Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer

Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López de Prado

***Abstract*—We solve a multi-period portfolio optimization problem using D-Wave Systems' quantum annealer. We derive a formulation of the problem, discuss several possible integer encoding schemes, and present numerical examples that show high success rates. The formulation incorporates transaction costs (including permanent and temporary market impact), and, significantly, the solution does not require the inversion of a covariance matrix. The discrete multi-period portfolio optimization problem we solve is significantly harder than the continuous variable problem. We present insight into how results may be improved using suitable software enhancements and why current quantum annealing technology limits the size of problem that can be successfully solved today. The formulation presented is specifically designed to be scalable, with the expectation that as quantum annealing technology improves, larger problems will be solvable using the same techniques.**

***Index Terms*—Optimal trading trajectory, portfolio optimization, quantum annealing.**

I. The problem

A. Introduction

CONSIDER an asset manager wishing to invest K dollars in a set of N assets with an investment horizon divided into T time steps. Given a forecast of future returns and the risk of each asset at each time step, the asset manager must decide how much to invest in each asset at each time step, while taking into account transaction costs, including permanent and temporary market impact costs.

One approach to this problem is to compute the portfolio that maximizes the expected return subject to a level of risk at each time step. This results in a series of "statically optimal" portfolios. However, there is a cost to rebalancing from a portfolio that is locally optimal at t to a portfolio that is locally optimal at $t+1$. This means that it is highly likely that there will be a different series (or, a "trajectory") of portfolios that will be "globally optimal" in the sense that its risk-adjusted returns will be jointly greater than the combined risk-adjusted returns from the series of "statically optimal" portfolios.

Mean-variance portfolio optimization problems are traditionally solved as continuous-variable problems. However, for assets that can only be traded in large lots, or for asset managers who are constrained to trading large blocks of assets, solving the continuous problem yields an approximation, and a discrete solution is expected to give better results. For example, institutional investors are often limited to trading "even" lots (due to a premium on "odd" lots), that is, lots that are an integer multiple of a standard lot size, in which case the problem becomes inherently more discrete as the trade size increases versus the lot size. This could occur, for example, due to the trading of illiquid assets. Two common examples of block trading are ETF-creation and ETF-redemption baskets, which can only be traded in large multiples, such as fund units of 100 000 shares each.

The discrete problem is non-convex due to the fragmented nature of the domain, and is therefore much harder to solve than a similar continuous problem. Furthermore, our formulation allows the covariance matrix to be ill-conditioned or degenerate. This complicates the finding of a solution using traditional optimizers since a continuous relaxation would still be non-convex, and therefore difficult to solve.

B. Previous work

The single-period discrete portfolio optimization problem has been shown to be NP-complete, regardless of the risk measure used [1], [2]. Jobst *et al.* [3] showed that the efficient frontier of the discrete problem is discontinuous and investigated heuristic methods of speeding up an exact branch-and-bound algorithm for finding it. Vielma *et al.* presented a branch-and-bound algorithm and results for up to 200 assets [4]. Heuristic approaches, including an evolutionary algorithm, were investigated by other authors [1], [5], [6].

Bonami and Lejeune [7] solved a single-period problem with integer trading constraints, minimizing the risk given a probabilistic constraint on the returns (with no transaction costs), and finding exact solutions via a branch-and-bound method for problems with up to 200 assets. They considered four different methods, of which one was able to solve the largest problems to optimality in 83% of the cases, but the other three failed for all problems (the average run time for the largest problems was 4800 seconds). They found that solving the integer problem was harder than solving a continuous problem of the same size with cardinality constraints or minimum buy-in thresholds.

Manuscript received September 30, 2015; revised April 20, 2016; accepted May 14, 2016. Date of publication June 01, 2016; date of current version August 12, 2016. This work was supported by 1QB Information Technologies (1QBit) and Mitacs. The guest editor coordinating the review of this manuscript was Daniel. P. Palomar.

G. Rosenberg and P. Goddard are with 1QBit, Vancouver, BC V6C 2B5, Canada (e-mail: gili.rosenberg@1qbit.com; phil.goddard@1qbit.com).

P. Haghnegahdar is with the Department of Physics and Astronomy University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: phaghneg@phas.ubc.ca).

P. Carr is with the Courant Institute of Mathematical Sciences, New York University (NYU), New York, NY 10012 USA (e-mail: Peter.P.Carr@morganstanley.com).

K. Wu is with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: kwu@lbl.gov).

M. López de Prado is with the Guggenheim Partners LLC, New York, NY 10017 USA and also with the Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: Marcos.LopezDePrado@guggenheimpartners.com).

Digital Object Identifier 10.1109/JSTSP.2016.2574703

Gârleanu and Pedersen [8] solved a continuous multi-period problem via dynamic programming, deriving a closed-form solution when the covariance matrix is positive definite, thereby offering insight on the properties of the solutions to the multi-period problem. A multi-period trade execution problem was treated analytically by Almgren and Chriss [9], motivating our inclusion of both temporary and permanent price-impact terms.

The connection between spin glasses and Markowitz portfolio optimization was shown by Galluccio *et al.* [10]. The discrete multi-period problem was suggested by López de Prado [11] as being amenable to solving using a quantum annealer. The contribution of this paper is to investigate the implementation and solution of a similar discrete multi-period problem on the D-Wave quantum annealer.

C. Integer formulation

The portfolio optimization problem described above may be written as a quadratic integer optimization problem. We seek to maximize returns, taking into account the risk and transaction costs, including temporary and permanent market impact (the symbols are defined in the Appendix),

$$w = \text{argmax}_w \sum_{t=1}^{T} \Big(\mu_t^T w_t - \frac{\gamma}{2} w_t^T \Sigma_t w_t - \Delta w_t^T \Lambda_t \Delta w_t + \Delta w_t^T \Lambda_t' w_t \Big), \quad (1)$$

subject to the constraints that the sum of holdings at each time step be equal to K,

$$\forall t : \sum_{n=1}^{N} w_{\text{nt}} = K, \quad (2)$$

and that the maximum allowed holdings of each asset be K',

$$\forall t, \forall n : w_{\text{nt}} \leq K'. \quad (3)$$

The first term in Eq. 1 is the sum of the returns at each time step, which is given by the forecast returns μ times the holdings w. The second term is the risk, in which the forecast covariance tensor is given by Σ, and γ is the risk aversion. The third and fourth terms encapsulate transaction costs. Specifically, the third term includes any fixed or relative direct transaction costs, as well as the temporary market impact, while the fourth term captures any permanent market impact caused by trading activity. The transaction cost term is square in the change in the holdings, so it penalizes changes in the holdings if the corresponding entry in Λ_t is positive [8]. The permanent market impact term allows for the fact that increasing a large holding requires executing a large buy order, which increases the price, and hence the returns [9].

D. Extensions

A straightforward extension can be made to solve optimal trade execution problems. For example, in order to solve a problem in which the asset manager has K units invested and would like to liquidate them over T time steps, the constraint in Eq. 2 would change to

$$\forall t : \sum_{n=1}^{N} w_{\text{nt}} \leq K \quad (4)$$

and the sum in the transaction cost and permanent impact terms would extend to time step $T+1$ with $w_{T+1} = 0$ (the zero vector).

The risk term in Eq. 1 requires the estimation of a covariance matrix for each time step. There are cases in which this is problematic: for example, if not enough data exists for a good estimate or if some of the assets were not traded due to low liquidity. An alternative and more direct way to quantify risk is via the variance of the returns stream of the proposed trajectory. This avoids the issues in the estimation of covariance matrices. An additional advantage of this method is that it does not assume a normal distribution of returns. The disadvantage is that if the number of time steps T is small, the estimate of the true variance of the proposed trajectory will be poor. A high variance of returns is penalized regardless of whether it occurs due to positive or negative returns.

The variance is quadratic in the returns, so it is a suitable term to include in a quadratic integer formulation. We use the identity $\text{Var}(r) = \langle r^2 \rangle - \langle r \rangle^2$, and note that the returns stream is given by the vector $r[w] = \text{diag}(\mu^T w)$. We find the alternative risk term

$$\text{risk}[w] = \frac{\gamma}{T} \sum_{t=1}^{T} \left[\left(\mu_t^T w_t \right)^2 - \frac{1}{T} \sum_{t'=1}^{T} \left(\mu_t^T w_t \right) \left(\mu_{t'}^T w_{t'} \right) \right]. \quad (5)$$

II. Solution using a quantum annealer

A. Quantum Annealing

Quantum annealing is a process which can be used to find the optimal solution to optimization problems, if these problems can be encoded as a Hamiltonian [12], [13]. To this end, the quantum system is first prepared such that it represents a trivial problem, and is in the ground state of that problem, which is an equally weighted superposition of all possible states. The system is then transformed continuously to the point that it represents the optimization problem that we want to solve. If this process is done slowly enough, the adiabatic theorem guarantees that the system will remain in the ground state, as long as external disturbances are absent. The state of the system is then read, and in the ideal case it would correspond to the optimal solution of the optimization problem we wish to solve [14]. This process is referred to as "adiabatic quantum computation." In a real device, external interference always exists to some degree, so the result is probabilistic, and annealing the same problem multiple times increases the probability of finding the optimum. Therefore, quantum annealers are effectively heuristic solvers.

It has been argued that quantum annealing has an advantage over classical optimizers due to quantum tunnelling. Quantum tunnelling allows an optimizer to more easily search the solution space of the optimization problem, thereby having a higher probability of finding the optimal solution. This might provide a

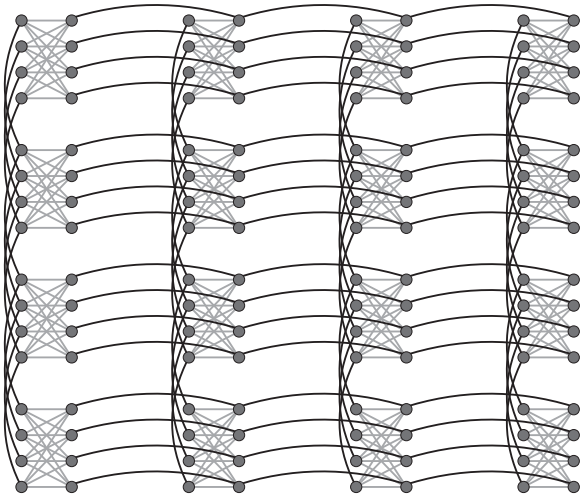

Fig. 1. An example hardware graph, showing the connectivity of the qubits for a Chimera graph with $s = 4$ unit cells in each row/column, giving a total of $q = 128$ qubits.

speed improvement over classical optimizers, at least for certain problem classes [12], [13], [15]–[17].

D-Wave Systems has developed a scalable quantum annealer. Mathematically, this is a device which minimizes unconstrained binary quadratic functions,

$$\min x^T Q x \qquad (6)$$
$$s.t.\, x \in \{0,1\}^N,$$

where $Q \in \mathbb{R}^{N\times N}$ [18]–[20]. In order to keep external disturbances to a minimum, D-Wave's quantum annealer is cooled to 15 mK (about 180 times colder than interstellar space), is shielded from RF signals due to its being housed inside a metal enclosure, is shielded from external magnetic fields larger than 1 nT (about 50 000 times less than Earth's magnetic field), and operates in a high-vacuum environment in which the pressure is 10 billion times lower than atmospheric pressure [21].

There is strong evidence that the D-Wave machine is indeed quantum [22], [23]. Recently, there has been significant interest in benchmarking the D-Wave machines using different metrics, and often against classical solvers [24]–[29]. There is an ongoing debate on how to define quantum speedup, and on which problems a noisy quantum annealer would be expected to show such a speedup [30]–[32]. Recently, Denchev *et al.* claimed a 10^8 speedup over simulated annealing when solving a specially constructed class of problems on a single-core machine [33]. It is still an open question whether D-Wave's quantum annealer shows a quantum speedup. We expect new results to shed light on this in the near future.

The connectivity of the qubits in D-Wave's quantum annealer is currently described by a square Chimera graph [34]. This hardware graph is composed of a lattice of bipartite unit cells containing eight qubits. Qubits in adjacent unit cells are connected if they are in the same position in the unit cell (see Fig. 1 for an example).

If we label the number of unit cells along an edge s, then the total number of qubits is $q = 8s^2$. The hardware graph is sparse and in general does not match the problem graph, which is defined by the adjacency matrix of the problem matrix Q. In order to solve problems that are denser than the hardware graph, we identify multiple physical qubits with a single logical qubit (a problem known as "minor embedding" [35], [36]), at the cost of using many more physical qubits. For square Chimera hardware graphs, the size V of the largest fully dense problem that can be embedded on a chip with q qubits is $V = \sqrt{2q} + 1 = 4s + 1$, assuming no faulty qubits or couplers. For example, the latest chip is the D-Wave 2X,[1] which has $s = 12$ unit cells along each side, giving $q = 1152$ qubits, for which we get $V = 49$. Lower-density problems of significantly larger size can be embedded. For example, on one of the annealers used in this study, which has 1100 qubits, problems with $V_b \simeq 140$ and a density of $\simeq 0.1$ were able to be embedded.

TABLE I
ENCODINGS: $f(d)$ IS THE ENCODING FUNCTION AND D IS THE BIT DEPTH

Encoding	$f(d)$	D
Binary	2^{d-1}	$\log_2(K'+1)$
Unary	1	K'
Sequential	d	$\left(\sqrt{1+8K'}-1\right)/2$

B. From Integer to Binary

To solve this problem using the D-Wave quantum annealer, the integer variables of Eq. 1 must be recast as binary variables, and the constraints must be incorporated into the objective function. We have investigated four different encodings: binary, unary, sequential, and partitioning. The first three can be described by writing the integer holdings as a linear function of the binary variables,

$$w_{\rm nt}[x] = \sum_{d=1}^{D} f(d) x_{\rm dnt}, \qquad (7)$$

where $x_{\rm dnt} \in \{0,1\}$ and the encoding function $f(d)$ and the bit depth D for each encoding are given in Table I.

The fourth encoding involves finding all partitions of K into N assets with K' or less units in each asset, and assigning a binary variable to each encoding at each time step.

We summarize the properties of each of the four encodings described above in Table II. Which encoding is preferred will depend on the problem being solved and the quantum annealer being used. Table III presents the number of variables required for some example multi-period portfolio optimization problems for each of these encodings.

For binary, unary, and sequential encodings, there is a trade-off between the efficiency of the encoding (the number of binary variables needed to represent a given problem) and the largest integer that can be represented. The reason is that for an encoding to be efficient, large coefficients will typically be introduced, limiting the largest integer representable (due to the noise level in the quantum annealer). For example, the binary encoding is

[1] as of June 2016.

TABLE II
COMPARISON OF THE FOUR ENCODINGS DESCRIBED IN SECTION II-B

Encoding	Variables	Largest integer	Notes
Binary	$TN\log_2(K'+1)$	$\lfloor 2n \rfloor - 1$	Most efficient in number of variables; allows representing of the second-lowest integer.
Unary	TNK'	No limit	Biases the quantum annealer due to differing redundancy of code words for each value; encoding coefficients are even, giving no dependence on noise, so it allows representing of the largest integer.
Sequential	$\frac{1}{2}TN\left(\sqrt{1+8K'}-1\right)$	$\frac{1}{2}\lfloor n \rfloor(\lfloor n \rfloor+1)$	Biases the quantum annealer (but less than unary encoding); second-most-efficient in number of variables; allows representing of the second-largest integer.
Partition	$\leq T\binom{K+N-1}{N-1}$	$\lfloor n \rfloor$	Can incorporate complicated constraints easily; least efficient in number of variables; only applicable for problems in which groups of variables are required to sum to a constant; allows representing the lowest integer.

The column "Variables" refers to the number of binary variables required to represent a particular problem. The column "Largest integer" refers to a worst-case estimate of the largest integer that could be represented based on the limitation imposed by the noise level ϵ and the ratio between the largest and smallest problem coefficients δ and $n \equiv 1/\sqrt{\epsilon\delta}$.

TABLE III
DEPENDENCE OF THE NUMBER OF BINARY VARIABLES REQUIRED ON THE NUMBER OF UNITS K, NUMBER OF ASSETS N, AND THE NUMBER OF TIME STEPS T FOR SOME EXAMPLE VALUES (HERE WE ASSUMED $K' = K/3$)

N	T	K	K'	V_b	V_u	V_s
5	5	15	5	75	125	75
10	10	15	5	300	500	300
10	15	15	5	450	750	450
20	10	15	5	600	1000	600
50	5	15	5	750	1250	750
20	15	15	5	900	1500	900
50	10	15	5	1500	2500	1500
50	15	15	5	2250	3750	2250

The number of variables is given for the three linear encodings: binary V_b, unary V_u, and sequential V_s.

the most efficient of the three (that is, it requires the fewest binary variables); however, it is the most sensitive to noise, and hence can represent the smallest integer of the three, given some number of qubits. Conversely, the unary encoding is the worst of the three in efficiency, but can represent the largest integer. An additional consideration is that some encodings introduce a redundancy that biases the quantum annealer towards certain solutions. Briefly, each integer can be encoded in multiple ways, the number of which is (in general) different for each integer. In this scenario, the quantum annealer is biased towards integers that have a high redundancy.

The partition encoding is different in that it requires an exponential number of variables; however, it allows a straightforward formulation of complicated constraints, like cardinality, by excluding partitions that break the constraints, which also lowers the number of variables required. For the other three encodings, constraints can be modelled through the encoding (for example, a minimum or maximum holdings constraint), or through linear or quadratic penalty functions. We note that the actual number of physical qubits required could be much larger than the number of variables indicated in Table III due to the embedding (see Section II-A).

In many cases, the maximum holdings K', which is also the largest integer to be represented, will not be exactly encodable using the binary and sequential encodings. For example, using a binary encoding one can encode the values 0 to 7 using three bits, and 0 to 15 using four bits, but integers with a maximum value between 7 and 15 are not exactly encodable. In order to avoid encoding infeasible holdings, these can be penalized by an appropriate penalty function. However, this penalty function will typically be a high-order polynomial and require many auxiliary binary variables in order to reduce it to a quadratic form. Instead, we propose to modify the encoding by adding bits with the specific values needed. For example, $\{1, 1, 2, 2, 4\}$ represents a modified binary encoding for the values 0 to 10.

The constraints of Eq. 2 can be incorporated into the objective function by rearranging the equations, squaring them, and summing the result, obtaining the penalty term

$$\text{penalty}[w] = -M\sum_{t=1}^{T}\left(K - \sum_{n=1}^{N} w_{\text{nt}}\right)^2, \tag{8}$$

where $M > 0$ is the strength of the penalty. In theory, M can be chosen large enough such that all feasible solutions have a higher value than all infeasible solutions. In practice, having an overly large M leads to problems due to the noise in the system (see Section II-D), so we choose it empirically by trial and error. In the future, it might be possible to include equality constraints of this form via a change in the quantum annealing process, allowing us to drop this term [37].

We note that an alternative approach, involving the tiling of the integer search space with binary hypercubes, was investigated by [38]; however, it requires an exponential number of calls to the quantum annealer.

C. Numerical Results

The results for a range of portfolio optimization problems are presented in Table IV. For each problem 200 random instances were generated. Each instance was solved using one query to the quantum annealer, with 1000 reads per query (which involved either 1 call or 5 calls if averaging over gauges).[2] All results were obtained from chips with a hardware graph with either 512 or 1152 qubits. (The number of active qubits was a little smaller.) For validation purposes, each instance was also solved using an exhaustive integer solver to find the optimal solution. For the

[2]We have observed that the success rate rises when increasing the number of reads, for a fixed problem size. If the number of reads is fixed, the success rate is expected to decrease as problem size increases, as the results show.

TABLE IV
RESULTS USING D-WAVE'S 512-QUBIT QUANTUM ANNEALER (WITH 200 INSTANCES PER PROBLEM)

N	T	K	Encoding	Vars	Density	Qubits	Chain	$S(0)$	$S(1)$	$S(2)$
2	3	3	Binary	12	0.52	31	3	100.00	100.00	100.00
2	2	3	Unary	12	0.73	45	4	97.00	99.50	100.00
2	4	3	Binary	16	0.40	52	4	96.00	100.00	100.00
2	3	3	Unary	18	0.53	76	5	94.50	99.50	100.00
2	2	7	Binary	12	0.73	38	4	90.50	100.00	100.00
2	5	3	Binary	20	0.33	63	4	89.00	100.00	100.00
2	6	3	Binary	24	0.28	74	4	50.00	97.50	99.50
3	2	3	Unary	18	0.65	91	6	38.50	72.50	91.50
3	3	3	Binary	18	0.45	84	5	35.50	66.50	82.50
3	4	3	Binary	24	0.35	106	6	9.50	50.50	84.50

N is the number of assets, T is the number of time steps, K is the number of units to be allocated at each time step and the maximum allowed holding (with $K' = K$), "Encoding" refers to the method of encoding the integer problem into binary variables, "Vars" is the number of binary variables required to encode the given problem, "Density" is the density of the quadratic couplers, "Qubits" is the number of physical qubits that were used, "Chain" is the maximum number of physical qubits identified with a single binary variable, and $S(\alpha)$ refers to the success rate given a perturbation magnitude $\alpha\%$ (explained in the text).

larger problems, a heuristic solver was run a large number of times in order to find the optimal solution with high confidence.

As a solution metric, we used the percentage of instances for each problem for which the quantum annealer's result fell within perturbation magnitude $\alpha\%$ of the optimal solution, denoted by $S(\alpha)$. This metric was evaluated by perturbing each instance at least 100 times, by adding Gaussian noise with standard deviation given by $\alpha\%$ of each eigenvalue of the problem matrix Q. Each perturbed problem was solved by an exhaustive solver, and the optimal solutions were collected. If the quantum annealer's result for that instance fell within the range of optimal values collected, then it was deemed successful (within a margin of error). This procedure was repeated for each random problem instance, giving a total success rate for that problem. For the case $\alpha = 0$, this reduces to defining success as the finding of the optimal solution.

The chosen approach relaxes the success metric in a problem-instance-specific way. An alternative would be to define success as finding a solution within ϵ of the optimum. However, this alternative success metric has the disadvantage that ϵ could be small or large compared to the energy scale of a particular problem instance, and so the metric can be misleading for problems of the type being solved here.

Although the variance of the success rate would also be interesting to observe, the number of experimental runs required for this metric to be statistically significant were not able to be performed due to a limited availability of machine time.

In order to investigate the quality of the solution on software enhancements, we replaced the "embedding solver" supplied with the D-Wave quantum annealer with a proprietary embedding solver developed by 1QBit, tuned the identification coupling strength, and combined results from calls with multiple random gauges. A gauge transformation is accomplished by assigning $+1$ or -1 to each qubit, and flipping the sign of the coefficients in the problem matrix accordingly, such that the optimization problem remains unchanged. We found a large improvement for all problems. Results with these improvements are presented in Table V.

TABLE V
RESULTS USING D-WAVE'S 512-QUBIT QUANTUM ANNEALER, WITH CUSTOM-TUNED PARAMETERS AND SOFTWARE (WITH 200 INSTANCES PER PROBLEM): AN IMPROVED EMBEDDING SOLVER, FINE-TUNED IDENTIFICATION COUPLING STRENGTHS, AND AVERAGING OVER 5 RANDOM GAUGES (200 READS PER GAUGE, GIVING A TOTAL OF 1000 READS PER CALL)

N	T	K	Encoding	Vars	Density	Qubits	Chain	$S(0)$	$S(1)$	$S(2)$
2	3	3	Binary	12	0.52	31	3	100.00	100.00	100.00
2	4	3	Binary	16	0.40	52	4	99.50	100.00	100.00
3	2	3	Unary	18	0.65	91	6	99.00	100.00	100.00
2	3	3	Unary	18	0.53	76	5	98.50	99.50	100.00
2	5	3	Binary	20	0.33	63	4	96.00	100.00	100.00
2	6	3	Binary	24	0.28	74	4	80.00	100.00	100.00
2	4	3	Unary	24	0.41	104	6	70.50	96.50	99.50
3	4	3	Binary	24	0.35	106	6	44.00	92.50	99.00
3	3	3	Binary	18	0.45	84	5	65.50	98.00	100.00
3	6	3	Binary	36	0.24	196	7	0.50	74.50	99.00
4	4	3	Binary	32	0.32	214	8	1.50	13.50	60.50
4	5	3	Binary	40	0.26	281	10	0.00	3.00	24.00

Columns are as in Table IV.

TABLE VI
RESULTS USING D-WAVE'S 1152-QUBIT QUANTUM ANNEALER (WITH 200 INSTANCES PER PROBLEM), WITH LOWER NOISE AND HIGHER YIELD, AND WITH CUSTOM-TUNED PARAMETERS AND SOFTWARE (AS IN TABLE V)

N	T	K	Encoding	Vars	Density	Qubits	Chain	$S(0)$	$S(1)$	$S(2)$
3	2	3	Unary	18	0.65	86	5	99.50	100.00	100.00
3	3	3	Binary	18	0.45	61	4	83.50	99.00	100.00
3	3	3	Unary	27	0.46	146	7	81.50	92.50	97.00
3	4	3	Binary	24	0.35	87	5	75.50	100.00	100.00
3	4	3	Unary	36	0.36	209	8	40.50	52.50	60.50
4	4	3	Binary	32	0.32	157	6	27.00	46.50	64.50
3	6	3	Binary	36	0.24	143	5	15.50	59.00	78.50
4	5	3	Binary	40	0.26	210	7	4.00	34.50	64.00
5	5	3	Binary	50	0.25	321	8	4.00	13.00	27.00
6	5	3	Binary	60	0.24	492	10	2.00	20.50	49.50
6	6	3	Binary	72	0.20	584	11	0.00	6.50	21.00

Columns are as in Table IV.

To investigate the dependence of the success rate on the noise level and qubit yield of the quantum annealer, several problems were solved on two different quantum annealing chips. Results for the second chip, the D-Wave 2X, which has 1152 qubits, a lower noise level and fewer inactive qubits and couplers, are presented in Table V. We found an increase in all success rates, and were able to solve larger problems.

These investigations highlight the importance of having an in-depth understanding of D-Wave's quantum annealer in order to be able to achieve the best results possible.

D. Discussion

The success rate and the ability to solve larger problems are affected by certain hardware parameters of the quantum annealer. First, there is a level of intrinsic noise which manifests as a misspecification error—the coefficients of the problem that the quantum annealer actually solves differ from the problem

coefficients by up to ϵ[3]. For future chip generations the expectation is that ϵ will decrease, and hence the success rate will increase by virtue of the quantum annealer solving a problem that is closer to the problem passed to it. In addition, the problem coefficients on the chip have a defined coefficient range, and if the specified problem has coefficients outside this range, the entire problem is scaled down. This can result in coefficients becoming smaller than ϵ, affecting the success rate. These factors are especially relevant for high-precision problems such as the multi-period portfolio optimization problem solved here.

The quantum annealer has a hardware graph that is currently very sparse and in general does not match the problem graph. In order to solve problems that are denser than the hardware graph, multiple physical qubits are identified with a single binary variable, at the cost of using more physical qubits. In order to force the identified qubits to all have the same value, a strong coupling is needed. If the required coupling is outside of the range of the couplings in the problem, the result will be an additional scaling, possibly reducing additional coefficients to less than ϵ, again impacting the success rate. Generally, the denser the hardware graph is, the fewer identifications are needed, and the weaker the couplings are required to be to identify the qubits.

To solve larger problems, the number of qubits must be greater. More qubits would also allow the use of an integer encoding scheme that is less sensitive to noise levels (such as unary encoding versus binary encoding). The fabrication process is not perfect, resulting in inactive qubits and couplers. The more inactive qubits and couplers there are on a chip, the lower the effective density and the higher the number of identifications required, which typically reduces the success rate.

In addition, custom tuning, through software, can be used to enhance the results. In particular, when the problem to be solved has a graph that differs from the hardware graph, a mapping, referred to as an "embedding," must be found from the problem graph to the hardware graph. The development of sophisticated ways to find better embeddings (for example, with fewer identified qubits) would be expected to increase the success rate—often the structure of the problem can be exploited in order to find better embeddings. In addition, when an embedding is used, there are different ways in which the couplings of the identified qubits should be set, controlled by the "embedding solver." For example, the strength of the couplings could be fine-tuned further to give higher success rates, or tuned separately for each set of identified qubits [39].

The issue of which embedding properties are most desirable is still under active research. Here the pi-elite metric was used to select the best scaling of the problem versus the strength of the qubit identification chains, and to choose the best embedding amongst the highest-ranked embeddings [39]. The pi-elite score is determined by comparing the mean energy of the best ("elite") states (for example, the lowest 2%) found by using each scale/embedding. The embeddings were ranked using a scheme that combines equal weights based on the shortest of the longest chains, the total number of qubits, and the variance of the lengths of the chains, after which the highest-ranked embeddings were compared using their pi-elite scores.

In addition, it has been observed that there is a gauge transformation under which the problem and solution remain invariant, but the success rates vary strongly (due to imperfections in the annealing chip). Combining results from multiple calls to the solver with random gauges, as we did, could provide a large improvement [39]. Software error correction, such as majority voting or energy minimization, which we employed when the identified physical qubits do not agree, as well as calibration, could also lead to improved solutions [40]–[46]. We also note that it may be possible to use the quantum annealer to find good local minima, which could then be used to speed up deterministic or heuristic classical solvers [47], [48].

Although the core contributions of this paper are the formulation of the general multi-period optimization strategy and discussion of the issues involved in solving that problem using available quantum annealing hardware, a brief comment regarding the time taken to calculate a solution is warranted. In general, benchmarking of the time to solution of a D-Wave quantum annealer against classical hardware is an area of ongoing and active research. For the small-scale problems solved in this study, the time to solution, on both classical hardware and using the quantum annealer, is comparable. The recent results by Denchev *et al.* [33], which show that for a specific class of problems the D-Wave machine has the potential to provide a speedup over computations on classical hardware, are encouraging. However, only after quantum speedup has been demonstrated for general problems, and specifically those requiring a high precision of couplings, is it expected that a quantum speedup for the optimal trading trajectory problem will be observed.

III. Conclusion

In this limited experiment we have demonstrated the potential of D-Wave's quantum annealer to achieve high success rates when solving an important and difficult multi-period portfolio optimization problem. We have also shown that it is possible to achieve a considerable improvement in success rates by fine-tuning the operation of the quantum annealer.

Although the current size of problems that can be solved is small, technological improvements in future generations of quantum annealers are expected to provide the ability to solve larger problems, and at higher success rates.

Since larger problems are expected to be intractable on classical computers, there is much interest in solving them efficiently using quantum hardware.

Appendix
Definition of Symbols

The symbols used above are defined in Table VII. In addition, we use w_t to denote the t-th column of the matrix w (and similarly for μ), and Σ_t to denote the covariance matrix ($N \times N$) which is the t-th page of the tensor Σ. For convenience of notation, the temporary transaction costs c are represented using the tensor Λ, where $\Lambda_{tnn'} = c_{\mathrm{nt}}\delta_{nn'}$ (and similarly for the

[3]The intrinsic noise for the current generation of chip is estimated to be around 2%–4% of the full scale.

TABLE VII
DEFINITION OF SYMBOLS

Symbol	Type	Description
K	$\mathbb{N}_1$	Number of units to be allocated at each time step
K'	$\mathbb{N}_1$	Largest allowed holding for any asset
N	$\mathbb{N}_1$	Number of assets
T	$\mathbb{N}_1$	Number of time steps
μ	$\mathbb{R}^{N\times T}$	Forecast mean returns of each asset at each time step
γ	$\mathbb{R}$	Risk aversion
Σ	$\mathbb{R}^{T\times N\times N}$	Forecast covariance matrix for each time step
c'	$\mathbb{R}^{N\times T}$	Permanent market impact coefficients for each asset at each time step
c	$\mathbb{R}^{N\times T}$	Transaction cost coefficients for each asset at each time step
w_0	$\mathbb{N}_0^N$	Initial holdings for each asset
w_{T+1}	$\mathbb{N}_0^N$	Final holdings for each asset
w	$\mathbb{N}_0^{N\times T}$	Holdings for each asset at each time step (the trading trajectory)

permanent price impact c' and Λ'). The difference in holdings between two time periods is defined as $\Delta w_t \equiv w_t - w_{t-1}$.

ACKNOWLEDGMENT

The authors would like to thank M. Bucyk for editing a draft of this paper, and R. Foerster, P. Ronagh, M. Rounds, and A. Zaribafiyan for their useful comments. C. Adolphs and D. Marchand provided invaluable help in obtaining the numerical results, and M. Vazifeh's code was used for the Chimera-graph figure.

REFERENCES

[1] H. Kellerer, R. Mansini, and M. G. Speranza, "Selecting portfolios with fixed costs and minimum transaction lots," *Ann. Oper. Res.*, vol. 99, nos. 1–4, pp. 287–304, 2000.

[2] R. Mansini and M. G. Speranza, "Heuristic algorithms for the portfolio selection problem with minimum transaction lots," *Eur. J. Oper. Res.*, vol. 114, no. 2, pp. 219–233, 1999.

[3] N. J. Jobst, M. D. Horniman, C. A. Lucas, G. Mitra, and G. Mitra, "Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints," *Quantitative Finance*, vol. 1, no. 5, pp. 489–501, 2001.

[4] J. P. Vielma, S. Ahmed, and G. L. Nemhauser, "A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs," *INFORMS J. Comput.*, vol. 20, no. 3, pp. 438–450, 2008.

[5] M. Corazza and D. Favaretto, "On the existence of solutions to the quadratic mixed-integer mean–variance portfolio selection problem," *Eur. J. Oper. Res.*, vol. 176, no. 3, pp. 1947–1960, 2007.

[6] F. Streichert, H. Ulmer, and A. Zell, "Evolutionary algorithms and the cardinality constrained portfolio optimization problem," in *Proc. Oper. Res*. 2004, pp. 253–260.

[7] P. Bonami and M. A. Lejeune, "An exact solution approach for portfolio optimization problems under stochastic and integer constraints," *Oper. Res.*, vol. 57, no. 3, pp. 650–670, 2009.

[8] N. Gârleanu and L. H. Pedersen, "Dynamic trading with predictable returns and transaction costs," *J. Finance*, vol. 68, no. 6, pp. 2309–2340, 2013.

[9] R. Almgren and N. Chriss, "Optimal execution of portfolio transactions," *J. Risk*, vol. 3, pp. 5–40, 2001.

[10] S. Galluccio, J.-P. Bouchaud, and M. Potters, "Rational decisions, random matrices and spin glasses," *Physica A, Statist. Mech. Appl.*, vol. 259, no. 3, pp. 449–456, 1998.

[11] M. López de Prado, "Generalized optimal trading trajectories: A financial quantum computing application," 2015. [Online]. Available: http://ssrn.com/abstract=2575184

[12] A. Finnila, M. Gomez, C. Sebenik, C. Stenson, and J. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," *Chem. Phys. Lett.*, vol. 219, no. 5, pp. 343–348, 1994.

[13] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, no. 5, 1998, Art. no. 5355.

[14] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem," *Science*, vol. 292, no. 5516, pp. 472–475, 2001.

[15] P. Ray, B. Chakrabarti, and A. Chakrabarti, "Sherrington-Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations," *Phys. Rev. B*, vol. 39, no. 16, p. 11828–11832, 1989.

[16] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, "Theory of quantum annealing of an Ising spin glass," *Science*, vol. 295, no. 5564, pp. 2427–2430, 2002.

[17] D. A. Battaglia, G. E. Santoro, and E. Tosatti, "Optimization by quantum annealing: Lessons from hard satisfiability problems," *Phys. Rev. E*, vol. 71, no. 6, 2005, Art. no. 066707.

[18] R. Harris *et al.*, "Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor," *Phys. Rev. B*, vol. 82, no. 2, 2010, Art. no. 024511.

[19] M. Johnson *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.

[20] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, "Experimental signature of programmable quantum annealing," *Nature Commun.*, vol. 4, 2013.

[21] "The D-Wave 2x quantum computer technology overview," D-Wave Systems Inc., Burnaby, BC, Canada, Tech. Rep., Aug. 2015.

[22] T. Lanting *et al.*, "Entanglement in a quantum annealing processor," *Phys. Rev. X*, vol. 4, no. 2, 2014, Art. no. 021041.

[23] S. Boixo *et al.*, "Computational multiqubit tunnelling in programmable quantum annealers," *Nature Comm.*, vol. 7, 2016.

[24] C. C. McGeoch and C. Wang, "Experimental evaluation of an adiabiatic quantum system for combinatorial optimization," in *Proc. ACM Int. Conf. Comput. Frontiers*, 2013, Art. no. 23.

[25] S. Boixo *et al.*, "Evidence for quantum annealing with more than one hundred qubits," *Nature Phys.*, vol. 10, no. 3, pp. 218–224, 2014.

[26] H. G. Katzgraber, F. Hamze, and R. S. Andrist, "Glassy Chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines," *Phys. Rev. X*, vol. 4, no. 2, 2014, Art. no. 021008.

[27] I. Hen, J. Job, T. Albash, T. F. Rønnow, M. Troyer, and D. A. Lidar, "Probing for quantum speedup in spin-glass problems with planted solutions," *Phys. Rev. A*, vol. 92, Oct. 2015, Art. no. 042325. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.92.042325

[28] A. D. King, "Performance of a quantum annealer on range-limited constraint satisfaction problems," arXiv:1502.02098, 2015.

[29] J. King, S. Yarkoni, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch, "Benchmarking a quantum annealing processor with the time-to-target metric," arXiv:1508.05087, 2015.

[30] T. F. Rønnow *et al.*, "Defining and detecting quantum speedup," *Science*, vol. 345, no. 6195, pp. 420–424, 2014. [Online]. Available: http://www.sciencemag.org/content/345/6195/420.abstract

[31] V. Martin-Mayor and I. Hen, "Unraveling quantum annealers using classical hardness," *Sci. Rep.*, vol. 5, 2015.

[32] E. Crosson and A. W. Harrow, "Simulated quantum annealing can be exponentially faster than classical simulated annealing," arXiv:1601.03030, 2016.

[33] V. S. Denchev *et al.*, "What is the computational value of finite range tunneling?" arXiv:1512.02206, 2015.

[34] P. I. Bunyk *et al.*, "Architectural considerations in the design of a superconducting quantum annealing processor," *IEEE Trans. Appl. Supercond.*, vol. 24, no. 4, pp. 1–10, Aug. 2014.

[35] V. Choi, "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem," *Quantum Inf. Process.*, vol. 7, no. 5, pp. 193–209, 2008.

[36] V. Choi, "Minor-embedding in adiabatic quantum computation: I. Minor-universal graph design," *Quantum Inf. Process.*, vol. 10, no. 3, pp. 343–353, 2011.

[37] I. Hen and F. M. Spedalieri, "Quantum annealing for constrained optimization," *Phys. Rev. Appl.*, vol. 5, Mar. 2016, Art. no. 034007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevApplied.5.034007

[38] P. Ronagh, "Integer optimization toolbox," 2013. [Online]. Available: http://1qbit.com/whitepaper/integer-optimization-toolbox/

[39] A. Perdomo-Ortiz, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, "A performance estimator for quantum annealers: Gauge selection and parameter setting," arXiv:1503.01083, 2015.

[40] A. Mishra, T. Albash, and D. Lidar, "Performance of two different quantum annealing correction codes," *Quantum Informat. Proc.*, vol. 15, no. 2, pp. 609–636, 2016.

[41] K. L. Pudenz, T. Albash, and D. A. Lidar, "Error-corrected quantum annealing with hundreds of qubits," *Nature Commun.*, vol. 5, 2014.
[42] K. L. Pudenz, T. Albash, and D. A. Lidar, "Quantum annealing correction for random Ising problems," *Phys. Rev. A*, vol. 91, no. 4, 2015, Art. no. 042302.
[43] W. Vinci, T. Albash, G. Paz-Silva, I. Hen, and D. A. Lidar, "Quantum annealing correction with minor embedding," *Phys. Rev. A*, vol. 92, Oct. 2015, Art. no. 042310. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.92.042310
[44] A. Perdomo-Ortiz, B. O'Gorman, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, "Determination and correction of persistent biases in quantum annealers," arXiv:1503.05679, 2015.
[45] F. Pastawski and J. Preskill, "Error correction for encoded quantum annealing," *Phys. Rev. A*, vol. 93, no. 5, May 2016, doi: 10.1103/PhysRevA.93.052325.
[46] W. Lechner, P. Hauke, and P. Zoller, "A quantum annealing architecture with all-to-all connectivity from local interactions," *Sci. Adv.*, vol. 1, no. 9, 2015, Art. no. e1500838.
[47] H. G. Katzgraber, F. Hamze, Z. Zhu, A. J. Ochoa, and H. Munoz-Bauza, "Seeking quantum speedup through spin glasses: The good, the bad, and the ugly*," *Phys. Rev. X*, vol. 5, Sep. 2015, Art. no. 031206. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevX.5.031026
[48] A. Zaribafiyan, "Integer programming via quantum binary programming and its application in finance," *In Preparation*, 2016.

Gili Rosenberg received the B.Sc. degree from Ben-Gurion University, Beersheba, Israel, and the Ph.D. degree from the University of British Columbia, both in physics, in 2004 and 2012. He is a Senior Researcher at 1QBit. His research focuses on the application of quantum annealing technology to solve problems in computational finance that are traditionally formulated using machine learning techniques such as neural networks, and heuristic algorithms such as simulated annealing.

.

Poya Haghnegahdar received the B.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, and the M.Sc. degree from the University of British Columbia, Vancouver, BC, Canada, both in physics, in 2005 and 2009, respectively. He is currently working toward the Ph.D. degree in physics at the University of British Columbia where his research focuses on the measurement-based model of quantum computing, Monte Carlo simulation techniques, and percolation theory.

He was formerly the Applications and QA Lead at 1QBit and has practical experience in machine learning and data analysis techniques.

Phil Goddard received the B.Eng. degree from the University of Western Australia, Perth, W.A., USA, and the Ph.D. degree in engineering from Cambridge University, Cambridge, U.K., in 1991 and 1995, respectively. He is the Head of Research at 1QBit. His research focuses on the application of heuristic and machine learning algorithms to topics found in computational finance and traditional engineering. Working with clients and partners spanning a range of business sectors, he leads teams focused on applying quantum annealing technology to a variety of traditional optimization problems.

With 25 years of experience in designing and developing custom software applications for modelling, simulation, data analysis, and visualization, he is also the President and Principal Consultant at Goddard Consulting. He was formerly a Senior Consultant at The MathWorks, where he worked with clients in the financial services, automotive, aerospace, telecommunications, and process industries. He has also worked for British Aerospace (Dynamics) and Woodside Offshore Petroleum. Since 2006, he has been working as a Visiting Lecturer within the Beedie School of Business, Simon Fraser University, where he lectures on topics in numerical analysis and software development for computational finance.

Peter Carr received the Ph.D. degree from UCLA. He currently serves as the Executive Director of the Math Finance program at NYU's Courant Institute. Formerly a Managing Director and Global Head of Market Modeling at Morgan Stanley, he has approximately 20 years of experience in the financial services industry. He is also a Trustee for the Museum of Mathematics in New York.

Prior to joining the financial industry, he was a Finance Professor for 8 years at Cornell University. He has more than 75 publications in academic and industry-oriented journals and serves as an Associate Editor for 8 journals related to mathematical finance. He was selected as Quant of the Year by Risk Magazine in 2003 and Financial Engineer of the Year by IAFE and Sungard in 2010. For the last 4 years, he was included in Institutional Investor's Tech 50, an annual listing of the 50 most influential people in financial technology.

Kesheng Wu received the B.Sc. degree in physics from Nanjing University, the M.Sc. degree in physics from the University of Wisconsin-Milwaukee, and the Ph.D. degree in computer science from the University of Minnesota. He is the Group Leader of the Scientific Data Management Group at Lawrence Berkeley National Laboratory, U.S. Department of Energy's Office of Science. His current research focuses on indexing technology for searching large data sets, including improving bitmap index technology with compression, encoding, and binning. He is the key developer of FastBit bitmap indexing software, which is used in a number of applications including high-energy physics, combustion, network security, and query-driven visualization. He has also worked on various scientific computing projects including developing the Thick-Restart Lanczos algorithm for solving eigenvalue problems and devising statistical tests for deterministic effects in broad band time series.

Marcos López de Prado received the Ph.D. degrees in financial economics and mathematical finance from Complutense University, Madrid, Spain, in 2003 and 2011, respectively. He is a Senior Managing Director at Guggenheim Partners.

He is also a Research Fellow at Lawrence Berkeley National Laboratory's Computational Research Division (U.S. Department of Energy's Office of Science), where he conducts unclassified research in the mathematics of large-scale financial problems and supercomputing. Previously, he was the Head of Quantitative Trading and Research at Hess Energy Trading Company (the trading arm of Hess Corporation, a Fortune 100 company) and the Head of Global Quantitative Research at Tudor Investment Corporation. In addition to his 17 years of trading and investment management experience at some of the largest corporations, he has received several academic appointments, including Postdoctoral Research Fellow of RCC at Harvard University and a Visiting Scholar at Cornell University.

He received the National Award for Excellence in Academic Performance by the Government of Spain (National Valedictorian, 1998) among other awards, and was admitted into American Mensa with a perfect test score. He serves on the Editorial Board of the *Journal of Portfolio Management* (IIJ), the *Journal of Investment Strategies* (Risk) and the *Big Data and Innovative Financial Technologies Research Series* (SSRN). He has collaborated with many leading academics, resulting in some of the most read papers in finance (SSRN), four international patent applications on high-frequency trading, three textbooks, and numerous publications in top mathematical finance journals.