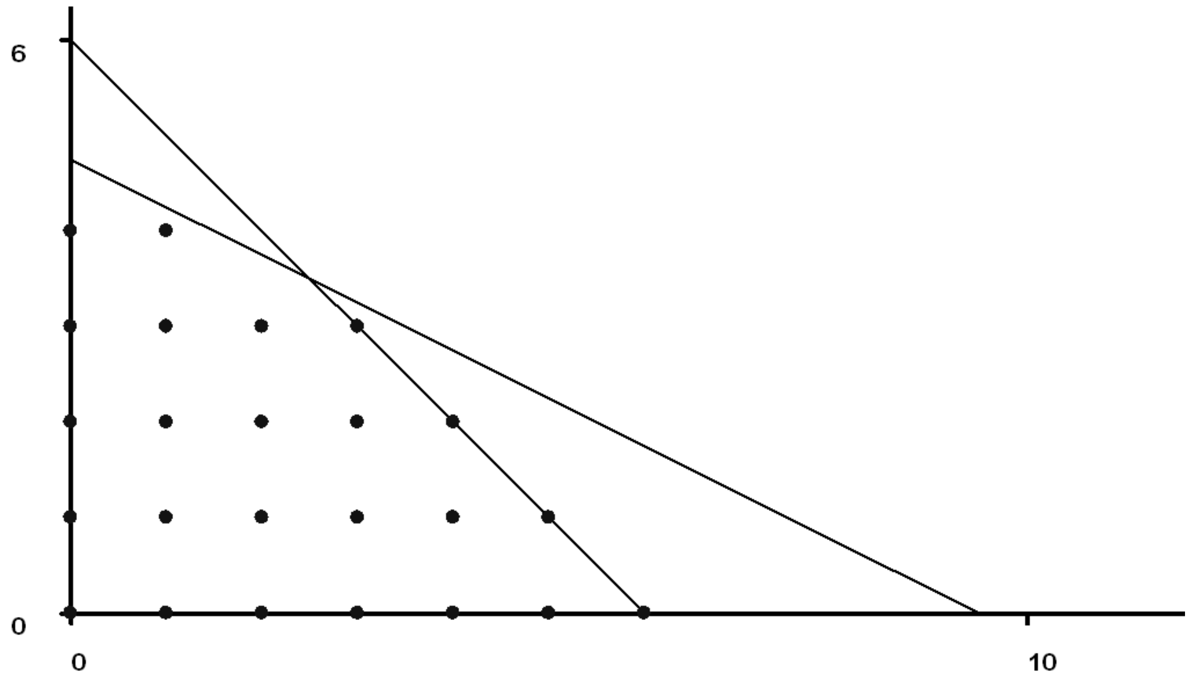
Branch-and-Bound

Background Information on VBA Programming in Business Economics by Sanne Wøhlk

Linear Programming Problems (LPs) can be solved relatively fast using the simplex algorithm, but that algorithm cannot be used to solve problems where variables are required to have integer values, the so-called Integer Linear Programming Problems (ILPs). An intuitive explanation is that the simplex algorithm moves from corner point (intersection of constraints) to corner point until it reaches an optimal solution. However, due to the integrality constraint corner points need not be feasible solutions for ILPs. This is illustrated by the below figure.



It is widely believed that it is not possible to construct a polynomial time algorithm that can solve general ILPs, but it has not been possible to prove this belief. Algorithms exist that can obtain near-optimal solutions to some ILPs in polynomial time, but the solutions are not guaranteed to be optimal. Other algorithms will eventually solve ILPs to optimality – if they do not run out of time or computer memory in the meantime, but it may – in theory - take years. The Branch-and-Bound algorithm is of the latter type: It will – in theory – solve any ILP to optimality but for many real size problems it will reach time or memory constraints before it reaches optimality.

Nevertheless, most commercial software for solving ILPs is based on a Branch-and-Bound algorithm. For those purposes, the algorithm is often combined with other algorithms, for example cutting planes.

We consider the LP relaxation of the problem, which is the problem obtained by removing the integrality constraint. Let S be the set of feasible solutions to the LP relaxation of a maximization problem and let f be the objective function. Hence, we are looking for a solution $s \in S \cap \mathbb{Z}$ such that $\{s \in S \cap \mathbb{Z} \mid \forall s' \in S \cap \mathbb{Z} : f(s') \leq f(s)\}$. The idea is to solve the LP relaxation of the problem and thereby obtain a solution $r \in S$ such that $\{r \in S \mid \forall r' \in S : f(r) \leq f(r)\}$. If $r \in S \cap \mathbb{Z}$ then the solution is optimal to the

ILP problem. Assume to the contrary that r is fractional and hence not feasible for the ILP problem. In this case, the set S is partitioned into two sets S' and S'' such that $r \notin S' \cup S''$ but $(S' \cup S'') \cap \mathbb{Z} = S \cap \mathbb{Z}$. This means that the fractional solution *r* is removed but all feasible integer constraints are preserved.

In practice the partitioning is often done as follows: Let *x* be a variable that has a fractional value in the solution r. Let the value of *x* be *k*. To construct S', add the constraint $x \leq \lfloor k \rfloor$ to the constraint set that formed S. On the other hand, to construct S'', add the constraint $x \geq \lceil k \rceil$ to the constraint set that formed S. Formally we have $S' = S \cap \{s \mid x \leq \lfloor k \rfloor\}$ and $S'' = S \cap \{s \mid x \geq \lceil k \rceil\}$. The process is now repeated with the solution of the LP relaxation based on each of these two sets. This process of solving LP relaxations and partitioning sets into two sets is repeated until the following is true for each set: Either the optimal solution of the LP relaxations is feasible for the ILP problem (i.e. it is integral) or it is proved that the globally optimal solution cannot be contained in the set. The latter is the case if the value of the LP relaxation is worse than the best known feasible solution.

An outline of the basic Branch-and-Bound algorithm is shown below, where *node* refers to a structure storing the separate constraint sets.

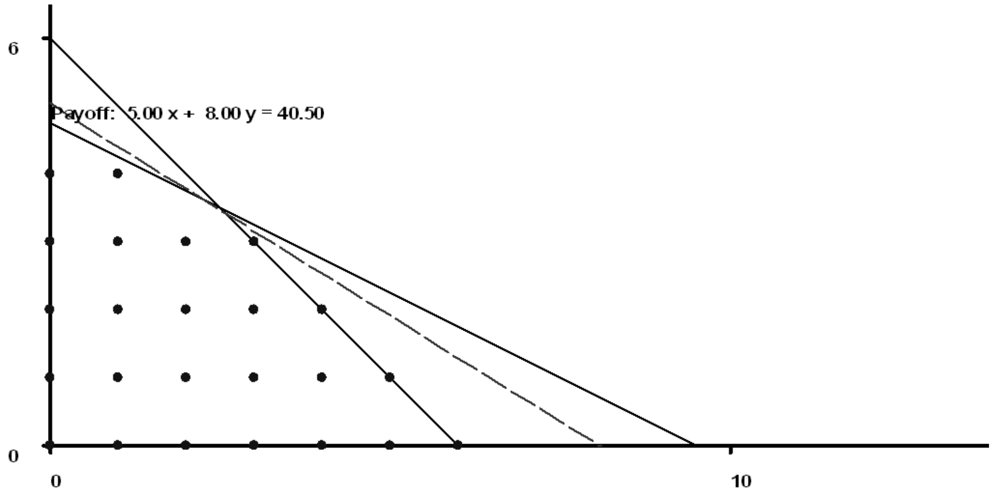
As long as there are still active nodes, repeat:

- Select an active node A
- Solve LP-relaxation of A
- If LP-relax is infeasible -> close A
- If the solution in A is integral, then
  - Close node
  - Update best feasible solution if this solution is better
- Else (the solution is fractional)
  - If LP-relax value ≤ best value, then
    - Bound node
  - Else Branch, i.e. partition A into sub nodes and make them active

For an illustration of the idea in the Branch-and-Bound algorithm, consider the following integer linear programming problem:

$$
\begin{aligned}
max \quad & 5x + 8y \\
s.t. \quad & x + y \leq 6 \\
& x + 2y \leq 10 \\
& x, y \geq 0 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

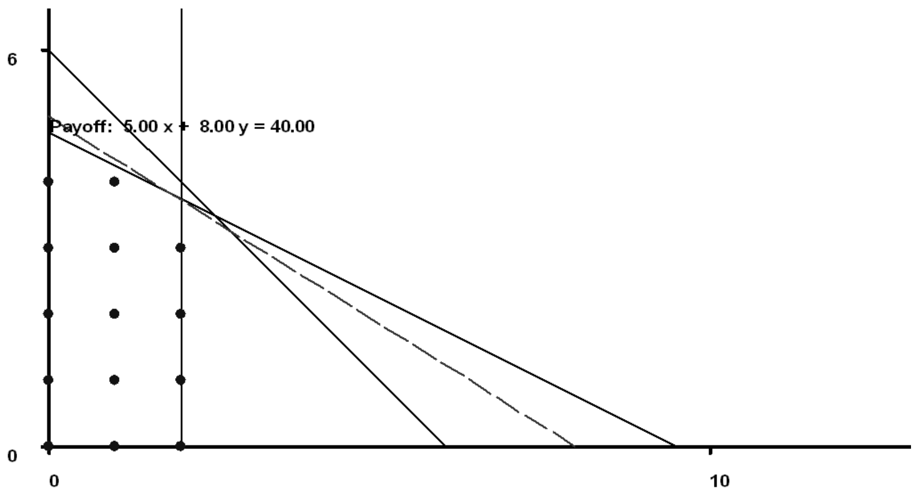
To solve the problem we first consider the LP relaxation of the problem by ignoring the integrality constraints. This is illustrated below where the optimal values of the variables in the LP relaxation are (x;y) = (2.5 ; 3.5) and the objective value is 40.5. The valid integer solutions are indicated in the figure but it should be stressed that these are not known directly when the solution process is initiated.



Optimal Decisions(x,y): (2.50, 3.50)
: 1.00x + 1.00y <= 6.00
: 1.00x + 2.00y <= 9.50
: 1.00x + 0.00y >= 0.00
: 0.00x + 1.00y >= 0.00

This solution is not valid with respect to the integrality constraints. Hence, we select a variable of fractional value, for instance x with value 2.5 and use that to partition the problem into two smaller problems. To do that we include the constraint $x \leq 2$ in one problem and $x \geq 3$ in the other. Let us refer to the two resulting problems as P_1 and P_2. Now, the process is repeated for each of these problems.
With the new constraint regarding *x*, P_1 is defined as follows with a graphical illustration below:

$$
\begin{aligned}
max \quad & 5x + 8y \\
s.t. \quad & x + \ \ y \leq 6 \\
& x + 2y \leq 10 \\
& x \qquad \leq 2 \\
& x, y \geq 0 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$



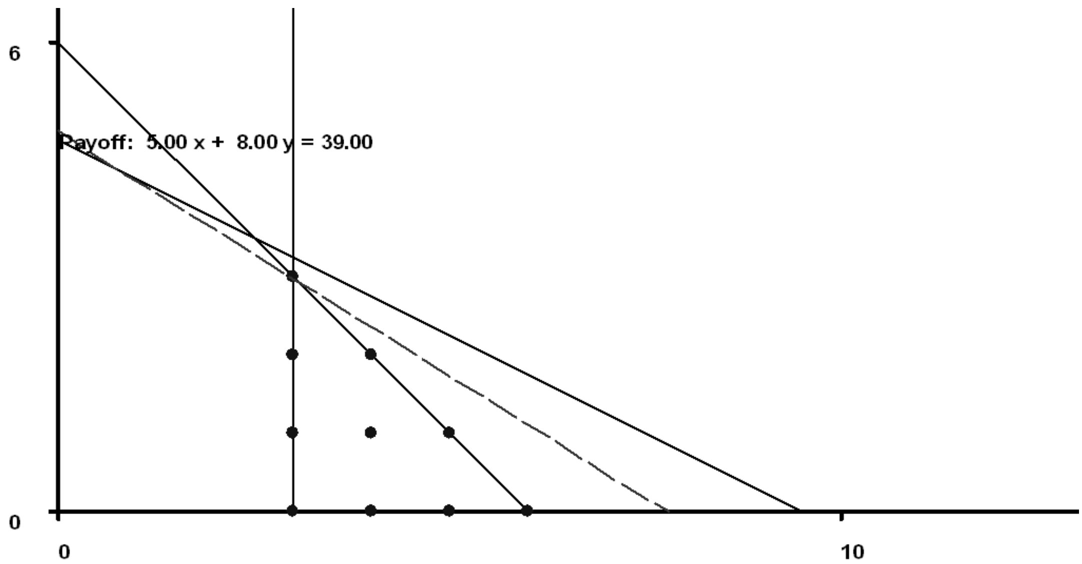
Optimal Decisions(x,y): (2.00, 3.75)
: 1.00x + 1.00y <= 6.00
: 1.00x + 2.00y <= 9.50
: 1.00x + 0.00y >= 0.00
: 0.00x + 1.00y >= 0.00
: 1.00x + 0.00y <= 2.00

The optimal solution of the LP relaxation is (x;y) = (2.0 ; 3.75). Because this solution again contains a variable of fractional value, we partition P_1 into two new problems by including the constraint $y \leq 3$ in one problem and $y \geq 4$ in the other. Let us refer to the two resulting problems as P_3 and P_4, respectively. Again these two problems must be considered separately.

At this point, three problems are left for consideration: P_2, P_3, and P_4. The order in which they are considered can be chosen freely. Many selection criteria have been experimentally tested but we will not go into detail about this. Instead we arbitrarily select P_2 to be considered first.

With the added constraint regarding *x*, P_2 is defined as follows with the graphical illustration given below:

$$
\begin{array}{ll}
max & 5x + 8y \\
s.t. & x + \ \ y \leq 6 \\
 & x + 2y \leq 10 \\
 & x \qquad \geq 3 \\
 & x, y \geq 0 \\
 & x, y \in \mathbb{Z}
\end{array}
$$

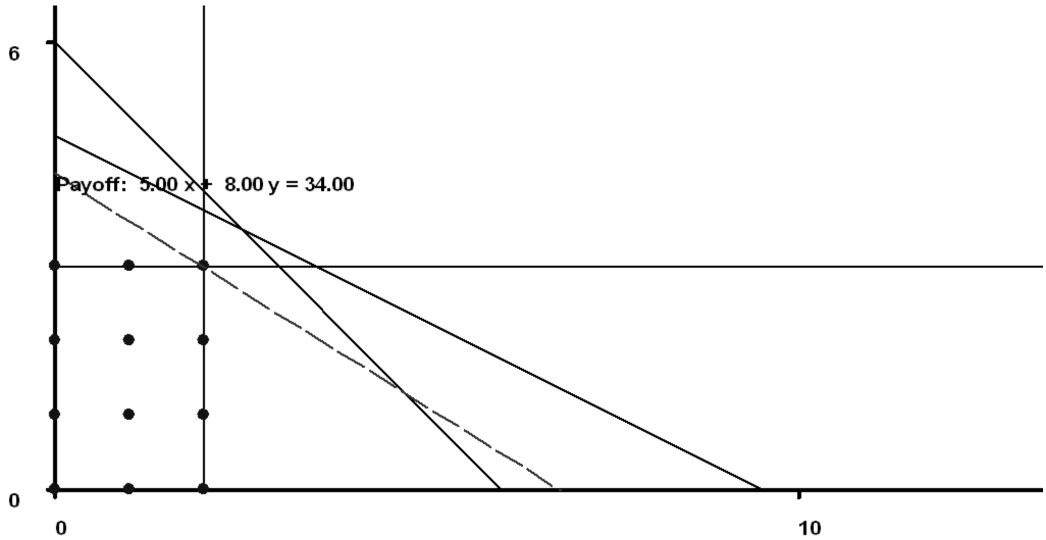


Optimal Decisions(x,y): (3.00, 3.00)
: 1.00x + 1.00y <= 6.00
: 1.00x + 2.00y <= 9.50
: 1.00x + 0.00y >= 0.00
: 0.00x + 1.00y >= 0.00
: 1.00x + 0.00y >= 3.00

Note that the optimal value of the LP relaxation of this problem does respect the integrality constraint. Hence we have found the first feasible solution: (x;y) = (3.0 ; 3.0) with an objective value of 39. We store this as the best solution obtained so far.

We choose to consider P_3 next. With the extra constraint added regarding *x* and the extra one added regarding *y*, the problem is stated as follows with the graphical illustration given below:

$$
\begin{aligned}
max \quad & 5\,x + 8\,y \\
s.t. \quad & x + \quad y \le 6 \\
& x + 2y \le 10 \\
& x \qquad \le 2 \\
& \qquad y \le 3 \\
& x, y \ge 0 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$



Optimal Decisions(x,y): (2.00, 3.00)
: 1.00x + 1.00y <= 6.00
: 1.00x + 2.00y <= 9.50
: 1.00x + 0.00y >= 0.00
: 0.00x + 1.00y >= 0.00
: 1.00x + 0.00y <= 2.00
: 0.00x + 1.00y <= 3.00

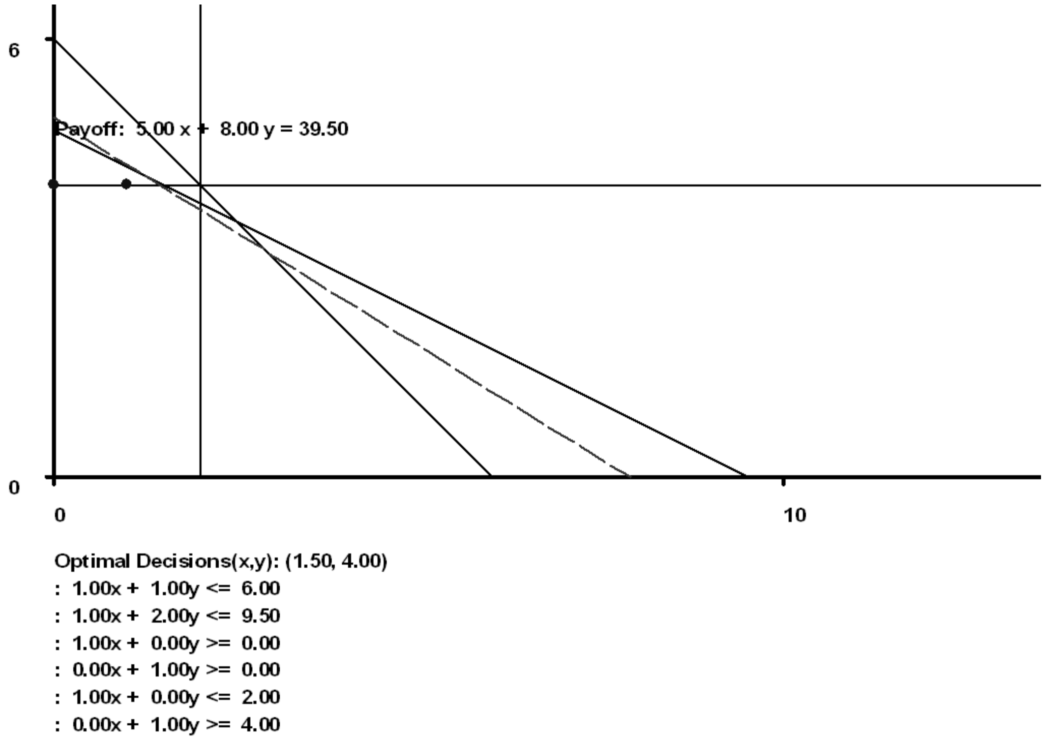
The optimal solution of the LP relaxation of P_3, (x;y) = (2.0 ; 3.0) is again integral and hence feasible for the original problem. The value of the objective function is, however, lower than the current best feasible solution. Therefore this new solution is worse and can be ignored.

The only problem left to consider is P_4 which is stated as follows with the graphical illustration given below:

$$
\begin{aligned}
max \quad & 5\,x + 8\,y \\
s.t. \quad & x + \quad y \le 6 \\
& x + 2y \le 10 \\
& x \qquad \le 2 \\
& \qquad y \ge 4 \\
& x, y \ge 0 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

The optimal solution of P_4 is (x;y) = (1.5 ; 4.0). This solution is fractional and hence the problem should be partitioned into two new problems. But the objective value of P_4 is 39.5. Inspection of the objection function teaches us that any feasible objection value must be integral because the coefficients are so. Therefore we can round the 39.5 down to 39 which is the highest possible value we can hope for when partitioning P_4 further.

Above we learned that the best feasible solution obtained so far has a value of 39 as well. Hence there is no need to consider P_4 further because we know now that it is not possible to obtain a better solution in this case by adding further constraints to P_4. We say that P_4 is bounded.



There are no more problems left for consideration and the algorithm terminates with (x;y) = (3.0 ; 3.0) being the optimal solution.