ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

Real-time Urban Flood Management Using an Improved Genetic Algorithm-Assisted Model Predictive Control Approach

Zhimei Tang[1,*]

[1]College of Environmental Science and Engineering, Tongji University, Shanghai, China

[*] Corresponding author

Abstract

With the increasing severity of urban flooding and related disasters, dynamic management of urban drainage systems has become increasingly critical in addressing these challenges. Model Predictive Control (MPC), as an advanced real-time control technique, has shown considerable promise; however, its performance is highly dependent on the computational efficiency of the optimization algorithm. Traditional genetic algorithms (GA) often suffer from slow convergence and low computational efficiency, which limits the real-time application of MPC. To address these challenges, this study proposed a real-time control framework for urban drainage systems that integrates an improved genetic algorithm (IGA) with MPC to enable efficient and accurate system control. The proposed approach enhanced algorithmic convergence by introducing an adaptive crossover and mutation mechanism, while significantly improving computational efficiency via parallel computing techniques. The improved genetic algorithm is embedded within the MPC framework to generate dynamic control strategies. The effectiveness of the proposed method was validated through its application to a real-world urban drainage system. The results show that: (1) The genetic algorithm enhanced with an adaptive mechanism achieved faster convergence and demonstrated more stable convergence toward the optimal solution in the later stages of evolution. (2) The computational time of the improved GA was reduced from 305.01 minutes to 18.34 minutes under a 32-core processor environment, representing a 93.99% improvement in efficiency. (3) The MPC approach achieved an 18.93% improvement in flooding reduction, effectively mitigating the risk of urban flooding. This study provides an effective technical solution for the intelligent management of urban drainage systems, offering significant theoretical and practical value for enhancing urban flood resilience.

Keywords

Real-time control; Urban drainage system; Model predictive control; Improved genetic algorithm; Adaptive mechanism.

1. INTRODUCTION

Urban pluvial flooding has emerged as a pressing global issue, posing significant threats to public health and economic stability [1]. The expansion of impervious surfaces in urban areas, coupled with the intensifying impacts of climate change, has led to an increased frequency of extreme rainfall events and substantial disruption of the natural hydrological cycle. Traditional drainage systems are increasingly challenged by these evolving conditions. Therefore, there is an urgent need to develop adaptive and cost-effective control strategies capable of responding to complex and rapidly changing urban environments.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

Relying on the construction or expansion of physical infrastructure to enhance the treatment capacity of urban drainage systems entails high capital investment, long payback periods, and significant spatial constraints, rendering such approaches difficult to scale across urban areas. In contrast, non-structural interventions—such as establishing drainage monitoring networks, improving flood early-warning systems, and developing optimization-based operational scheduling models—have proven to be effective alternatives for enhancing system performance. With the growing complexity of urban water environment management, the dynamic behavior, multi-objective nature, and inherent uncertainties of drainage systems have become increasingly pronounced [2]. However, current planning and operational practices remain largely grounded in static designs and rule-based control, which can either underutilize available infrastructure, resulting in inefficiencies, or overload system capacity, leading to urban flooding and combined sewer overflows (CSOs). Therefore, it is essential to adopt dynamic control strategies that can fully exploit the potential of existing infrastructure while achieving key objectives such as CSO reduction and flood mitigation. A substantial body of empirical evidence and engineering practice has confirmed that Real-time control (RTC) provides a viable and effective approach for optimizing the performance of urban drainage systems [3-5].

Typically, the development of RTC strategies involves the application of either rule-based control (RBC) or optimization-based algorithms to fulfill local or global control objectives [6]. Model predictive control (MPC), an advanced optimization-driven real-time control technique, has garnered significant attention in recent years for its potential in urban drainage system management [7]. MPC operates on the principle of receding horizon optimization, wherein future system states are forecasted using a predictive model, and an optimal control sequence is determined by solving a constrained optimization problem. However, the performance of MPC is highly dependent on both the efficiency and quality of the optimization solver. This challenge is particularly pronounced in complex urban drainage systems characterized by multivariable interactions and numerous constraints, where rapidly and accurately solving the optimization problem becomes critical for the practical implementation of MPC [6].

With the advancement of artificial intelligence algorithms, a wide array of intelligent optimization techniques has been increasingly applied to the operational control of urban drainage systems. Among these, the genetic algorithm (GA), a representative population-based heuristic method, has been frequently employed for system optimization [8, 9]. However, conventional GA approaches often suffer from limitations such as slow convergence, susceptibility to local optima, and suboptimal computational efficiency, which hinder their broader application in real-world drainage system management [10].

To address the aforementioned challenges, this study proposed a real-time control framework for urban drainage systems that integrated an improved genetic algorithm with MPC, aiming to enhance the timeliness and adaptability of drainage system operations. The main contributions of this work are as follows: (1) A sigmoid-function-based adaptive crossover and mutation mechanism was introduced to improve the convergence behavior and solution quality of the genetic algorithm. (2) A parallel computing strategy was incorporated to optimize the overall computational architecture, thereby significantly enhancing computational efficiency. (3) The proposed improved genetic algorithm was embedded within the MPC framework to effectively manage urban pluvial flooding, enabling intelligent and efficient scheduling of the drainage system.

2. METHODOLOGY

2.1. Model Predictive Control Approach

Model predictive control (MPC) is a control strategy based on receding horizon optimization. Its fundamental principle lies in predicting the system's dynamic response over a future time horizon using a process model, upon which an optimization problem is formulated to determine the optimal control sequence. This approach integrates both predictive and feedback capabilities, enabling dynamic adjustments and allowing the system to maintain stable and near-optimal performance in the presence of uncertainties and disturbances. The MPC method is particularly well-suited for control scenarios involving pronounced time variability and constrained multivariable interactions, such as the coordinated scheduling of multiple facilities in urban drainage systems.

2.1.1. Main Components

In the practical implementation of MPC for urban drainage systems, the framework typically comprises three main components (Figure 1). The system model characterizes the hydraulic and hydrological behavior of the drainage network. The process model is responsible for forecasting the future evolution of the system states. Based on these predictions, the optimization model determines the optimal control strategy.

The first component of the MPC framework is the system model, which encompasses both system states and control variables [11]. The system states include hydraulic variables such as node water levels and pipe flow velocities, as well as hydrological conditions such as catchment soil moisture and surface runoff. In real-world applications, these states are typically monitored using real-time sensors. The control aspect of the system is realized through actuators, which receive and execute the control commands generated by the MPC at each control time step.

The second component of the MPC framework is the process model, which is used to simulate the future states of the urban drainage system over the prediction horizon [12]. This model takes the current system states—retrieved from the system model—as its initial condition and incorporates external disturbances such as rainfall as inputs to generate forecasts of future system responses. In this study, the process model is constructed using the Storm Water Management Model (SWMM), enabling high-fidelity simulation of the dynamic behavior of the drainage system.

The third component of the MPC framework is the optimization model, whose primary function is to evaluate the performance of various control strategies under predefined objectives and constraints, and to identify the optimal control solution. In this study, an improved genetic algorithm is incorporated into the optimization model to enhance search capability and computational efficiency, thereby enabling the generation of highly adaptive control strategies.

2.1.2. Simulation and Prediction

At each control time step t, the system model first updates the current system state based on the latest rainfall data and actuator settings. This updated state, along with forecasted rainfall and other future input information, is then fed into the process model. The process model utilizes this information, in conjunction with a candidate control sequence over the specified prediction horizon, to dynamically simulate the future evolution of the system states.

In this study, the constructed process model takes as input a state matrix representing the drainage system, which includes both the forecasted rainfall sequence and the control sequence of actuators. The model employs SWMM to simulate system behavior, enabling dynamic prediction of the system's evolution over the future control horizon. The final output consists of the projected flooding conditions within the drainage network. The inputs and outputs of the model are defined as follows:

$$y(t) = f\left(R, U_1, \ldots, U_i, \ldots, U_m\right) \tag{1}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

$$R = [r_t, \ldots r_{t+F}] \quad (2)$$

$$U_i = [u_{i,t}, \ldots, u_{i,t+F}] \quad (3)$$

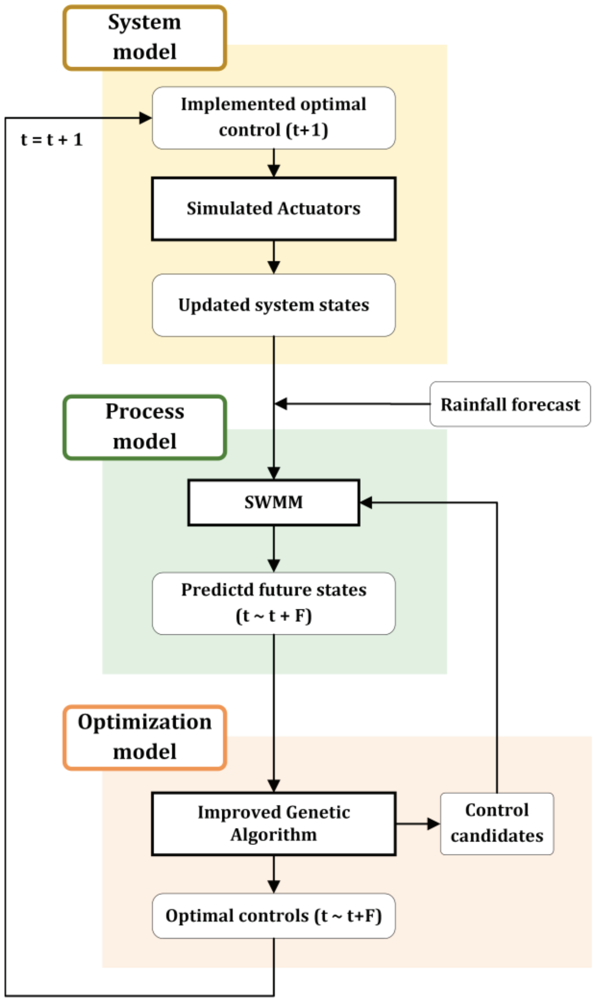

Figure 1. Main components of MPC for urban drainage systems.

where R is the rainfall sequence; U_1, ..., U_i, ..., U_m are the control state sequences of the actuators; r_t is the rainfall intensity at time step t; $u_{i,t}$ is the control state of actuator i at time step t; t is the current time step; F is the length of the future control horizon.

The output variable y of the process model is determined by the objective function specified within the optimization model. In this study, the optimization objective was centered on minimizing the total flood volume within the system.

2.1.3. Optimization Model

To achieve coordinated and optimal operation of the drainage system, thereby effectively mitigating urban flooding risk, an optimization-based scheduling model was developed in this study. The core function of this model is to identify the actuators that require regulation (e.g., gates, pump stations) and to determine the appropriate timing and manner of their operation. To fulfill the objective of flood control, the total system flooding volume is defined as the objective function of the optimization model, expressed as follows:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

$$\min\ y = \sum_{\tau=t}^{t+F} \sum_{j=1}^{N_V} V_{j,\tau} \tag{4}$$

where is the flooding volume at $V_{j,\tau}$ node j over the time step t; t is the current time step; F is the number of time steps within the control horizon; N_V is the total number of nodes in the drainage system.

In the optimization model, the decision variables to be determined correspond to the control actions of each actuator at every control time step within the control horizon. The combination of control horizon length and time step size determines the dimensionality of the control strategy, which directly affects the size of the search space for the Genetic Algorithm and, consequently, the computational complexity of the scheduling model. In this study, the control horizon ws set to 1 hour, with a control time step of 20 minutes.

The system under consideration consisted of seven actuators, including two storage tank valves (STV1 and STV2) and five pumps (PU1–PU5). Each control action was encoded as a binary digit, representing the closed (0) or open (1) status of the actuator. The MPC updated every 20 minutes and performed dynamic optimization over the upcoming 1-hour horizon, determining control actions for three future time steps for all seven actuators. As a result, the optimization problem involved a total of 21 binary decision variables.

2.2. Improved Genetic Algorithm

The genetic algorithm (GA) is an optimization technique inspired by the process of natural evolution. It simulates the evolutionary dynamics of biological populations through operations such as selection, crossover, and mutation. During the iterative process, individuals with low fitness are gradually eliminated, allowing those with higher fitness to emerge and dominate, ultimately leading to the identification of the optimal solution. In this study, the traditional GA was enhanced through two key modifications: (1) An adaptive crossover and mutation strategy was employed to improve solution quality and enhance convergence performance. (2) A parallel computing framework was introduced to accelerate the computational process. The overall procedure of the improved Genetic Algorithm is illustrated in Figure 2.

2.2.1. Adaptive crossover and mutation

The crossover probability P_c and mutation probability P_m are critical parameters that influence both the convergence speed and the exploration capability of Genetic Algorithms. When P_c and P_m are set to relatively high values, population diversity is enhanced, which helps prevent premature convergence; however, this may also disrupt high-quality genetic material within superior individuals. Conversely, setting these probabilities too low reduces population diversity, increasing the risk of the algorithm becoming trapped in local optima. To address this trade-off and improve both search capability and convergence performance, an adaptive strategy was employed in this study. This approach dynamically adjusted the values of P_c and P_m based on the evolutionary status of the population, thereby allowing the crossover and mutation processes to operate in a more rational and effective manner throughout different stages of evolution.

Several researchers have introduced the sigmoid function into the formulation of P_c and P_m, demonstrating notable improvements in algorithm performance [13]. Owing to its continuity, nonlinearity, and monotonicity, the sigmoid function plays a significant role in designing adaptive adjustment mechanisms for scheduling problems. The mathematical form of the sigmoid function is expressed as follows:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

$$\varphi(x) = \frac{1}{1+\exp(-x)} \tag{5}$$

The value of the function monotonically increases most often from 0 to 1.

According to the sigmoid function, the corresponding adaptive crossover operator (P_c) and mutation operator (P_m) were designed as follows:

$$P_c = \begin{cases} \dfrac{P_{c\max} - P_{c\min}}{1+\exp\left(5\times\dfrac{f^{'} - f_{avg}}{f_{\max} - f_{avg}}\right)} + P_{\mathrm{cmin}} & f^{'} \ge f_{avg} \\ P_{c\max} & f^{'} < f_{avg} \end{cases} \tag{6}$$

$$P_m = \begin{cases} \dfrac{P_{m\max} - P_{m\min}}{1+\exp\left(5\times\dfrac{f - f_{avg}}{f_{\max} - f_{avg}}\right)} + P_{\mathrm{mmin}} & f^{'} \ge f_{avg} \\ P_{m\max} & f^{'} < f_{avg} \end{cases} \tag{7}$$

where P_{cmax} and P_{cmin} are maximum and minimum probabilities of crossover operator, respectively; P_{mmax} and P_{mmin} are maximum and minimum probabilities of mutation operator, respectively; f' is the larger fitness value of the two individuals to be crossed; f is the fitness value of the individual to be mutated; f_{avg} and $f_{\max}$ are the average and maximum fitness value of current population, respectively.

In this study, the parameter settings for the adaptive crossover and mutation probabilities were determined based on the empirical insights provided by Zhu et al. [14], with appropriate adjustments made to suit the characteristics of the present optimization problem. The final values were set as follows: P_{cmax} = 0.95, P_{cmin} = 0.6, P_{mmax} = 0.1, P_{mmin} = 0.01.

When an individual's fitness is lower than the average fitness of the current population, higher crossover and mutation probabilities are applied to accelerate its evolutionary progress. Conversely, when an individual's fitness exceeds the population average, both crossover and mutation probabilities are gradually reduced as fitness increases. This allows high-performing individuals to maintain low crossover and mutation rates, thereby preserving superior genetic material and preventing the disruption of optimal gene structures.

2.2.2. Parallel Computing

Taking advantage of the inherent parallelism of genetic algorithms, a multi-threaded parallel computing technique was introduced to optimize the overall computational architecture and improve computational efficiency. The primary implementation process is as follows: First, by defining SWMM project objects, independent model input files (.inp) were generated to receive the new population parameters produced in each generation of the genetic algorithm. Then, SWMM was used to carry out simulation computations. Upon completion, the required simulation results were extracted from the output report files (.rpt), and the objective function values were calculated. Finally, these values were fed back to the genetic algorithm to proceed with the subsequent evolutionary operations.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

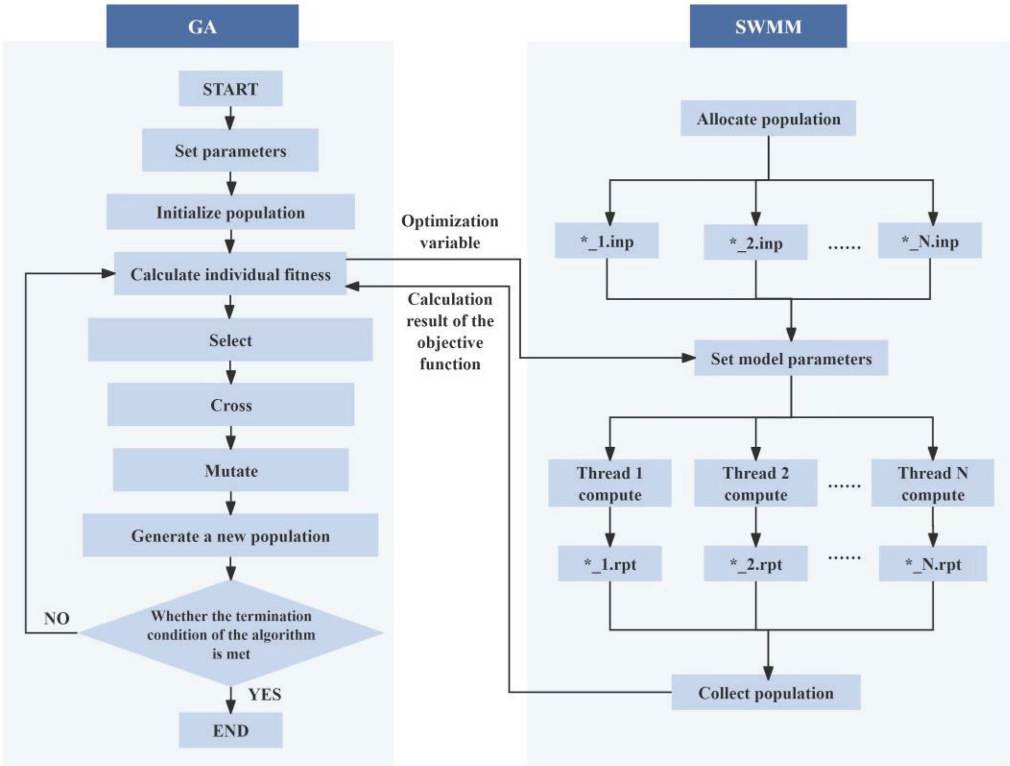

Figure 2. The flowchart of the improved genetic algorithm (IGA).

2.3. Real-time operation model

In this study, a real-time control model was developed based on the MPC framework, which has been widely applied in the real-time management of urban drainage systems.

The workflow of the MPC procedure is as follows:

(1) Acquire the latest operational state from the physical system.

(2) Input rainfall forecasts into the process model (SWMM) to predict future system states.

(3) Apply the optimization algorithm to determine the optimal control trajectory over the control horizon (t - t + F).

(4) Implement the optimized control action for the decision time t + 1.

(5) Advance to the next time step and update the system state.

(6) Repeat steps (1 - 5) until the end of the control process.

3. CASE STUDY

In this study, a drainage area in an eastern city of China was selected as the research object (Figure 3). The area covers approximately 150 hectares, with land use types mainly including roads, rooftops, and green spaces. The study area adopts a separate stormwater drainage system, without considering sewage inflows. In the drainage system model built using SWMM, a total of 594 sub-catchments were divided, with 659 nodes and 714 pipes configured. Originally, the area was equipped with one stormwater pumping station and an outfall. With the adjustment of urban functional planning, two additional stormwater storage facilities are proposed to be constructed to collect initial runoff and alleviate local flooding. Under this background, the number of control facilities increases significantly, and it becomes urgent to develop a scientific control strategy to realize coordinated scheduling of storage tanks and the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

pumping station, thereby maximizing the system's flood control capacity and operational efficiency.

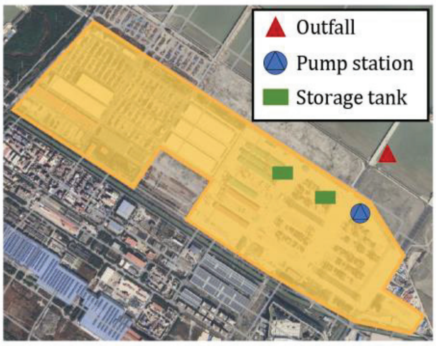

Figure 3. Overview of the study area.

The adopted rainfall data is a designed rainfall event generated based on the Chicago rainfall pattern, with a recurrence period of 10 years and a duration of 2 hours, as shown in Figure 4.

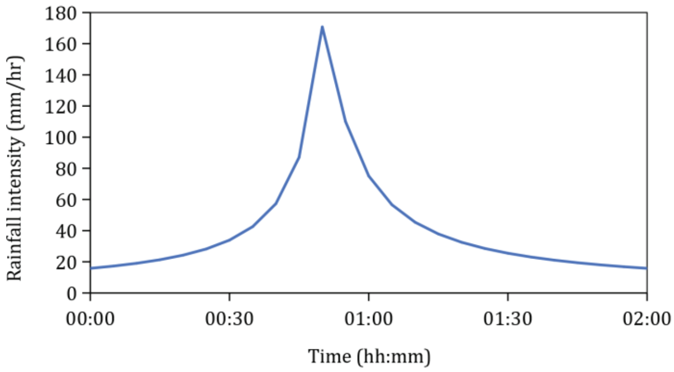

Figure 4. Rainfall intensity curve of the selected rainfall event.

4. RESULTS AND DISCUSSION

4.1. Analysis of Improved Genetic Algorithm

4.1.1. Convergence process

Figure i illustrates the convergence processes of different algorithms, where the improved genetic algorithm (IGA) demonstrated superior convergence performance. Specifically, the adaptive crossover and mutation mechanism in IGA accelerated convergence in the early stages, while the gradual reduction of crossover and mutation rates in later stages helped maintain solution stability. This design enabled rapid convergence while avoiding excessive perturbations. In terms of final solution quality, the IGA also outperformed the traditional GA, reflecting a more effective global search capability. The conventional GA, constrained by fixed crossover and mutation parameters, lacked the ability to adapt during different evolutionary phases. As a result, it tended to exhibit substantial random disturbances even in the later stages of evolution, which hampers stable convergence.

Overall, the introduction of adaptive crossover and mutation rates allowed the IGA to converge quickly in the early phase and remain stable in the later phase. Consequently, IGA

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202506_11(6).0012

achieved significantly better performance than the traditional GA in terms of convergence speed, stability, and final solution quality.

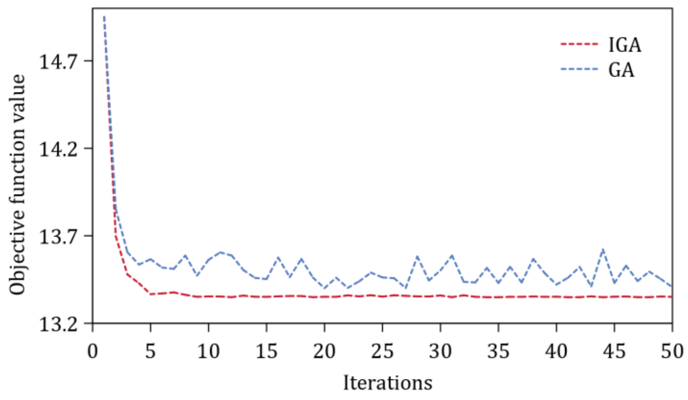

Figure 5. The convergence process of IGA and GA.

4.1.2. Computational Efficiency

Figure 6 presents the variation in computation time with respect to the number of processor cores. Parallel computing demonstrated a significant advantage in efficiency compared to serial computation. Under a 32-core configuration, the optimization process was completed in 18.34 minutes, representing a 93.99% reduction in computation time relative to the serial approach, which required 305.01 minutes.

As observed in Figure 6, computation time exhibited a clear decreasing trend with the increase in the number of processor cores, while speedup efficiency displayed a typical pattern of diminishing marginal returns. This phenomenon primarily resulted from the constraints of Amdahl's Law and the cumulative effects of parallel overhead. Specifically, certain inherently serial components of the genetic algorithm—such as selection operations, global best solution updates, and convergence checks—could not be parallelized, thereby imposing an upper limit on the theoretical speedup. Moreover, as the number of cores increased, parallel overheads, including inter-process communication, load balancing complexity and memory access conflicts, grew nonlinearly. When the growth in these overheads exceeded the performance gains from parallel computation, the speedup efficiency began to decline significantly, eventually reaching saturation and even leading to performance degradation in some cases.

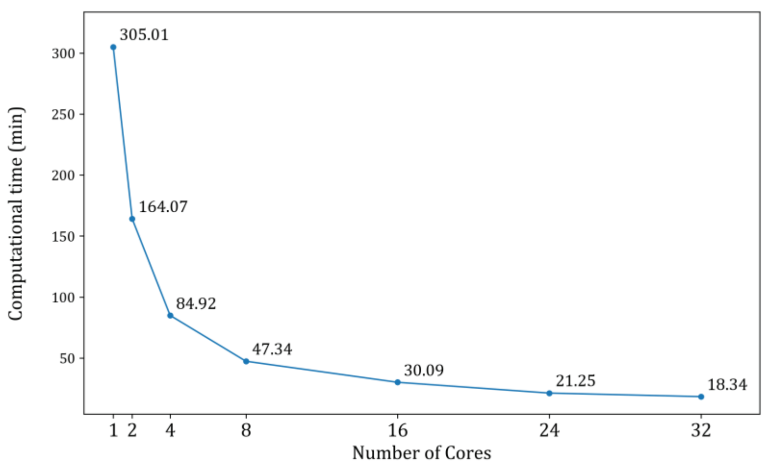

Figure 6. Computational time at different numbers of cores.

In addition, it is important to note that in practical applications of MPC, the time required to complete the optimization within a single control time step must be less than the length of the control interval itself. Otherwise, the optimal control actions for the next time step cannot be determined in time for execution. In this study, the control time step was set to 20 minutes. As shown in Figure 6, the optimization process, when accelerated using 32 computational cores, was completed in 18.34 minutes, which meets the requirements for real-time application.

4.2. Analysis of MPC Application Effectiveness

To illustrate the effectiveness of the MPC model, passive control and rule-based control (RBC) were introduced as comparative strategies.

In passive control, all actuators—such as valves and pumps—are fixed at predefined static settings, which are assumed to remain unchanged throughout the operation.

RBC is typically expressed in an "if-then" format. The control rules are established offline, i.e., they are predefined prior to rainfall events based on the experience of practitioners or statistical analysis of historical rainfall data. The performance of RBC is highly dependent on expert knowledge and lacks the flexibility to adapt to unforeseen events [15].

Figure 7 illustrates the total system flooding volume under the three control approaches. Compared to passive control, rule-based control achieved an 11.63% reduction in flooding volume. The MPC strategy further improved flooding reduction, achieving an 18.93% improvement over passive control and an 8.26% improvement over RBC. Among the three methods, MPC performed the best, demonstrating a significant enhancement in the flood mitigation capacity of the urban drainage system.

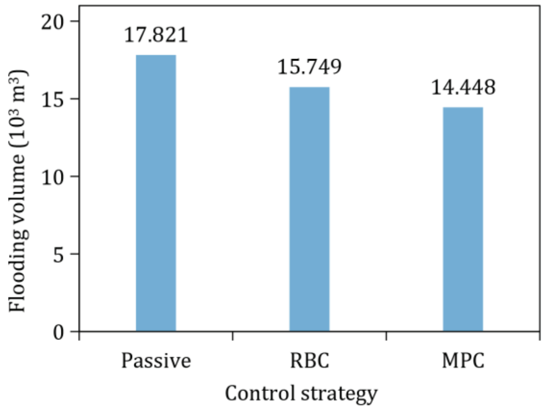

Figure 7. Comparison of flooding volume under passive control, RBC, and MPC.

The superior performance of the MPC strategy is primarily attributed to its ability to account for future system conditions within a defined control horizon when making decisions. This allows for more accurate anticipation and timely adjustment of control actions, resulting in improved flood volume management. Figure 8 shows the water level trajectories of storage tank ST2 under the three control strategies. It can be observed that RBC initiated inflow into the tank earlier than MPC; however, this occurred well before the rainfall peak. The premature use of storage capacity failed to alleviate flooding and instead occupied valuable buffer volume, which led to a larger total flooding volume. In contrast, MPC delayed the inflow into the storage tank, preserving capacity for the peak rainfall period. This effectively reduced the drainage pressure on the system and significantly decreased the total flood volume.

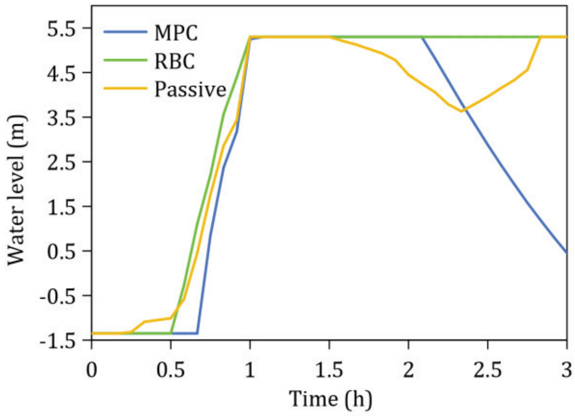

Figure 8. Water level in the storage tank ST2.

5. CONCLUSIONS

In this study, a real-time optimization and scheduling model for urban drainage systems was developed based on the MPC framework, and the optimization efficiency was enhanced through the incorporation of the improved genetic algorithm. The effectiveness and practicality of the proposed approach were validated through a case study. The main conclusions are as follows.

(1) The improved genetic algorithm proposed in this study achieved significant advancements in two key aspects. First, the introduction of a sigmoid-function-based adaptive crossover and mutation mechanism enabled IGA to dynamically adjust parameters according to individual fitness, resulting in marked improvements in convergence speed, stability, and solution quality compared to the conventional GA. Second, by incorporating multi-threaded parallel computing, the computation time was reduced from 305.01 minutes to 18.34 minutes under a 32-core processor environment, representing a 93.99% improvement in efficiency.

(2) The proposed MPC method demonstrated significant real-time control effectiveness. Compared to passive control, MPC achieved an 18.93% improvement in reducing system overflow volume; relative to RBC, the improvement was 8.26%. By accounting for the anticipated system evolution over the future control horizon, MPC enabled proactive and adaptive adjustment of control strategies, resulting in more effective and efficient real-time operation.

This study successfully developed a real-time control model for urban drainage systems by integrating an improved genetic algorithm with MPC, providing effective technical support for enhancing urban flood resilience and reducing flooding risks. However, this study did not account for the impact of uncertainties, including those arising from observed data, rainfall forecasts, and the process model. Future research could explore the performance of models under uncertain conditions.

REFERENCES

[1] X. Dong, L. K. Jiang, S. Y. Zeng, R. Guo, and Y. N. Zeng, Vulnerability of urban water infrastructures to climate change at city level, Resources Conservation and Recycling, Vol. 161 (2020).

[2] A. Casal-Campos, S. M. K. Sadr, G. T. Fu, and D. Butler, Reliable, Resilient and Sustainable Urban Drainage Systems: An Analysis of Robustness under Deep Uncertainty, Environmental Science & Technology, Vol. 52 (2018) No. 16, p. 9008-9021.

[3] J. G. Langeveld et al., Impact-based integrated real-time control for improvement of the Dommel River water quality, Urban Water Journal, Vol. 10 (2013) No. 5, p. 312-329.

[4] N. S. V. Lund, M. Borup, H. Madsen, O. Mark, and P. S. Mikkelsen, CSO Reduction by Integrated Model Predictive Control of Stormwater Inflows: A Simulated Proof of Concept Using Linear Surrogate Models, Water Resources Research, Vol. 56 (2020) No. 8.

[5] M. E. Jean, C. Morin, S. Duchesne, G. Pelletier, and M. Pleau, Optimization of Real-Time Control With Green and Gray Infrastructure Design for a Cost-Effective Mitigation of Combined Sewer Overflows, Water Resources Research, Vol. 57 (2021) No. 12.

[6] N. S. V. Lund, A. K. V. Falk, M. Borup, H. Madsen, and P. S. Mikkelsen, Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management, Critical Reviews in Environmental Science and Technology, Vol. 48 (2018) No. 3, p. 279-339.

[7] A. Castelletti et al., Model Predictive Control of water resources systems: A review and research agenda, (in English), Annual Reviews in Control, Vol. 55 (2023), p. 442-465.

[8] J. M. Sadler, J. L. Goodall, M. Behl, M. M. Morsy, T. B. Culver, and B. D. Bowes, Leveraging open source software and parallel computing for model predictive control of urban drainage systems using EPA-SWMM5, Environmental Modelling & Software, Vol. 120 (2019).

[9] G. Yi and W. H. Zou, Optimal operation of urban tidal drainage pumping station based on genetic algorithm coupled with head-water level successive approximation, (in English), Frontiers in Energy Research, Vol. 10 (2023).

[10] K. Hao, J. L. Zhao, Z. S. Li, Y. L. Liu, and L. Zhao, Dynamic path planning of a three-dimensional underwater AUV based on an adaptive genetic algorithm, (in English), Ocean Engineering, Vol. 263 (2022).

[11] L. X. Sun, J. Xia, and D. X. She, Integrating Model Predictive Control With Stormwater System Design: A Cost-Effective Method of Urban Flood Risk Mitigation During Heavy Rainfall, (in English), Water Resources Research, Vol. 60 (2024) No. 4.

[12] Y. Chen et al., Model predictive control and rainfall Uncertainties: Performance and risk analysis for drainage systems, Journal of Hydrology, Vol. 630 (2024).

[13] H. Lu and F. Qiao, An efficient adaptive genetic algorithm for energy saving in the hybrid flow shop scheduling with batch production at last stage, (in English), Expert Systems, Vol. 39 (2022) No. 2.

[14] J. Y. Zhu, J. C. Wang, and X. C. Li, Optimal scheduling of water-supply pump stations based on improved adaptive genetic algorithm, (in English), Proceedings of the 35th Chinese Control Conference 2016, (2016), p. 2716-2721.

[15] K. Miszta-Kruk, Reliability and failure rate analysis of pressure, vacuum and gravity sewer systems based on operating data, Engineering Failure Analysis, Vol. 61 (2016), p. 37-45.