Faculty of Mathematics and Information Science Warsaw University of Technology

Algorithms and Computability Project

prepared by:
Kuśmierczyk Aleksander
Sławińska Martyna
Żaba Kornel

Version no. 1.0

Date: 19.10.2017

History of changes

Example:

Date	Author	Description	Version
19.10.2017	Kuśmierczyk Aleksander Sławińska Martyna Żaba Kornel	First version	1.0

Table of Contents

1 Problem definition

1.1 Description of a problem

The problem is to assign a set of experts to a set of projects. The assignment must be done in a way that covers the most of the specialization fields required by each project.
The expert can be assigned to only one project, and only one its specialization might be used for that project. The project can have many experts assigned to it.
The aim is to minimize the number of unused experts or to minimize the number of unfinished projects, i.e. the number of projects that are still lacking some experts.

1.2 What is a project

The project is a vector of values, which belong to natural numbers. The number of elements in this vector depends on the features number declared in the input file.
Each element of the project vector represents a specialization field required by the project, and its value is the number of experts needed in that domain.

1.3 What is an expert

The expert is a vector of Boolean values. The number of elements in this vector depends on the features number declared in the input file.
Each element of the expert vector represents a specialization field, which belongs to the expert's area of expertise. The value of 1 signifies that the expert specializes in a given feature, while 0 indicates the lack of knowledge in given field.

1.4 Description of a problem using mathematical notions

u – number of projects
v – number of experts
w – number of features
SP – set of pairs (Expert, Project) representing that the expert is working on the project

A pair (E_k,P_i) represents that the k^{th} expert working in a i^{th} project.

A pair (E_k,p_i) represents that the k^{th} expert working in a i^{th} field of a project.

$$\forall_{i \in N,\, i \le v}\ E_i = [e_1, e_2, \dots, e_w]^{-1} \quad \forall_j\ e_j \in \{0,1\}$$

$$\forall_{i \in N,\, i \le u}\ P_i = [p_1, p_2, \dots, p_w]^{-1} \quad \forall_j\ p_j \in N$$

$$\forall_{k \in N,\, k \le v}\ \forall_{i,j \in N,\, i \le u,\, j \le u,\, i \ne j}\ (E_k, P_i) \in SP \Rightarrow (E_k, P_j) \notin SP$$

$$\forall_{k \in N,\, k \le v}\ \forall_{i,j \in N,\, i \le w,\, j \le w,\, i \ne j}\ \exists_{l \in N,\, l \le u}\ p_i \in P_l \quad (E_k, p_i) \Rightarrow$$

$$\sim\exists_{j \in N,\, j \le w}\ p_j \in P_l\ (E_k, p_j)$$

The solution to the problem is the result of a function, which minimizes the number of not assigned experts.

$F(u,v,w,E,P) = \min(\#x : x \in E \wedge x \notin SP)$

2 Solution in the form of a pseudo-code

E – set of expert vectors

P – set of project vectors

sum – vector of summed values of expert vectors

```
calculatingWeight(E)    // results are saved in the experts

sort( E )    // according to occurrence and weight

sum = sum( E )

indProj = 0    // index of the project to which experts are being currently assigned

while(oldSum != sum && (experts are not finished || projects are not finished))

{

        oldSum = sum

        indSmall = indexOf( min( sum ) )    // index of the smallest element from vector sum

        if( P[ indProj ] requires feature with index indSmall )

        {
```

```
        if P[ indProj ][ indSmall ] < sum[ indSmall ]
            diff = sum[ indSmall ] - P[ indProj ][ indSmall ]
            sum[ indSmall ] -= P[ indProj ][ indSmall ]
            P[ indProj ][ indSmall ] = 0
        else
            diff = P[ indProj ][ indSmall ] - sum[ indSmall ]
            P[ indProj ][ indSmall ] -= sum[ indSmall ]
            sum[ indSmall ] = 0
    }
    // E.remove() returns the list of deleted experts
    usedExperts = E.remove(elements with highest weight and feature with index
            indSmall) x diff
    sum.remove(usedExperts)    // deleting vectors of used experts from the sum
    indProj++
    if( indProj == P.length - 1 )
        indProj = 0
}
```