Strategic Earning on Tokenized Platforms via Model-based Decision Making

Extended Results

Tianyi Li *Department of Decisions, Operations and Technology, CUHK*

Extended Result A: Strategy parameters

Parameters of laboring strategies (*SL**) are experimented with (Figure A7). Four cases are shown on each plot. Invariant parameter values are (a) $g_L = 1$, (b) $T_D = 2$, (f) $p_0^{F,+g_L} = 0.6$, (g) $g_L = 0.01$.

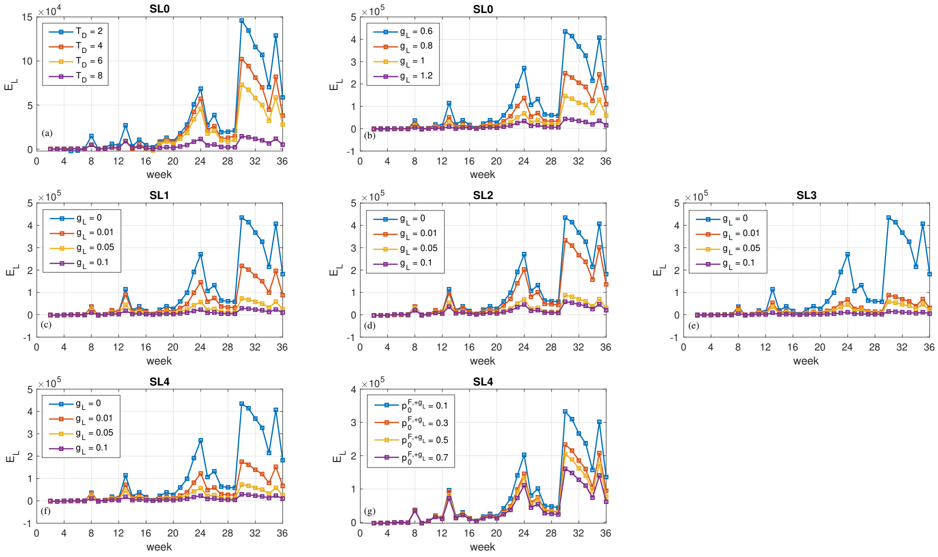

Figure A7: Parameters for strategies on laboring intensity (*SL0* - *SL4*).

Parameters of investment strategies (*SP**, E_P-dependent) are experimented with. Four cases are shown on each plot. Invariant parameter values are:

Figure A8: (a) $g_{P+} = 1$, $g_{P-} = 0.8$, (b) $T_D = 6$, $g_{P-} = 0.8$, (c) $T_D = 2$, $g_{P+} = 1.5$, (d) $g_{P-} = 0.5$, (e) $g_{P+} = 1.5$, (f) $g_{P+} = 1$, $g_{P-} = 0.8$, (g) $T_D = 6$, $g_{P-} = 0.6$, (h) $T_D = 6$, $g_{P+} = 1.5$, (i) $g_{P+} = 1$, $g_{P-} = 0.2$, (j) $T_D = 3$, $g_{P-} = 0.6$, (k) $T_D = 5$, $g_{P+} = 1.5$, (l) $g_{P+} = 2$, $g_{P-} = 0.2$, (m) $T_D = 4$, $g_{P-} = 0.8$, (n) $T_D = 2$, $g_{P+} = 1.5$.

Figure A9: (a) $g_{P-} = 0.8$, $p_0^{F,+P+/-P-} = 0.8/0.3$, (b) $g_{P+} = 1.5$, $p_0^{F,+P+/-P-} = 0.8/0.3$, (c) $g_{P+/P-} = 1.5/0.8$, $p_0^{F,-P-} = 0.3$, (d) $g_{P+/P-} = 1.5/0.8$, $p_0^{F,+P+} = 0.8$, (e) $g_{P-} = 0.8$, $p_0^{F,+P+/-P-} = 0.8/0.3$, $T_D = 3$, (f) $g_{P+} = 1.5$, $p_0^{F,+P+/-P-} = 0.8/0.3$, $T_D = 6$, (g) $g_{P+/P-} = 1.5/0.8$, $p_0^{F,-P-} = 0.3$, $T_D = 6$, (h) $g_{P+/P-} = 1.5/0.8$, $p_0^{F,+P+} = 0.8$, $T_D = 6$, (i) $g_{P+/P-} = 1.5/0.8$, $p_0^{F,+P+/-P-} = 0.8/0.3$.

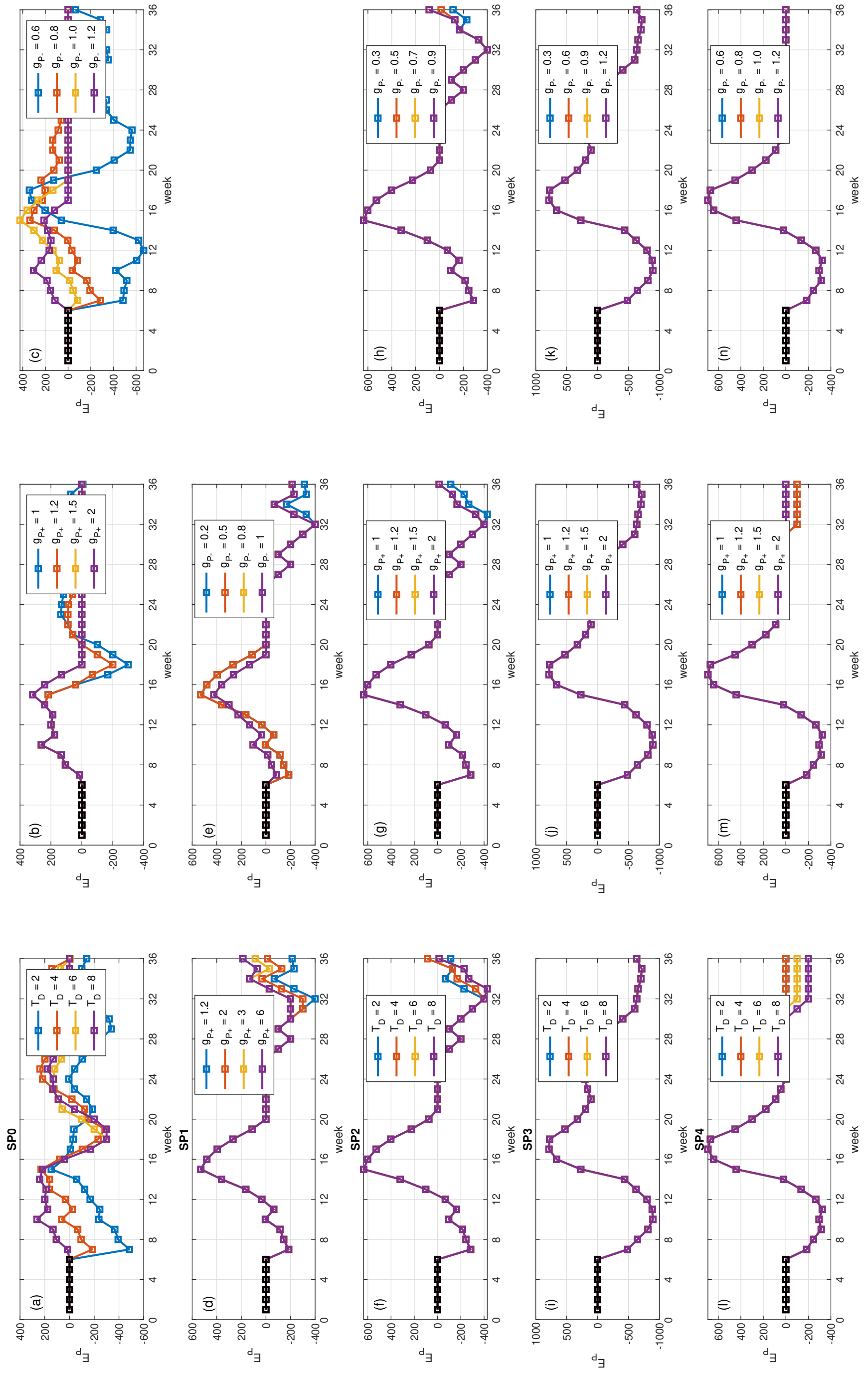

Figure A8: Parameters for strategies on investment intensity (*SP0* - *SP4*).

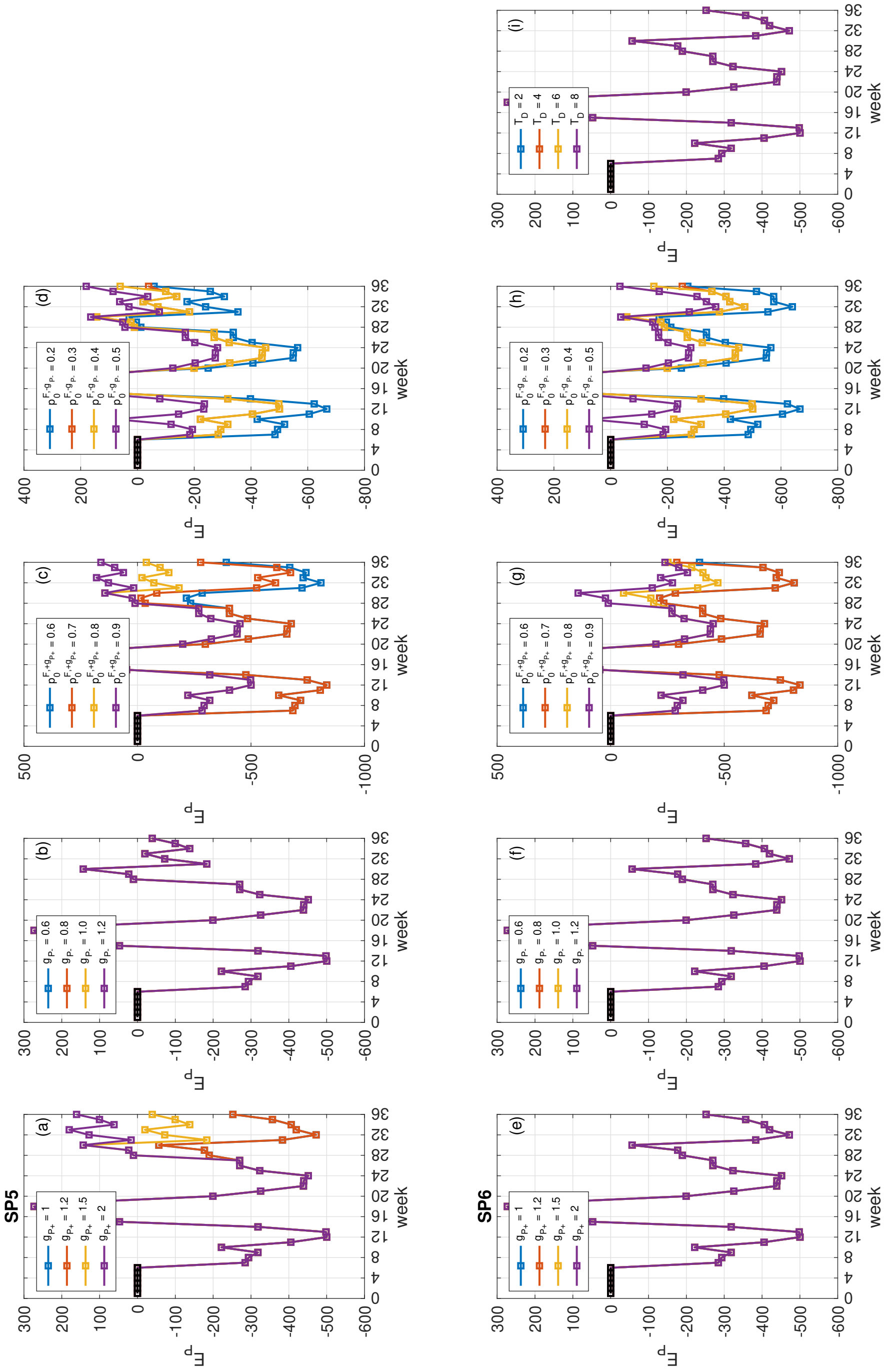

Figure A9: Parameters for strategies on investment intensity (*SP5* - *SP6*).

Extended Result B: Robustness checks at metric-based decision-making

We conduct robustness checks to supplement main results at the metric-based decision-making (Section 7).

Strategies on investment intensity - η^β dependent. We investigate investment strategies based on metrics for projected price η^β, instead of projected E_P (Figure A10). Model-based strategies have X in the suffix; model-free *SP0* remains the same. Other conditions are the same as the main result. Under the same strategy parameters, PI-based strategies are less profitable than the main result (Figure A8); even the conservative strategy is not securing much earning. This derives from price series's higher variance than E_P series; the latter integrates price over a $t_{p,ind}$ interval and is thus less volatile. Strategy based on historical mean, *SP2X*, performs the best and outperforms the model-free *SP0*.

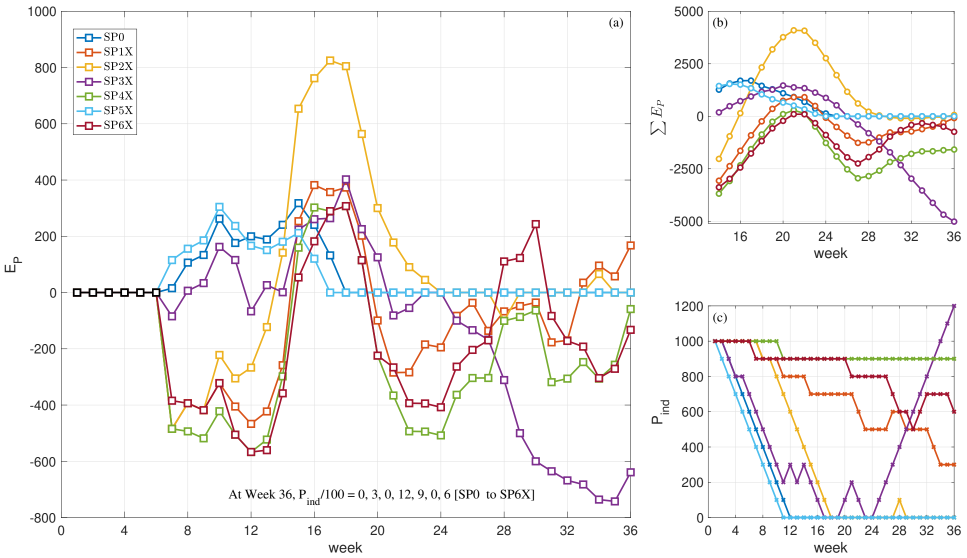

Figure A10: Strategies on investment intensity (η^β-dependent). Comparing model-free strategy *SP0* and model-based strategies *SP1X* to *SP6X*. Notations same as Figure 8.

Strategy parameters are experimented with. Four cases are shown on each plot. Invariant parameter values for Figure A11 and A12 are the same as for Figure A8 and A9.

Fixed amount execution at the investment decision. We analyzing six strategies *SP*1* to *SP*6* and the model-free *SP*0* (OS-C). Start with zero investment; at each period t, the action a_P to be made is invest a fixed amount \$100 ($a_P = 1, P_{ind}(t) = 100$), or not invest ($a_P = 0, P_{ind}(t) = 0$). For both strategies based on E_P projection (Figure A13) and strategies based on price projection (Figure A14), results are consistent with main results as fixed increment execution. For example, the aggressive strategy gains much earning during the first mid-term, but suffers great loss in the second mid-term.

Choice on system parameters. Parameters in model estimation, in model projection, and in strategy and evaluation (Table 2) are experimented with extensively. Main results are shown with appropriate parameter values. Model estimation and projection are robust to uncertainty parameters; time window parameters behave consistently under different values; strategy parameters lead to desired behavior of strategies. Choice of parameter values is nonetheless subject to future discussion.

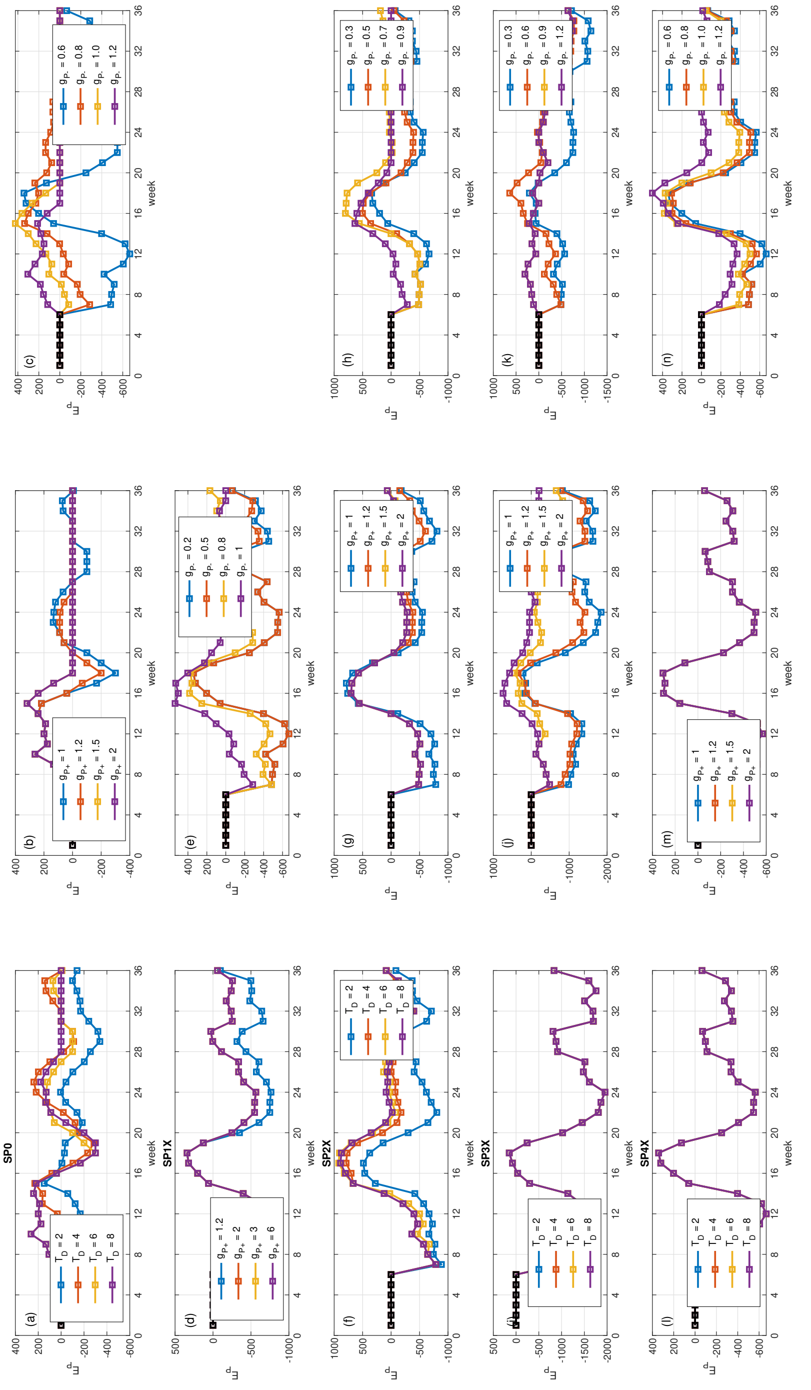

Figure A11: Parameters for strategies on investment intensity (*SP0*, *SP1X* - *SP4X*).

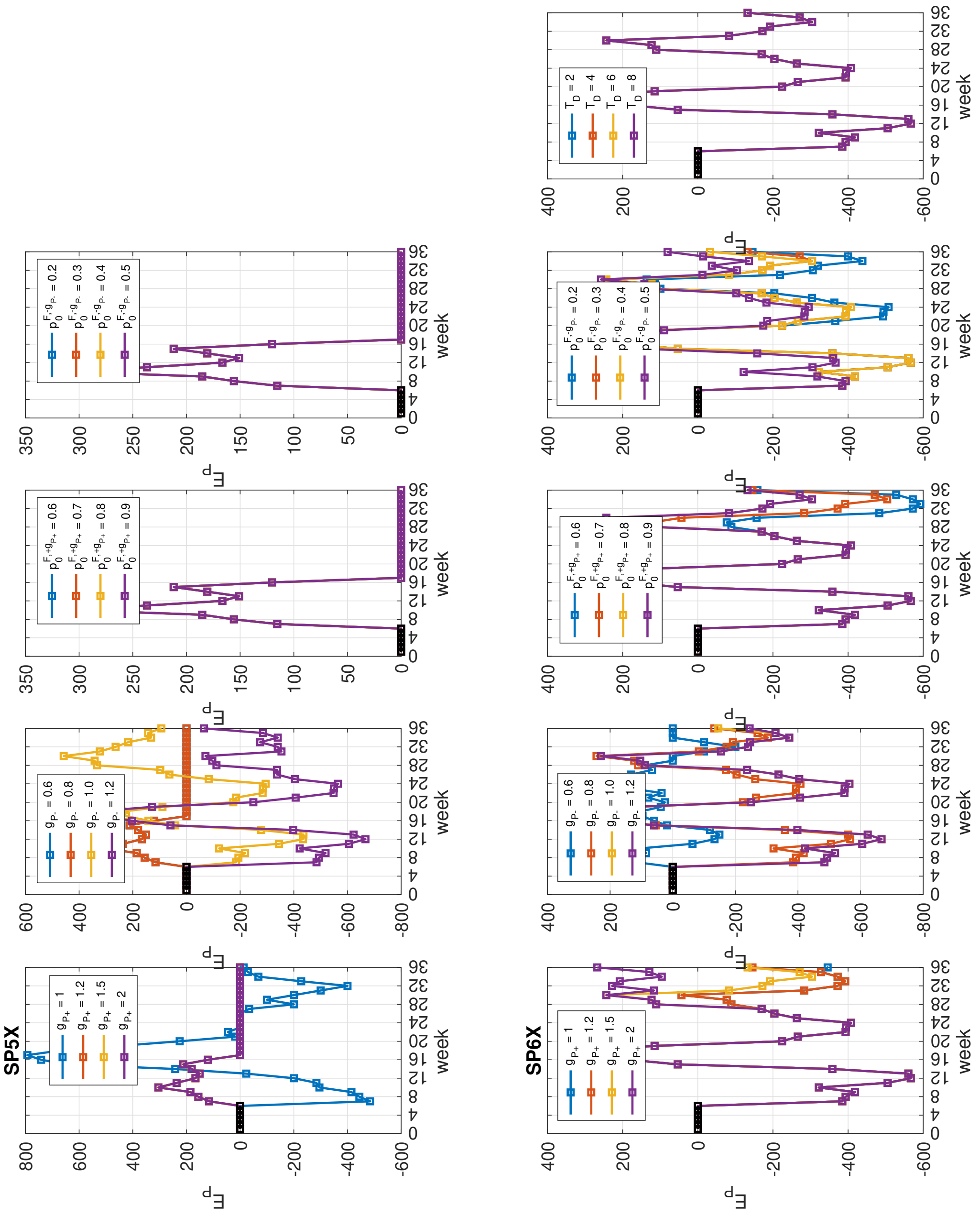

Figure A12: Parameters for strategies on investment intensity (*SP5X* - *SP6X*).

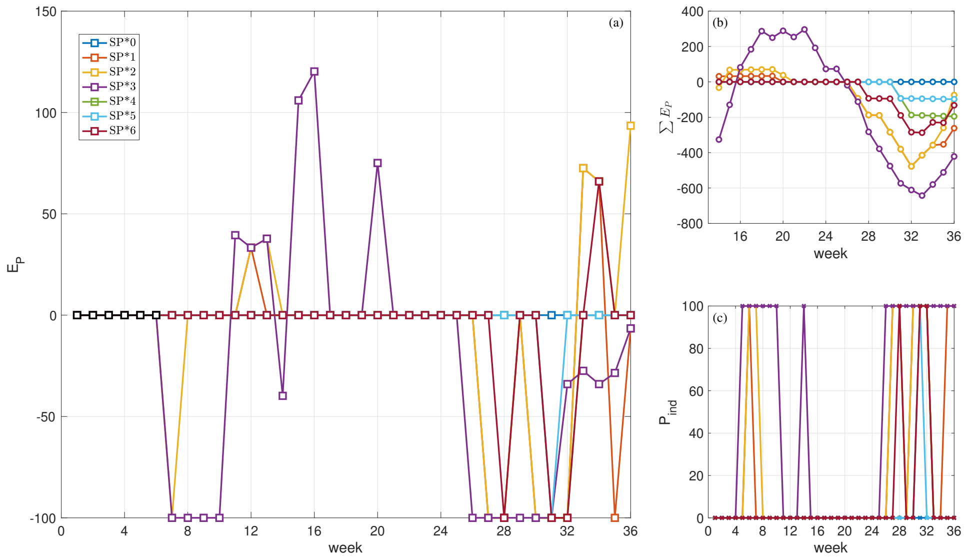

Figure A13: Strategies on investment intensity (E_P-dependent) under fixed amount execution (OS-C). Comparing model-free strategy *SP* 0* and model-based strategies *SP* 1*–*SP* 6*. Notations same as Figure 8.

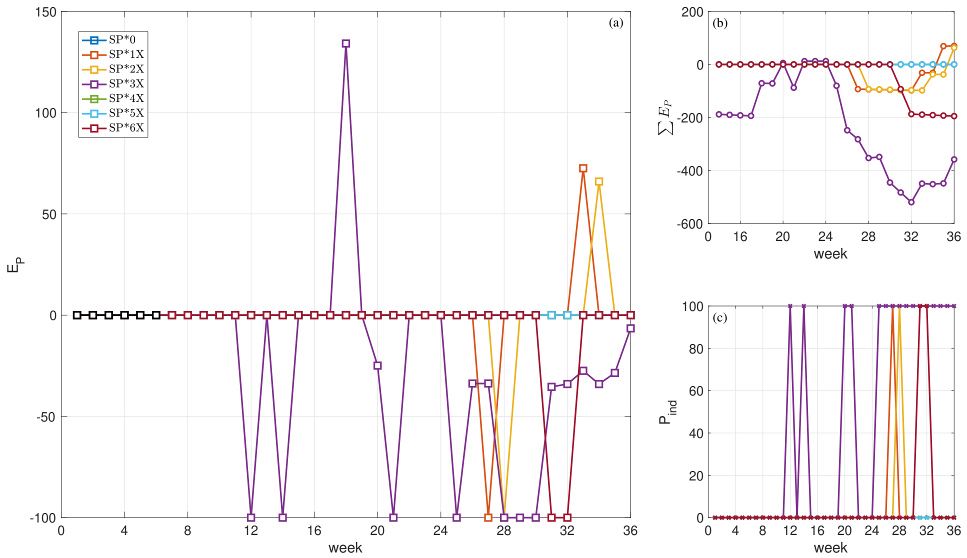

Figure A14: Strategies on investment intensity (η^β-dependent) under fixed amount execution (OS-C). Comparing model-free strategy *SP* 0* and model-based strategies *SP* 1X*–*SP* 6X*. Notations same as Figure 8.

Extended Result C: Related literature

Besides grounded in the literature on platform economy and on blockchains, this study is related to the literature on strategic participation on digital platforms, on uncertainties in forecast-based decision-making, on optimal execution and algorithm trading, on deep RL and in particular RL applications in management sciences.

Strategic participation on digital platforms

The development of digital platforms asks for sustained participation from customers (*Oh et al.*, 2016), on the target platform as well as on external platforms such as social media (*Rishika et al.*, 2013). The success of online communities "depends on participants' willingness to invest their time and attention" (*Bateman et al.*, 2011) on the platform.

Along the advancement of individuals' daily operations on digital platforms, participants have evolved to demonstrate strategic behavior during on-platform interactions. For example, strategic activities of individual contestants on a competitive software development platform (TopCoder.com) are documented (*Archak*, 2010), among the broad strategic participation in crowdsourcing contests (*Zheng et al.*, 2011). Conceivably, participation on platforms, especially developing platforms, evolves under platform's different governance modes with varied access and control conditions (*O'Mahony and Karp*, 2022).

Digital platforms are carefully responding to this transition. For example, platform can implement fast-fashion systems (*Cachon and Swinney*, 2011; *Aviv et al.*, 2019) that combine quick response production capabilities with enhanced product design capabilities, to both design "hot" products that capture the latest consumer trends and exploit minimal production lead times to match supply with uncertain demand; on e-commerce platforms, different discount pricing strategies of online coupons can bring various effects to strategic consumers (*Li et al.*, 2020); at the platform ecosystem scale, user preferences for ecosystem innovativeness and ecosystem size, and demand-based economies of scale, can shape the strategic interactions between the platform provider and the complementors (*Panico and Cennamo*, 2022).

On blockchain platforms, driven by considerable economic incentives, participants are highly strategic, at mining pool selection (*Liu et al.*, 2018), block reward maximization (*Hacioglu et. al*, 2021), token investment decision (*Alessandretti et al.*, 2018), and at portfolio management across platforms (*Brauneis and Mestel*, 2019) etc. Analyzing participants' contemplated behavior on blockchains contributes to the discussion of strategic participation on crowdsourcing, crowdfunding, and general digital platforms.

Uncertainties in forecast-based decision-making

For decision-making based on model projection, uncertainty of forecast can be indicated with prediction intervals (PI) around the point forecast. A set of PI consists of an upper bound and a lower bound, where a certain fraction (e.g., 95%) of forecast instances lie within this interval. Different methods for constructing PIs are invented in various fields: at wind-power forecasting (*Yuan et al.*, 2019), energy management (*Valencia et al.*, 2015), electricity price forecast (*Shrivastava et al.*, 2015) etc. As summarized in *Shrestha and Solomatine* (2006), uncertainty in model forecast can be characterized from different approaches: construct forecast using the Bayesian methodology (*Bishop*, 1995), estimate uncertainty with statistical properties of model's error structure (e.g., Gaussian errors in linear regression), use re-sampling or Monte-Carlo methods to yield forecast ensembles (*Stine*, 1985; *Heskes*, 1996), or adopt fuzzy theory based methods with the non-probabilistic logic (*Sáez et al.*, 2014; *Valencia et al.*, 2015; *Kavousi-Fard et al.*, 2015). Along the recent advance of machine learning, PIs are also constructed using various machine learning models, e.g., neural networks (*Hwang and Ding*, 1997; *Khosravi et al.*, 2010), LSTM (*Yuan et al.*, 2019), SVM (*Shrivastava et al.*, 2015). See *Marín et al.* (2019) for a latest review of PI modeling.

Among different approaches, Monte-Carlo methods for PI construction are widely adopted, due to its simplicity and general applicability, despite the associated computational cost. In this approach, an ensemble of model forecasts are collected, and PIs are derived directly from the statistic of this ensemble. Without further assumption on the uncertainty's structure, Monte-Carlo methods help determine PIs in a straightforward way.

Optimal execution and algorithm trading

The optimal execution problem (*Bertsimas and Lo*, 1998; *Almgren and Chriss*, 2001) determines the optimal execution of orders (i.e., liquidation) over a finite trading horizon, minimizing the trading cost. With a model describing price dynamics, this problem can be cast into a stochastic optimal control problem and solved using dynamic programming (DP). Besides traditional DP solvers, *Nevmyvaka et al.* (2006) presented the first application of RL on optimal execution. Within a time horizon, system optimizes the execution of a target trading volume upon interaction with the market. Through the discretization of trading unit and trading interval, the problem is formulated as a Q-learning procedure, which can be solved greedily thanks to the relatively small state-action space. They showed that learned policies have a large reward margin over heuristics such as submit-and-leave or immediate-execution.

Their work inspired subsequent studies of RL-based trading as well as portfolio optimization (e.g., *Eilers et al.*, 2014; *Fischer*, 2018; *Liu et al.*, 2018; *Wu et al.*, 2020; *Zhang et al.*, 2020). People adopt different learning methods: value-based (SARSA (*Pendharkar and Cusatis*, 2018), Q-learning (*Hendricks and Wilcox*, 2014; *Pendharkar and Cusatis*, 2018), deep Q-learning (*Jeong and Kim*, 2019), DDQN (*Ning et al.*, 2018; *Dabérius et al.*, 2019)), policy based (PPO (*Dabérius et al.*, 2019), DDPG (*Xiong et al.*, 2018)), or actor-critic methods (*Yang et al.*, 2020). Algorithms are applied to different trading targets: stock (*Hendricks and Wilcox*, 2014), foreign exchange (*Carapuço et al.*, 2018), or cryptocurrency (*Jiang and Liang*, 2017), and RL solutions are compared to heuristic trading strategies such as time-weighted average price (TWAP) (*Dabérius et al.*, 2019) or immediate-execution (*Nevmyvaka et al.*, 2006), to model-based strategies (*Hendricks and Wilcox*, 2014), and among different RL systems (*Deng et al.*, 2016). This line of research gains popularity along the recent progress of deep learning (*LeCun et al.*, 2015), where deep RL is combined with other deep learning modules such as fuzzy layer (*Deng et al.*, 2016), recurrent structure (*Jiang et al.*, 2017), and autoencoder (*Li et al.*, 2019). In the past decade, RL has developed into a key component of algorithm trading (*Cartea et al.*, 2015) that constitutes an important domain in machine behaviour (*Rahwan et al.*, 2019).

When the agent makes an action (trading), it brings impact to the environment (market price). The impact can be decomposed into permanent price impact, which persists during the liquidation horizon, and temporary price impact, which only exists in the current period (*Bertsimas and Lo*, 1998; *Almgren and Chriss*, 2001). Linear price impact function is often used (*Dabérius et al.*, 2019), where the slope can be called Kyle's Lambda (*Kyle*, 1985). When agent action's influence on market price is negligible (e.g., *Nevmyvaka et al.*, 2006), the price impact function is omitted.

During optimal execution, different reward functions can be considered besides accumulated return minus execution cost, such as risk-adjusted return (*Almgren and Chriss*, 2001), ratio of average return to standard deviation over execution horizon (*Deng et al.*, 2016), and additional return with respect to execution at the starting price (*Dabérius et al.*, 2019).

Deep reinforcement learning

Reinforcement learning (RL) considers the Markov decision process (MDP) where an agent interacts with the environment in discrete time steps (*Arulkumaran et al.*, 2017; *Sutton and Barto*, 2018). It is rooted in value iteration (*Bellman*, 1957) and temporal difference (*Sutton*, 1988) methods. At each time step t, the agent is in state $\boldsymbol{S}_t \in \mathbb{S}$ that describes its own status and the status of the environment. According to a policy π that determines agent's action under the specific state, the agent performs action $\boldsymbol{A}_t \in \mathbb{A}$, under which state $\boldsymbol{S}_t$ transits to $\boldsymbol{S}_{t+1}$ for the next time step. A reward r_t is assigned to the agent depending on action $\boldsymbol{A}_t$. The goal is to learn an optimal policy π^* that maximizes the expectation of the discounted cumulative return R over a time horizon t_H: $R = \sum_{t=1}^{t_H} \epsilon_T^{t-1} r_t$, where ϵ_T is the discount factor.

Best policy π^* can be learned via value function methods, direct policy search, or the actor-critic approach that is the hybrid of the two (*Arulkumaran et al.*, 2017). Value function methods utilize (i) state-value function $V^\pi(\boldsymbol{S})$ which specifies the expected return of each state $\boldsymbol{S}$ under a certain policy π, (ii) state-action value function $Q^\pi(\boldsymbol{S}, \boldsymbol{A})$ which specifies the expected return of each state-action pair $(\boldsymbol{S}, \boldsymbol{A})$ under π, and (iii) advantage function $A^\pi(\boldsymbol{S}, \boldsymbol{A}) = Q^\pi(\boldsymbol{S}, \boldsymbol{A}) - V^\pi(\boldsymbol{S})$ which records different actions' value difference.

Bellman equation (*Bellman*, 1952) exploits the Markov property of $Q^\pi(\boldsymbol{S}, \boldsymbol{A})$:

$$Q^\pi(\boldsymbol{S}_t, \boldsymbol{A}_t) = \mathbb{E}_{\boldsymbol{S}_{t+1}}[r_{t+1} + \epsilon_T Q^\pi(\boldsymbol{S}_{t+1}, \pi(\boldsymbol{S}_{t+1}))], \tag{1}$$

which suggests that Q^π can be solved gradually through minimizing the temporal difference (TD) error, and then policy π is derived from a determined Q^π. Depending on the TD learning format, Q-learning (off-policy) (*Watkins and Dayan*, 1992) and SARSA (on-policy) (*Rummery and Niranjan*, 1994) are constructed.

The state-action table form of $Q^\pi(\boldsymbol{S}, \boldsymbol{A})$ can be replaced with a neural network that approximates more complex value functions, leading to deep Q-learning (DQN) (*Mnih et al.*, 2015). With its delicate engineering design (e.g., experience replay and target networks), DQN starts the era of deep reinforcement learning (DRL) and inspires a series of design (e.g., double DQN (*Van Hasselt et al.*, 2016), asynchronous training (*Mnih et al.*, 2016), dueling architecture (*Wang et al.*, 2016), prioritized experience replay (*Schaul et al.*, 2015)), where deep networks help RL achieve beyond-human performance on many gaming tasks such as Atari games (*Mnih et al.*, 2015) and Go (*Silver et al.*, 2016). For more real-world tasks such as performing physics experiments (*Denil et al.*, 2016) or indoor navigation (*Zhu et al.*, 2017), which are often related to robotics and are often model-free, DRL is shown to achieve performance comparable to human. Policy-based methods (e.g., PPO (*Schulman et al.*, 2017), TRPO (*Schulman et al.*, 2015)) and actor-critic methods (e.g., DDPG (*Lillicrap et al.*, 2015), multi-agent DDPG (*Lowe et al.*, 2017)) also enjoy the benefits of deep networks and join DRL as distinct branches besides value function methods. Recent efforts explore multi-agent RL (*Littman*, 1994), inverse RL (*Ng and Russell*, 2000), hierarchical RL (*Barto and Mahadevan*, 2003), and Meta RL (*Wang et al.*, 2018) within the realm of DRL (see *Arulkumaran et al.* (e.g., 2017) for a detailed discussion).

Parallel to model-free RL is model-based RL, which is closely related to the planning problem (*Sutton*, 1990). Planning typically uses local solutions and focuses on one state or a subset of states, while RL derives global solutions over the entire state space. Despite many successful applications (e.g., *Doya et al.*, 2002; *Polydoros and Nalpantidis*, 2017; *Kaiser et al.*, 2019; *Wang et al.*, 2019), there are various challenges with model-based RL solvers (*Moerland et al.*, 2020), such as stochasticity, limited data, partial observability, multi-step learning, non-stationarity of MDP etc., some of which also apply to model-free RL. For both model-based and model-free RL, abstraction and interpretation of learned policies require sophisticated representation techniques (e.g., saliency map (*Simonyan et al.*, 2013)) to generate practical insights in the strategy space.

RL applications in management sciences

Optimal control (*Bertsekas*, 2012) is an important topic in operations research and management sciences. Sequential decision-making problems under uncertainty can be modeled as Markov decision process (MDP) (*Bellman*, 1957; *Howard*, 1960) or semi-Markov decision process (SMDP) (*Das et al.*, 1999; *Gosavi*, 2004), which can be solved with dynamic programming (DP). DP nevertheless often requires a complete and accurate model of the environment and may be sufficiently feasible only in a constrained state-action space, suffering from the *curse of dimensionality* (*Gosavi*, 2004). When the MDP employs a large state space or the environment is only partially observed (i.e., POMDP (*Burnetas and Katehakis*, 1997; *Kaelbling et al.*, 1998)), people use RL to generate near-optimal solutions (*Sutton and Barto*, 2018).

RL and DRL witness applications on a wide range of topics in management studies. A key application domain is supply chain management and inventory control (*Das et al.*, 1999; *Gosavi*, 2004; *Oroojlooyjadid et al.*, 2021). When inventory optimization is in single period, the problem is bandit learning and can be solved with DP; when inventory control is more complicated, such as supervising multiple periods or with unknowns, conducting DP is difficult and RL methods are employed (e.g., *Chen and Simchi-Levi*, 2004a,b; *Chen et al.*, 2021). For example, in a seminal work, *Das et al.* (1999) proposed an algorithm for continuous-time SMDP and applied it to optimizing preventative maintenance in a production inventory system. Along this line of research, *Li et al.* (2012) applies RL to joint pricing, lead-time and scheduling decisions in make-to-order systems. More recently, *Dai and Gluzman* (2021) applies and advances PPO and TRPO methods to the stochastic control of queueing networks; *Oroojlooyjadid et al.* (2021) applies DQN to the beer game, a classic demonstration of the bullwhip effect in supply chain management, extending earlier works (e.g., *Chaharsooghi et al.*, 2008).

Besides inventory control management and trading optimization (as mentioned above), RL is applied to other management domains. *Kokkodis and Ipeirotis* (2021) applies RL to career path recommendation, relying on a Markov decision process to operate on a graph of feasible actions and dynamically recommend profitable career paths. *Chen et al.* (2021) reports combining DRL and prediction models to develop a multiobjective markdown system for many merchandising units at Walmart. *Liang et al.* (2022) reports applying DRL to laptop manufacturing in a major personal computer manufacturing company (Lenovo). RL is also frequently adopted in transportation science. *Oda and Joe-Wong* (2018) employs DQN in a model-free setting to study fleet management. *Shou et al.* (2020) employs a MDP to model idle e-hailing drivers' optimal sequential decisions in passenger-seeking, leveraging an inverse RL technique in a multi-agent setting. *Qin et al.* (2020) reports applying DRL to ride-hailing order dispatching at a large-scale ride-hailing platform (DiDi). *Li et al.* (2021) applies multi-agent RL to adaptive traffic signal control through knowledge-sharing protocol.

Extended Result D: Addressing challenges in real-world RL applications

Dulac-Arnold et al. (2019) laid out nine challenges when applying RL to real-world problems. Some challenges can be addressed in our RL framework, through which we show that the current decision-making problem is suitable for a model-based RL application. Unattended challenges point to future works.

- *Off-line and off-policy training.* Our training can be either on-policy or off-policy. When we consider slow deviation of platform development from the projected trajectory, i.e., slow change in platform parameters, policies can be determined off-line. The development model's descriptive power is limited, however, and relies heavily on estimation, especially for platforms with a short history.

- *Sample efficiency.* The model-based framework does not suffer from the problem of insufficient training samples. However, boostrapping is confined to a small set of platform parameters, and sufficient care is needed to slow down training and prevent local optima (*Botvinick et al.*, 2019). It is useful to explore shared training across different platforms to derive platform-agnostic policies, in addition to current platform-specific policies. Techniques such as episodic memory (*Gershman and Daw*, 2017) and meta-learning (*Finn et al.*, 2017) can be experimented with.

- *High-dimensional state/action space.* Our state vector $\boldsymbol{S}$ has an unfixed length (determined by τ and t_p). To reduce the state space, continuous state values can be rendered discrete at different resolutions. Action space $\boldsymbol{A}$ is discrete and small. It is useful to explore larger and more detailed action space.

- *Safety constraints.* Safety issues are not applicable to our setting. Strategies on investment and laboring intensity should be taken as mere reference. The platform development model does not specify exit mechanisms, thus the system may fail in projecting platform's collapse.

- *Partial observability.* To describe platform's state, only η is used, which is transformed from the public token price. w_m indicates platform's development stage, requiring some trial-and-error when training the agent for the first time; at re-training, its value can derive from the previous value.

- *Reward function.* Discounted earning accumulated over a finite horizon is a natural reward for our RL environment. We can consider investment earning and laboring earning separately, or combine them. Time-varying reward functions (*Nagabandi et al.*, 2018) can be studied in future works.

- *Policy explainability.* Policies that govern state-action transitions are more interpretable with value functions (i.e., Q-learning) and less with neural network architectures (i.e., DQN). Summarize and visualize learned policies is a key challenge for our framework.

- *Real-time operations.* Current analysis assume weekly operations of decision-making and do not suffer from the real-time constraint. It is useful to investigate the framework's applicability to high-frequency decision-making, addressing the constraints for training on-the-fly.

- *System delays.* A finite history of η is recorded in the state vector to capture system delay. Delay in investment return is a fundamental real-world feature and is carefully addressed in the model. This asks good investment policy to maintain sufficient foreseeing capacity. Our RL agents demonstrate this ability in the results.

References

Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018), Anticipating cryptocurrency prices using machine learning, *Complexity*, 2018.

Almgren, R., & Chriss, N. (2001), Optimal execution of portfolio transactions, *Journal of Risk*, 3, 5-40.

Archak, N. (2010, April), Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com, In *Proceedings of the 19th International Conference on World Wide Web* (pp. 21-30).

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017), Deep reinforcement learning: A brief survey, *IEEE Signal Processing Magazine*, 34(6), 26-38.

Aviv, Y., Wei, M. M., & Zhang, F. (2019), Responsive pricing of fashion products: The effects of demand learning and strategic consumer behavior, *Management Science*, 65(7), 2982-3000.

Barto, A. G., & Mahadevan, S. (2003), Recent advances in hierarchical reinforcement learning, *Discrete Event Dynamic Systems*, 13(1), 41-77.

Bateman, P. J., Gray, P. H., & Butler, B. S. (2011), Research note—the impact of community commitment on participation in online communities, *Information Systems Research*, 22(4), 841-854.

Bellman, R. (1952), On the theory of dynamic programming, *PNAS*, 38(8), 716.

Bellman, R. (1957), A Markovian decision process, *Journal of Mathematics and Mechanics*, 679-684.

Benjaafar, S., & Hu, M. (2020), Operations management in the age of the sharing economy: What is old and what is new? *Manufacturing & Service Operations Management*, 22(1), 93-101.

Bertsekas, D. (2012), Dynamic programming and optimal control: Volume I (Vol. 1)., *Athena Scientific*.

Bertsimas, D., & Lo, A. W. (1998), Optimal control of execution costs, *Journal of Financial Markets*, 1(1).

Bishop, C. M. (1995), *Neural networks for pattern recognition*, Oxford university press.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019), Reinforcement learning, fast and slow, *Trends in Cognitive Sciences*, 23(5), 408-422.

Brauneis, A., & Mestel, R. (2019), Cryptocurrency-portfolios in a mean-variance framework, *Finance Research Letters*, 28, 259-264.

Burnetas, A. N., & Katehakis, M. N. (1997), Optimal adaptive policies for Markov decision processes, *Mathematics of Operations Research*, 22(1), 222-255.

Cachon, G. P., & Swinney, R. (2011), The value of fast fashion: Quick response, enhanced design, and strategic consumer behavior, *Management Science*, 57(4), 778-795.

Carapuço, J., Neves, R., & Horta, N. (2018), Reinforcement learning applied to Forex trading, *Applied Soft Computing*, 73, 783-794.

Cartea, Á., Jaimungal, S., & Penalva, J. (2015), *Algorithmic and high-frequency trading*, Cambridge University Press.

Chen, X., & Simchi-Levi, D. (2004a), Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case, *Operations Research*, 52(6), 887-896.

Chaharsooghi, S. K., Heydari, J., & Zegordi, S. H. (2008), A reinforcement learning model for supply chain ordering management: An application to the beer game, *Decision Support Systems*, 45(4), 949-959.

Chen, X., & Simchi-Levi, D. (2004b), Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The infinite horizon case, *Mathematics of Operations Research*, 29(3), 698-723.

Chen, B., Simchi-Levi, D., Wang, Y., & Zhou, Y. (2022), Dynamic pricing and inventory control with fixed ordering cost and incomplete demand information, *Management Science*, 68(8), 5684-5703.

Chen, Y., Mehrotra, P., Samala, N. K. S., Ahmadi, K., Jivane, V., Pang, L., ... & Pleiman, S. (2021), A multiobjective optimization for clearance in walmart brick-and-mortar stores, *INFORMS Journal on Applied Analytics*, 51(1), 76-89.

Dabérius, K., Granat, E., & Karlsson, P. (2019), Deep execution-value and policy based reinforcement learning for trading and beating market benchmarks, *Available at SSRN 3374766.*

Dai, J. G., & Gluzman, M. (2022), Queueing network controls via deep reinforcement learning, *Stochastic Systems*, 12(1), 30-67.

Das, T. K., Gosavi, A., Mahadevan, S., & Marchalleck, N. (1999), Solving semi-Markov decision problems using average reward reinforcement learning, *Management Science*, 45(4), 560-574.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016), Deep direct reinforcement learning for financial signal representation and trading, *IEEE Transactions on Neural Networks and Learning Systems*, 28(3).

Denil, M., Agrawal, P., Kulkarni, T. D., Erez, T., Battaglia, P., & De Freitas, N. (2016), Learning to perform physics experiments via deep reinforcement learning, arXiv:1611.01843.

Doya, K., Samejima, K., Katagiri, K. I., & Kawato, M. (2002), Multiple model-based reinforcement learning, *Neural Computation*, 14(6), 1347-1369.

Dulac-Arnold, G., Mankowitz, D., & Hester, T. (2019), Challenges of real-world reinforcement learning, arXiv:1904.12901.

Eilers, D., Dunis, C. L., von Mettenheim, H. J., & Breitner, M. H. (2014), Intelligent trading of seasonal effects: A decision support algorithm based on reinforcement learning, *Decision Support Systems*, 64, 100-108.

Finn, C., Abbeel, P., & Levine, S. (2017, July), Model-agnostic meta-learning for fast adaptation of deep networks, In *International Conference on Machine Learning (pp. 1126-1135)*, PMLR.

Fischer, T. G. (2018), Reinforcement learning in financial markets-a survey (No. 12/2018), *FAU Discussion Papers in Economics.*

Gershman, S. J., & Daw, N. D. (2017), Reinforcement learning and episodic memory in humans and animals: an integrative framework, *Annual Review of Psychology*, 68, 101-128.

Gosavi, A. (2004), Reinforcement learning for long-run average cost, *European Journal of Operational Research*, 155(3), 654-674.

Hacioglu, U., Chlyeh, D., Yilmaz, M. K., Tatoglu, E., & Delen, D. (2021), Crafting performance-based cryptocurrency mining strategies using a hybrid analytics approach, *Decision Support Systems*, 142, 113473.

Hendricks, D., & Wilcox, D. (2014, March), A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution, In *2014 IEEE Conference on CIFEr (pp. 457-464)*, IEEE.

Heskes, T. (1996), Practical confidence and prediction intervals, *Advances in Neural Information Processing Systems*, 9.

Howard, R. A. (1960), Dynamic programming and markov processes.

Hwang, J. G., & Ding, A. A. (1997), Prediction intervals for artificial neural networks, *Journal of the American Statistical Association*, 92(438), 748-757.

Jeong, G., & Kim, H. Y. (2019), Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning, *Expert Systems with Applications*, 117, 125-138.

Jiang, Z., & Liang, J. (2017, September), Cryptocurrency portfolio management with deep reinforcement learning, In *2017 Intelligent Systems Conference (IntelliSys)* (pp. 905-913). IEEE.

Jiang, Z., Xu, D., & Liang, J. (2017), A deep reinforcement learning framework for the financial portfolio management problem, arXiv:1706.10059.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998), Planning and acting in partially observable stochastic domains, *Artificial Intelligence*, 101(1-2), 99-134.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., ... & Michalewski, H. (2019), Model-based reinforcement learning for atari, arXiv:1903.00374.

Kavousi-Fard, A., Khosravi, A., & Nahavandi, S. (2015), A new fuzzy-based combined prediction interval for wind power forecasting, *IEEE Transactions on Power Systems*, 31(1), 18-26.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2010), Lower upper bound estimation method for construction of neural network-based prediction intervals, *IEEE Transactions on Neural Networks*, 22(3), 337-346.

Kokkodis, M., & Ipeirotis, P. G. (2021), Demand-aware career path recommendations: A reinforcement learning approach, *Management Science*, 67(7), 4362-4383.

Kyle, A. S. (1985), Continuous auctions and insider trading, *Econometrica*, 1315-1335.

LeCun, Y., Bengio, Y., & Hinton, G. (2015), Deep learning, *Nature*, 521(7553), 436-444.

Li, C., Chu, M., Zhou, C., & Zhao, L. (2020), Two-period discount pricing strategies for an e-commerce platform with strategic consumers, *Computers & Industrial Engineering*, 147, 106640.

Li, X., Wang, J., & Sawhney, R. (2012), Reinforcement learning for joint pricing, lead-time and scheduling decisions in make-to-order systems, *European Journal of Operational Research*, 221(1), 99-109.

Li, B., Wang, J., Huang, D., & Hoi, S. C. (2018), Transaction cost optimization for online portfolio selection, *Quantitative Finance*, 18(8), 1411-1424.

Li, Z., Yu, H., Zhang, G., Dong, S., & Xu, C. Z. (2021), Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning, *Transportation Research Part C: Emerging Technologies*, 125, 103059.

Li, Y., Zheng, W., & Zheng, Z. (2019), Deep robust reinforcement learning for practical algorithmic trading, *IEEE Access*, 7, 108014-108022.

Liang, Y., Sun, Z., ... & Bai, P. (2022), Lenovo Schedules Laptop Manufacturing Using Deep Reinforcement Learning, *INFORMS Journal on Applied Analytics*, 52(1), 56-68.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015), Continuous control with deep reinforcement learning, arXiv:1509.02971.

Littman, M. L. (1994), Markov games as a framework for multi-agent reinforcement learning, In *Machine Learning Proceedings 1994* (pp. 157-163), Morgan Kaufmann.

Liu, X., Wang, W., Niyato, D., Zhao, N., & Wang, P. (2018), Evolutionary game for mining pool selection in blockchain networks, *IEEE Wireless Communications Letters*, 7(5), 760-763.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017), Multi-agent actor-critic for mixed cooperative-competitive environments, *Advances in Neural Information Processing Systems*, 30.

Marín, L. G., Cruz, N., Sáez, D., Sumner, M., & Núñez, A. (2019), Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks, *Expert Systems with Applications*, 119, 128-141.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June), Asynchronous methods for deep reinforcement learning, In *International Conference on Machine Learning (pp. 1928-1937)*, PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015), Human-level control through deep reinforcement learning, *Nature*, 518(7540), 529-533.

Moerland, T. M., Broekens, J., & Jonker, C. M. (2020), Model-based reinforcement learning: A survey, arXiv:2006.16712.

Nagabandi, A., Finn, C., & Levine, S. (2018), Deep online learning via meta-learning: Continual adaptation for model-based RL, arXiv:1812.07671.

Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006, June), Reinforcement learning for optimized trade execution, In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 673-680).

Ng, A. Y., & Russell, S. J. (2000, June), Algorithms for inverse reinforcement learning, *ICML* (Vol. 1, p. 2).

Ning, B., Lin, F. H. T., & Jaimungal, S. (2018), Double deep q-learning for optimal execution, arXiv:1812.06600.

Oda, T., & Joe-Wong, C. (2018, April), Movi: A model-free approach to dynamic fleet management, In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 2708-2716)*, IEEE.

Oh, W., Moon, J. Y., Hahn, J., & Kim, T. (2016), Research note—Leader influence on sustained participation in online collaborative work communities: A simulation-based approach, *Information Systems Research*, 27(2), 383-402.

O'Mahony, S., & Karp, R. (2022), From proprietary to collective governance: How do platform participation strategies evolve? *Strategic Management Journal*, 43(3), 530-562.

Oroojlooyjadid, A., Nazari, M., Snyder, L. V., & Taká č, M. (2021), A deep q-network for the beer game: Deep reinforcement learning for inventory optimization, *Manufacturing* & *Service Operations Management*.

Panico, C., & Cennamo, C. (2022), User preferences and strategic interactions in platform ecosystems, *Strategic Management Journal*, 43(3), 507-529.

Pendharkar, P. C., & Cusatis, P. (2018), Trading financial indices with reinforcement learning agents, *Expert Systems with Applications*, 103, 1-13.

Polydoros, A. S., & Nalpantidis, L. (2017), Survey of model-based reinforcement learning: Applications on robotics, *Journal of Intelligent* & *Robotic Systems*, 86(2), 153-173.

Qin, Z., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H., & Ye, J. (2020), Ride-hailing order dispatching at didi via reinforcement learning, *INFORMS Journal on Applied Analytics*, 50(5), 272-286.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J. F., Breazeal, C., ... & Wellman, M. (2019), Machine behaviour, *Nature*, 568(7753), 477-486.

Rishika, R., Kumar, A., Janakiraman, R., & Bezawada, R. (2013), The effect of customers' social media participation on customer visit frequency and profitability: an empirical investigation, *Information Systems Research*, 24(1), 108-127.

Rummery, G. A., & Niranjan, M. (1994), *On-line Q-learning using connectionist systems* (Vol. 37, p. 14), Cambridge, UK: University of Cambridge, Department of Engineering.

Sáez, D., Ávila, F., Olivares, D., Cañizares, C., & Marín, L. (2014), Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids, *IEEE Transactions on Smart Grid*, 6(2), 548-556.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015), Prioritized experience replay, arXiv:1511.05952.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June), Trust region policy optimization, In *International Conference on Machine Learning (pp. 1889-1897)*, PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017), Proximal policy optimization algorithms, arXiv:1707.06347.

Shrestha, D. L., & Solomatine, D. P. (2006), Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks*, 19(2), 225-235.

Shrivastava, N. A., Khosravi, A., & Panigrahi, B. K. (2015), Prediction interval estimation of electricity prices using PSO-tuned support vector machines, *IEEE Transactions on Industrial Informatics*, 11(2).

Shou, Z., Di, X., Ye, J., Zhu, H., Zhang, H., & Hampshire, R. (2020), Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning, *Transportation Research Part C: Emerging Technologies*, 111, 91-113.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016), Mastering the game of Go with deep neural networks and tree search, *Nature*, 529(7587), 484-489.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013), Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv:1312.6034.

Stine, R. A. (1985), Bootstrap prediction intervals for regression, *Journal of the American Statistical Association*, 80(392), 1026-1031.

Sutton, R. S. (1988), Learning to predict by the methods of temporal differences, *Machine Learning*, 3(1), 9-44.

Sutton, R. S. (1990), Integrated architectures for learning, planning, and reacting based on approximating dynamic programming, In *Machine Learning Proceedings 1990 (pp. 216-224)*, Morgan Kaufmann.

Sutton, R. S., & Barto, A. G. (2018), Reinforcement learning: An introduction, *MIT press.*

Valencia, F., Collado, J., Sáez, D., & Marín, L. G. (2015), Robust energy management system for a microgrid based on a fuzzy prediction interval model, *IEEE Transactions on Smart Grid*, 7(3), 1486-1494.

Van Hasselt, H., Guez, A., & Silver, D. (2016, March), Deep reinforcement learning with double q-learning, In *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).*

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., ... & Ba, J. (2019), Benchmarking model-based reinforcement learning, arXiv:1907.02057.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., ... & Botvinick, M. (2018), Prefrontal cortex as a meta-reinforcement learning system, *Nature Neuroscience*, 21(6), 860-868.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June), Dueling network architectures for deep reinforcement learning, In *ICML (pp. 1995-2003)*, PMLR.

Watkins, C. J., & Dayan, P. (1992), Q-learning, *Machine Learning*, 8(3), 279-292.

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020), Adaptive stock trading strategies with deep reinforcement learning methods, *Information Sciences*, 538, 142-158.

Xiong, Z., Liu, X. Y., Zhong, S., Yang, H., & Walid, A. (2018), Practical deep reinforcement learning approach for stock trading, arXiv:1811.07522.

Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020, October), Deep reinforcement learning for automated stock trading: An ensemble strategy, In *First ACM International Conference on AI in Finance* (pp. 1-8).

Yuan, X., Chen, C., Jiang, M., & Yuan, Y. (2019), Prediction interval of wind power using parameter optimized Beta distribution based LSTM model, *Applied Soft Computing*, 82, 105550.

Zhang, Z., Zohren, S., & Roberts, S. (2020), Deep reinforcement learning for trading, *Journal of Financial Data Science*, 2(2), 25-40.

Zheng, H., Li, D., & Hou, W. (2011), Task design, motivation, and participation in crowdsourcing contests, *International Journal of Electronic Commerce*, 15(4), 57-88.

Zhu, Y., Mottaghi, R., ..., & Farhadi, A. (2017, May), Target-driven visual navigation in indoor scenes using deep reinforcement learning, In *2017 IEEE-ICRA (pp. 3357-3364)*, IEEE.