BMW Quantum Challenge: Optimizing the Production of Test Vehicles

Robert M. Parrish[1,*]

[1] *QC Ware Corporation, Palo Alto, CA 94301, USA*

Notes of the formulation and implementation of BMW test vehicle production optimization problem.

FORMULATION

Notes on Problem Input Refinement

Group Feature Collision

Their consultant is just trying to be an asshole:

Group 40 (28 elements): `[245, 246, 247, 250, 251, 252, 253, 254, 255, 256, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 284]`

Group 41 (46 elements): `[245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 285, 286, 287, 288, 289, 290, 291, 292, 293]`

Union (48 elements): `[245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293]`

Intersection (26 elements): `[245, 246, 247, 250, 251, 252, 253, 254, 255, 256, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282]`

In Group 40 but not Group 41 (2 elements): `[266, 284]`

In Group 41 but not Group 40 (20 elements): `[248, 249, 257, 258, 259, 260, 261, 262, 263, 264, 283, 285, 286, 287, 288, 289, 290, 291, 292, 293]`

This is hugely vexing for efficient enumeration of group-feature-satisfying vehicle candidates.

This can be overcome by (1) redefining Group 40 to be `[266, 284]` and then (2) adding a new global (added for all types) rule to the build rules:

```
F266 | F284 => !F245 & !F246 & !F247 & !F250 & !F251 & !F252 & !F253 & !F254 & !F255 & !F256 & !F267 & !F268 & !F269 & !F270 & !F271 & !F272 & !F273 & !F274 & !F275 & !F276 & !F277 & !F278 & !F279 & !F280 & !F281 & !F282
```

If the group features are chosen randomly, uniformly, and independently, this rule has a probability of $2/(1+2)$ to be activated (if 266 xor 284 are true). The probability of the rule being violated is $\sim 26/(1+46) \sim 0.55$. Therefore the joint probability of the rule being activated and failing is $(2/3)*(26/47) \sim 0.37$. Note that this high success probability is somewhat accidental, and is only due to the fact that the in-40-but-not-in-41 subset is small relative to the intersection *and* the in-41-but-not-in-40 subset is large relative to the the intersection. In future, it is recommended that collisions between feature groups be avoided at all costs in the formulation of this problem, insofar as is possible.

Notes on Problem Input Specialization

Notes on Problem Rules

We consider certain classes of rules observed in the pilot problem inputs to be universal, and denote such choices here:

1. Type allowed feature indices are nondegenerate and sorted during parsing to be stored in ascending order.
2. Group feature indices are nondegenerate and sorted during parsing to be stored in ascending order.
3. The union of Group feature indices is nondegenerate after refinement.
4. Not all Types have all Groups active.
5. Not all Type features are contained in a Group.
6. Not all Group features are valid for an intersecting Type.

Notes on Problem Sizes

1. The number of Types n_T is 25.
2. The number of Features n_F is 469.

Notes on Problem Seeding

1. A guess of $\vec{0}$ does not satisfy build rules for any type. Between 2 and 4 (inclusive) build rules are

violated, depending on the type. All implications of these build rules are "any on" types. The cardinalities are 2: 3, 3: 15, 4: 6.

2. Valid cars of all types can be found by chasing the $\vec{0}$ build rules zero levels deep to fix the predicates of the violated rules. There are between 4 and 609 valid cars per type.

3. Valid Type 2, 13, 24 cars (the smallest $\vec{0}$ rule violators with $2\times$ rules each) can be found by chasing the $\vec{0}$ build rules violations one level deep to fix the implications of the violated rules. There are 150, 175, and 175 valid cars, respectively.

4. Valid Type 5 cars (the smallest type) can be found efficiently by random search.

---

* Electronic address: rob.parrish@qcware.com