_n,
#         },
#       "name2" :
#         {
#           "property_1" : value_1,
#           "property_2" : value_2,
#           "property_n" : value_n,
#         }
#     }
# }

def main
  str_data = File.read('input.json',
    :external_encoding => 'utf-8',
    :internal_encoding => 'cp866') # hack for encoding. Don't knot what it is
  data = JSON.parse(str_data)

  # Calculate Convolutions
  wrong_chooses_count = 0
  data['choices'].each do |name, choice|
    convolution = 1
    data['weights'].each do |property, weight|
      begin
        convolution *= choice[property] ** weight
      rescue Exception => e # Some of alternatives have no some properties
        convolution = 0
        wrong_chooses_count += 1
        break
      end
    end
    choice['weight'] = convolution * 100
  end

  IO.write('output.json', nice_output_2( data ))
  print instructions
  puts "There are #{wrong_chooses_count} wrong alternatives"
end

# Output
def nice_output( data )
  choices = data['choices'].sort_by {|name, choice| choice['weight']}.reverse!
  JSON.pretty_generate(choices)
end

def nice_output_2( data )
  output = ''
  choices = data['choices'].sort_by {|name, choice| choice['weight']}.reverse!
  choices.each_with_index do |choice, index|
    output += "#{index+1}) #{choice[0]} : #{choice[1]['weight']}\n"
  end
  output
end

def print_instructions
  puts 'See weight in output.json'
  puts 'Results are sorted by weight'
end

main
```

Выводы

Были получены навыки реализации свертки и выбор ее варианта.
Попутно был получен опыт парсинга сайта и работы с json форматом.