ISC610A (Operations Research) Exam, First term 2008/2009

Leo Liberti, Antonio Mucherino, Fabien Tarissan

LIX, École Polytechnique, F-91128 Palaiseau, France
Email:`{liberti,mucherino,tarissan}@lix.polytechnique.fr`

8 january 2009

General notes

This exam paper consists of three exercises. If you do one of them *perfectly*, you will have full marks. As perfection is in the eye of the beholder (us, the authors of this exam paper), don't be fooled into thinking your exam will be perfect. So aim to complete one exercise and try to start some other exercise with your spare time. But *do try and concentrate on one exercise first*.

Each exercise requires you to formulate a problem by its mathematical programming formulation using the AMPL[1] language, choose an appropriate solver (you can use `cplex` for LP/MILPs and convex quadratic MIPs, `ipopt/snopt` for cNLPs and local solutions of NLPs, `bonmin` for cMINLPs and local solutions of MINLPs, and `boncouenne` for global solutions of NLPs/MINLPs), write three separate files (model, data, and run) and obtain a solution on the computer. If the formulation is nonlinear, then you should reformulate it so that it becomes linear; aim to use the same data file as for the original problem, and CPLEX as a solver.

You are free to use whatever didactical material (paper or online) you see fit. But don't waste too much time on studying other material.

For each exercise you attempt (or complete), you will submit: three ASCII files (`surname.mod`, `surname.dat`, `surname.run`) encoding the model, data and run commands for the original problem. In case your original formulation was nonlinear, you will also submit two ASCII files (`surname-ref.mod`, `surname-ref.run`) encoding the model and run commands for the reformulated problem. You should use the same data file as for the original problem. In all cases, **replace `surname` in the file name with your own surname!** In order to correct the electronic part of your exam papers (the AMPL files), we will simply run these files on our computers, so please do make sure that typing "`ampl surname.run`" and "`ampl surname-ref.run`" yield the desired solution. If we find that your run file does not work, you automatically lose some points (even if it was because of a stupid syntax error).

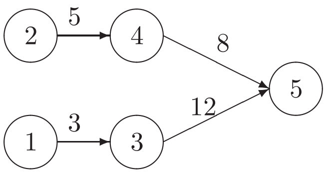
[1] **Warning**: if AMPL/CPLEX give you some license errors, use the "student" versions: at the command prompt, type `setenv PATH "/usr/local/AMPL:$PATH"`, then proceed as usual. In order to re-set to the licensed version (if the license trouble goes away), simply exit the shell and open another terminal, or at worst log out and log back into your workstation.

1 Scheduling with delays

A complex job consisting of 5 threads must be scheduled to run on a computer with 2 CPUs, each of which has dedicated RAM. The threads have the following length, in milliseconds:

thread	1	2	3	4	5
length	2	3	5	8	4

Since the threads are parts of the same job, some data are passed bewteen threads. In the following directed graph, vertices represent threads, and there is an arc between thread i and j if data need to be passed from thread i to thread j; the weight w_{ij} of the arc is the amount of data needing to be passed (in MB).



Thus, if the arc (i, j) is in the graph, thread j cannot start before thread i has finished its execution.

If two threads i, j, such that the arc (i, j) is in the graph, are scheduled to be executed on the same CPU, then the data is kept in the CPU's dedicated RAM and no data transfer is necessary. By contrast, if i, j are scheduled to be executed on different CPUs, then the data must be transferred: this transfer takes time equal to γw_{ij} milliseconds, where γ is a proportionality constant equal to 0.1.

Write a mathematical program deciding the scheduling of threads to CPUs (i.e. whether thread i is assigned to CPU k and at what time t_i does each thread start) such that the total time taken by the job to complete is minimized, and solve the problem using AMPL and an appropriate solver. If your formulation is nonlinear, also supply and solve a linear exact reformulation, checking that the schedulings obtained by the two formulations are equal.

> *Hint*: a transfer delay occurs only if two jobs are scheduled on different CPUs, but this information is decided during the optimization process, so it is not known in advance.

2 Contaminated alligators

Year 2050. The State of Florida has built several small atomic power stations on its territory. Many of them are so small that they are able to give electricity supplies to few surrounding counties, and they are often managed by locals. In Lake City, a town in the northern part of Florida, corrupted politicians allowed the contamination of the waters of Alligator Lake with very small parts of radioactive material which was buried very deeply in the lake. For 3 years it seemed that there were no pollution, but suddenly alligators living in the lake started to increase in size and weight, and they started to reproduce themselves quickly. FBI agents were warned by local citizens, who swore they saw big alligators with a very weird shape in the lake. Therefore, the FBI decided to ask the mayor of Lake City to send two alligators from Alligator Lake to the laboratories of Gatorland, in Orlando, within 10 days, where the animals could be examined.

The mayor of Lake City is the one who allowed the contamination of the lake. After an emergency meeting in City Hall, it was decided to take the following steps in order to avoid a scandal. The alligators of the contaminated lake will be killed and their skins will be used for the black leather market. Alligators from the close lakes Hamburg, Isabella and Harper will be used for repopulating Alligator Lake. In order to minimize the reproduction in contaminated waters, only male alligators will be moved to the contaminated lake, as some experts suggested. Other two alligators from these three lakes will be chosen and sent to Gatorland.

The plan is perfect to cheat the FBI, but the mayor is obviously worried about the cost of the operation. He decides to hire a mathematician for organizing the whole operation with a minimum cost. The total cost is due to the cost c_1 for moving the alligators to the contaminated lake, and the cost c_2 for sending two alligators to Gatorland. The cost c_1 is proportional to the distances between the lakes and to the weights of the alligators. The cost c_2 is proportional to the distance between Lake City and Gatorland and to the weights of the two alligators.

The mathematician, before accepting the task, requires some details from the mayor:

Lake name	Alligator Lake (AL)	Lake Hamburg	Lake Isabella	Lake Harper
Number of alligators	8	6	5	4
Distance from AL	-	1.9 mi	1.5 mi	3.0 mi

The weight (in pounds) and sex of each alligator is as in the following tables:

Alligator Lake	1	2	3	4	5	6	7	8
weight	784	844	803	695	788	678	912	736
sex	f	m	m	f	f	f	m	m

Lake Hamburg	1	2	3	4	5	6
weight	744	988	812	733	676	793
sex	f	m	f	m	m	m

Lake Isabella	1	2	3	4	5
weight	977	853	511	645	780
sex	m	f	f	m	f

Lake Harper	1	2	3	4
weight	325	698	324	678
sex	f	f	m	m

The mathematician accepts the task, and asks his friend computer scientist for help. His friend suggests to formulate the problem as an optimization problem, where the objective function represents the total cost $c_1 + c_2$ of the operation. The computer scientist advises his friend about the constraints to be used: *(i)* as required by the mayor, all of the alligators must be replaced in the contaminated lake; *(ii)* as required by the FBI, two alligators must be sent to Gatorland; *(iii)* as suggested by the experts, only male alligators must be moved to Alligator Lake; *(iv)* obviously, the same alligator cannot be sent to Gatorland and Alligator Lake in the same time.

The mathematician writes a mathematical model in AMPL. He uses BONMIN as solver because there is a nonlinearity in the constraint *(iv)*. He removes the nonlinearity, solves the problem by using CPLEX, and checks if it is able to find the same exact solution. Happy for the obtained result, he runs and explains everything to the mayor. The mayor is happy for the solution as well. Satisfied, he gives the start to the operations, just after having ensured that the mathematician will never be able to tell about this operation to anybody.

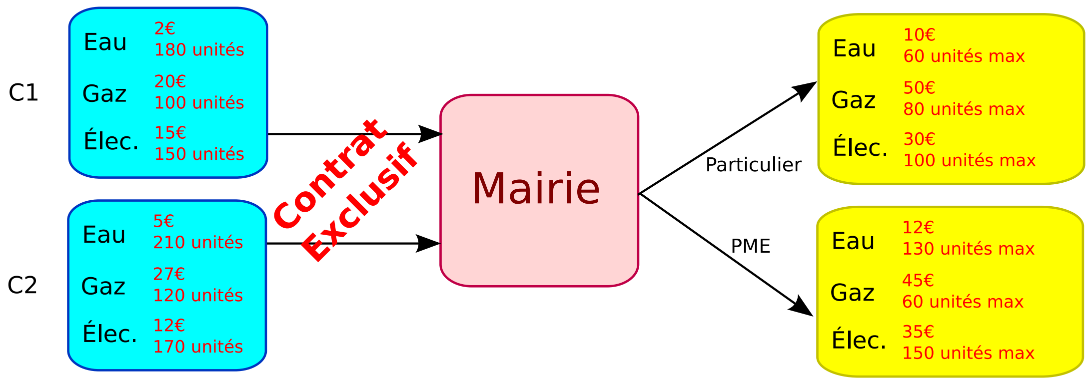
Follow the same steps as you would be the mathematician. Give your solution to the teacher, and not to any mayor!

3 Gestion des ressources

Une mairie décide d'optimiser la gestion en eau, gaz et électricité de sa commune. Elle s'occupe en effet de redistribuer l'ensemble de ces services à une partie de sa population. La mairie peut se fournir chez deux consortiums différents. Le premier propose l'unité d'eau à 2€, l'unité de gaz à 20€ et l'unité d'électricité à 15€. Le second consortium, quant à lui, vend respectivement ces unités à 5€, 27€ et 12€ (cf. tableau 1). Mais les deux consortiums réclament **l'exclusivité du contrat** ! Autrement dit, si la mairie décide d'acheter l'un des éléments chez un consortium, elle ne peut plus acheter les autres éléments chez le consortium concurrent.

La marie revend les unités achetées à un prix différent aux particuliers et aux PME. Les particuliers achètent l'unité d'eau à 10€, l'unité de gaz à 50€ et l'unité d'électricité à 30€. Les PME les achètent respectivement à 12€, 45€, et 35€ (cf. tableau 2).

La capacité de production des différents consortiums est répartie comme suit : le consortium 1 peut produire jusqu'à 180 unités d'eau, 100 de gaz et 150 d'électricité tandis que le consortium 2 peut produire respectivement 210, 120 et 170 unités (cf. tableau 3). De même, la capacité d'achat des différents acteurs est limitée comme suit : les particuliers peuvent acheter jusqu'à 60 unités d'eau, 80 de gaz et 100 d'électricité. Les PME peuvent absorber 130 unités d'eau, 60 unités de gaz et 150 unités d'électricité (cf. tableau 4).



Partie 1 : Programme non-linéaire. Proposez un programme mathématique (non linéaire) qui représente le problème et permette d'optimiser le profit réalisé. Vous éditerez pour cela 3 fichiers : un *.mod*, un *.dat* et un *.run*.

Partie 2 : linéarisation. Proposez une seconde modélisation qui soit linéaire[2]. Vous implémenterez cette linéarisation en 2 fichiers *.mod* et *.run*. Le fichier de données *.dat* doit rester inchangé.

	Eau (€/unités)	Gaz (€/unités)	Électricité (€/unités)
Consortium 1	2	20	15
Consortium 2	5	27	12

TAB. 1 – Prix d'achat en fonction des consortiums

[2]Indice : le produit xy entre une variable binaire x et une variable entière/continue $y \in [L, U]$ peut être linéarisé en remplaçant le produit xy par une variable entière/continue z et en ajoutant les contraintes suivantes :

1. $z \leq Ux$
2. $z \geq Lx$
3. $z \leq y - (1 - x)L$
4. $z \geq y - (1 - x)U$

	Eau (€/unités)	Gaz (€/unités)	Électricité (€/unités)
Particuliers	10	50	30
PME	12	45	35

Tab. 2 – Prix de vente en fonction des acheteurs

	Eau (unités)	Gaz (unités)	Électricité (unités)
Consortium 1	180	100	150
Consortium 2	210	120	170

Tab. 3 – Capacités de production des deux consortiums

	Eau (unités)	Gaz (unités)	Électricité (unités)
Particulier	60	80	100
PME	130	60	150

Tab. 4 – Capacités d'achats des particuliers et PME