Imperial College London
BUSINESS SCHOOL

MSc Business Analytics 2020/21
Optimisation and Decision Models
Wolfram Wiesemann

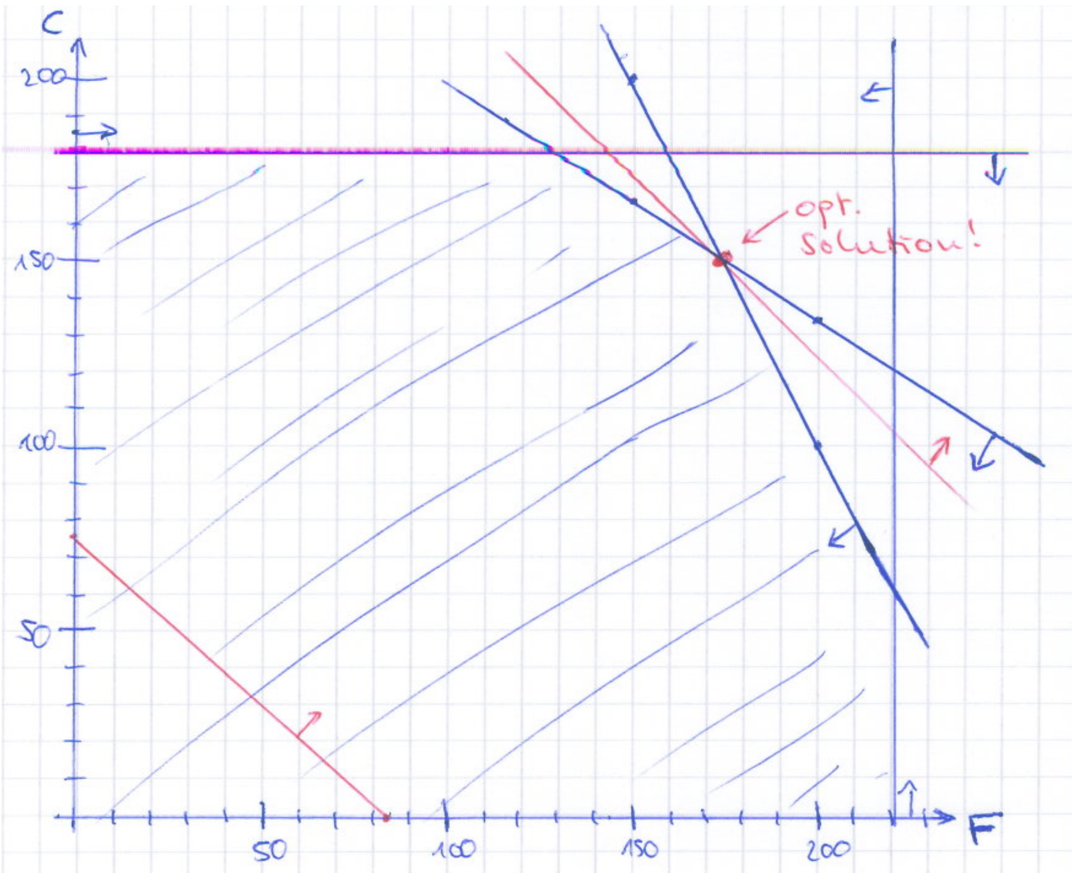
Solution to Assignment 1

Solution to (1) (a): Using the decision variables F and C for full-size and compact microwave ovens, respectively, the linear program reads as follows:

$$
\begin{array}{ll}
\text{maximise} & 120F + 130C \\
\text{subject to} & 2F + 1C \leq 500 \\
& 2F + 3C \leq 800 \\
& F \leq 220,\ C \leq 180 \\
& F,\ C \geq 0
\end{array}
$$

Here, the objective function maximises earnings, whereas the constraints ensure (in order) the satisfaction of the general assembly labour budget, the electronic assembly labour budget, the demand bounds and non-negativity of the production schedule.

Solution to (1) (b): The graphical solution looks as follows:



The optimal solution seems to be something like $F \approx 172$ and $C \approx 150$ with an objective value of

$$120 * 172 + 130 * 150 = 40{,}140.$$

The two labour constraints are binding at the optimal solution.

Solution to (1) (c): Please refer to the Excel sheet for the model. The optimal solution produces 175 full-size ovens and 150 compact ovens with a total earnings of £40,500.

Solution to (1) (c): Resolving the problem with 510 general assembly hours instead of 500 yields the higher earnings of £40,750, that is, an increase of £250. If we divide £250 by 10 to obtain the increase per hour, we obtain £25. In other words, Magnetron should pay at most £25 for each additional hour of general assembly time (within a certain range, as we will discuss later).

Solution to (2) (a): Using the decision variables S, B, H and V for the production amounts of stir fry, barbecue, hearty mushrooms and veggie crunch, respectively, the linear program reads as follows:

$$
\begin{array}{lrrrrl}
\text{maximise} & 0.22S + & 0.20B + & 0.18H + & 0.18V & \\
\text{subject to} & 62.5S + & 50B + & & 62.5V & \leq 3{,}750{,}000 \\
 & 75S + & & 100H & & \leq 2{,}000{,}000 \\
 & 62.5S + & 50B + & 75H + & 62.5V & \leq 3{,}375{,}000 \\
 & 50S + & 75B + & 75H + & 62.5V & \leq 3{,}500{,}000 \\
 & & 72B + & & 62.5V & \leq 3{,}750{,}000 \\
 & S,\ B,\ H,\ V \geq 0 & & & &
\end{array}
$$

Here, the objective function maximises earnings, whereas the constraints ensure (in order) the satisfaction of the carrots, mushrooms, green peppers, broccoli and corn constraints, as well as the non-negativity of the production schedule.

Solution to (2) (b): The AMPL model could look like this (see also the attached model):

```
# decision variables
var S;
var B;
var H;
var V;

# objective function
maximize earnings: 0.22 * S + 0.20 * B +  0.18 * H + 0.18 * V;

# constraints
subject to carrots: 62.5 * S + 50 * B + 62.5 * V <= 3750000;
subject to mushrooms: 75 * S + 100 * H <= 2000000;
subject to green_peppers: 62.5 * S + 50 * B + 75 * H + 62.5 * V <=
3375000;
subject to broccoli: 50 * S + 75 * B + 75 * H + 62.5 * V <= 3500000;
subject to corn: 72 * B + 62.5 * V <= 3750000;
subject to nn_S: S >= 0;
subject to nn_B: B >= 0;
subject to nn_H: H >= 0;
subject to nn_V: V >= 0;
```

The optimal solution is to produce 26,666.67 bags of Stir Fry, 18,333.33 bags of Barbecue and 12,666.67 bags of Veggie Crunch, with overall earnings of £11,813.33:

```
ampl: solve;
MINOS 5.51: optimal solution found.
3 iterations, objective 11813.33333

ampl: display S, B, H, V;
S = 26666.7
B = 18333.3
H = 0
```

```
V = 12666.7
```

We can verify which constraints are binding as follows:

```
ampl: display carrots.slack, mushrooms.slack, green_peppers.slack,
broccoli.slack, corn.slack;
carrots.slack = 375000
mushrooms.slack = 0
green_peppers.slack = 0
broccoli.slack = 0
corn.slack = 1638330
```

Thus, the mushrooms, the green peppers and the broccoli are fully used, whereas there are carrots and corn left.

Solution to (2) (c): There are two ways to answer this question. The first one is to resolve the problem with the updated quantity of green peppers:

```
# decision variables
var S;
var B;
var H;
var V;

# objective function
maximize earnings: 0.22 * S + 0.20 * B +  0.18 * H + 0.18 * V;

# constraints
subject to carrots: 62.5 * S + 50 * B + 62.5 * V <= 3750000;
subject to mushrooms: 75 * S + 100 * H <= 2000000;
subject to green_peppers: 62.5 * S + 50 * B + 75 * H + 62.5 * V <=
3475000;
subject to broccoli: 50 * S + 75 * B + 75 * H + 62.5 * V <= 3500000;
subject to corn: 72 * B + 62.5 * V <= 3750000;
subject to nn_S: S >= 0;
subject to nn_B: B >= 0;
subject to nn_H: H >= 0;
subject to nn_V: V >= 0;
```

We see that the overall earnings increase to £11,877.33, that is, the value of 100kg of additional green peppers is £11,877.33 - £11,813.33 = £64:

```
ampl: solve;
MINOS 5.51: optimal solution found.
3 iterations, objective 11877.33333
ampl: display S, B, H, V;
S = 26666.7
B = 14333.3
H = 0
```

```
V = 17466.7
```

Alternatively, we will soon see in class that we can use the sensitivity information to answer the question:

```
ampl: option solver cplex;
ampl: option cplex_options 'sensitivity';
ampl: solve;
CPLEX 12.6.3.0: sensitivity
CPLEX 12.6.3.0: optimal solution; objective 11877.33333
3 dual simplex iterations (1 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;

ampl: display green_peppers.down, green_peppers, green_peppers.up;
green_peppers.down = 3111110
green_peppers = 0.00064
green_peppers.up = 3750000
```

The report shows that the Green peppers constraint has a shadow price of £0.00064/g. Since an increase by 100kg is within the allowable increase, the answer is 100,000g * £0.00064/g = £64. *(Please ignore this part of the answer until we have covered the topic of sensitivity analysis!)*