

Anexo 11. DATOS1SOL

Este programa se encarga de calcular los parámetros Nadir y Utópicos para las dos funciones objetivo. Estos parámetros son necesarios para la implementación de la función multiobjetivo. Las siguientes líneas de código, describen el proceso de cálculo de estos parámetros y el proceso de cómo son agregados a los archivos .dat resultantes del programa DATOS del anexo 10.

```
1  using namespace std;
2
3  #include <algorithm>
4  #include <iostream>
5  #include <fstream>
6  #include <string>
7  #include <vector>
8  #include <locale>
9  #include <sstream>
10 #include <numeric>
11 #include <filesystem>
12 #include "ampl/ampl.h"
13
14 void dat(string name1, string name2){
15
16     string nombre1 = name1;
17     string nombre2 = name2;
18     fstream f;
19     double z;
20
21
22     try {
23         ampl::Environment env("D:\\Documentos\\PROGRAMAS\\AMPL\\ampl_mswin64");
24         ampl::AMPL ampl(env);
25
26         //Leer el modelo Modelos\\MODELO_SIN_ENTEROS1.mod y los archivos de datos
27         ampl.read("C:\\Users\\User\\Escritorio\\PD C++\\Modelos\\" + name2);
28         ampl.readData("C:\\Users\\User\\Escritorio\\PD C++\\Instancias1 .dat\\" +
29             name1 + ".dat");
30         ampl.setOption("solver", "cplex");
31
32         //Resolver
33         ampl.solve();
34         cout << "AMPL fin." << endl << endl;
35
36         //Obtener el valor objetivo del modelo
37         ampl::Objective FO = ampl.getObjective("z");
38
39         z = FO.value();
40         ampl.close();
41
42         string dire = ("C:\\Users\\User\\Escritorio\\PD C++\\Instancias2 .dat\\" +
43             + name1 + ".dat");
44         f.open(dire, std::ios_base::app);
45
46         if (f.is_open() == false) {
47             cout << "No se puede abrir el archivo de lectura.\n";
48             cout << dire << endl;
49         }
50
51         //A través de las siguientes líneas de código se agregan los paramewtros
52         calculados a los archivos .dat
```

```
50
51     double tetta = 0.2;
52
53     if (name2 == "MODELO_SIN_ENTEROS1.mod") {
54         f << endl;
55         f << "param P1:= " << tetta << ";" << endl;
56         f << endl;
57         f << "param NadirMax := " << z << ";" << endl;
58         f << endl;
59     }
60
61     if (name2 == "MODELO_SIN_ENTEROS2.mod") {
62         f << "param UtopiaMax:= " << z << ";" << endl;
63         f << endl;
64     }
65
66     if (name2 == "MODELO_SIN_ENTEROS3.mod") {
67         f << "param UtopiaMin:= " << z << ";" << endl;
68         f << endl;
69     }
70
71     if (name2 == "MODELO_SIN_ENTEROS4.mod") {
72         f << "param NadirMin:= " << z << ";" << endl;
73         f << endl;
74     }
75
76     f.close();
77
78 }
79
80 catch (const exception& e) {
81     cout << "Error: " << e.what() << "\n";
82     cin.get();
83 }
84
85 }
86
87 #include "Header.h"
88
89 int main() {
90     vector<string> vName;
91     vector<string> vName1; // Vector de strings que será lleno con los nombres de las instancias
92     string aux;
93     fstream p("C:\\Users\\User\\Escritorio\\PD C++\\Modelos\\nombres1.txt", fstream::in);
94     fstream f("C:\\Users\\User\\Escritorio\\PD C++\\nombres2.txt", fstream::in); // Se lee el .txt con los nombres de cada una de las instancias
95
96
97     while (p >> aux) {
98         vName1.push_back(aux); // Lee todos los nombres de las instancias y se
```

```
        van poniendo en el vector de strings vName.
    99     }
    100
    101     while (f >> aux) {
    102         vName.push_back(aux);    // Lee todos los nombres de las instancias y se
        van poniendo en el vector de strings vName.
    103     }
    104
    105     for (int i = 81; i < 82; i++) {
    106         for (int b = 0; b < vName1.size(); b++) {
    107             dat(vName[i], vName1[b]);    // Llama la acción "crearDatos" para
        cada uno de los nombres presentes en el vector de strings
    108         }
    109     }
    110     cout << "fin." << endl; // Muestra "fin." en pantalla cuando haya terminado
        de ejecutarse el programa.
    111     cin.get();
    112     return 0;
    113 }
    114
```