Quantum-Enhanced Control of a Tandem Queue System

George T. Stamatiou[1(✉)] and Kostas Magoutis[1,2]

[1] Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH), Heraklion, Greece
stamatiou@ics.forth.gr, magoutis@csd.uoc.gr

[2] Computer Science Department, University of Crete, Heraklion, Greece

Abstract. Controlling computer systems in an optimal way using quantum devices is an important step towards next generation infrastructures that will be able to harness the advantages of quantum computing. While the implications are promising, there is a need for evaluating new such approaches and tools in comparison with prevalent classical alternatives. In this work we contribute in this direction by studying the stabilization and control of a tandem queue system, an exemplary model of a computer system, using model predictive control and quantum annealing. The control inputs are obtained from the minimization of an appropriately constructed cost function and the optimal control problem is converted into a quadratic unconstrained binary optimization problem to be solved by the quantum annealer. We find that as the prediction horizon increases and the core optimization problem becomes complicated, the quantum-enhanced solution is preferable over classical simulated annealing. Moreover, there is a trade-off one should consider in terms of variations in the obtained results, quantum computation times and end-to-end communication times. This work shows a way for further experimentation and exploration of new directions and challenges and underscores the experience gained through utilization of the state-of-the-art quantum devices.

Keywords: Model predictive control · Quantum annealing · Tandem queue system

1 Introduction

Adaptive computing systems, such as robots that plan their motion [30] and data-center management systems that enforce end-to-end quality-of-service [8,15,22], operate in a real-time feedback loop setting with appropriate control inputs to achieve their goals optimally, requiring the repeated solution of an optimization problem within a specific time interval to take appropriate action in time. A prevalent method for optimal control is Model Predictive Control

E. Kalyvianaki and M. Paolieri (Eds.): VALUETOOLS 2023, LNICST 539, pp. 99–114, 2024.
https://doi.org/10.1007/978-3-031-48885-6_7

(MPC)[5,25,27,30], an advanced control strategy widely used in engineering and industrial processes. MPC involves creating a mathematical model that describes the system's behavior and formulating an optimization problem to determine optimal control actions. The aim is to minimize an objective function that captures both tracking performance (deviation from desired reference points) and control effort (penalizing excessive control actions or changes). MPC is also designed to handle constraints effectively. Constraints can represent physical limitations or operational requirements of the system. These constraints can be imposed on inputs, outputs, or states and can be hard or soft. Hard constraints must be strictly satisfied, while soft constraints can be violated within certain bounds. By incorporating constraints into the optimization problem, MPC ensures that the control actions respect these limitations, ensuring feasibility and safety in real-time control. A main appeal of MPC is that rather than optimizing the whole process it operates over a finite time horizon, usually called prediction horizon N_s, which specifies how far into the future the system behavior is predicted. By applying only the first control action of the optimal sequence, MPC allows for adjustments at each time step, creating a receding horizon effect and the process is re-initialized for the next time window. This enables MPC to react to changes in the system and adapt the control actions based on updated predictions and measurements. A key challenge with MPC feedback loops is that the computational effort involved in the MPC optimization problem increases dramatically as the control period becomes shorter and/or the optimization problem increases in complexity by considering longer task horizon for example. Consequently, adaptive behavior for large distributed systems aiming for short sample times, in the order of tens of milliseconds, currently resort to using overly simple models, short prediction horizons, and often an explicit solution of the optimization problem by solving the problem in an offline fashion, in advance. Although approximations to the MPC optimization problem are commonly used, they are still inadequate for large configuration search spaces.

Quantum computers available today, known as Noisy - Intermediate - Scale Quantum (NISQ) devices [24], though not a commodity yet, are in the direction of solving real-world problems. A quantum computer is a device that exploits properties of quantum physics, such as superposition and entanglement, to perform computations. The main building block of a quantum device is the qubit, which is the quantum analog of a classical bit. While a classical bit can be in either state 0 or 1, a qubit can exist in a superposition of both states, allowing it to represent and process multiple possibilities simultaneously. The main approaches towards utility quantum computing currently include but not limited to *gate-based* models and *quantum annealers*. Gate-based quantum computing is based on the concept of a quantum circuit, a sequence of quantum gates which manipulate and transform the qubits. The combinations of gates perform operations on the qubits, such as rotations, entanglement, and measurements, allowing for complex computations and the development of quantum algorithms. Quantum annealers, used in our work, are based on the concept of

quantum annealing, a computational optimization technique that utilizes principles of quantum mechanics to solve optimization problems where one needs to search over a large space of possible solutions and find an optimal one [14,28,35]. Thus, it is a specialized application of quantum computing that focuses on finding the global minimum (or maximum) of a given objective function. Classical simulated annealing [17,31] is inspired by the annealing process in metallurgy, where a material is heated and then slowly cooled to obtain a desired structure with minimal defects. Similarly, in optimization the process involves starting with a high-energy state and gradually cooling the system to reach a low-energy state corresponding to the optimal solution. In quantum annealing this concept is evolved by leveraging quantum mechanical effects, such as quantum tunneling, to explore the solution space in a more efficient way than classical optimization algorithms. Examples of quantum annealers include the D-Wave 2000Q and its next-generation successor, the Advantage System. To work with a quantum annealer, the problem to be optimized is mapped onto the configuration of qubits. The objective function of an optimization problem, which is typically expressed as an Ising model, defines the problem's energy landscape [20]. The Ising model is a mathematical model used to describe the interactions between spins in a physical system, where spins can take values of either $+1$ or -1. The energy landscape of the problem is encoded in the system's Hamiltonian, which is a mathematical operator representing the total energy of the system. The Hamiltonian consists of two components: initial Hamiltonian (H_0) and the problem Hamiltonian (H_p),

$$\begin{aligned} H &= \Big(1 - \frac{t}{T}\Big) H_0 + \frac{t}{T} H_p \\ &= \Big(1 - \frac{t}{T}\Big) \sum_i \hat{\sigma}_i^x + \frac{t}{T} \Big(\sum_i h_i \hat{\sigma}_i^z + \sum_{i>j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z \Big). \end{aligned} \tag{1}$$

where $\hat{\sigma}_i$ are the Pauli matrices operating on qubit i, h_i is the on-site energy (local field) of qubit i and J_{ij} represents the pairwise interaction strength between qubits i and j determining how strongly the states of two qubits influence each other in the objective function being optimized. Quantum annealing relies on the adiabatic theorem of quantum mechanics which states that if we start in the ground state (the state with the lowest energy) of the initial Hamiltonian H_0, by slowly modifying time t transforming H_0 into H_p, the system remains in its ground state, providing us with the solution for our problem by measuring the quantum state at a future time $t = T$. To solve an optimization problem using quantum annealing, we first need to formulate the objective function as a quadratic unconstrained binary optimization (QUBO) problem equivalent to (1) where the optimization variables are translated into binary variables resulting to the expression

$$f(b_1, \ldots, b_n) = \sum_i C_i b_i + \sum_{i,j} J_{ij} b_i b_j. \tag{2}$$

where $b_i \in \{0, 1\}$ and C_i, J_{ij} are parameters. In this form the problem can be mapped onto the graph structure of the quantum processor chip (QPU).

The evolution of quantum devices gives rise to new possibilities for solving hard computational problems with higher accuracy and speed compared to existing classical solutions. In this work, we investigate the *quantum-enhanced* solution of an MPC problem using quantum annealers designed and built by D-Wave Systems. We choose the *tandem queue system* as a simple performance model of a computer system network. The choice is also made as to provide insight into expressing a complex MPC problem and its respective cost function into forms amenable for quantum annealing. Accelerating optimization tasks at the core of MPC feedback loops via quantum annealing opens up new avenues for adaptive computing systems to become more intelligent, by virtue of being able to solve difficult optimization problems as their environment and external inputs change rapidly.

The remainder of this paper proceeds as follows. In Sect. 2 we review previous work related to the current research. In Sect. 3 we present the model of the tandem queue system expressed as a control problem in a state-space representation, the construction of the cost function and a way to transform the optimal control problem to a QUBO expression which is used as input for the quantum annealer. Section 4 presents our analysis and evaluation results for the D-Wave 2000Q and Advantage System. Finally, in Sect. 5 we discuss implications of our work, future directions and conclude the paper.

2 Related Work

Performance modeling and analysis of computer systems based on queueing models has been thoroughly studied through the years [1,3,18]. A tandem queue system is used to model systems in which customers must pass through a connected series of service stations (or queues) in order to complete a transaction or process. In a tandem queue system, the output of one queue becomes the input to the next queue, forming a sequence or chain of queues. Tandem queue systems have been considered appropriate performance models for many applications ranging from call centers, healthcare systems, manufacturing processes to packet-switched computer networks, stream-processing systems, and multi-tier computer systems in general [1,4,18,19]. Previous research efforts have studied performance properties of tandem queues, such as throughput and total queueing delay, under various assumptions for the distributions of arrivals and service times at all servers, fixed buffers (blocking) vs. infinite buffers at each queue, number of servers at each stage, etc. Another area of interest around tandem queues and multi-stage pipelines concerns their controllability, namely the ability to affect system outputs such as end-to-end response time via control inputs such as buffer size. Optimal control of computer systems, a key capability to achieving adaptive, self-managing systems has also received attention in recent years [6,11,12]. Linear quadratic regulators (LQR) are a classical approach to optimal control embodying an optimization problem with an appropriately crafted cost function [2,12,16]. Mathematical aspects concerning the optimal control of finite queues and tandem queues can be found in older works

[21,26,32]. There have also been recent works that investigate MPC in queueing networks [10,29,34], and others that explore the benefits of QDs comparing quantum annealing over simulated annealing for network optimization tasks [33]. A recent work [13] proposes an MPC algorithm for using the D-Wave 2000Q quantum annealer tested on mechanical-oriented system models for a fixed prediction horizon. Another, promising direction combining MPC and quantum annealing is the efficient control of HVAC systems in buildings [9]. However, there is no prior work on investigating the advantages of combining MPC and the utilization of quantum devices for the control of queueing-network models, experimenting with different prediction horizons, or with more recent quantum-annealing devices such as the D-Wave Advantage. In this paper we design, implement, and experimentally evaluate quantum-enhanced MPC control of a tandem queue system, improving over previous research in this space by means of tackling a system model with significant practical relevance, and carry out a broad experimental evaluation (various prediction horizons and number of samples per quantum device invocation) exhibiting key trade-offs in state-of-the-art quantum devices.

The main contributions in this paper are:

- Implementation of a MPC approach to stabilize and control an unstable tandem queue system by minimizing a designed cost function using both quantum-enhanced (quantum annealing) and classical (exact, simulated annealing) methodologies.
- Evaluation of quantum-enhanced MPC solution on state-of-the-art D-Wave Advantage System and 2000Q QDs investigating different predictions horizons and number of states (output solutions) to read from the QD solver.
- Highlighting the efficiency of quantum-enhanced MPC vs. classical simulation approximations for longer MPC horizons.

3 Methodology

3.1 Mathematical Model

The tandem queue system, as shown in Fig. 1, consists of two M/M/1/K queues Q_1, Q_2 where the inter-arrival times of the incoming requests are exponentially distributed with mean $\frac{1}{\lambda_i}, (i = 1, 2)$, the service times are exponentially distributed with mean $\frac{1}{\mu_i}, (i = 1, 2)$. There is 1 server in each queue and $K_i, (i = 1, 2)$ is the finite buffer size. The incoming requests arrive at Q_1 and the service begins immediately. In the case that the server is not available, the requests wait in the buffer of Q_1 until the server is available again. If the incoming request finds that the buffer is full, then the request is discarded. The departures from the Q_1 become arrivals at Q_2 which handles the requests on the same manner. Finally, the departures from the queueing system Q_2 are outgoing requests. For the matter of simplicity, we assume that the buffer size of Q_2 is sufficiently large ($K_2 >> K_1 \equiv K$) so that arrivals in Q_2 are never discarded. We focus on the average response times for the requests entering-leaving Q_1 and Q_2 denoted

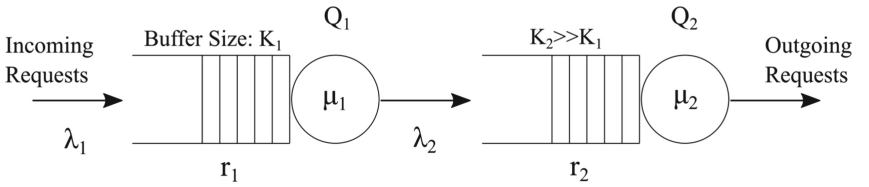

Fig. 1. Architecture diagram of a tandem queue system.

by R_1 and R_2, respectively. The main objective is to control the end-to-end response time defined as $R = R_1 + R_2$ by optimally tuning the buffer size K of the first queue Q_1. The control input of the system is the buffer size K for system Q_1 and the measured system output is the total response time R.

The dynamics of each one of the queueing systems Q_1, Q_2 can be represented using a first-order difference equation model

$$\begin{aligned} r_1(k+1) &= a_{11} r_1(k) + b u(k) \\ r_2(k+1) &= a_{21} r_1(k) + a_{22} r_2(k) \end{aligned} \tag{3}$$

where $a_{11}, a_{21}, a_{22}, b$ are parameters to be determined through system identification and $k \in \mathbb{N}$ denotes the discrete time. For the dynamics we use the lowercase $r_1(k), r_2(k), u(k)$ denoting the offset values for the response times and the buffer size defined as

$$\mathbf{r}(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} = \begin{bmatrix} R_1(k) - \langle R_1 \rangle \\ R_2(k) - \langle R_2 \rangle \end{bmatrix}, \quad u(k) = K(k) - \langle K \rangle \tag{4}$$

where $\langle R_1 \rangle, \langle R_2 \rangle, \langle K \rangle$ are the respective mean values we define as the operating points -or the reference values- of the system. The state-space dynamics of the tandem queue system as a linear time-invariant (LTI) system is

$$\begin{aligned} \mathbf{r}(k+1) &= A\mathbf{r}(k) + B\mathbf{u}(k) \\ \mathbf{R}(k) &= C\mathbf{r}(k) \end{aligned} \tag{5}$$

where $\mathbf{r}(k) \in \mathbb{R}^n$ is the vector of state variables (the response times for each queue), $\mathbf{u}(k) \in \mathbb{R}^m$ is the vector of inputs (the buffer size), $\mathbf{R}(k) \in \mathbb{R}^l$ is the vector of outputs (the end-to-end response time). $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$ are constant matrices that encompass the system dynamics.

3.2 Model Predictive Control

The state and output predictions can be derived from (5) in a recursive manner up to the prediction horizon $N_s \in \mathbb{N}$ of future steps [27]. Thus, we obtain

$$\begin{bmatrix} R(k) \\ R(k+1) \\ R(k+2) \\ R(k+3) \\ \vdots \\ R(k+N_s-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_s-1} \end{bmatrix} \mathbf{r}(k) + \begin{bmatrix} 0 & 0 & 0 & \dots \\ CB & 0 & 0 & \dots \\ CAB & CB & 0 & \dots \\ CA^2B & CAB & CB & \dots \\ \vdots & \vdots & \vdots & \vdots \\ CA^{N_s-2}B & CA^{N_s-3}B & CA^{N_s-4}B & \dots \end{bmatrix} \mathbf{u}(k),$$

and (5) becomes in vectorized form

$$\mathbf{R}(k) = \mathbb{A}\mathbf{r}(k) + \mathbb{B}\mathbf{u}(k) \tag{6}$$

where $\mathbf{R}(k) = [R(k), R(k+1), \dots, R(k+N_s-1)]^T \in \mathbb{R}^{lN_s}$ are the system outputs, $\mathbf{u}(k) = [u(k), u(k+1), \dots, u(k+N_s-1)]^T \in \mathbb{R}^{mN_s}$ are the control inputs and $\mathbb{A} \in \mathbb{R}^{lN_s \times n}$, $\mathbb{B} \in \mathbb{R}^{lN_s \times mN_s}$. In order to implement the MPC optimization strategy to control the tandem queue system we first introduce a cost function (or objective function) that we aim to minimize

$$\mathcal{J} = \mathbf{R}(k)^T \mathcal{Q}\mathbf{R}(k) + \mathbf{u}(k)^T \mathcal{R}\mathbf{u}(k), \tag{7}$$

where $\mathcal{Q} \in \mathbb{R}^{lN_s \times lN_s}$, $\mathcal{R} \in \mathbb{R}^{mN_s \times mN_s}$ are symmetric diagonal positive definite weighting matrices which tune the trade-off between tracking performance and control effort. The optimal control problem aims to to find the appropriate control input sequence $u^*(k)$ which minimizes the cost function J. By substituting (6) to (7) we obtain after some matrix algebraic operations that

$$\mathcal{J} = \mathbf{u}(k)^T \Big[\mathbb{B}\mathcal{Q}\mathbb{B} + \mathcal{R}\Big]\mathbf{u}(k) + 2\mathbf{r}(k)^T \mathbb{A}^T \mathcal{Q}\mathbb{B}\mathbf{u}(k) + \mathbf{r}(k)^T \mathbb{A}^T \mathcal{Q}\mathbb{A}\mathbf{r}(k), \tag{8}$$

which is the cost function expressed in terms of states and control inputs.

3.3 QUBO Form of the Cost Function

In order to utilize quantum annealing (8) should be translated to a QUBO form. Given a finite ordered set $\mathcal{U}_m = \{u_1, u_2, u_3, \dots, u_M\}$ of fixed and equally spaced elements representing the control inputs that our system can receive, we want to express every element in terms of binary variables. It is convenient that the length of $\mathcal{U}_m$ is $M = 2^L$, where $L \in \mathbb{N}$ is the number of binary variables[13]. First, we normalize the elements according to the rule: $u_i \longrightarrow u_i - u_{\frac{M}{2}+1}$. In general, for m-inputs in our system one may write each element u_i as a function of L binary variables in vector form as

$$\mathbf{u_i} = \begin{bmatrix} p_1(-2^{L-1}b_{1,L} + \sum_{j=1}^{L-1} 2^{j-1}b_{1,j}) \\ \vdots \\ p_m(-2^{L-1}b_{m,L} + \sum_{j=1}^{L-1} 2^{j-1}b_{m,j}) \end{bmatrix}, \tag{9}$$

where $b_{i,j} \in \{0,1\} (i = 1, \ldots, m$ and $j = 1, \ldots, L)$ is a binary variable and $p_i \in \mathbb{R}$ is a scaling parameter. We provide an example to better illustrate the procedure. Let a system with $m = 2$ control inputs and given values $\mathcal{U}_1 = \{-2, 1, 4, 7\}$ and $\mathcal{U}_2 = \{150, 200, 250, 300\}$, where $L_1 = L_2 = L = 2$ so that $M_1 = M_2 = M = 2^L = 4$. The normalization of $\mathcal{U}_1, \mathcal{U}_2$ according to the aforementioned rule, results in $\mathcal{U}_1 = \{-6, -3, 0, 3\}$ and $\mathcal{U}_2 = \{-100, -50, 0, 50\}$. From (9) we obtain

$$\mathbf{u_i} = \begin{bmatrix} p_1(-2b_{1,2} + b_{1,1}) \\ p_2(-2b_{2,2} + b_{2,1}) \end{bmatrix} = \begin{bmatrix} 3b_{1,1} - 6b_{1,2} \\ 50b_{2,1} - 100b_{2,2} \end{bmatrix} \longrightarrow \begin{bmatrix} \{-6, -3, 0, 3\} \\ \{-100, -50, 0, 50\} \end{bmatrix}$$

where we set $p_1 = 3$, $p_2 = 50$ and take all the combinations of zeros and ones. Now, we can write (9) in matrix form as

$$\mathbf{u_i} = \begin{bmatrix} W_1 & \ldots & 0 \\ & W_2 & \vdots \\ \vdots & \ddots & \\ 0 & \ldots & W_m \end{bmatrix} \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,L} \\ \vdots \\ b_{m,1} \\ \vdots \\ b_{m,L} \end{bmatrix} = Wb \tag{10}$$

where $W \in \mathbb{R}^{m \times mL}$, $W_i \in \mathbb{R}^{1 \times L}$ and $b \in \{0,1\}^{mL}$. In terms of the previous example we have

$$\mathbf{u_i} = \begin{bmatrix} 3 & -6 & 0 & 0 \\ 0 & 0 & 50 & -100 \end{bmatrix} \begin{bmatrix} b_{1,1} \\ b_{1,2} \\ b_{2,1} \\ b_{2,2} \end{bmatrix}$$

Now, we proceed by incorporating the predictions for the control inputs for prediction horizon N_s as

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_s-1) \end{bmatrix} = \begin{bmatrix} W & & \\ & W & \\ & & \ddots \\ & & W \end{bmatrix} \begin{bmatrix} b(k) \\ b(k+1) \\ \vdots \\ b(k+N_s-1) \end{bmatrix} \Leftrightarrow \mathbf{u}(k) = \mathbb{W}\mathbf{b}(k), \tag{11}$$

where $\mathbb{W} \in \mathbb{R}^{mN_s \times mLN_s}$ and $\mathbf{b}(k) \in \{0,1\}^{mLN_s}$. The substitution of (11) to (8) reads

$$\mathcal{J} = \mathbf{b}(k)^T \left[\mathbb{W}^T \mathbb{B}^T \mathcal{Q} \mathbb{B} \mathbb{W} + \mathbb{W}^T \mathcal{R} \mathbb{W} \right] \mathbf{b}(k) + 2\mathbf{r}(k)^T \mathbb{A}^T \mathcal{Q} \mathbb{B} \mathbb{W} \mathbf{b}(k) + \mathbf{r}(k)^T \mathbb{A}^T \mathcal{Q} \mathbb{A} \mathbf{r}(k), \tag{12}$$

which is the cost function expressed in terms of states and binary control inputs. Equation (12) is minimized by the quantum annealer with respect to vector $\mathbf{b}(k)$ of binary variables. Notice that the last relation encompasses the dynamics of the initial system we aim to control along with the future predictions required for the MPC strategy.

4 Evaluation and Results

The tandem queue system simulation involves the utilization of *ciw* [23], a discrete event simulation Python library for queueing networks. The arrival rate for Q_1 is set to $\lambda_1 = 3.8$ reqs/sec and the service rate for both Q_1 and Q_2 is $\mu_1 = \mu_2 = 4.0$ reqs/sec. The steady-state response time for an M/M/1/K queue using Little's law ([3,12]) is given by

$$R_1 = \frac{\mathcal{N}_1}{\lambda_1} = \frac{\rho_1}{\lambda_1(1-\rho_1)} \cdot \frac{1-(K+1)\rho_1^K + K\rho_1^{K+1}}{1-\rho_1^{K+1}}, \tag{13}$$

where $\rho_1 = \lambda_1/\mu_1$ is the utilization and $\mathcal{N}_1$ is the expected number in the system Q_1. Our work is focused on small buffer sizes $K \in [0, 15]$ where the relation between response time and buffer size is closer to linear [12]. In this region the average response times of the system are calculated performing 50 different experiments for each K. The main objective is to examine how the change in the buffer size K affects the total response time R of the tandem queue system. In other words what are the optimal values for the control input K in order for the R to reach a desired value.

The dynamics of the tandem queue system is given by (3) and because this is a simple model, we choose to study a rather non-trivial case in which the system is unstable, in order to examine whether the MPC along with quantum annealing will manage to stabilize and control it showing how this procedure works in this case. From the state-space form of (5), our system has $n = 2$ state variables (average response times of the two queues), $m = 1$ control input (buffer size K) and $l = 1$ output (end-to-end average response time) and the constant matrices for the considered model parameters are

$$A = \begin{bmatrix} 1.00076975 & 0 \\ 0.78485803 & 0.33908979 \end{bmatrix}, \quad B = \begin{bmatrix} -0.00283586 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{14}$$

The system is unstable as the eigenvalues of matrix A are outside the unit circle. From the simulation and data analysis the obtained mean values are $\langle R_1 \rangle = 1.25$, $\langle R_2 \rangle = 1.83$ and $\langle K \rangle = 8$ which we assume to be the reference values of the system (see Sect. 3.1). In order for the system to operate near the reference points we aim for the offset values $r_i(k) = R_i(k) - \langle R_i \rangle, i = 1, 2$ and $u(k) = K(k) - \langle K \rangle$ to be near 0. The set of available control inputs is $\mathcal{U} = \{0, 1, 2, \ldots, 14, 15\}$. Based on methodology in Sect. 3.3 the set becomes $\mathcal{U} = \{-8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7\}$. The set $\mathcal{U}$ consists of values with constant intervals, its length is $M = 2^4 = 16$ and each element is expressed using $L = 4$ binary variables. In our case, the control input is $m = 1$ so (9) reads

$$u_i = \begin{bmatrix} b_{1,1} + 2b_{1,2} + 4b_{1,3} - 8b_{1,4} \end{bmatrix}.$$

where we set $p_1 = 1$. From (10) we obtain $W = W_1 = \begin{bmatrix} 1 & 2 & 4 & -8 \end{bmatrix}$ and $b = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \end{bmatrix}^T$. The last step completes the necessary ingredients needed for the cost function in (12) to be implemented to the quantum annealer.

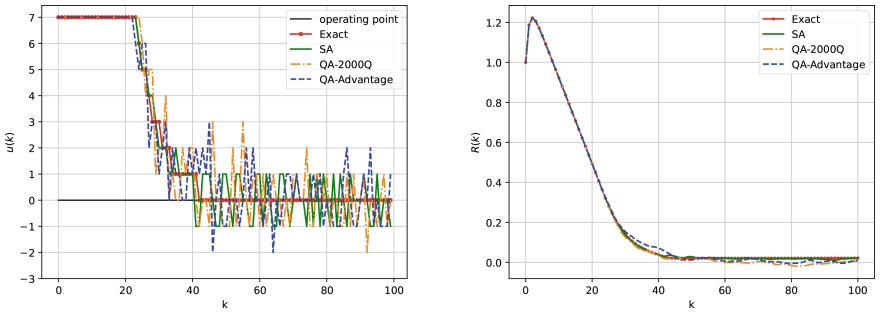

Fig. 2. (Left) Control inputs u as a function of time steps k and (Right) System output $R(k)$ using for prediction horizon $N_s = 6$ and initial conditions $(r_1(0), r_2(0)) = (0.6, 0.4)$. The *num-reads* parameter for all methods is set to default.

To perform the minimization of J we employ and compare different approaches. First, the exact solution which searches all the available space of possible solutions, second, the classical optimization method of simulated annealing and third, quantum annealing. The quantum annealers that are used in the present study are the D-Wave Advantage System as well as the previous generation D-Wave 2000Q. The Advantage_System5.3 we used is located in Germany whereas the D-Wave 2000Q machine is physically located in British Columbia, Canada. Our analysis is mostly focused on the Advantage System, the closest D-Wave infrastructure to our environment resulting in the lowest HTTP communication times between our MPC loop and the quantum device. To communicate with and execute our programs at the available devices we utilize the LEAP cloud-based service along with Ocean software [7] supporting Python provided by D-Wave. More specifically, for the exact solution we used the exact solver from *dimod* API for samplers, for the simulated annealing the *neal* library and for the quantum annealing the *dwave-system* tool. An important tuning parameter is the *num-reads* which indicates the number of states (output solutions) to read from the solver before the selection of the one with the minimum energy. The value of *num-reads* improves the solution profile at the cost of an increase in the QPU access times.

The experiment runs for 100 time steps and each time step is denoted by k. Initially, for quantum and simulated annealing all parameters (including *num-reads*) are set set to default values and the prediction horizon is set to $N_s = 6$. For all experiments, the diagonal elements of the weighting matrices $\mathcal{Q}$, $\mathcal{R}$ (see (7)) are set to $Q = 10^7$ and $R = 1$, respectively. Increasing the order of magnitude of values between the weight matrices may benefit performance and in that case the stabilization and control may be achievable even earlier. The specific values used in this evaluation were empirically chosen to yield satisfying convergence. To demonstrate our simulation results we choose the initial conditions $r_1(0) = 0.6$ and $r_2(0) = 0.4$ denoting a positive offset of the response times from the system's operating point. In Fig. 2 (left), we observe that the $u(k)$ values are discrete and

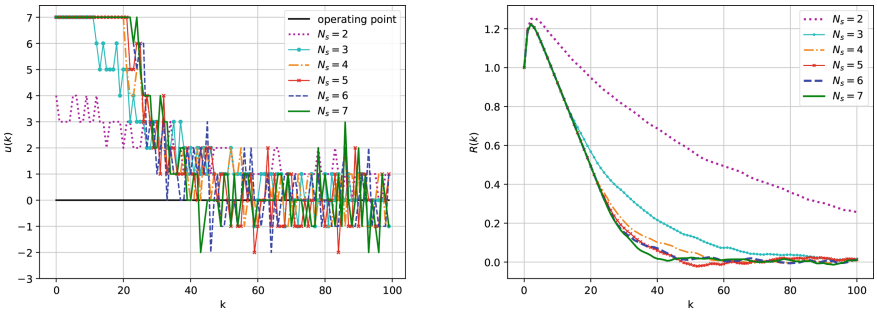

Fig. 3. Scaling of the solution with the MPC prediction horizons $N_s = 2, 3, 4, 5, 6, 7$ using the D-Wave Advantage System. (Left) Control inputs u as a function of time step k and (Right) System output $R(k)$. The initial conditions are $r_1(0) = 0.6$ and $r_2(0) = 0.4$ and the *num-reads* parameter is set to default.

Table 1. Elapsed times for MPC optimization *per time step* (*in ms*)

Methods	$N_s = 3$	$N_s = 4$	$N_s = 5$	$N_s = 6$	$N_s = 7$
Exact	6.88	77.80	1390.42	25210.21	timed out
SA	10.28	12.44	15.61	19.82	24.51
QA *sampler-call*	62.33	129.08	288.41	610.55	1061.57
QA *qpu-access*	15.94	15.96	15.98	16.02	16.36

equally spaced with respect to the previously defined control input set $\mathcal{U}$. In all cases, the control inputs are gradually decreasing and the unstable tandem queue system is stabilized reaching the desired operating point. However, although the solutions for the different methodologies are close with each other, the classical simulated annealing is found to perform better than the quantum counterparts in terms of fluctuations and proximity to the exact solution. Moreover, between the two quantum annealers, we found that 2000Q performs a little better than the Advantage System as can be seen in Fig. 2 (right) showing the resulted system output of end-to-end system response time.

Next, we examine in more detail different prediction horizons for the stabilization and control of the tandem queue system which is an aspect not studied in previous works. Figure 3 shows the scaling of the solution with MPC prediction horizon values $N_s = 2, 3, 4, 5, 6, 7$. The initial conditions are the same $r_1(0) = 0.6, r_2(0) = 0.4$ and the *num-reads* parameter is set to default. In Fig. 3 (left) we observe that an increasing prediction horizon leads to an improvement in the performance of the system. Indeed, during the first few tens of time steps, the obtained control inputs show a shifting to the right and the system is stabilized and controlled before the end of the experiment at time step 100. Specifically, there is quite a difference between the solution $N_s = 2$ which cannot converge during the simulation time and the solution with $N_s = 7$ which reaches the operating point in half the simulation time. The total response time

shown in Fig. 3 (right) is also improving showing a saturation as $N_s \to 7$. This indicates that for this particular case of the unstable tandem queue system a prediction horizon of $N_s = 5$ or 6 would suffice for an acceptable performance. On the other hand, a prediction horizon as small as $N_s = 2$ is found to return a response times far from the operating point within the time step limit. However, this improvement in performance comes with a cost which we try to address in the following.

In Table 1 the elapsed times measured for the different methods and for different prediction horizons N_s per time step are presented. All times are in milliseconds. For the exact solution, we see that for $N_s \geq 5$ solving the MPC optimization problem at each time step k can be very costly; for $N_s = 7$ the calculation does not terminate in our local system. For the quantum annealing method we examine two metrics: the *sampler-call* time and *qpu-access* time. The *sampler-call* metric indicates the time taken by each call to the D-Wave sampler (implemented in the SDK), which starts executing in our local system and interacts over REST-based remote calls with the quantum device, at each time step. The *qpu-access* metric is the QPU time spent at each invocation of the quantum device including programming and sample time. The latter depends on the number of samples requested. We note that *qpu-access* time is a sub-component of *sampler-call* time.[1] The time needed for the quantum-annealing process *per sample* is reported to be 20 μs independent of problem size [7]. This indicates that the evolution time of the quantum annealing process for each sample does not depend on N_s at least for the range considered in this work ($N_s \in [2, 7]$). From Table 1 we also see that the classical simulated annealing (SA) results in lower elapsed times than the QA *sampler-call* for all N_s. This is due to the fact that the former is executed locally whereas the latter includes communication overhead. The reason to show the *sampler-call* times is to have a sense for the timing of all involving parts (computation and communication) which is important to know if we aim to control a system in real-time. Another observation is that while for prediction horizons $N_s \leq 5$, SA is faster than the purely quantum part (*qpu-access*), the latter takes advantage as the MPC optimization problem gets harder ($N_s > 5$). This result is promising as it shows the quantum advantage compared to the classical case.

In Fig. 4 we show a visualization of the results from Table 1 focusing on the relation between QA *sampler-call* and QA *qpu-access* times as the *num-reads* parameter increases from 1 (default) to 100. We observe that the time spent in the quantum device (*qpu-access* time) is a small fraction of the overall *sampler-call* time (e.g. $< 3\%$ for $N_s = 6$). This fraction is seen to increase as *num-reads* varies from 1 to 100. The remaining percentage is classical end-to-end communication time between the D-Wave machine and our infrastructure. We attribute this overhead to progressively larger message sizes, as problems (and thus QUBO expressions) that are communicated to the quantum device become more complex with increasing N_s. Another aspect we study is the effect of a varying *num-reads* parameter. In Fig. 5 we show the impact of the *num-reads*

[1] https://docs.dwavesys.com/docs/latest/c_qpu_timing.html.

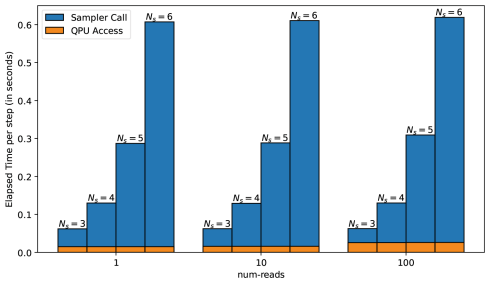

Fig. 4. Histogram showing *sampler-call* and *qpu-access* times for *num-reads* $= 1, 10, 100$ and $N_s = 3, 4, 5, 6$.

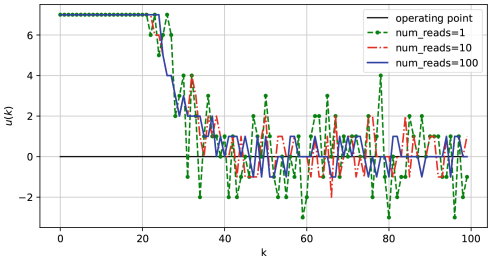

Fig. 5. The effect of *num-reads* parameter to the control input values for prediction horizon $N_s = 6$ using the Advantage System.

parameter on the quality of the obtained control inputs for the tandem queue system with *num-reads* $= 1, 10, 100$ for $N_s = 6$ using the Advantage System. We observe that the fluctuations are significantly reduced as *num-reads* increases. A comparison with the times presented in Fig. 4 shows first that a careful tuning of the parameters is required and second that in this case one may choose *num-reads* $= 10$ with a negligible difference concerning the time cost but a significant difference in the variance of the control input values. However, increasing *num-reads* leads into more expensive QPU access (Fig. 4), highlighting an important trade-off further discussed in the next section.

5 Conclusions

This work describes the construction and simulation of a tandem queue system, which we aim to stabilize and control using MPC in a quantum-enhanced manner. For this optimization problem a designed cost function is minimized leveraging the quantum annealing approach using state-of-the-art quantum devices, as well as classical methodologies such as the exact solution and simulated annealing. Specifically, we evaluate solutions from the D-Wave Advantage System and 2000Q machines investigating variations and providing insights for the significance of the prediction horizon N_s. A long prediction horizon improves the

overall model performance at the cost of an increase in computation resources. In our case, we show that a prediction horizon as long as $N_s = 6$ yields acceptable results and could be sufficient to control the system. Another aspect is the significance of the *num-reads* parameter. An important result from Table 1 is that with *num-reads* fixed, QPU access time is found to increase with increasing prediction horizons but at a much slower pace compared to classical simulated annealing. This highlights the efficiency of quantum-enhanced MPC compared to classical simulation approximations for longer prediction horizons. However, simulated annealing is found to result into less variance in control inputs compared to those obtained with any of the two D-Wave quantum annealers. This indicates that in some cases existent classical solutions perform well but this is case-dependent and further experimentation is needed. Tuning the *num-reads* parameter results in a trade-off of lower variations but higher QPU access times. Adaptive applications striving for a very short control period may want to settle for higher noise in their control inputs in exchange for shorter QPU access execution times per time step. Other applications may opt for more accurate control inputs but also more expensive quantum device invocations. The network communication part of the interaction with the quantum device is a key aspect affecting latency that should not be overlooked. This leads to an interesting question, whether real-time control is in fact possible with this approach of using quantum computing for optimization. This is important for the general functionality and faster response of complex adaptive computing systems or even mechanical/robotic systems. Our results indicate that the simulated annealing and the quantum-annealing methods for the tandem-queue system with $N_s = 6$ are able to return a solution every 19.82 ms and 16.02 ms, respectively (see Table 1) which is within 100ms, sampling time that highly-responsive adaptive systems may aim for.

As part of ongoing and future work, the current problem can be extended to a more complex generalized queueing network with multiple servers. In addition, one could experiment with other distributions for the arrival and service times as well as systems that exhibit non-linear behavior. From the perspective of quantum computing, a promising direction suitable for large problems instances is further experimentation with the available D-Wave's LEAP solvers. Our work in this paper evaluates the methodology and tooling needed to explore this promising new direction for adaptive computing systems.

Acknowledgements. We thankfully acknowledge funding by the Hellenic Foundation for Research and Innovation through the STREAMSTORE faculty grant (GrantID HFRI-FM17-1998)

References

1. Balsamo, S., De Nitto Personè, V., Inverardi, P.: A review on queueing network models with finite capacity queues for software architectures performance prediction. Perform. Eval. **51**(2), 269–288 (2003). https://doi.org/10.1016/S0166-5316(02)00099-8
2. Bertsekas, D.P.: Dynamic Programming and Optimal Control: Volumes I-II. Athena Scientific, Belmont, MA (1995)
3. Bhat, U.: An Introduction to Queueing Theory: Modeling and Analysis in Applications. Statistics for Industry and Technology, Birkhäuser Boston (2015)
4. Boxma, O., Resing, J.: Tandem queues with deterministic service times. Ann. Oper. Res. **49**, 221–239 (1994). https://doi.org/10.1007/BF02031599
5. Camacho, E., Bordons, C.: Model Predictive Control. Springer, London, UK (2004)
6. Cerf, S., Berekmeri, M., Robu, B., Marchand, N., Bouchenak, S.: Cost function based event triggered model predictive controllers application to big data cloud services. In: 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 1657–1662 (2016). https://doi.org/10.1109/CDC.2016.7798503
7. D-Wave: Ocean SDK Documentation (2023). https://docs.ocean.dwavesys.com
8. De Matteis, T., Mencagli, G.: Proactive elasticity and energy awareness in data stream processing. J. Syst. Softw. **127**, 302–319 (2017). https://doi.org/10.1016/j.jss.2016.08.037
9. Deng, Z., Wang, X., Dong, B.: Quantum computing for future real-time building hvac controls. Appl. Energy **334**, 120621 (2023). https://doi.org/10.1016/j.apenergy.2022.120621
10. Fang, Q., Wang, J., Gong, Q.: Qos-driven power management of data centers via model predictive control. IEEE Trans. Autom. Sci. Eng. **13**(4), 1557–1566 (2016). https://doi.org/10.1109/TASE.2016.2582501
11. Filieri, A., Maggio, M., Angelopoulos, K., D'Ippolito, N., Gerostathopoulos, I., Hempel, A.B., Hoffmann, H., Jamshidi, P., Kalyvianaki, E., Klein, C., Krikava, F., Misailovic, S., Papadopoulos, A.V., Ray, S., Sharifloo, A.M., Shevtsov, S., Ujma, M., Vogel, T.: Software engineering meets control theory. In: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 71–82 (2015). https://doi.org/10.1109/SEAMS.2015.12
12. Hellerstein, J., Diao, Y., Parekh, S., Tilbury, D.M.: Feedback Control of Computing Systems. Wiley Interscience Press (2004)
13. Inoue, D., Yoshida, H.: Model predictive control for finite input systems using the d-wave quantum annealer. Sci. Rep. 10(1591) (2020). https://doi.org/10.1038/s41598-020-58081-9
14. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Phys. Rev. E **58**, 5355–5363 (1998). https://doi.org/10.1103/PhysRevE.58.5355
15. Karniavoura, F., Magoutis, K.: Decision-making approaches for performance QOS in distributed storage systems: a survey. IEEE Trans. Parallel Distrib. Syst. (TPDS) **30**(8), 1906–1919 (2019). https://doi.org/10.1109/TPDS.2019.2893940. August
16. Kirk, D.E.: Optimal Control Theory: An Introduction. Prentice-Hall, Englewood Cliffs, N.J. (2004)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983) https://doi.org/10.1126/science.220.4598.671, https://www.science.org/doi/abs/10.1126/science.220.4598.671

18. Kobayashi, H., Konheim, A.: Queueing models for computer communications system analysis. IEEE Trans. Commun. **25**(1), 2–29 (1977). https://doi.org/10.1109/TCOM.1977.1093702
19. Le Gall, P.: The theory of networks of single-server queues and the tandem queue model. J. Appl. Math. Stoch. Anal. **10**(4), 363–381 (1997)
20. Lucas, A.: Ising formulations of many np problems. Front. Phys. **2** (2014). https://doi.org/10.3389/fphy.2014.00005
21. Neuts, F.M.: Two queues in series with a finite, intermediate waitingroom. J. Appl. Prob. **5**(1), 123–142 (1968). http://www.jstor.org/stable/3212081
22. Padala, P., Hou, K.Y., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A.: Automated control of multiple virtualized resources. In: Proceedings of the 4th ACM European Conference on Computer Systems (EuroSys). Nuremberg, Germany (2009)
23. Palmer, G.I., Knight, V.A., Harper, P.R., Hawa, A.L.: CIW: An open-source discrete event simulation library. J. Simul. **13**(1), 68–82 (2019). https://doi.org/10.1080/17477778.2018.1473909
24. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**(79) (2018)
25. Qin, S., Badgwell, T.: A survey of industrial model predictive control technology. Control. Eng. Pract. **93**(316), 733–764 (2003)
26. Rosberg, Z., Varaiya, P., Walrand, J.: Optimal control of service in tandem queues. IEEE Trans. Autom. Control **27**(3), 600–610 (1982). https://doi.org/10.1109/TAC.1982.1102957
27. Rossiter, J.A.: A First Course in Predictive Control. CRC Press (2018)
28. Santoro, G.E., Tosatti, E.: Optimization using quantum mechanics: quantum annealing through adiabatic evolution. J. Phys. A: Math. General **39**(36), R393 (2006). https://doi.org/10.1088/0305-4470/39/36/R01, https://dx.doi.org/10.1088/0305-4470/39/36/R01
29. Schoeffauer, R., Wunder, G.: Model-predictive control for discrete-time queueing networks with varying topology. IEEE Trans. Control Netw. Syst. **8**(3), 1528–1539 (2021). https://doi.org/10.1109/TCNS.2021.3074250
30. Schwenzer, M., Ay, M., Bergs, T., Abel, D.: Review on model predictive control: An engineering perspective. Int. J. Adv. Manuf. Technol. **117**, 1327–1349 (2021). https://doi.org/10.1007/s00170-021-07682-3
31. Suman, B., Kumar, P.: A survey of simulated annealing as a tool for single and multiobjective optimization. J. Oper. Res. Soc. **57**, 1143–1160 (2006). https://doi.org/10.1057/palgrave.jors.2602068
32. de Waal, P.R.: Performance analysis and optimal control of an mm1k queueing system with impatient customers, pp. 28–40. Springer, Berlin Heidelberg (1987). https://doi.org/10.1007/978-3-642-73016-0_3
33. Wang, C., Chen, H., Jonckheere, E.: Quantum versus simulated annealing in wireless interference network optimization. Sci. Rep. **6**(25797) (2016). https://doi.org/10.1038/srep25797
34. Xu, Q., Ma, G., Ding, K., Xu, B.: An adaptive active queue management based on model predictive control. IEEE Access **8**, 174489–174494 (2020). https://doi.org/10.1109/ACCESS.2020.3025377
35. Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: Introduction and review. Rep. Progr. Phys. **85**(10), 104001 (2022). https://doi.org/10.1088/1361-6633/ac8c54