Botlhale Village

Working together for ICT innovation and growth in Africa

BELGIUM CAMPUS iTversity

It's the way we're wired

ARE YOU READY?

COMBINATORIAL OPTIMISATION PROBLEM

Combinatorial optimisation is a topic often found in applied mathematics and theoretical computer science that consists of finding an optimal object from a finite set of objects. Loosely speaking, a combinatorial optimisation problem is any optimisation problem that has a finite number of feasible solutions. In many such problems, exhaustive search is not feasible. Combinatorial optimisation operates on the domain of those optimisation problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution.

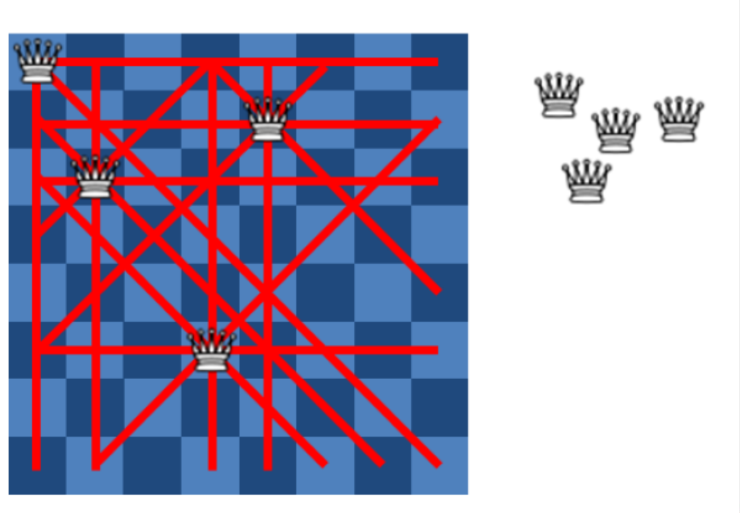
APPLICATION

Specific combinatorial optimisation problems include:

- Vehicle routing problem
- Travelling salesman problem
- Minimum spanning tree problem
- Linear programming, if the solution space is the choice of which variables to make basic
- Integer programming
- Eight queens puzzle, a constraint satisfaction problem
- Knapsack problem
- Cutting stock problem
- Assignment problem
- Weapon target assignment problem

THE EIGHT QUEENS PUZZLE

The eight queens puzzle is the problem of placing eight chess queens on an 8 × 8 chessboard so that no two queens attack each other. Thus, a solution requires that no two queens share the same row, column, or diagonal. This puzzle is an example of the more general n queens problem of placing n queens on an n × n chessboard, where solutions exist for all natural numbers n with the exception of 2 and 3.



SCENARIO

The problem can be computationally expensive as there are (64C8) > 4 × 1010 possible arrangements of eight queens on an 8 × 8 board, but only 92 solutions.

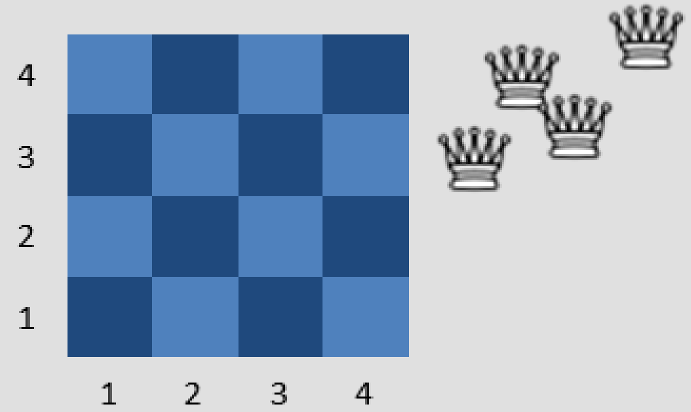
The Proposed **heuristic** and the explicit methods are interesting exercises to find solutions, but both methods of course find only one of the possible solutions to the particular problem.

Note: Remember from that there is only one unique solution with two distinct variations obtained from symmetry operations.

BRANCH-AND-BOUND FOR FOUR QUEENS PROBLEM

BELGIUM CAMPUS iTversity
It's the way we're wired

Use branch-and-bound to determine a way (if any exists) to place four queens on a 4 × 4 chessboard so that no queen can capture another queen.



Formulating the IP

- Decision variables:

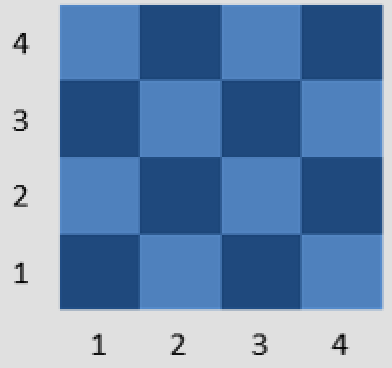
  $x_{ij} = If\ a\ queen\ is\ placed\ in\ row\ i\ column\ j\ (1) or\ not\ (0)\ where\ i = j = 1 - 4$

- Objective function:

$$\max z = x_{11} + x_{12} + x_{13} + x_{14} + \\ x_{21} + x_{22} + x_{23} + x_{24} + \\ x_{31} + x_{32} + x_{33} + x_{34} + \\ x_{41} + x_{42} + x_{43} + x_{44}$$

- Row constraints: One queen per row so that it cannot capture another queen

$$x_{11} + x_{12} + x_{13} + x_{14} \leq 1$$
$$x_{21} + x_{22} + x_{23} + x_{24} \leq 1$$
$$x_{31} + x_{32} + x_{33} + x_{34} \leq 1$$
$$x_{41} + x_{42} + x_{43} + x_{44} \leq 1$$



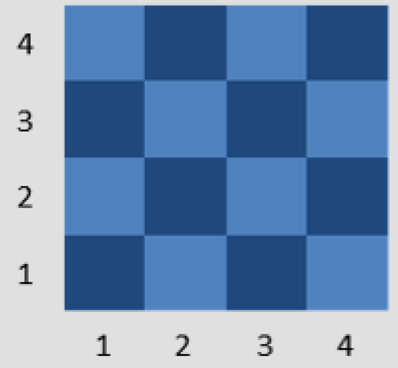
Formulating the IP

- Column constraints: One queen per column so that it cannot capture another queen

$$x_{11} + x_{21} + x_{31} + x_{41} \leq 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} \leq 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} \leq 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} \leq 1$$

- Diagonal / constraints: At most one queen per rising diagonal

$$x_{31} + x_{42} \leq 1$$
$$x_{21} + x_{32} + x_{43} \leq 1$$
$$x_{11} + x_{22} + x_{33} + x_{44} \leq 1$$
$$x_{12} + x_{23} + x_{34} \leq 1$$
$$x_{13} + x_{24} \leq 1$$



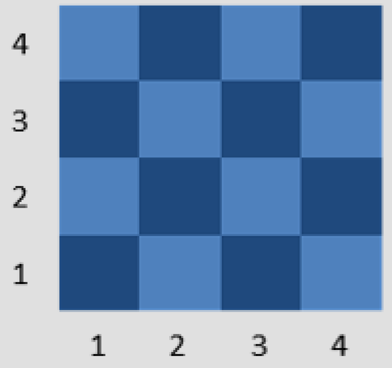
Formulating the IP

- Diagonal \ constraints: At most one queen per falling diagonal

$$x_{21} + x_{12} \leq 1$$
$$x_{31} + x_{22} + x_{13} \leq 1$$
$$x_{41} + x_{32} + x_{23} + x_{14} \leq 1$$
$$x_{42} + x_{33} + x_{24} \leq 1$$
$$x_{43} + x_{34} \leq 1$$

- Sign restrictions:

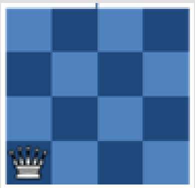
$$x_{ij} = 0 \; or \; 1$$



Branch & Bound Algorithm – Combinatorial method

Prob. 1

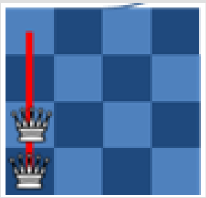
(x11 = 1)



Viable

Prob. 1.1

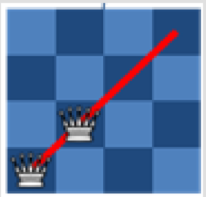
(x21 = 1)



Infeasible

Prob. 1.2

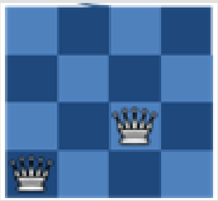
(x22 = 1)



Infeasible

Prob. 1.3

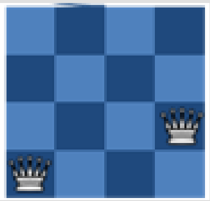
(x23 = 1)



Viable

Prob. 1.4

(x24 = 1)

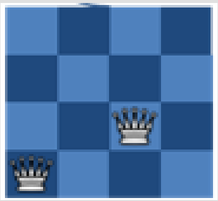


Viable

Branch & Bound Algorithm – Combinatorial method

Prob. 1.3

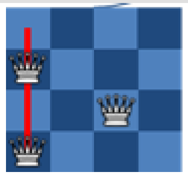
(x23 = 1)



Viable

Prob. 1.3.1

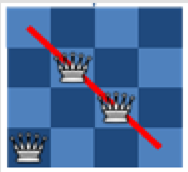
(x31 = 1)



Infeasible

Prob. 1.3.2

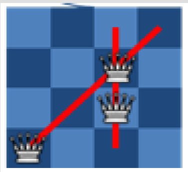
(x32 = 1)



Infeasible

Prob. 1.3.3

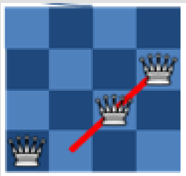
(x33 = 1)



Infeasible

Prob. 1.3.4

(x34 = 1)

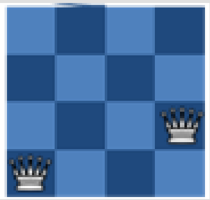


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 1.4

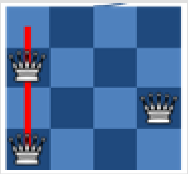
(x24 = 1)



Viable

Prob. 1.4.1

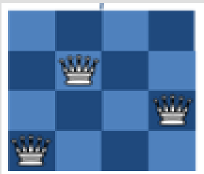
(x31 = 1)



Infeasible

Prob. 1.4.2

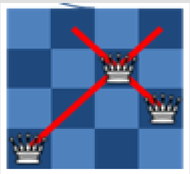
(x32 = 1)



Viable

Prob. 1.4.3

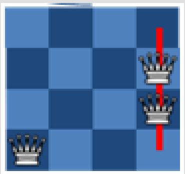
(x33 = 1)



Infeasible

Prob. 1.4.4

(x34 = 1)

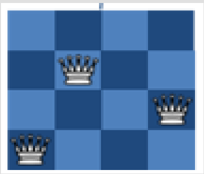


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 1.4.2

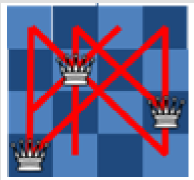
(x32 = 1)



Viable

Prob. 1.4.2.1

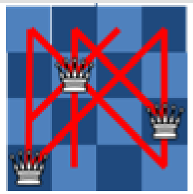
(x41 = 1)



Infeasible

Prob. 1.4.2.2

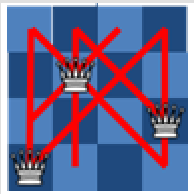
(x42 = 1)



Infeasible

Prob. 1.4.2.3

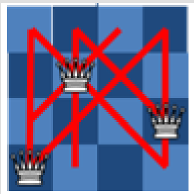
(x43 = 1)



Infeasible

Prob. 1.4.2.4

(x44 = 1)

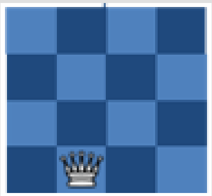


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 2

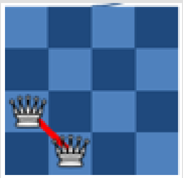
(x12 = 1)



Viable

Prob. 2.1

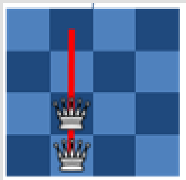
(x21 = 1)



Infeasible

Prob. 2.2

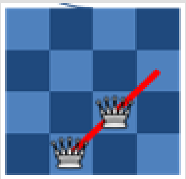
(x22 = 1)



Infeasible

Prob. 2.3

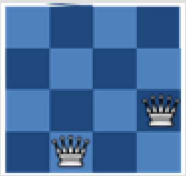
(x23 = 1)



Infeasible

Prob. 2.4

(x24 = 1)

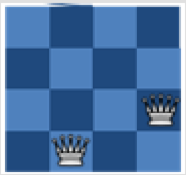


Viable

Branch & Bound Algorithm – Combinatorial method

Prob. 2.4

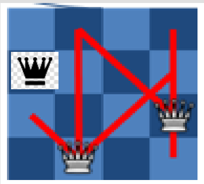
(x24 = 1)



Viable

Prob. 2.4.1

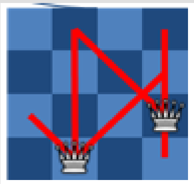
(x31 = 1)



Viable

Prob. 2.4.2

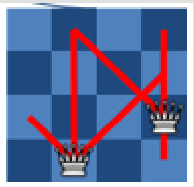
(x32 = 1)



Infeasible

Prob. 2.4.3

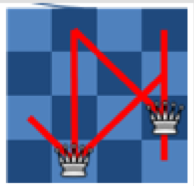
(x33 = 1)



Infeasible

Prob. 2.4.4

(x34 = 1)

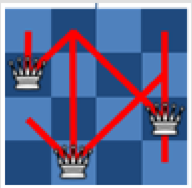


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 2.4.1

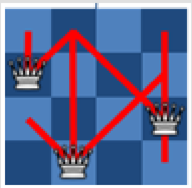
(x31 = 1)



Viable

Prob. 2.4.1.1

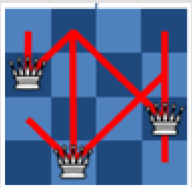
(x41 = 1)



Infeasible

Prob. 2.4.1.2

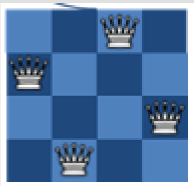
(x42 = 1)



Infeasible

Prob. 2.4.1.3

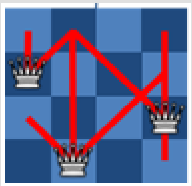
(x43 = 1)



Candidate

Prob. 2.4.1.4

(x44 = 1)

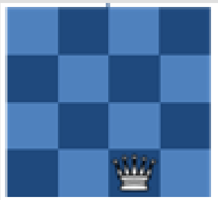


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 3

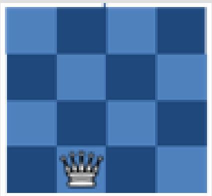
(x13 = 1)



Viable

Prob. 2

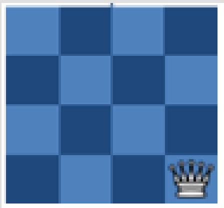
(x12 = 1)



Viable

Prob. 4

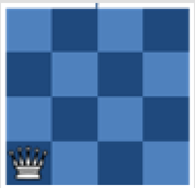
(x14 = 1)



Viable

Prob. 1

(x11 = 1)



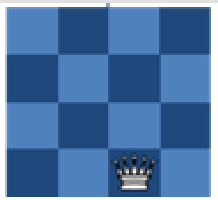
Viable

You need to continue with Problem 3 and 4.
Problem 4 will have no candidates like Problem 1.
Problem 3 will have a candidate like problem 2, but a symmetrical answer.

Branch & Bound Algorithm – Combinatorial method

Prob. 3

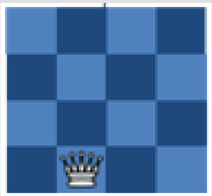
($x_{13} = 1$)



Viable

Prob. 2

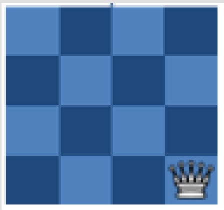
($x_{12} = 1$)



Viable

Prob. 4

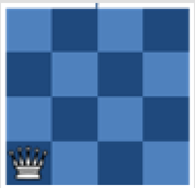
($x_{14} = 1$)



Viable

Prob. 1

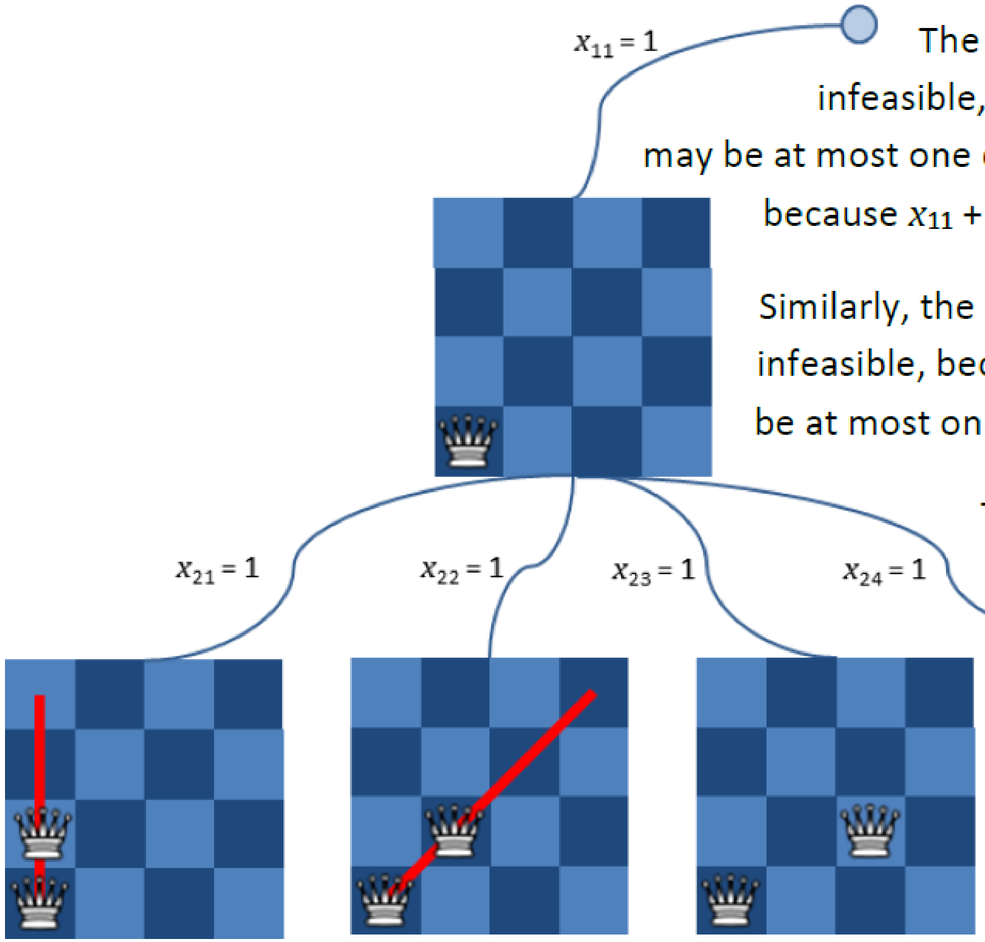
($x_{11} = 1$)



Viable

You need to continue with Problem 3 and 4.
Problem 4 will have no candidates like Problem 1.
Problem 3 will have a candidate like problem 2, but a symmetrical answer.

START WITH X11=1



The node associated with the arc $x_{11} = x_{21} = 1$ is infeasible, because it violates the constraint that there may be at most one queen in column 1, $x_{11} + x_{21} + x_{31} + x_{41} \leq 1$, because $x_{11} + x_{21} = 2 > 1$.

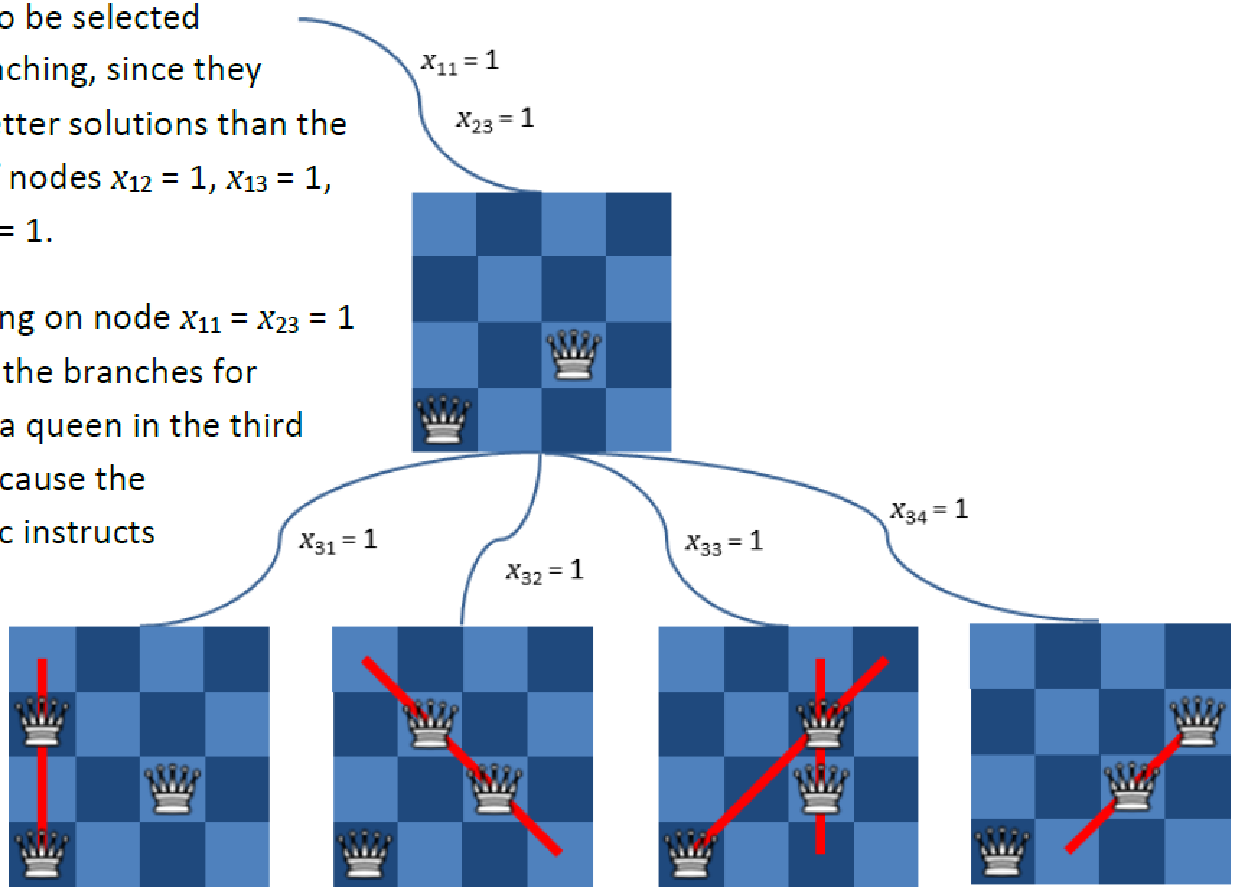
Similarly, the node associated with the arc $x_{11} = x_{22} = 1$ is infeasible, because it violates the constraint that there may be at most one on the diagonal, $x_{11} + x_{22} + x_{33} + x_{44} \leq 1$.

The branches associated with these infeasible nodes may be pruned, because it is known that they can only lead to infeasible solutions. These nodes are thus eliminated from consideration for further branching.

X11 TO X23

The nodes associated with the arcs $x_{11} = x_{23} = 1$ and $x_{11} = x_{24} = 1$ now have $Q = 2$ (two queens placed). Jumptracking now causes the selection of one of these nodes to be selected for branching, since they have better solutions than the $Q = 1$ of nodes $x_{12} = 1$, $x_{13} = 1$, and $x_{14} = 1$.

Branching on node $x_{11} = x_{23} = 1$ creates the branches for placing a queen in the third row, because the heuristic instructs that no

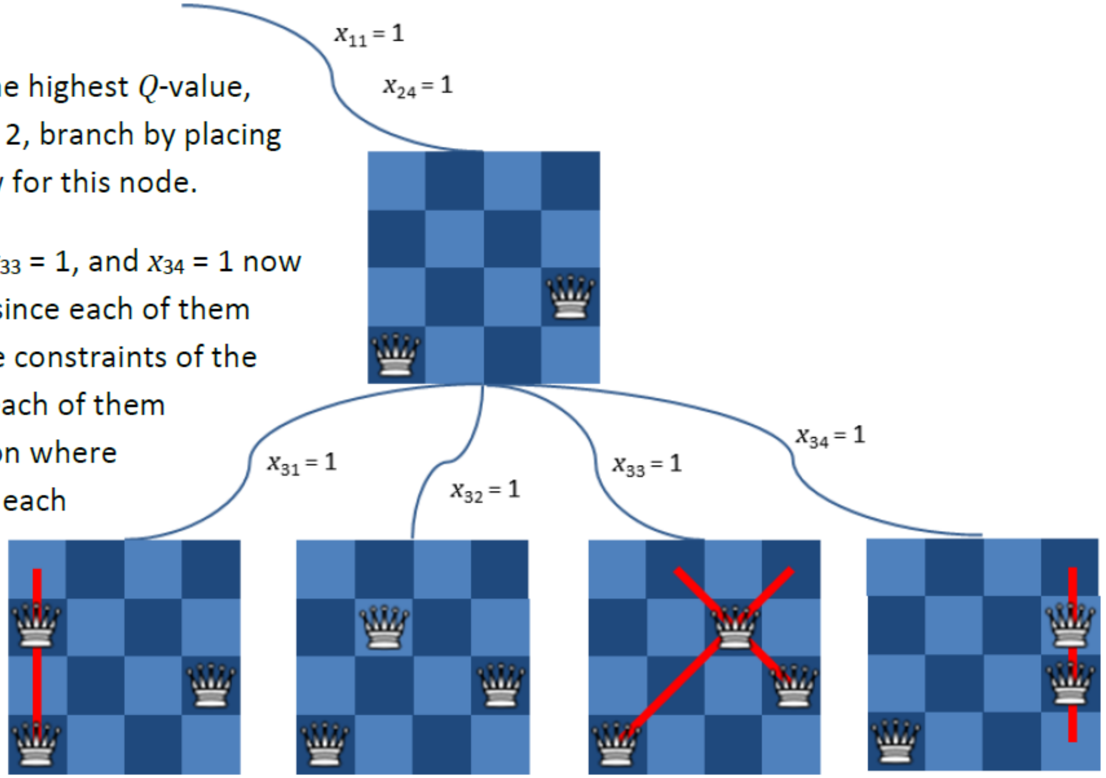


X11 TO X24

solutions.

Selecting the node with the highest Q-value, node $x_{11} = x_{24} = 1$ with $Q = 2$, branch by placing a queen into the third row for this node.

The branches for $x_{31} = 1$, $x_{33} = 1$, and $x_{34} = 1$ now turn out to be infeasible, since each of them violates at least one of the constraints of the problem, in other words each of them represents a board position where some queens can capture each other.

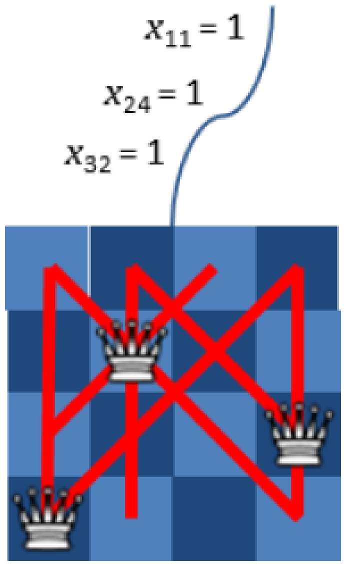


The node associated with the arc $x_{11} = x_{24} = x_{32} = 1$ represents a position where the queens already placed cannot capture each other, satisfying the constraints of the problem. This node is therefore retained and has a Q-value of 3, because it contains three queens successfully placed.

X11 TO X24 TO X32

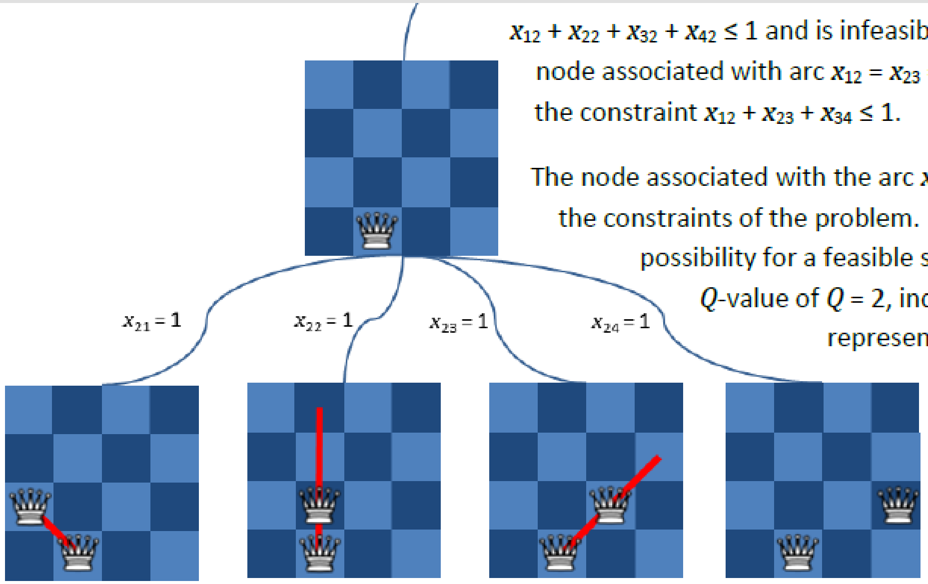
BELGIUM CAMPUS iTversity

It's the way we're wired



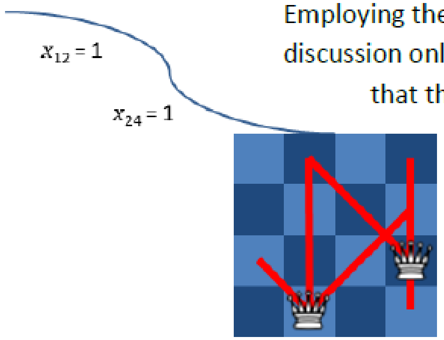
It is clear from the view on the left that the queens already placed for node $x_{11} = x_{24} = x_{32} = 1$ already cover every square in row 4, so all four possible branches from this node will lead to infeasible solutions. This idea may stimulate additional heuristics which may be applied this problem, like considering a node infeasible if a look-ahead reveals infeasibility of this type where no space remains available for future queen placement. The value of such a heuristic should be compared to the cost incurred to calculate it at every step. If too many calculations are required to look ahead effectively, the method may simply branch and discover the infeasibility of the subsequent nodes after branching.

X21 TO X24 TO X31

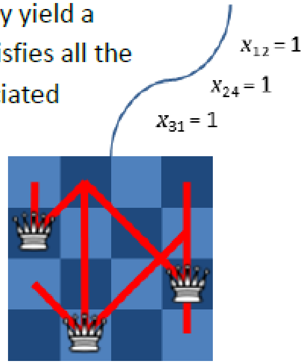


$x_{12} + x_{22} + x_{32} + x_{42} \leq 1$ and is infeasible as a result; as is the node associated with arc $x_{12} = x_{23} = 1$ because it violates the constraint $x_{12} + x_{23} + x_{34} \leq 1$.

The node associated with the arc $x_{12} = x_{24} = 1$ satisfies all the constraints of the problem. This node is thus a possibility for a feasible solution and has a Q-value of $Q = 2$, indicating that this node represents the successful placement of two queens that are not able to capture each other. Its Q-value is higher than the $Q = 1$ of the nodes $x_{13} = 1$ and $x_{14} = 1$, so the jumptracking principle indicates that this is the next node to branch on.

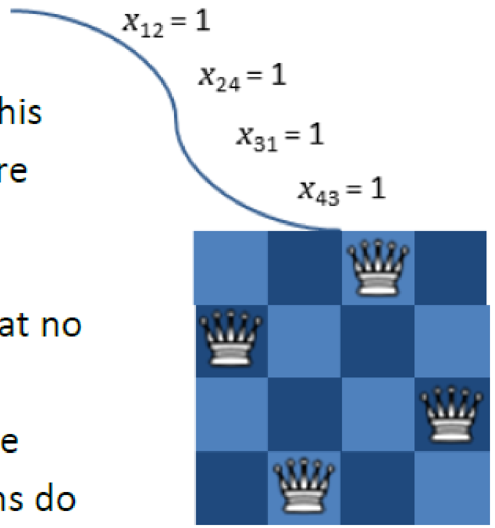


Employing the look-ahead heuristic from earlier (as a convenience for this discussion only – it will be too expensive to implement in general) it is seen that the only square in the third row that can possibly yield a feasible solution (or, the only branch that satisfies all the constraints) is branch $x_{31} = 1$. The node associated with this branch has $Q = 3$ (three queens placed) and thus the node $x_{12} = x_{24} = x_{31} = 1$ is the next node selected by jumptracking to branch on.

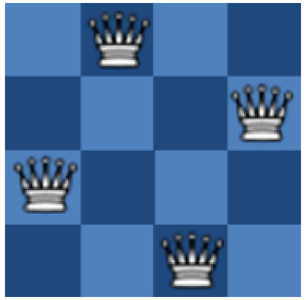


X21 TO X24 TO X31

The look-ahead shows that only branch $x_{43} = 1$ yields a feasible solution. The node associated with arc $x_{12} = x_{24} = x_{31} = x_{43} = 1$ is a feasible solution because it satisfies all the constraints of the problem. This node is thus a candidate solution with $Q = 4$, meaning that four queens are successfully placed with no queen able to capture any other queen.



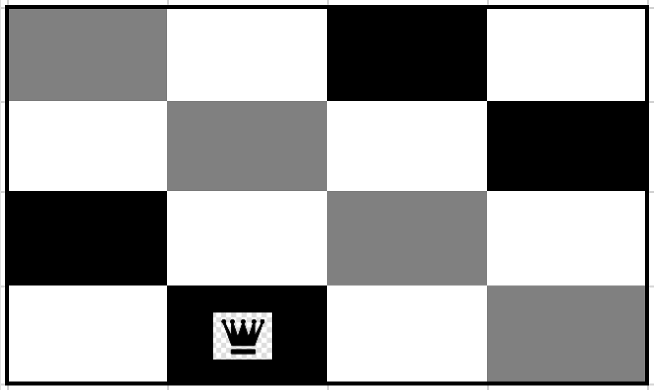
Heuristics indicate that this is an optimal solution, because it is known that no solution can exist with $Q > 4$. If the purpose is simply to find any optimal solution, the method may stop here. The distinct variations of this unique solution may be performed by performing symmetry operations; rotations do not yield new solutions, but both reflections yield the (same) distinct variation, $x_{13} = x_{21} = x_{34} = x_{42} = 1$. This solution also has $Q = 4$ and is an alternative optimal solution.



Alternatively, this solution may be found by branching on the unfathomed nodes $x_{13} = 1$ and $x_{14} = 1$. This will be necessary if an arbitrary problem space is being searched by the branch-and-bound method to find all feasible solutions.

SCENARIO – EXAMPLE 2

Use branch-and-bound to determine a way (if any exists) to place four queens on a 4 × 4 chessboard so that no queen can capture another queen. The queens cannot be placed in the gray areas of the chessboard and a queen cannot move over these areas. Start in x_{12}



Formulating the IP

- Decision variables:

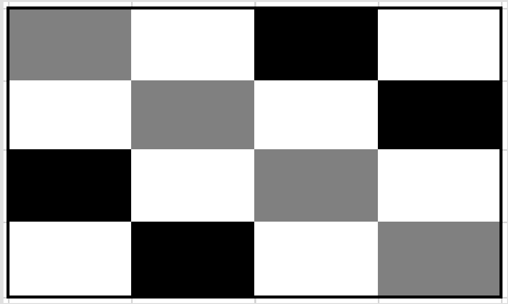
$x_{ij} = If\ a\ queen\ is\ placed\ in\ row\ i\ column\ j\ (1) or\ not\ (0)\ where\ i = j = 1 - 4$

- Objective function:

$$\max z = x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{24} + x_{31} + x_{33} + x_{34} + x_{42} + x_{43} + x_{44}$$

- Row constraints:

$$x_{11} + x_{12} + x_{13} \leq 1$$
$$x_{21} + x_{22} \leq 1$$
$$x_{33} + x_{34} \leq 1$$
$$x_{42} + x_{43} + x_{44} \leq 1$$



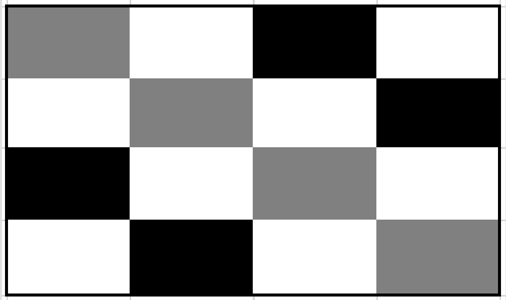
Formulating the IP

- Column constraints:

$$x_{11} + x_{21} + x_{31} \leq 1$$
$$x_{12} + x_{22} \leq 1$$
$$x_{33} + x_{43} \leq 1$$
$$x_{24} + x_{34} + x_{44} \leq 1$$

- Diagonal / constraints:

$$x_{31} + x_{42} \leq 1$$
$$x_{11} + x_{22} + x_{33} + x_{44} \leq 1$$
$$x_{13} + x_{24} \leq 1$$



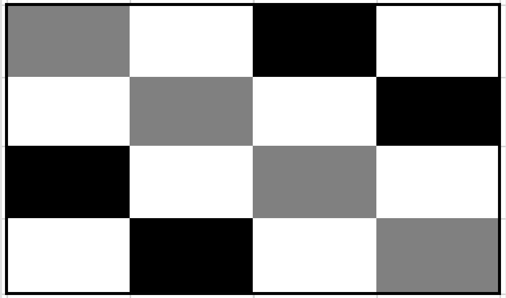
Formulating the IP

- Diagonal \ constraints:

$$x_{21} + x_{12} \leq 1$$
$$x_{31} + x_{22} + x_{13} \leq 1$$
$$x_{42} + x_{33} + x_{24} \leq 1$$
$$x_{43} + x_{34} \leq 1$$

- Sign restrictions:

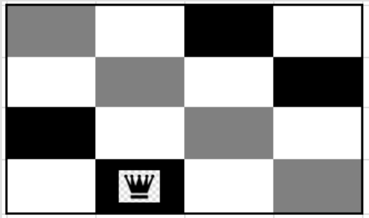
$$x_{ij} = 0 \text{ or } 1$$



Branch & Bound Algorithm – Combinatorial method

Prob. 1

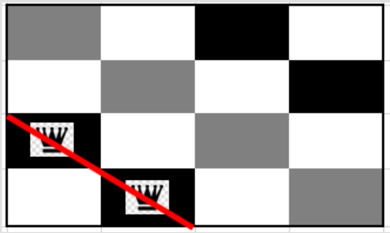
(x12 = 1)



Viable

Prob. 1.1

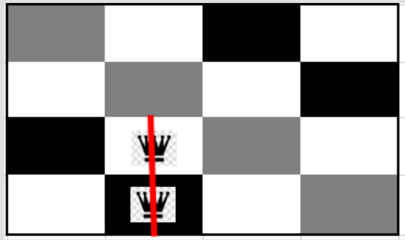
(x21 = 1)



Infeasible

Prob. 1.2

(x22 = 1)

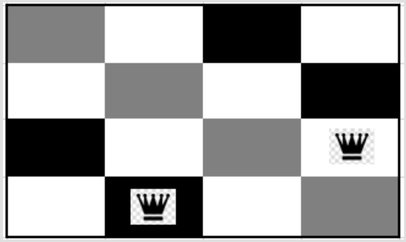


Infeasible

(x23 = 0)

Prob. 1.3

(x24 = 1)



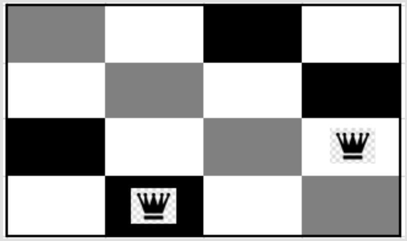
Viable

Branch & Bound Algorithm – Combinatorial method

BELGIUM CAMPUS iTversity
It's the way we're wired

Prob. 1.3

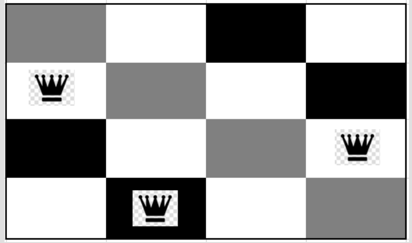
(x24 = 1)



Viable

Prob. 1.3.1

(x31 = 1)

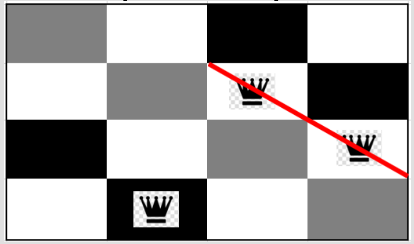


Viable

(x32 = 0)

Prob. 1.3.2

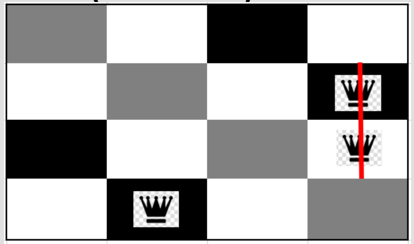
(x33 = 1)



Infeasible

Prob. 1.3.3

(x34 = 1)

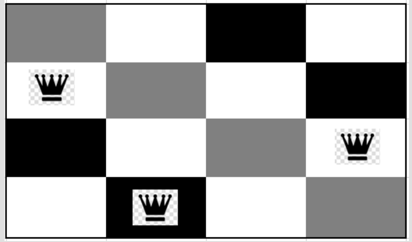


Infeasible

Branch & Bound Algorithm – Combinatorial method

Prob. 1.3.1

(x31 = 1)

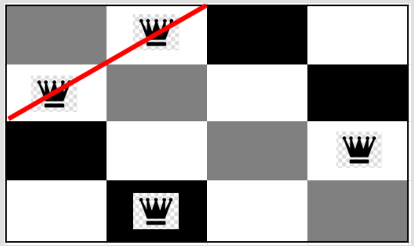


Viable

(x41 = 0)

Prob. 1.3.1.1

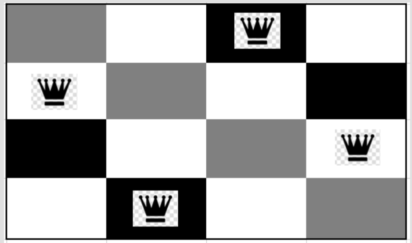
(x42 = 1)



Infeasible

Prob. 1.3.1.2

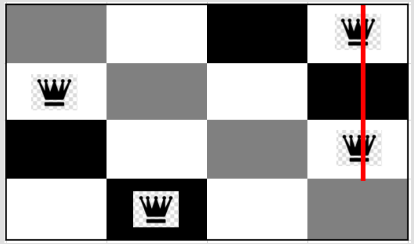
(x43 = 1)



Candidate

Prob. 1.3.1.3

(x44 = 1)

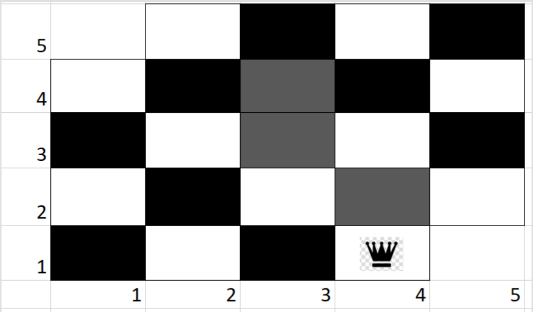


Infeasible

x12, x24, x31, x43 OR the reflection

Exercises

Five queens must be placed on a 5 x 5 chessboard, so that they cannot capture each other. A queen cannot be placed in x15, x51 and the gray areas. A queen cannot move over the gray areas.



- Formulate an Integer Programming Model that will solve the given problem.
- Solve the formulated Integer Programming Model using Solver.
- Solve the formulated Integer Programming Model using the Branch & Bound Combinatorial Algorithm.
- Solve the formulated Integer Programming Model using the Branch & Bound Simplex Algorithm.
- Solve the formulated Integer Programming Model using the Cutting Plane Algorithm.

END

CONNECT WITH US

info@belgiumcampus.ac.za

+27 10 593 5368

+27 (0) 12 543-1617

PO Box 60327,
Karenpark 0118,
South Africa

@BelgiumCampusSA

@BelgiumCampus

/Belgium Campus

Tshwane Campus
138 Berg Avenue
Heatherdale, Pretoria

Ekurhuleni Campus
45A Long Street
Kempton Park

Nelson Mandela Bay Campus
6 Uitenhage Road
North End, Port Elizabeth,