Lista de exercícios Laboratório 2 – Programação inteira
Entrega: 30 de abril 2012

Preparação do trabalho

- A ferramenta é (como no lab 1) o GNU Linear Programming Kit (GLPK) em http://gnuwin32.sourceforge.net/packages/glpk.htm. (Eventualmente o pacote deve ser instalado novamente.)
- A documentação sobre os formatos CPLEX lp e GNU MathProg (sub-linguagem de AMPL) é disponível em http://www.inf.ufrgs.br/~mrpritt/lang.pdf e nas notas de aula.

Para cada um dos seguintes problemas frmule como programa inteira (pura,0-1 ou misto), implementa o modelo em AMPL e resolve com GLPK.

Exercício 1 (Investimento, fácil)
Uma empresa projetou quatro novos tipos de produtos e quer decidir quais deles e quantos de cada um ela vai produzir. A produção de cada tipo de produto gera um custo inicial e cada item produzido tem um preço de venda conforme a tabela

Produto	1	2	3	4
Custo inicial [KR$]	50	40	70	60
Preço [R$/item]	70	60	90	80

Além disso, o marketing definiu as seguintes restrições:

- A empresa pode produzir no máximo dois tipos de produtos.
- Os produtos de tipo 3 e 4 só podem ser produzidos se um tipo dos produtos 1 ou 2 for produzido.
- Para cada produto existe um mercado de no máximo 2000 items.

Formule um programa inteiro com o objetivo de maximizar o lucro total.

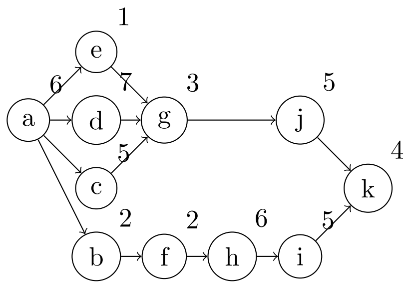
Exercício 2 (Quadrados latinos, médio)
Um *quadrado latino* de ordem n é um tabuleiro de tamanho $n \times n$ preenchido com n símbolos diferentes tal que toda linha ou coluna contém cada símbolo exatamente uma vez. Supõe que os símbolos são simplesmente o conjunto de números $[1, n]$. Um quadrado latino é em *forma normal* se os símbolos da primeira linha e coluna ocorrem em ordem crescente.

(a) Formule um programa linear, que gera um quadrado latino de tamanho n maximizando a soma dos elementos na diagonal principal.

(b) Estende a formulação tal que o quadrado latino gerado é em formal normal.

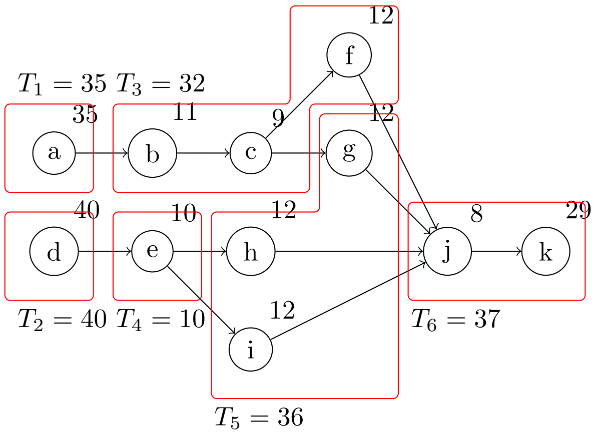
(c) Usa o solver para produzir um quadrado latino de tamanho 10×10.

Exercício 3 (Balanceamento de linhas de produção, difícil)
Uma linha de produção consiste numa sequencia linear de estacões de trabalho. Para montar um produto, uma serie de tarefas tem que ser executadas. Cada tarefa tem o seu próprio tempo de execução. Além disso, certas tarefas tem que ser executadas antes de outras. Isso é especificado através de um grafo direcionado sobre o conjunto de tarefas, tal que existe uma aresta (t_i, t_j) caso a tarefa t_i tem que ser executada antes da tarefa t_j. Veja um exemplo, com 11 tarefas:



(O tempo de execução de cada tarefa é o numero no seu lado.)
Para executar as tarefas numa linha de produção, temos que atribuí-las às estacões de forma que as restrições de precedência dado pelo grafo estão satisfeitas. Ou seja se uma tarefa t_i precede outra tarefa t_j, t_j tem que ser atribuído a mesma estação que t_i ou uma das estacões seguintes.
Dado uma atribuição, o tempo total de tarefas em cada estação é o *tempo de estação*. O maior tempo de estação define o *tempo de ciclo*. Dentro do tempo de ciclo cada estação e capaz de executar todas tarefas no produto atual e depois passar o produto parcialmente montado para a estação seguinte e receber um produto parcialmente montado da estação anterior.

a) Formula um programa linear que encontra para um dado número de estacões o menor tempo de ciclo possível. Veja mais um exemplo com a solução ótima:



(Nesse exemplo T_i e o tempo de estação da estação i. Observe como as estacões respeitam as precedências das tarefas. O tempo de ciclo dessa atribuição e 40, definido pela estação 2.)

b) Encontra a solução ótima para o primeiro exemplo acima usando o solver no caso de 3 estacões.

Exercício 4 (Sudoku, médio)
Formule um programa inteiro, que resolve Sudokus. Aplica o solver para resolver

							1	
					2			3
			4					
						5		
4		1	6					
		7	1					
	5					2		
				8			4	
	3		9	1				