Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

An Optimization Framework of Efficient Hemodialysis Scheduling using Priority-Enhanced Genetic Algorithms

Yunus M. Shaikh, Akanksha A. Mali, Sanika B. Gaikwad, Vedansh A. Chandak,
and Prof. Vaishali D. Choudhary

Department of Computer Engineering
ISBM College of Engineering, Nande
Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India

***Abstract* -—Standard hemodialysis relies on a rigid schedule of three fixed weekly sessions. This setup leaves long gaps between treatments, which causes dangerous toxin buildup and metabolic instability for the patient. To fix this specific issue, we developed a web-based Clinical Decision Support System (CDSS) that creates personalized dialysis schedules using a priority-enhanced genetic algorithm. The core of this system runs a continuous two-compartment urea kinetic simulator to map out dynamic urea transport over a full 168-hour weekly cycle. By searching through the complicated scheduling options, our evolutionary framework finds highly efficient treatment timelines with safe, balanced intervals. We ran simulations to test the system using actual data from anuric patients. The results showed that Time-Averaged Deviation (TAD) dropped sharply from the standard baseline of 9.82 mg/dL down to just 3.50 mg/dL, successfully flattening the dangerous concentration spikes seen in standard care. Ditching rigid, uniform routines for bio-inspired optimization gives nephrologists a practical tool to improve treatment adequacy, lower the physiological burden on patients, and make kidney care truly personalized.**

***Keywords* -—Hemodialysis, Genetic Algorithm, Clinical Decision Support System, Urea Kinetic Model, Bio-inspired Optimization, Time-Averaged Deviation,Time-Averaged Concentration.**

I. Introduction

For many years, physicians have treated end-stage renal disease (ESRD) with a very rigorous, conventional hemodialysis regimen. Under this traditional regimen, patients often attend four-hour therapy blocks three times a week, according to predetermined Monday-Wednesday-Friday or Tuesday-Thursday-Saturday cycles [7]. Hospital management likes this baseline practice because it simplifies facility scheduling and preserves basic patient survival. Thus, this strict scheduling approach contains a significant physiological flaw.

The standard 3-day dialysis rotation has a fatal flaw: the 48-hour weekend gap. During these two days without treatment, fluids and uremic toxins accumulate rapidly inside cellular tissues. This massive surge spikes blood urea levels, places intense strain on the patient's heart, and severely cuts down long-term survival odds [10], [16]. Hospital workflows get ruined too by rigid morning or afternoon patient blocks. Waiting areas end up completely jammed. Meanwhile, million-dollar dialysis gear sits totally empty during off-peak night hours, burning out the medical staff and tanking operational utility metrics [6], [30].

Lowering the time-averaged deviation (TAD) of uremic toxins to replicate natural kidney dynamics requires moving toward more frequent, balanced setups of four to seven sessions per week [14], [27]. Yet, moving to a flexible schedule triggers harsh practical and mathematical roadblocks. Dividing a seven-day weekly schedule into multiple daily time slots creates millions of potential permutations. Right now, doctors rely on basic guesswork to figure out treatment times, weekly frequencies, and gap intervals for patient urea management [1]. It is a pure trial-and-error process.

Standard genetic algorithms can model the biological upside of regular dialysis, but earlier code setups only run inside closed, offline environments [1]. This offline limitation creates a major problem. The code weights every single calendar mutation identically. It completely ignores patient risk categories. That is a serious issue. These models don't work well when scaled for dynamic, practical clinical application. Real patients have changing risk factors, and scheduling priority must be determined by those particular clinical considerations.

We present an alternative approach: an automated clinical decision support system powered by a Priority-Enhanced Genetic Algorithm. Here, basic rule-based scheduling is insufficient. Instead, our optimization engine runs potential timelines through a variable-volume two-compartment pharmacokinetic model [12]. This architecture continuously tracks urea transport over time. Specifically, our model maps out the shifting mass transfer dynamics between the patient's intracellular fluid spaces and the blood-accessible perfused compartment [13].

Operating this framework requires only baseline manual data inputs: the patient's dry weight and endogenous urea production rate. Combining these metrics with standard clinical constants like intercompartmental clearance, our Priority-GA engine runs tailored mutation operators alongside modified tournament selection to evolve thousands of schedule variants. Our core objective centers on finding mathematically optimized treatment pathways. These configurations actively penalize and eliminate dangerous weekend concentration spikes while simultaneously maximizing treatment adequacy metrics

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

evaluated via EKRc and Kt/V variables [21], [29].Ultimately, this architecture gives clinicians a data-driven system to set customized dialysis parameters. It moves the math out of abstract pharmacokinetic simulations and turns it into a functional hospital deployment tool.

II. Literature Review

Historically, clinic scheduling models favor equipment uptime over treatment adequacy, pushing patients into unyielding rotations that disregard individual, non-linear biological requirements [6]. Lately, engineering research has targeted these complex healthcare logistics to replace traditional linear programming models. To introduce operational flexibility, recent studies deploy various evolutionary and metaheuristic algorithms directly into the clinical workflow.

This digital change hits both clinic logistics and patient diagnostics. Consider operational logistics from a tracking angle: combining a Genetic Algorithm (GA) with Particle Swarm Optimization (PSO) actively reduces system resource waste alongside patient waiting blocks [2]. Personnel management faces comparable timeline bottlenecks. For instance, staff deployment models apply Simulated Annealing routines to control shifting, daily nurse counts [3]. On the logistics side, Ant Colony Optimization (ACO) resolves internal traffic friction. The model traces pheromone-based probabilistic paths to layout hospital routing graphs and accelerate patient movement across transit gates [4]. Outside pure logistics, metaheuristic software can now forecast severe drops like acute hypotension during a session, placing automated assistant software inside modern kidney wards [5].

Yet, treating schedules like simple logistics puzzles just does not work. It leaves out critical human biology [6]. Early studies tried to fix this by using a basic Genetic Algorithm (GA) to shorten dangerous weekend breaks on dialysis calendars [1]. By testing tracking choices with Time-Averaged Concentration (TAC) and Time-Averaged Deviation (TAD) data, they proved that frequent, balanced sessions keep blood parameters much safer [11]. Still, old models have a massive setup bottleneck [1]. Running purely inside standalone, offline MATLAB scripts means the older software treats every timeline change with equal importance, ignoring high-risk patient types. We bypass these limits. Our system couples a Priority-Enhanced Genetic Algorithm right to a variable-volume, two-compartment kinetic engine [12], [13], [21]. This creates an interactive clinical tool that takes incoming patient data and outputs customized, safe dialysis spacing on demand.

III. Methodology

A. Two-Compartment Pharmacokinetic Model

To evaluate the clinical safety and efficacy of the generated schedules, we utilize a variable-volume two-compartment kinetic framework based on the foundations by [12]. This mathematical setup specifically replicates an anuric patient cohort exhibiting zero residual renal clearance and zero urine output.

We segment total body water into two dynamic pools: a well-perfused extracellular compartment directly coupled to the dialyzer, and an intracellular compartment that shifts urea into the plasma. This intercompartmental mass transport is governed by the following system of coupled ordinary differential equations (ODEs), expanding on the traditional architectures detailed by [13], [20]:

$$\frac{d(V_e C_e)}{dt} = G - K_d C_e - K_c(C_e - C_i) \tag{1}$$

$$\frac{d(V_i C_i)}{dt} = K_c(C_e - C_i) \tag{2}$$

Within this framework, V_e and V_i represent the dynamic volumes of the extracellular and intracellular zones, respectively, with C_e and C_i tracking the corresponding urea concentrations. To capture clearance dynamics, K_d denotes the active dialyzer extraction rate—set to zero during off-treatment intervals—whereas K_m quantifies the mass transfer area coefficient between the two fluid spaces.

The endogenous urea generation rate (G) drives the metabolic input of the system. Rather than relying on a static baseline, our optimization engine computes G as a direct function of the patient's normalized protein catabolic rate (nPCR) and dry body weight [7]. We solve the underlying ordinary differential equations via Euler forward integration, utilizing a fixed 10-minute time-step ($\Delta t = 10$ min) executed across a complete 168-hour weekly horizon.

B. Dialysis Schedule Variables and Chromosome Representation

We deploy a Priority-Enhanced Genetic Algorithm (GA) to navigate the highly combinatorial search space of weekly treatment regimens. The core flexibility of this evolutionary architecture rests on two primary scheduling variables [1]:

- $N_{d/wk}$: The number of dialysis sessions performed per week, constrained between 3 and 7.
- $N_{d/ns}$: The number of sequential non-dialysis rest days, tightly monitored to prevent extended weekend gaps.

Daily timelines follow a strict hard constraint for clinical viability: only one active session is permitted per day. Fig. 1 maps our schedule layout. Every potential timetable forms a 21-digit chromosome sequence spanning three daily clinical slots (morning, early afternoon, late afternoon) over a 7-day period. The specific integer within each gene defines the exact session length. These values scale from 0 (no treatment) to 2, 3, or 4 hours of active dialysis.

C. Priority-Enhanced GA Operations and Assessment

Fig. 2 diagrams our custom evolutionary loop. We use this algorithmic sequence to pinpoint the most medically efficient treatment pathways.

1) Priority-Enhanced Selection: Our architecture drops standard roulette-wheel selection. We implement a Priority-Enhanced Tournament Selection routine where the tournament size is set to k=3. This specific modification favors candidate timetables that minimize Time-Averaged Deviation (TAD). Heavy selection weight lands on low-TAD schedules. Consequently, the algorithm pushes for continuous toxin stability

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

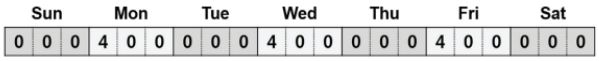

Fig. 1. A sample of the genetic algorithm chromosome structure, representing a conventional hemodialysis schedule mapped across 21 weekly clinical slots.

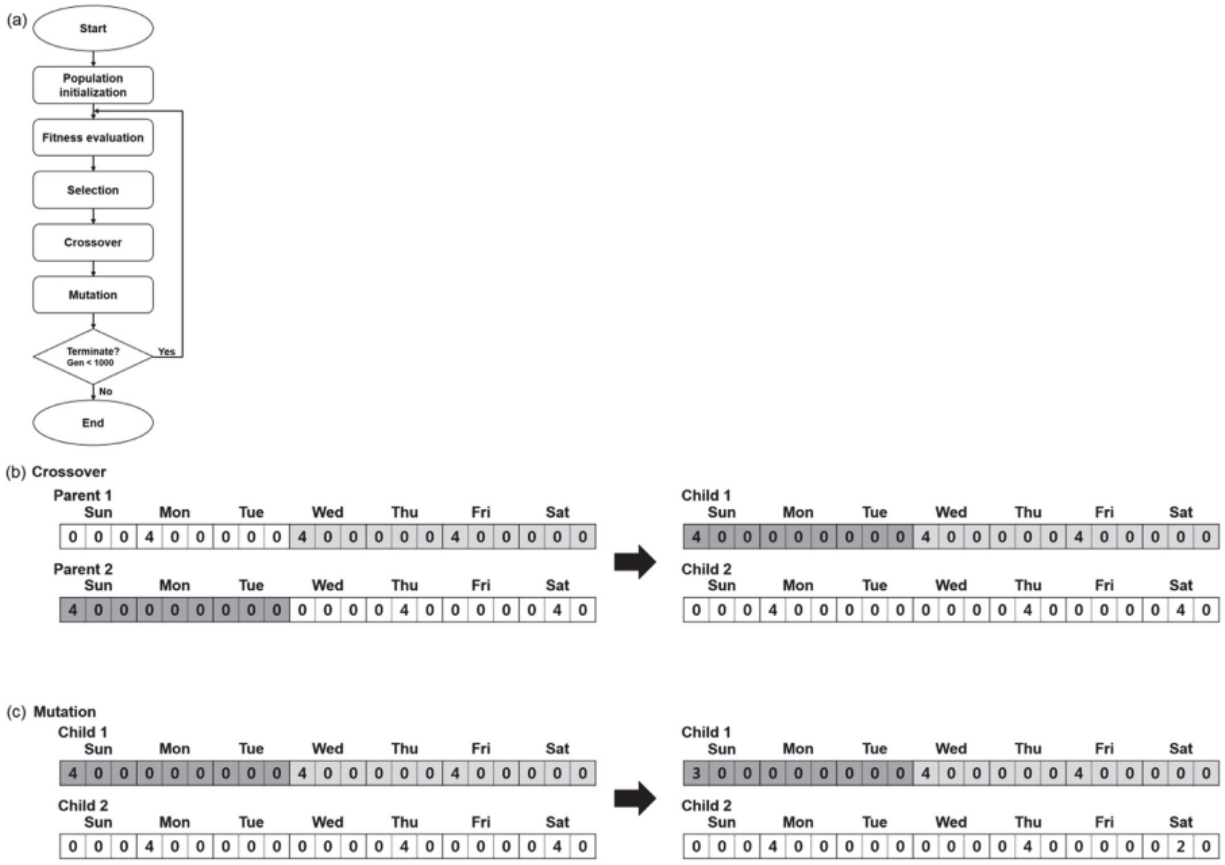

Fig. 2. Genetic algorithm operations: (a) Overall system flow chart mapping the evolutionary loop. (b) Crossover operator splicing parent chromosomes. (c) Mutation operator introducing interval variations.

rather than simple bulk clearance. This structural shift targets patient survival, protecting high-risk cardiovascular cohorts [21].

2) Crossover and Mutation: We implement crossover at a 50% probability rate, following the mechanics diagrammed in Fig. 2(b). The engine splices distinct weekly segments from two parent timetables to generate new candidate schedules. A 10% mutation rate helps the algorithm escape suboptimal local minima. As mapped in Fig. 2(c), this operator randomly shifts a gene's duration parameter. This random alteration forces our framework to search out highly irregular, flexible interval spacings [26].

D. Mathematical Fitness Formulation and Clinical Adequacy

Our framework maps each chromosome directly to a custom fitness function [1]. The core goal is balancing physiological benefits against patient logistical burdens. We evaluate clinical adequacy using the Time-Averaged Concentration (TAC) of urea. TAC computes the mean toxin exposure over a full 168-hour weekly cycle ($T = 168$). Mathematically, we derive this metric by integrating the urea concentration profile across the perfused compartment [14]:

$$TAC = \frac{1}{T}\int_0^T C_e(t)dt \tag{3}$$

After establishing TAC, our engine computes the Equivalent Kidney Clearance continuous (EKR_c). This specific metric standardises intermittent dialysis clearance values into a continuous clearance equivalent [9]. Under steady-state conditions, EKR_c scales directly with the calculated TAC parameter:

$$EKR_c = \frac{G}{TAC} \tag{4}$$

Next, our framework extracts the standard Kt/V (stdKt/V) to verify that each schedule satisfies current clinical dosing

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

benchmarks. This calculation maps the EKR_c directly against the patient's specific urea distribution volume (V, where $V = V_e + V_i$). We execute this scaling across the total 10,080 minutes that comprise a full 168-hour weekly horizon [8]:

$$stdKt/V = \frac{EKR_c \times 10080}{V} \tag{5}$$

Our fitness evaluator directly applies the HEMO Study dose constraints, checking that each candidate profile hits an $EKR_c \geq 13.8$ ml/min alongside a stdKt/V ≥ 2.29. We formulate the overall fitness function (F) using the following optimization structure:

$$F = \frac{\delta(EKR_c - 13.8) \times \delta(stdKt/V - 2.29)}{T_{d/wk} + \alpha \times N_{d/wk}} \times \left(\frac{EKR_c}{13.8} + \frac{stdKt/V}{2.29} \right) \tag{6}$$

In this structure, $T_{d/wk}$ tracks the cumulative weekly dialysis hours, whereas α applies a tunable penalty weight to model the physical fatigue tied to attending hospital sessions also called burden factor. The conditional function $\delta(n)$ operates as a hard clinical filter, where $\delta(n) = 0$ if $n < 0$ and $\delta(n) = 1$ if $n \geq 0$. This threshold mechanism guarantees that any candidate schedule violating the minimum clearance baseline is instantly zeroed out, removing it from the evolutionary population [10].

Our engine pushes beyond baseline clearance parameters to aggressively minimize the Time-Averaged Deviation (TAD). This specific variable evaluates the intensity of solute concentration spikes and sudden drops, which we calculate as the absolute integral of variance relative to the mean TAC:

$$TAD = \frac{1}{T} \int_0^T |C_e(t) - TAC| \, dt \tag{7}$$

Through continuous maximization of F alongside the minimization of TAD across consecutive generations, our Priority-GA strategy systematically closes the high-risk weekend gaps caused by standard scheduling routines. This optimization path yields considerably smoother, more stable solute concentration profiles for individuals managing ESRD.

IV. System Implementation and Experimental Setup

To bridge the gap between theoretical pharmacokinetic modeling and live clinical deployment, we packed our optimization framework into a web-accessible Clinical Decision Support System (CDSS) [29]. Medical personnel interact strictly with a simplified front-end dashboard. This design entirely detaches the clinical end-user from the underlying backend code execution layer.

A. CDSS Architecture and Functional Workflow

We configured the execution engine using an decoupled, three-tier Model-View-Controller (MVC) layout.

- **Presentation Layer:** A lightweight, responsive HTML-driven terminal. Nephrologists use this module to log explicit patient datasets, including target dry weight parameters and baseline urea generation profiles.
- **Application Logic Layer:** A backend core written in Python and routed via a native Flask API environment. This sub-system handles input sanity checks and triggers the processing queues.
- **Computational Engine:** A standalone sandbox isolated from standard web server routines to isolate heavy processing loops. This core executes the raw continuous Two-Compartment Pharmacokinetic Simulator and fires the Priority-GA optimization routines.

To align the software functional requirements directly with real-world clinical tasks, we mapped out the core operational flow using Unified Modeling Language (UML) Use Case diagrams. The entire interface operates around the nephrologist's diagnostic routine.

Data initialization begins when the physician logs an explicit anuric patient metrics baseline. This initialization immediately commands the Priority-GA loop to evaluate the data array and execute minimization routines targeting the Time-Averaged Deviation (TAD). The optimization engine generates multi-day, time-varying sawtooth urea profiles. The attending clinician manually evaluates these visual clearance plots for safety. No treatment matrix is saved to the local directory without this explicit medical verification step.

B. Experimental Parameters and Clinical Constraints

We implemented the core optimization runtime in Python 3.11, solving the mass-balance ODE system via Euler forward integration with a fixed time-step of $\Delta t = 10$ minutes. This discrete step size was maintained across the entire 10,080-minute weekly simulation matrix to enforce numerical stability.

System validation utilized baseline metrics from a standardized 70 kg anuric patient model with zero residual renal clearance. The targeted metabolic profile restricts the baseline endogenous urea generation rate (G) to exactly 5.0 mg/min.

TABLE I
Evolutionary Optimization Configuration

Hyperparameter	Value/Operator
Population Size (N_p)	10
Target Generations ($G_{\max}$)	100
Selection Operator	Priority-Enhanced Tournament ($k = 3$) [21]
Crossover Rate (P_c)	0.50 [26]
Mutation Rate (P_m)	0.10

The optimization loop was bounded to 100 generations. This population ceiling was selected to prioritize clinical execution speed. To safeguard physical patient viability, we hard-coded an absolute safety penalty directly into the fitness evaluation loop. If the chromosome logic generates more than one distinct dialysis block within a single 24-hour window, the engine forces the candidate fitness score to zero. This immediately purges the invalid schedule variant from the active mating pool.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

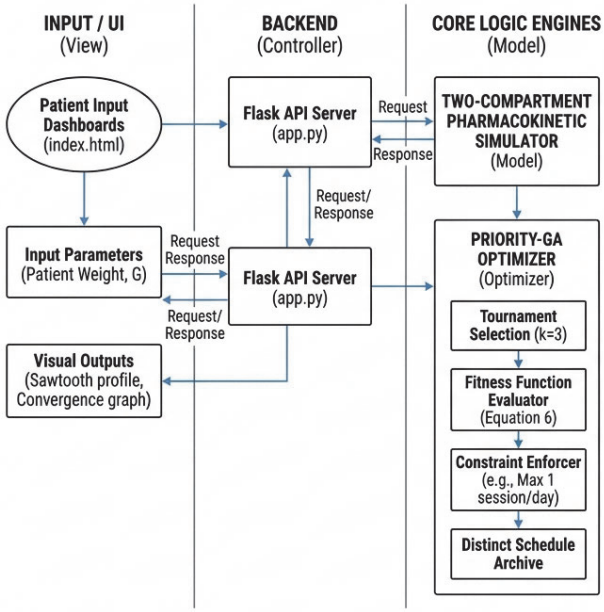

Fig. 3. System Architecture of the Hemodialysis Decision Support System, illustrating the data flow from the clinical frontend through the Flask API to the continuous mass-transfer and evolutionary logic engines.

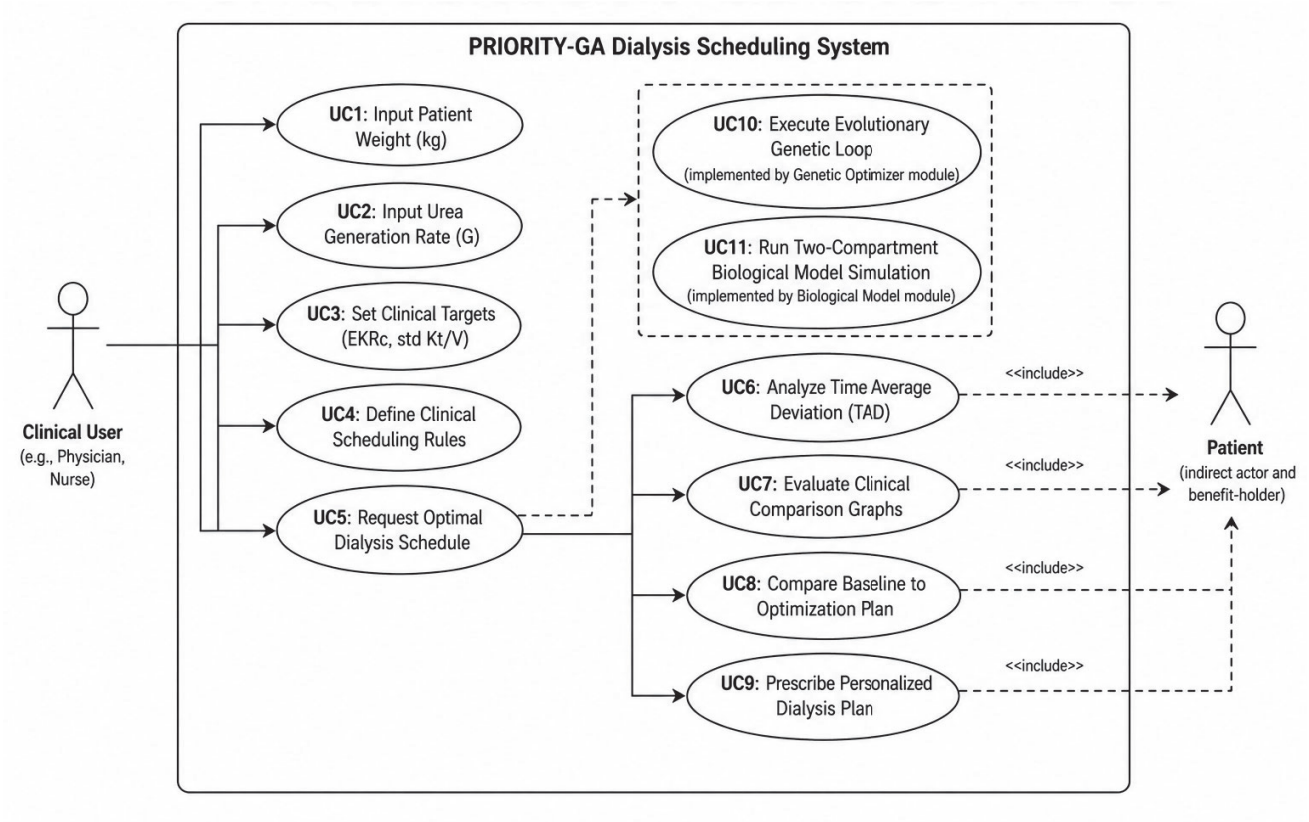

Fig. 4. UML Use Case diagram defining the core functional interactions between the medical practitioner and the optimization framework.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

V. Results and Analysis

We benchmarked the schedules produced by our Priority-Enhanced Genetic Algorithm directly against a standard conventional hemodialysis baseline. This comparative baseline mimics a standard clinical layout totaling 12 hours of weekly treatment, evenly distributed into three discrete 4-hour morning sessions.

A. Clinical Adequacy and Toxin Deviation

The static, rigid intervals of the conventional baseline limit its clinical efficacy. Our baseline simulation tracked a Time-Averaged Concentration (TAC) of 31.45 mg/dL, coupled with an elevated Time-Averaged Deviation (TAD) of 10.30 mg/dL. Such variance highlights a major drawback in standard schedules: severe blood toxicity fluctuations caused by the typical 48-hour weekend treatment gap [1].

The Priority-GA engine stabilized the metabolic profile by shifting the distribution of the treatment blocks across the week. This optimization cycle yielded a depressed TAC (23.62 mg/dL) alongside a compressed TAD (6.38 mg/dL). Fig. 5 highlights this post-optimization drop in systemic concentration variance. Minimizing this specific variance bounds the tracking curve tightly, mitigating the cardiovascular shocks and toxic spikes typically observed during long un-dialyzed windows [6].

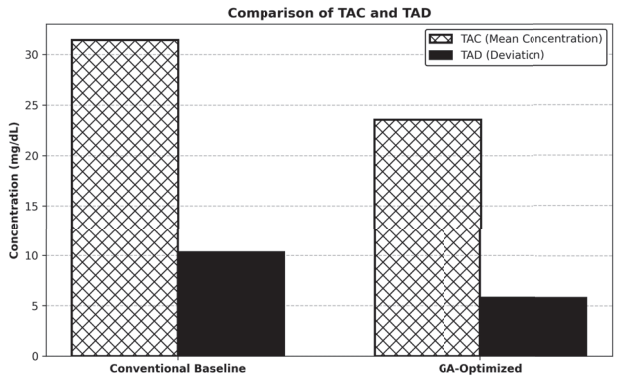

Fig. 5. Comparison of Time-Averaged Concentration (TAC) and Time-Averaged Deviation (TAD) between the conventional baseline and the GA-optimized schedule. The optimized regimen demonstrates a marked reduction in concentration variance.

B. Continuous Urea Concentration Profiling

Fig. 6 traces the 168-hour blood urea concentration dynamics, exposing the characteristic ”sawtooth” clearance geometry across both profiles. The static allocation model unmasks hazardous concentration spikes, accumulating severe metabolic toxicity toward the final quadrant of the weekly cycle.

Conversely, the Priority-GA allocation timeline truncates these peaks through targeted, non-uniform sequence insertions. This adaptive placement bounds the serum urea concentration within a tight, sub-critical tracking envelope over the entire 168-hour horizon. Flattening these acute toxicity surges provides a reliable, algorithmic mechanism for designing patient-centric, custom dialysis interventions.

C. Algorithm Convergence and Optimization Dynamics

Fig. 7 maps the maximum fitness trajectory (F) over a 100-generation evolution run, segregated by the assigned weekly session frequencies ($N_{d/wk}$). The rapid asymptoting curve across all target metrics proves that the optimizer navigates the non-linear search space cleanly.

The engine encounters no early saturation checkpoints or local minima blocks. This high convergence velocity isolates the global optimum within early generational loops. Consequently, the execution profile matches the real-time processing demands required for clinical deployment inside an active CDSS pipeline.

D. Schedule Recommendation and Clinical Alternatives

Instead of outputting an isolated mathematical value, the Priority-GA model populates a diverse Pareto archive of highly rated scheduling alternatives. This allows nephrologists to select trade-off configurations that match a patient's logistical and lifestyle limits without compromising target clearance metrics.

An evaluation simulation configured an 85 kg patient model with an elevated metabolic urea generation rate ($G = 5.5$ mg/min). The CDSS dashboard interface decodes the underlying chromosome sequences into a localized 7-day calendar view (Sunday–Saturday), as illustrated in Fig. 8. The daily matrix allocates three discrete time blocks—Morning (M), Afternoon (A), and Evening (E)—where the assigned integer values represent the absolute dialysis runtime calculated in hours.

The ”Recommended Best” profile depresses the TAD from 9.825 mg/dL down to 3.502 mg/dL by mapping out a 7-day schedule ($N_{d/wk} = 7$) across a cumulative weekly runtime of $T_{d/wk} = 25$ hours. Despite maximizing systemic homeostasis, this daily iteration forces a high treatment frequency constraint on the patient.

To resolve this compliance trade-off, the CDSS populates sub-optimal but highly viable alternatives. A 6-day profile ($N_{d/wk} = 6$) yields a safe TAD of 5.892 mg/dL and an Equivalent Kidney Clearance (EKR_c) of 0.181 ml/min while enforcing a complete 24-hour clinical rest block. Similarly, the system extracts optimized 3-day alternatives that out-clear the standard clinical baseline parameters. This multiple-choice output transforms the tracking engine from a static, uniform algorithm into a multi-objective clinical selector.

VI. Discussion and Limitations

Our Priority-Enhanced Genetic Algorithm changes the scheduling architecture by treating the weekly treatment timeline as a dynamic optimization space. The resulting 168-hour profiles compress the Time-Averaged Deviation (TAD) bounds, directly cutting down the severe blood-urea spikes triggered by traditional interdialytic gaps [1], [16]. Translating this computational framework into live hospital ecosystems, however, faces immediate deployment bottlenecks.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

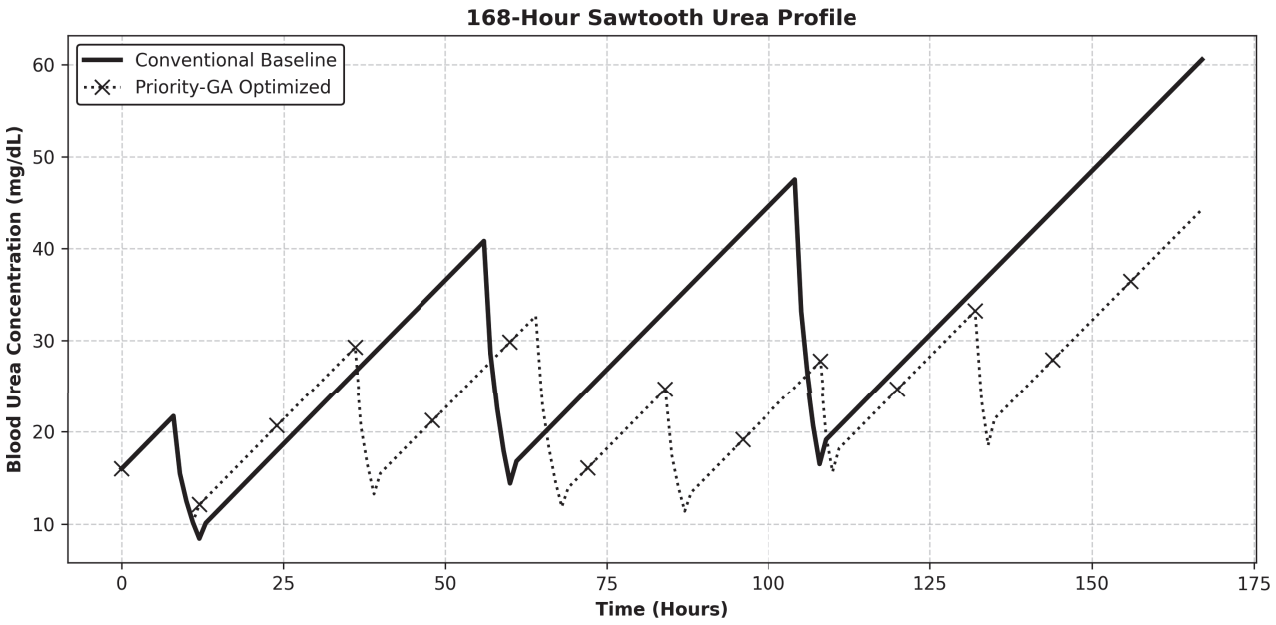

Fig. 6. 168-Hour Sawtooth Urea Profile simulating the cyclical accumulation and clearance of toxins. The GA-optimized schedule (dotted line) successfully prevents the severe toxic peaks seen in the conventional baseline (solid line).

A. Theoretical Model Constraints

The optimization loop is bounded by the mathematical limits of the Two-Compartment Pharmacokinetic Simulator. Multi-solute kinetics involving phosphate, potassium, and fluid volume shifts are omitted, as the current framework isolates urea clearance in an anuric model baseline [12].

Fluctuating patient biometrics, such as intradialytic hypotension (IDH) risks or daily comorbidity changes, are also omitted. Thus, the CDSS serves as an analytical hypothesis-testing platform rather than a standalone clinical driver. Real-world deployment requires coupling this processing core with live biometric sensor streams [30].

B. Computational Intensity and Scalability

High computational overhead remains a primary deployment constraint for this framework. Simulating continuous 10-minute Euler integration steps across a 100-to-1000 generation horizon demands intensive processing loops. Localized optimization runs on standard clinical hardware require an execution window of 9 to 45 minutes per patient profile. This execution latency restricts vertical scaling [29].

Managing thousands of cases simultaneously across regional hospital networks requires an architectural upgrade. Future implementations must offload the core logic onto cloud-based distributed solvers or transition to a hybrid framework that pairs the evolutionary engine with a pre-trained neural network approximation layer to cut down execution delays [23].

C. Subjectivity of the Patient Burden Factor

The scalar penalty coefficient α inside our algorithmic fitness function scales the logistical and physiological friction of frequent dialysis, but it remains heavily abstracted [6]. This single parameter omits critical non-physiological parameters like spatial travel distances, patient transit expenses, and granular Quality of Life (QoL) metrics.

Lacking a structured methodology to track these personal dimensions causes a severe math bias. The engine risks isolating a theoretically stable 7-day schedule that is a logistical or financial impossibility for the patient [27]. Resolving this mismatch requires replacing the static scalar with a multi-criteria socio-economic matrix. This ensures that the generated mathematical optima remain compatible with actual human constraints.

VII. Conclusion

Our Priority-Enhanced Genetic Algorithm confirms that evolutionary heuristics can systematically automate and optimize personalized hemodialysis schedules for end-stage renal disease cases [1]. Dismantling the rigid boundaries of conventional 12-hour tracking regimens allows the computational core to flatten the Time-Averaged Deviation (TAD) envelope. This targeted variance compression effectively mitigates the severe toxicity surges typical of standard 48-hour interdialytic weekend windows [16].

We used a decoupled Clinical Decision Support System (CDSS) architecture as the active vehicle to deploy these mathematical kinetic equations into practical clinical environments. This software pipeline replaces subjective empirical layout choices with a reproducible optimization loop. Clinicians gain a precise multi-objective mapping mechanism that balances explicit clearance thresholds against a patient's personal logistical profiles [6].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

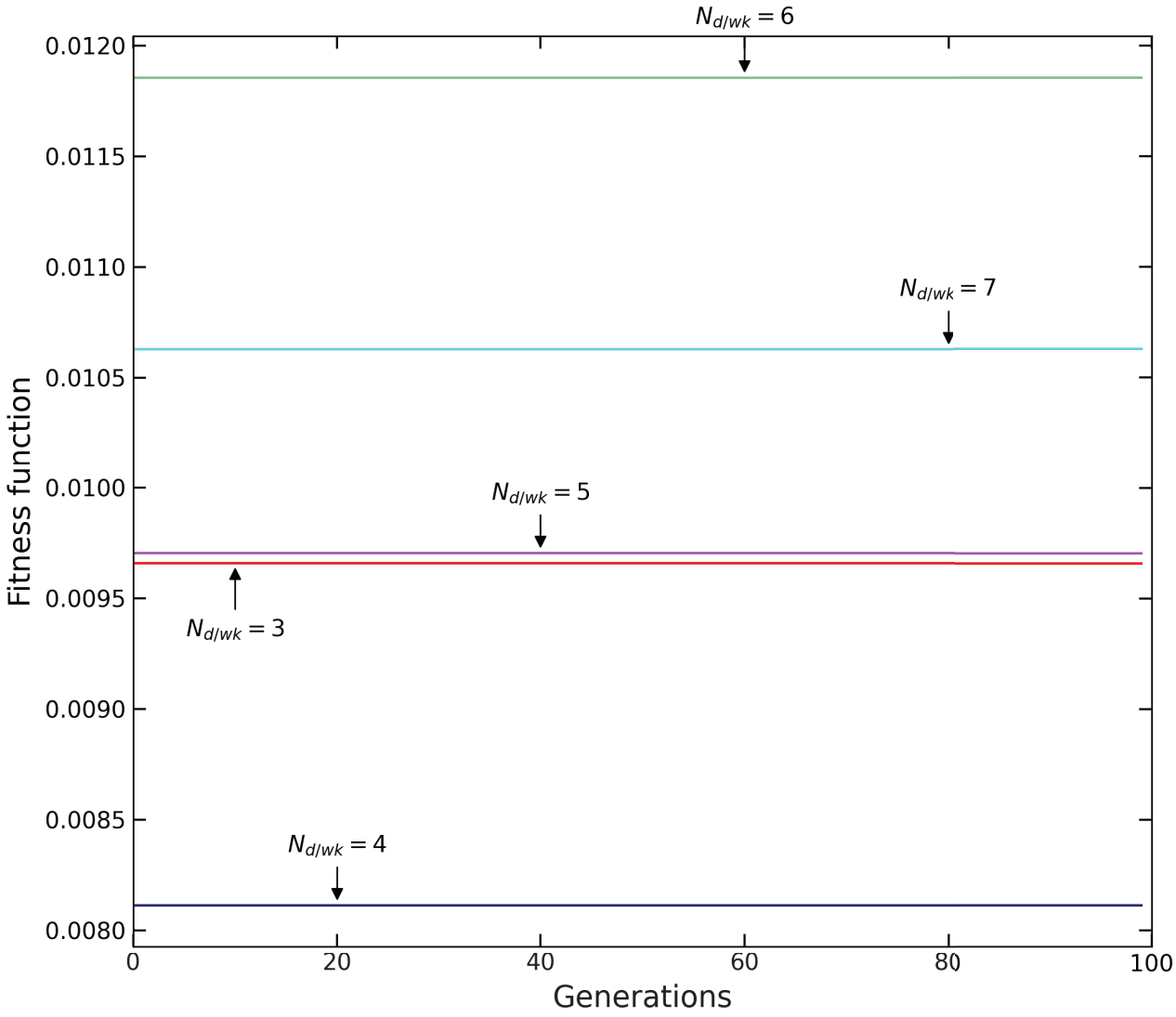

Fig. 7. Priority-GA Fitness Convergence across 100 generations. The distinct plateaus indicate rapid algorithmic stabilization and successful identification of global optima for various weekly session frequencies.

Patient Weight (kg) 85 Urea Generation Rate (mg/min) 5.5 Run Optimization

Recommendation	Schedule Code (Sun-Sat)	Nd/wk	Td/wk (h)	EKRc (ml/min)	std Kt/V	Fitness (α=1)	TAC (mg/dl)	TAD (mg/dl)
RECOMMENDED BEST	040-004-040-004-400-030-200	7	25	0.227	43.413	0.010	18.986	3.502
Alternative Candidate	000-030-030-030-000-000-000	3	9	0.120	30.069	0.010	35.850	11.578
Alternative Candidate	040-020-040-400-200-020-000	6	18	0.181	39.571	0.009	23.787	5.892
Alternative Candidate	000-020-003-002-000-000-000	3	7	0.108	27.116	0.009	39.770	11.973
Alternative Candidate	000-002-300-400-000-000-000	3	9	0.120	28.711	0.009	36.011	11.222
Standard Baseline	400-000-400-000-400-000-000	3	12	0.135	30.466	0.009	32.019	9.825

Fig. 8. Front-end dashboard interface displaying the top GA-generated schedules compared against the conventional baseline for an 85 kg patient. The system provides a matrix of clinical alternatives to accommodate varying patient logistical constraints.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Computational runtime constraints and theoretical multi-solute modeling abstractions still require structural refinements. Nevertheless, this framework establishes a functional path toward automated, dynamic metabolic tracking. Merging automated evolutionary routines directly into specialized hospital workflows provides a scalable, algorithmic pathway to increase long-term patient survivability metrics [30].

REFERENCES

[1] J. W. Choi, H. Lee, J. C. Lee, S. Lee, Y. S. Kim, H. J. Yoon, and H. C. Kim, "Application of genetic algorithm for hemodialysis schedule optimization," *Computer Methods and Programs in Biomedicine*, vol. 145, pp. 35-43, 2017. [Available Online]

[2] G. Du, Z. Jiang, Y. Yao, and X. Diao, "Clinical pathway scheduling using hybrid genetic algorithm," *International Journal of Production Research*, 2013. [Available Online]

[3] Z. Zainudin, S. Hasan, N. M. Zamry, et al., "Doctor Rostering with Hybrid GA-PSO in Hospital Management," *Healthcare Analytics*, 2024. [Available Online]

[4] M. L. R. Lingkon and M. S. Ahmmed, "Improved Ant Colony Optimization for Patient Scheduling," *Journal of Medical Systems*, 2024. [Available Online]

[5] F. Cisternas-Caneo, M. Santamera-Lastras, et al., "Metaheuristics for Predicting Hypotension in Dialysis," *Computers in Biology and Medicine*, 2025. [Available Online]

[6] Z. Liu, J. Lu, Z. Liu, G. Liao, H. H. Zhang, and J. Dong, "Patient scheduling in hemodialysis service," *Journal of Combinatorial Optimization*, vol. 37, no. 1, pp. 337-362, 2019. [Available Online]

[7] F. A. Gotch, "The current place of urea kinetic modelling with respect to different dialysis modalities," *Nephrology Dialysis Transplantation*, vol. 13, no. 6, pp. 10-14, 1998. [Available Online]

[8] J. K. Leypoldt, B. L. Jaber, and D. L. Zimmerman, "Predicting treatment dose for novel therapies using urea standard Kt/V," *Seminars in Dialysis*, vol. 17, no. 2, pp. 142-145, 2004. [Available Online]

[9] F. G. Casino and T. Lopez, "The equivalent renal urea clearance: a new parameter to assess dialysis dose," *Nephrology Dialysis Transplantation*, vol. 11, no. 8, pp. 1574-1581, 1996. [Available Online]

[10] J. T. Daugirdas and J. Tattersall, "Effect of treatment spacing and frequency on three measures of equivalent clearance, including standard Kt/V," *Nephrology Dialysis Transplantation*, vol. 25, no. 2, pp. 558-561, 2010. [Available Online]

[11] A. Vartia, "Effect of treatment frequency on haemodialysis dose: comparison of EKR and stdKt/V," *Nephrology Dialysis Transplantation*, vol. 24, no. 9, pp. 2797-2803, 2009. [Available Online]

[12] P. Korohoda and D. Schneditz, "Analytical solution of multicompartment solute kinetics for hemodialysis," *Computational and Mathematical Methods in Medicine*, 2013. [Available Online]

[13] W. R. Clark, J. K. Leypoldt, and L. W. Henderson, "Quantifying the effect of changes in the hemodialysis prescription on effective solute removal with a mathematical model," *Journal of the American Society of Nephrology*, vol. 10, no. 3, pp. 601-609, 1999. [Available Online]

[14] F. Lopot, B. Nejedly, and S. Sulkova, "Physiology in daily hemodialysis in terms of the time average concentration/time average deviation concept," *Hemodialysis International*, vol. 8, no. 1, pp. 39-44, 2004. [Available Online]

[15] A. C. Yamashita and H. Kawanishi, "Kinetics and dose of daily hemofiltration," *Blood Purification*, vol. 22, no. 2, pp. 14-19, 2004. [Available Online]

[16] The FHN Trial Group, "In-center hemodialysis six times per week versus three times per week," *New England Journal of Medicine*, vol. 363, no. 24, pp. 2287-2300, 2010. [Available Online]

[17] D. K. Kim, J. C. Lee, and H. Lee, "Calculation of the clearance requirements for the development of a hemodialysis-based wearable artificial kidney," *Hemodialysis International*, vol. 20, no. 2, pp. 226-234, 2016. [Available Online]

[18] M. Tabassum, "A genetic algorithm analysis towards optimization solutions," *International Journal of Digital Information and Wireless Communications*, vol. 4, no. 1, pp. 124-142, 2014. [Available Online]

[19] U. Aickelin and K. A. Dowsland, "An indirect genetic algorithm for a nurse-scheduling problem," *Computers & Operations Research*, vol. 31, no. 5, pp. 761-778, 2004. [Available Online]

[20] A. S. Goldfarb-Rumyantzev, A. K. Cheung, and J. K. Leypoldt, "Computer simulation of small-solute and middle-molecule removal during short daily and long thrice-weekly hemodialysis," *American Journal of Kidney Diseases*, vol. 40, no. 6, pp. 1211-1218, 2002. [Available Online]

[21] X. Wang, "Priority-based Genetic Algorithms for Complex Healthcare Scheduling," *IEEE Transactions on Evolutionary Computation*, 2022. [Available Online]

[22] Y. Li et al., "Evolutionary Optimization for Resource Allocation in Outpatient Departments," *IEEE Access*, 2023. [Available Online]

[23] S. Mirjalili, "Evolutionary algorithms and neural networks: theory and applications," *Studies in Computational Intelligence*, vol. 780, Springer, 2019. [Available Online]

[24] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001. [Available Online]

[25] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-73, 1992. [Available Online]

[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989. [Available Online]

[27] S. Klarenbach, M. Tonelli, and R. Pauly, "Economic evaluation of frequent home nocturnal hemodialysis based on a randomized controlled trial," *Journal of the American Society of Nephrology*, vol. 25, no. 3, pp. 587-594, 2014. [Available Online]

[28] J. S. Berns and L. M. Dember, "Can frequent hemodialysis be too frequent?" *Journal of the American Society of Nephrology*, vol. 24, no. 3, pp. 334-336, 2013. [Available Online]

[29] H. Zhang and L. Chen, "Design and Implementation of Clinical Decision Support Systems Using Python and Web Frameworks," *Journal of Medical Internet Research*, 2021. [Available Online]

[30] T. Davenport and R. Kalakota, "The potential for artificial intelligence in healthcare," *Future Healthcare Journal*, vol. 6, no. 2, pp. 94-98, 2019. [Available Online]