

Attention

- Use a separate folder for each problem.
- Create a project (.mzp) for each problem, when there are multiple model or data files involved.
 - Add the model files (*.mzn) and the data files (*.dzn)
- Configure the solver to obtain the solution statistics and to set a time limit (300 seconds).
- Use commas when reporting big numbers. E.g.,
 - 976474 instead of 976,474
- Submit one single zip file.

N-Queens

- With the alldifferent model (without symmetry breaking), search for **a solution** for $N = 30, 35, 45, 50$, using the following 6 variable - value ordering heuristics of Gecode:
 - input order – min value
 - input order – random value
 - min domain size – min value
 - min domain size – random value
 - domWdeg – min value
 - domWdeg – random value
- Record the number of failures ('-' for timed out instances) in a table.
- For each instance, indicate the best results in bold.
- Observe your results.

Comments

- Present briefly your observations. Pay attention to the following points.
 - For a given variable ordering, how does random value choice change the performance?
 - For a given value ordering, how does a dynamic variable ordering change the performance?
 - Are there any heuristics that behave similarly? Explain the reason.

Poster Placement

- Use the global model and the instances provided in the data files.
- Order the main decision variables X and Y as $[X_1, Y_1, X_2, Y_2, X_3, Y_3, \dots, X_n, Y_n]$.
- Based on this ordering, search for **a solution** using the following 4 variable - value ordering heuristics of Gecode:
 - input order – min value
 - input order – random value
 - domWdeg – min value
 - domWdeg – random value
- Record the number of failures ('-' for timed out instances) in a table.
- For each instance, indicate the best results in bold.

Poster Placement

- Then, re-order the rectangles in the data file in decreasing order by their perimeter and repeat the previous experiments.
- Record the number of failures ('-' for timed out instances) in a table.
- For each instance, indicate the best results in bold.

Comments

- Present briefly your observations. Pay attention to the following points.
 - For a given variable ordering, how does random value choice change the performance? Explain the reason.
 - Which heuristic reveals the best performance? Explain the reason.
 - Which heuristic is not affected much by the order of the rectangles? Explain the reason.

Quasigroup Completion Problem

- Implement a model using alldifferent constraints.
- Search for **a solution** to the instances given in the data files using Gecode and experiment with:
 - default search
 - domWdeg – random value
 - domWdeg – random value + restarting (employing the Luby strategy with $L = 250$)
- Record the number failures and time ('-' for timed out instances).
- For each instance, indicate the best results in bold.

Comments

- Present briefly your observations. Pay attention to the following points.
 - When does programming search do not improve the default search?
 - When restarting the domWdeg – random value heuristic degrade the performance, what could be the reason?
 - Which search approach is the best overall? Justify your answer.