

9 Domino

Ci sono n tessere t_1, \dots, t_n costituite ognuna da una coppia di numeri di valore compreso da 1 a k . Ci possono essere tessere ripetute. Le tessere (e k e un intero h di cui parliamo sotto sono l'input). Denotiamo con $t_i(\ell)$ il numero a sinistra e con $t_i(r)$ il numero più a destra della tessera t_i (l'ordine conta, non come nel domino in cui sono rappresentati da puntini).

Si vuole trovare una sequenza i_1, \dots, i_ℓ tale che per ogni $j = 1, \dots, \ell - 1$ $t_{j+1}(r) = t_j(\ell)$ di lunghezza ℓ maggiore possibile. Ogni tessera può essere usata al massimo una volta nella sequenza.

C'è una ulteriore difficoltà: hai a disposizione una scacchiera di lato h (più piccolo di ℓ) e ogni tessera usa due celle e può essere messa in orizzontale o in verticale. Non puoi sovrapporre le celle.

1. Write a Minizinc **and** an ASP program capable of finding an optimal solution consisting with the constraints.
2. Prepare a battery of benchmark instances in the following way. For $n = 20, 30, 40, 50, 60$ and $k = n/5$ and $h = 7, 8, 9, 10, 11$ (or different values if the instances seem too easy / too difficult) generate random instances. 100 instances in total.
3. Run both the Minizinc and the ASP encoding on all the instances, possibly exploring different search strategies, with a timeout of 5 minutes for each test ("configuration" option in Minizinc, -time-limit option in clingo, in both cases you can use linux tricks).
4. Write a 6–10 pages report containing your models (and the reasons for some choices) and a presentation of the execution results. Prepare the programs and the benchmark instances used in a unique zip file.