# BMW Quantum Challenge: Optimizing the Production of Test Vehicles

Robert M. Parrish[1,*] and Rachael Al-Saadon[1]

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



A complete and completely classical solution of the industrial challenge problem is presented. Additional gains are potentially possible for extended versions of this problem and/or for similar problems if hybrid quantum/classical algorithms are considered - we present some ideas along these lines.

## RESULTS

The problem statement(s) variously ask for optimization of the constituents of a set or "constellation" of $n_\mathrm{C}$ test vehicles, with each test vehicle taken from a state space of $\sim 459$ binary dimensions (this and other dimensions quoted below to vary in future problem sizes), and with each test vehicle satisfying hard "feature-group" and "type-build rule" constraints corresponding to $\sim 25$ basic test vehicle types. The problem statement(s), predicated by the hard constraints, specifically ask for (1) **SAT:** For a given $n_\mathrm{C}$, does there exist, for a given set of $n_\mathrm{test} \sim 644$ tests depending through binary expressions on the state space of each test vehicle, a set of $n_\mathrm{C}$ test cars for which the $n_\mathrm{test}$ tests can be separately evaluated, with the caveat that there need be $K_I \sim 1-5$ distinct test vehicles required to satisfy test $I$ for $I \in [0, n_\mathrm{test})$? (2) **Weighted MAX-SAT:** For a given $n_\mathrm{C}$, what is the optimal constellation of test vehicles such that the weighted sum of satisfied $n_\mathrm{test}$ tests, each requiring $K_I$ distinct test vehicles, is maximized? and (3) **Scheduling (not precisely specified):** For a given set of $n_\mathrm{test}$ tests and corresponding set of $n_\mathrm{C}$ test vehicles satisfying said tests including $\{K_I\}$ multiplicity constraints in a MAX-SAT formalism of (2), what is the optimal scheduling of said vehicles into a test sequence with at most $n_\mathrm{slot} \sim 10$ tests performed on distinct cars in each timeslot and with tests assigned to integer test groups with definite sorting of test groups within each car?

Within the problem statement document, solutions to the above problems were attempted using existing industry-standard sat solvers and constraint satisfaction solvers. The SAT problem of (1) was easily solved: "For 100 cars, the problem can be solved in a few seconds. A linear search counting down from 100 revealed the solution that at least 60 cars are needed to perform all the specified 750 tests." However the weighted MAX-SAT problem of (2) was not solvable: "On the other hand, the MAX-SAT problem was not solvable in a reasonable time with the chosen approach. Additionally, the scheduling problem of (3) was not solvable: "[O]n the test laptop, the full problem with 700 tests wasn't solvable in less than 24 hours."

**We provide what we believe under the rules of the problem statement represents a complete and tangible solution to all three specified problems.** Specifically, we developed a custom C++/Python code library to represent the details of the problem in a natural format. The combination of customized classical solution environment and high performance implementation allows for very rapid exploration of the hard-constraint-satisfying parameter space unique to this problem class. Within this environment, we developed a powerful and simple set of heuristics to approximately solve the MAX-SAT variant of the problem. This heuristic MAX-SAT solver produces nested constellations of test cars with increasing $n_\mathrm{C}$ and concomitant increasing MAX-SAT scores. The MAX-SAT solutions coming from this heuristic achieve saturation of all specified 644 tests (including multiplicity considerations) at the same $n_\mathrm{C} = 60$ bound determined by standard SAT solvers for problem (1) in the problem statement document. Thus our MAX-SAT solution provides a tight bound solution for problem (1) in the process of providing approximate solutions for (2). For values of $n_\mathrm{C} \ll 60$, we believe our heuristic MAX-SAT solutions are within a few percent of the global optimum. For the scheduling problem of (3) we develop additional heuristics to schedule the test sequence from the MAX-SAT optimized constellation fo $n_\mathrm{C} = 60$ cars while respecting the hard constraints of distinct cars within each time slot, strict ordering of randomly-specified test groups within cars, and separate cars used within the multiplicity considerations of each test. With the multiplicity considerations included, there are 766 separate test-car pairs required, mandating a theoretical floor of 77 test slots. Our heuristic solution provides a nearly dense scheduling with 78 test slots required, i.e., within 1.3% of dense scheduling.

## FORMULATION

*Initial Guess*

* Electronic address: rob.parrish@qcware.com