Optimal Control of an Ultraviolet Water Disinfection System

Viorel MINZU 1,*, Saïd RIAHI 1,2 and Eugen RUSU 3

1 Control and Electrical Engineering Department, “Dunarea de Jos” University, 800008 Galati, Romania; riahizsaid@gmail.com
2 UR-LAPER, Faculty of Sciences of Tunis, University of Tunis El Manar, 1068 Tunis, Tunisia
3 Mechanical Engineering Department, “Dunarea de Jos” University, 800008 Galati, Romania; evcrusu@gmail.com
* Correspondence: viorel.minzu@ugal.ro

Featured Application: Our work can be applied to the optimal control of ultraviolet water disinfection systems. The paper proposes a bi-criteria optimal problem statement, an algorithm to solve the problem, a detailed implementation using a metaheuristic algorithm, and a comprehensive analysis of the results. Our presentation of these findings is slightly dependent on the system’s dynamic model. Therefore, they can be applied mutatis mutandis to different ultraviolet water disinfection systems.

Abstract: The ultraviolet water disinfection process has numerous advantages in comparison with other water treatments. This study deals with this process’s optimal control and presents all its stages, from problem statement to analysis of results. We consider a typical disinfection water system and data presented in the literature. In this case, the control inputs are the water flow rate and the UV light intensity. A nonlinear dynamic model for the UV disinfection system is constructed and generates a method to calculate the bacterial reduction. An optimal control problem (OCP) is stated using the dynamic model and two optimal criteria: the disinfection and energy criterion. The optimal solutions are yielded by a metaheuristic algorithm based on hybrid topology particle swarm optimization. The algorithm determines the optimal flow rate and light intensity for an imposed bacterial reduction. The implemented program was carried out for a wide range of input values for comprehensive analysis. The analysis of the results establishes when an optimal solution exists and the extent to which the disinfection or energy criterion is fulfilled. The method to calculate the bacterial reduction, OCP statement, optimization algorithm, and results’ analysis may be considered findings of this work.

Keywords: ultraviolet water disinfection process; nonlinear model; optimal control; multi-criteria optimization algorithm; particle swarm optimization; hybrid topology; data analysis

1. Introduction

Disinfection of drinking water is clearly a highly important sanitary objective. This treatment may be accomplished through chemical or physical processes. Among these, UV treatment is widely used due to its numerous advantages: The odor, PH, taste, and other properties of the water are not affected; it does not affect minerals in water; it needs relatively small exposure time to UV radiation; it does not add chemicals to the water and does not yield by-products (or yields a very small amount, see [1]); and, a simple operational procedure is required for the UV system. The most important advantage of UV disinfection of water is its effectiveness against wide-ranging pathogens, including protozoans and chlorine-resistant viruses. Comprehensive action against viruses may be an essential quality that compels the use of UV disinfection. The UV rays damage the genetic components of viruses, parasites, bacteria, or molds, thus preventing reproduction, i.e., they are inactivated. UV disinfection of drinking water is widely used and has been the subject of many research studies because of these advantages.
UV disinfection is a complex process whose model requires the use of multidisciplinary, scientific aspects. Previous studies have applied chemical and physical approaches (see [2,3]) and generated models that allow mainly qualitative analysis. When authors model the process and propose equipment to perform disinfection, they adopt a discourse based on biological and engineering considerations (see [4–6]). Numerical techniques such as computational fluid dynamics are also used to jointly solve and simulate the flow, radiation, and disinfection equations (see for example [7,8]).

To systematically construct a UV system model, the paper [9] proposes a bond graph approach that can cope with the nonlinear characteristic of a pilot disinfection water system. The resulting model allows parameters and the water disinfection kinetics to be computed. Other studies address the static optimization of the reactor’s configuration through the evaluation of potential performance indexes (see, for example [10]).

This study regards the optimal control of a UV disinfection system (UVDS) that places our work at the optimization level, namely above the basic automation level (if any). The latter aims to automatically adjust the values of the UVDS control inputs to equal the setpoint values to ensure the inactivation of pathogens. The UVDS control inputs are, in most cases, the water flow rate and the UV light intensity. Sometimes these physical parameters are manually controlled because the basic automation level does not exist.

The optimization level has just the task of establishing the setpoint values jointly for the control inputs. An optimization algorithm considers optimum criteria (e.g., minimizing energy consumption or maximizing the water quality) and searches for the optimal values using a specific technique to fulfil this task.

To study the optimal control of a UVDS, the work presented in this study tackled the following mandatory matters: dynamic model of the UVDS; statement of the optimization problem; optimization algorithm; and examination of optimal solutions.

The dynamic model of the UVDS depends obviously on the equipment’s structure and its technical performances. To obtain a dynamic model for a specific UVDS is a complex task related to our work’s main objective: the disinfection model. The task is yet more complex in the case of flow reactors (like in our paper). An integrated UV disinfection process model should involve the modelling of fluence rate, flow field, and microbial inactivation kinetics. Computational fluid dynamics has been used to simulate the UV disinfection process (see [8,11]). Usually, this methodology is used by research specialists during scientific investigations. Certain specialists consider that it is not adequate for practitioners working with water systems because of its lack of flexibility (it is tailored for a specific single case). As a result, theoretical models have been developed that concern the variation of both the fluid velocity field and UV radiation intensity according to the distance from the radiation source (see [7]). These models can be used in the design of UV systems and to assist CFD simulations. They can be used to calculate, for example, the received radiation dose with good accuracy at any point in the irradiation room at a certain time. The models consider many parameters of the UVDS and the water quality (e.g., transmittance).

In conjunction with a kinetic model and validation and calibration measurements, these models can determine the bacterial reduction (abatement) that should be inside certain limits. For large UV systems, this approach seems to be unavoidable.

When the UVDS has a small irradiation room, and some constraints are met (including water quality), practical identification can result in a dynamic model whose output variable is bacterial reduction (see [5]). The control inputs may be, for example, the water flow rate and the UV light intensity of the source because they achieve a control action. Data obtained by conducting a series of experimental measurements are used with an identification procedure. Usually, specific software assists the identification method to construct the dynamic model. The result is a pair of transfer functions corresponding to the two influence channels.

The current study adopted a typical disinfection water system, described in [5]. In addition to the equipment’s technical parameters, we also used data presented in the paper
mentioned above, which proposed a linear dynamic model. For this UVDS, Section 2 presents a nonlinear dynamic model, which was used to generate a method that calculates the bacterial abatement (reduction).

Optimal control requires a well-stated optimization problem, which includes components that must be defined: the dynamic model, the optimum criterion, and types of constraints (see [12,13]). Section 3 proposes the statement of an optimal control problem devoted to the UVDS under consideration. It is a bi-criteria optimization (see [14]) problem whose solution has to meet two constraints:

- the final value of bacterial abatement should be equal to an imposed value (disinfection criterion);
- the control inputs minimize the energy consumed for irradiation (energy criterion).

To transform our problem into a standard optimization problem, we adopted a new optimum criterion that is a weighted combination of the two above criteria. In this new description, the optimal control problem is denoted by UV-OCP1. This statement may be considered a contribution of this work, particularly because it can be easily adapted to a different UVDS.

Section 4 proposes an optimization algorithm devoted to solving UV-OCP1. It is a meta-heuristic algorithm (see [15–17]) based on hybrid topology particle swarm optimization (HTPSO) [18,19], an improved version of PSO (see [20,21]) with better communication abilities among particles. In addition to the “social neighborhood”, other known techniques were used to improve the algorithm: adaptation of particle speed, a method to detect the convergence, and a simple deterministic method to initialize the particle position.

Section 4.2 and the files attached to this paper provide a detailed description of this algorithm and its implementation, which could be easily adapted to a different UVDS.

The proposed algorithm called HTPSO_UV1 receives, as input data, the imposed bacterial abatement value and determines the setpoint optimal values: the flow rate and UV light intensity. The implemented program also yields the values of the two optimal criteria.

The examination of optimal solutions supposes that HTPSO_UV1 has been carried out for a range of abatement values. Execution of the series was conducted to answer the following questions: Is there an optimal solution for any given bacterial abatement value? To what extent is the disinfection criterion met? To what extent is the energy criterion met?

The investigation presented in Section 5 also considers changes in the performance index due to the relaxation of the bacterial abatement requirement. A new problem, UV-OCP2, is defined and solved. Technical constraints are also considered: a fixed flow rate value (which generates UV-OCP3) or a fixed UV intensity value (which defines UV-OCP4). Execution of the series was carried out to answer the above questions, and the results are analyzed and illustrated.

2. A Disinfection UV System

2.1. Structure of the UV System

This work adopted a typical UVDS, i.e., the pilot disinfection water system described in [5] and illustrated in Figure 1. In the proposed modification, in addition to the equipment’s technical parameters, we also used data presented in the paper mentioned above. The treated water can be recycled through a recycling circuit that allows several passages through the irradiation room.

In most cases, the UVDS uses tubular germicidal lamps surrounded by a quartz tube submerged in a chamber through which the water flows. This is a cylindrical stainless reactor with an annular section, a length of 70 cm, and an internal diameter of 6 cm. The volume of the irradiation chamber is V = 2 L. The UV radiation source is usually a low-pressure mercury arc lamp that produces a UV shortwave with a wavelength of 253.7 nm. An electronic ballast ensures the starting of the discharge lamp and controls the current through the discharge lamp. The experiments used water whose UV transmittance was around 90% to diminish the influence of the decrease in the UV radiation intensity due to the growing distance from the UV lamp (see [7]).
The disinfection unit was precisely designed to facilitate the modification of UV intensity and freshwater flow rate. In this study, only UV intensity (I) and the freshwater flow rate (Q) were considered to be control inputs, meaning that they achieve the control action, which is to irradiate the water with a certain UV dose. The power module can establish the setpoints for the UV light intensity and the flow rate inside the minimum and maximum values as below:

\[ I_{\text{min}} = 5 \text{ mW/cm}^2; \quad I_{\text{max}} = 10 \text{ mW/cm}^2; \quad Q_{\text{min}} = 0.2 \text{ L/s}; \quad Q_{\text{max}} = 0.8 \text{ L/s}; \]

These physical parameters are manually controlled because the basic automation level does not exist; namely, there are no control closed-loops that adjust the control inputs to equal the setpoint values. This study regards the optimal control level that establishes the setpoint values for the control inputs of the UVDS taking into account optimum criteria.

### 2.2. The Dynamic Model of the UV Disinfection System

The modeling of the UV system as a dynamic system is subject to the control objective. Our work’s main objective was to study the possibility of controlling the process to ensure freshwater quality (a given disinfection degree) and, eventually, to minimize the energy consumed by the UV irradiation. Essentially, this model is used in the following sections to state and solve optimization problems to obtain the setpoints of the control inputs. Hence, we adopt a simple and easy to use dynamic model.

For our control objective, we chose the UV dose delivered by the source and the bacterial abatement as output variables, denoted respectively by D and A. These variables are dependent on several physical parameters, such as the intensity of UV light, the freshwater’s UV transmittance, the freshwater’s temperature, pH, and the hydraulic characteristics of the disinfection unit (see Figure 2).
In this study, only UV intensity and the freshwater flow rate were considered to be control inputs because they solely achieve the control action. Thus, the disinfection unit was precisely designed to facilitate the modification of $I$ and $Q$. The other input variables will be regarded as parameters of the system’s dynamic model, particularly because their values may be considered to be constant during the control horizon.

**Remark 1.** The desired model has two control inputs, $I(t)$ and $Q(t)$, and two output variables, $D(t)$ and $A(t)$. This model has to be appropriate for the simulation of the UVDS over the simulation horizon:

$$t \in [0, t_f], \ t_f : \text{the final time.} \quad (1)$$

**Remark 2.** Taking into account the needs of the optimization problems stated in the following sections, the dynamic model of the UVDS has a simplified task. It should calculate the output variables $D$ and $A$ for control inputs with constant values $I(t) = I_0$ and $Q(t) = Q_0$ over the simulation interval.

### 2.2.1. Calculation of the UV Dose Delivered by the Discharge Lamp

We are interested in calculating the UV dose delivered by the UV lamp during only one passage through the irradiation room. Its value will be used in the sequel when we try to minimize energy consumption. Let us note that the delivered UV dose ($D$) is different from the Received Radiation Dose (denoted by UVD in [7]). The latter is the UV dose received (absorbed) by the mass of the fluid elements (including microorganisms). The greater the UVD, the more efficient the UV system. Its calculus is a complex problem presented in [7].

The initial time $t = 0$ corresponds to the moment when the freshwater enters the irradiation room with the volume $V$. The moment when it exits the disinfection room depends on the flow rate $Q$ and the volume $V$. This is called exposure time ($t_{\text{exp}}$) and has the following value:

$$t_{\text{exp}} = \frac{V}{Q}, \quad (2)$$

supposing that $Q$ is constant.

Even for this flow reactor, the dose delivered by the UV lamp depends on the radiation intensity of the source $I_0$ and the exposure time, as follows:

$$D = I_0 \cdot t_{\text{exp}}, \quad (3)$$

where $I_0$ has a constant value.

For example, $D$ and $I_0$ are expressed respectively in mW·s/cm$^2$ and mW/cm$^2$. If $I_0$ and $Q = Q_0$ have constant values, Equations (2) and (3) allow the delivered UV dose to be calculated as:

$$D(I_0, Q_0) = I_0 \cdot \frac{V}{Q_0}. \quad (4)$$

Equation (4) shows that the output variable $D(t)$ has a nonlinear dependence on $Q(t)$. This represents an integral action with saturation that occurs in a moment that depends on $Q_0$.

**Remark 3.** Generally speaking, the nonlinear systems have the property that their responses depend not only on the shape but also on the amplitude of the input variables. Our case is a good example; the response to a step variation of the input variable $Q$ depends on the value of $Q_0$ because the saturation occurs at the moment $t = V/Q_0$.

Therefore, this nonlinear dependence cannot be characterized by a transfer function matrix (with the exception of an anterior system’s linearization), but Equation (4) is a direct and simple approach to calculate the $D$ dose according to Remark 2.
2.2.2. Bacterial Abatement Calculation

Well-known disinfection kinetic models have been proposed or analyzed in many papers (see [8,9,22]) to model the germicidal action of UV light. Numerous papers have used simple kinetic models to calculate the bacterial abatement defined below:

$$A(t) = \log_{10} \frac{N_0}{N(t)},$$  \hspace{1cm} (5)

$N_0$ and $N(t)$ are, respectively, the concentrations of viable organisms (expressed in cross flow ultra filtration/100 mL) before and after the germicidal action of UV light.

Instead of using a disinfection process model to calculate $A(t_{exp})$, we considered and improved a dynamic model for the particular UV system presented in [5]. This paper describes a series of experimental measurements that yielded data concerning the dynamic behavior of the UVDS under consideration. Starting from this data, we propose a nonlinear model whose output is $A(t)$ and input variables are $I(t)$ and $Q(t)$.

Using an identification procedure that exploited the stored measurements, the bacterial abatement is expressed according to a linear dependence (see the linear system of Figure 3). If we denote the Laplace transforms of $I(t)$, $Q(t)$, and $A(t)$ by $I(s)$, $Q(s)$, and $A(s)$, respectively, it holds that:

$$A(s) = T_I(s) \cdot I(s) + T_Q(s) \cdot Q(s).$$  \hspace{1cm} (6)

![Figure 3. The computation of the bacterial abatement.](image)

$T_I(s)$ and $T_Q(s)$ are the transfer functions of the two control channels. In the sequel, we adopt the two transfer functions obtained by system identification:

$$T_I(s) = \frac{k_I \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}; \hspace{0.5cm} T_Q(s) = \frac{k_Q}{1 + \tau s}. \hspace{1cm} (7)$$

The values of $I(t)$ and $Q(t)$ meet the following constraints:

$$I_{\min} \leq I(t) \leq I_{\max}; \hspace{0.5cm} Q_{\min} \leq Q(t) \leq Q_{\max} \hspace{1cm} (8)$$

The parameters $k_I$, $k_Q$, $\zeta$, $\tau$, $\omega_n$, $I_{\min}$, $I_{\max}$, $Q_{\min}$, $Q_{\max}$ have the usual definitions, with values given in Appendix A.

Considering Remark 3, the output $A(t)$ has a saturated evolution at the moment $t_{exp}$. Therefore, our goal is to construct a nonlinear model that calculates the value $A(t)$. At the same time, according to Remark 2, this model will be devoted only to control inputs with constant values $I(t) = I_0$ and $Q(t) = Q_0$ over the simulation interval.

This model’s construction is based on four considerations:

- The value $A(t)$ has two terms, like in Equation (6), corresponding to the two inputs.
- The dynamic behavior of the two terms is retained, i.e., we retain the values of the poles for the two transfer functions from Equation (7).
- We consider different static gains for each pair of values $(I_0, Q_0)$ as a direct consequence of Remark 3. These static gains are $k_1 (I_0, Q_0)$ and $k_2 (I_0, Q_0)$, which are different from $k_I$ and $k_Q$.
- The static gains are determined using experimental measurements. This is based on data extracted from the characteristic curves presented in paper [5] and reproduced in Appendix A.
Applying these considerations, Equation (6) takes the nonlinear form (9), which expresses the Laplace transform of the response to the step variation of the two inputs:

\[ A(s, I_0, Q_0) = k_1(I_0, Q_0) \cdot T_1(s) \cdot \frac{1}{s} + k_2(I_0, Q_0) \cdot T_2(s) \cdot \frac{1}{s}, \]  

(9)

where

\[ T_1(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}; \quad T_2(s) = \frac{1}{1 + \tau s} \]

(10)

and

\[ I_{\min} \leq I_0 \leq I_{\max}; \quad Q_{\min} \leq Q_0 \leq Q_{\max} \]

If the gain values \( k_1(I_0, Q_0) \) and \( k_2(I_0, Q_0) \) are known, the bacterial abatement \( A(I_0, Q_0) \) can be calculated by integrating Equation (9). An outline of the computation of bacterial abatement is presented in Table 1.

**Table 1. Outline of function \( A(I_0, Q_0, t) \).**

| Start \( A(I_0, Q_0, t) \) |
|----------------------------|
| /* \( y_1, y_2, \) and \( y \) are vectors with real elements */ |
| /* \( k_1(I_0, Q_0) \) and \( k_2(I_0, Q_0) \) are real functions */ |
| 1. \( y_1 \leftarrow \) step response of \( T_1(s) \); |
| 2. \( y_2 \leftarrow \) step response of \( T_2(s) \); |
| 3. \( A(I_0, Q_0, t) \leftarrow k_1(I_0, Q_0) \cdot y_1 + k_2(I_0, Q_0) \cdot y_2 \) |
| 4. \( \text{return} \ A(I_0, Q_0, t) \) |

Generally speaking, the values of \( k_1(I_0, Q_0) \) and \( k_2(I_0, Q_0) \) can be deduced by an interpolation procedure that processes the available experimental measurements.

In our work, these data were not available. Therefore, we exploited a peculiar aspect of the characteristic curves (presented in paper [5] and Appendix B) that depict the bacterial reduction versus flow rate. An additional hypothesis can be made: the curves are partial branches of parabolas. Because the determination of the gain values \( (k_1(I_0, Q_0) \) and \( k_2(I_0, Q_0)) \) is based upon this hypothesis, one may consider that our approach has an empirical characteristic. Hence, only a few details are given in Appendix A concerning the determination of the gain values.

**Remark 4.** The procedure that calculates \( A(I_0, Q_0, t) \) is essential for solving the optimization problems of the following sections. Nevertheless, the high degree of accuracy of this calculation is not crucial to define and solve the optimization problems, which address the UVDS under consideration. Finally, our main objective is to study these problems and propose good approximative methods to solve them. Obviously, for real-time implementation of optimal solutions, high calculation accuracy is needed.

The procedure implementing the calculation \( A(I_0, Q_0, t) \) was written using the MATLAB language (and system), like all of the algorithms presented in this paper.

Figure 4 depicts the static gains generated using procedure \( A(I_0, Q_0, t) \), which adopted the additional hypothesis. Compared with experimental measurements, these curves reveal high accuracy of the calculations. Hence, we have an appropriate tool to address the nonlinear dependence from Figure 3.
In our work, these data were not available. Therefore, we exploited a peculiar aspect of the characteristic curves (presented in paper [5] and Appendix B) that depict the bacterial reduction versus flow rate. An additional hypothesis can be made: the curves are partial branches of parabolas. Because the determination of the gain values ($k_{I0}$ and $k_{Q0}$) is based upon this hypothesis, one may consider that our approach has an empirical characteristic. Hence, only a few details are given in Appendix A concerning the determination of the gain values.

Remark 4. The procedure that calculates $A(I_0, Q_0, t)$ is essential for solving the optimization problems of the following sections. Nevertheless, the high degree of accuracy of this calculation is not crucial to define and solve the optimization problems, which address the UVDS under consideration. Finally, our main objective is to study these problems and propose good approximative methods to solve them. Obviously, for real-time implementation of optimal solutions, high calculation accuracy is needed.

The procedure implementing the calculation $A(I_0, Q_0, t)$ was written using the MATLAB language (and system), like all of the algorithms presented in this paper.

Figure 4 depicts the static gains generated using procedure $A(I_0, Q_0, t)$, which adopted the additional hypothesis. Compared with experimental measurements, these curves reveal high accuracy of the calculations. Hence, we have an appropriate tool to address the nonlinear dependence from Figure 3.

Figure 4 confirms the physical behavior of the UVDS: the bacterial abatement diminishes with the increase in the flow rate and increases with the increase in the UV intensity.

3. Disinfection System’s Optimal Control

As mentioned in Section 1, our study regards the optimization level of the UVDS under consideration. More specifically, this level considers the following aspects:

- The freshwater particles are exposed to UV radiation during a period expressed by Equation (2). However, the control inputs, $I$ and $Q$, have constant values ($I = I_0$ and $Q = Q_0$) for much longer periods.
- The setpoints $I_0$ and $Q_0$ will determine the dynamic evolution of UVDS, which means the outputs $D(I_0, Q_0, t)$ and $A(I_0, Q_0, t)$ evolve towards their steady values given by the nonlinear models described in the previous section.
- A performance criterion can be associated with the dynamic evolution of the disinfection system. An optimization algorithm yields the optimal solution: the reference values, $I_0$ and $Q_0$, which minimize or maximize the performance criterion.

The task to control the inputs $I$ and $Q$, such that they take the values $I_0$ and $Q_0$, is fulfilled by the electronic ballast for powering the UV discharge lamp and the actuation of the water pump (see Figure 1).

3.1. Optimum Criteria

Obviously, the most important optimum criterion refers to the level of inactivation of microbiological pathogens, i.e., a certain value or a minimum value for the bacterial abatement $A$. We can call it the disinfection criterion. The following constraints may express this criterion:

$$A(t_f) = A_0$$  \hspace{1cm} (11)

or

$$A(t_f) \geq A_0.$$  \hspace{1cm} (12)

The notation $t_f$ is the final moment of the control (simulation) horizon $[0 t_f]$.

As previously mentioned in Section 2.2.1, the bacterial abatement enters into the
saturation zone at the moment \( t_{\text{exp}} \) given by Equation (2). Consequently, the function \( A(I_0, Q_0, t) \) has to meet the equality:

\[
A(I_0, Q_0, t_f) = A(I_0, Q_0, t_{\text{exp}}), \text{ for } t_f \geq t_{\text{exp}}. \tag{13}
\]

Another optimum criterion that may be considered is to minimize the consumption of energy necessary to provide the UV dose. We can call it the energy criterion. The energy is used for powering the UV discharge lamp. Considering the unit of irradiation dose, i.e., mW·s/cm\(^2\), one can identify an equivalent surface \( S \) such that:

\[
\Delta E \approx D \cdot S. \tag{14}
\]

\( \Delta E \) denotes the energy consumed for irradiation in a single passage. It can be expressed in mW·s, if \( S \) is given in cm\(^2\). The equivalent surface \( S \) depends on the constructive parameters of the UVDS under consideration. Equation (14) allows us to state that \( \Delta E \) is proportional to the UV dose. Therefore, the minimization of energy is equivalent to minimizing the dose \( D \) received during the exposure time.

### 3.2. Statement of Optimal Control Problem

This section addresses the optimal control of the UVDS, tackling simultaneously the two optimal criteria presented in the previous section. Hence, a bi-criteria optimization problem is stated in the sequel; roughly speaking, the optimization algorithm should determine the values \( I_0 \) and \( Q_0 \) such that:

- the final value of bacterial abatement should be \( A_0 \) (an imposed value);
- the control inputs minimize the energy consumed for irradiation.

Hereafter, we present the constitutive elements of this optimal control problem denoted by UV-OCP1 (UV-Optimal Control Problem 1).

(a) The dynamic model of the UVDS

This relates to the simplified model presented in Section 2.2, that is \( I(t) = I_0 \) and \( Q(t) = Q_0 \) have constant values according to Remark 2. This simplified model is composed of Equations (2)–(4), (9), and (10) and completed with functions \( A(I_0, Q_0, t) \), \( k_1(I_0, Q_0) \), and \( k_2(I_0, Q_0) \).

(b) Constraints

Time horizon:

\[
t \in [0, t_f] \tag{15}
\]

\[
t_f = \frac{V}{Q_0} = t_{\text{exp}}
\]

The value \( t_f \) denotes the final time (the water passage time through the irradiation room). Bound constraints:

\[
I_{\text{min}} \leq I(t) \leq I_{\text{max}}, \quad Q_{\text{min}} \leq Q(t) \leq Q_{\text{max}}
\]

(c) Performance index

\[
\min_{I_0, Q_0} J(I_0, Q_0) = w_1 \cdot D(I_0, Q_0, t_f) + w_2 \cdot \left| A(I_0, Q_0, t_f) - A_0 \right|, \tag{16}
\]

\( w_1, w_2 > 0 \)

Expression (16) of the objective function \( J(I_0, Q_0) \) is a weighted combination of the two criteria and transforms our problem into a standard optimization problem with a single criterium. The numerical weights \( w_1, w_2 \) specify the relative importance we attach to the two criteria and, at the same time, allowing the ranges of both values to be compared. We can consider that \( w_1 \) also includes the equivalent surface \( S \). Hence, the first term
corresponds to energy consumption. The second term indicates that the final abatement should approach the value $A_0$ as much as possible.

Instead of using an objective function with two numerical weights, we can use an equivalent expression as below:

$$J(I_0, Q_0) = D(I_0, Q_0, t_f) + \alpha \cdot A(I_0, Q_0, t_f) - A_0$$  \hspace{1cm} (17)

where $\alpha = w_2/w_1$. The choice of value $\alpha$ needs careful, practical analysis.

4. Algorithm Using Hybrid Topology Particle Swarm Optimization

4.1. Hybrid Topology Particle Swarm Optimization

This section describes characteristics of hybrid topology particle swarm optimization. An optimization algorithm can be designed and implemented, even by those who are newcomers in the field of this metaheuristic (see [15–17]). The information given in the modification is supplemented in Appendix C and the attached files, which contain real implementations of the proposed algorithms.

PSO algorithms have significant theoretical support concerning their convergence, stability, and complexity (see [11,18,23,24]). Implementing an optimal criterion is also a very important topic, especially when the problem is a multi-objective optimization (see [16], Chapter 4).

4.1.1. General Description of PSO

A particle swarm system can optimize a given function with $n$ decision variables, called the objective function, through a simulated movement of the swarm inside the space where the function is defined. The objective function can model the “environment” profile, i.e., the altitude of “valleys” and “hills”, which is the basic topology of the optimization problem (OP). Each particle “flies” over new regions of the search space. In the PSO’s initial version, the particles communicate among themselves through the intermediary of a global variable, memorizing the best position “encountered” by the swarm.

As the search process proceeds, the exploration decreases, and the exploitation is intensified. Finally, the particles converge to the global best solution of the OP. Paper [18] presents the main aspects concerning the analysis of PSO algorithms. The convergence of PSO algorithms is treated in [11,24], which guides the choice of algorithm parameters.

The swarm is compounded by $N$ particles, with each particle coded through a three-component vector [20,21], usually denoted by $(X_i, V_i, P_{best_i})$. Each component is an $n$-dimensional vector representing, respectively, the position, the speed, and the best personal position reached in the search process (see [20]). It holds as:

$$X_i = (x_i^1, x_i^2, \ldots, x_i^n), V_i = (v_i^1, v_i^2, \ldots, v_i^n), P_{best_i} = (p_{best_i}^1, \ldots, p_{best_i}^n), \; i = 1, 2, \ldots, N$$

The evolution of the particle swarm is implemented through an iterative process. Let $P_{gbest}$ be the particles’ best position until the current step, called the “global best” position. At the end of the algorithm, i.e., when convergence is reached, this is the solution of the OP.

The essence of the PSO algorithm is updating the particle speed and position at each step of the search process. This update is performed using the following equations:

$$\left\{ \begin{array}{l}
V_i(t + 1) = w \times V_i(t) + C_1 \times rand_1 \times (P_{best}(t) - X_i(t)) + C_2 \times rand_2 \times (P_{gbest}(t) - X_i(t)) \\
X_i(t + 1) = X_i(t) + V_i(t + 1)
\end{array} \right.$$  \hspace{1cm} (18)

where $w$ is inertia weight; $C_1$ and $C_2$ are acceleration coefficients; $rand_1$, $rand_2$ are random numbers in the interval [0, 1]; and $t$ is the step number. This initial version often leads to premature convergence.
4.1.2. Hybrid Topology PSO

The algorithm described in the sequel is based on hybrid topology particle swarm optimization (HTPSO) [18,19], which is an improved version of the PSO metaheuristic with better communication abilities among particles. It is also endowed with adaptive control of the particle speed through a continuous change of some of the algorithm’s parameters.

The proposed HTPSO algorithm is devoted to solving UV_OCP1 and is called HTPSO_UV1.

HTPSO also uses a local topology of the swarm, regarded as a communication network. The local topology involves the existence, for any particle \( i \), of a “social neighborhood,” i.e., a set of 3–5 particles that inform particle \( i \) about their best personal experience. These neighborhoods are decided in a deterministic or random manner at each step of the algorithm (see [23]). The particle will determine the local best position \( P_{lbest_i} \) as the best experience of the particles belonging to the “social neighborhood” and the particle itself. It holds that:

\[
P_{lbest_i} = (p_{lbest_i}^1, \ldots, p_{lbest_i}^d, \ldots, p_{lbest_i}^n); d = 1, \ldots, n;
\] (19)

A new term appears in the speed equation containing \( C_3 \) and \( rand_3 \) (see [18]). Hence, the update of the speed and the position is performed using the following equations:

\[
\begin{align*}
    v_d^i(t + 1) &= w \times v_d^i(t) + C_1 \times rand_1 \times \left( p_{lbest_i}^d - x_d^i(t) \right) + \\
                &+ C_2 \times rand_2 \left( p_{lbest_i}^d - x_d^i(t) \right) + C_3 \times rand_3 \left( p_{gbest}^d - x_d^i(t) \right); d = 1, \ldots, n \\
    x_d^i(t + 1) &= x_d^i(t) + v_d^i(t + 1)
\end{align*}
\] (20)

4.1.3. Adaptation of Particle Speed

The HTPSO can be enhanced with an efficient technique: adaptation of particle speed. This technique modifies the coefficients \( C_1, C_2, C_3, \) and \( w \) during the searching process (see [18,19]). The objective is to adapt them to the phase of the search process and prepare the algorithm’s convergence. The adaptation is achieved by a linear increase in the coefficients \( C_1, C_2, \) and \( C_3, \) between their minimum and maximum values. At the same time, the parameter \( w \) decreases as:

\[
\begin{align*}
    C_1(t) &= C_{1min} + (C_{1max} - C_{1min})t/T; \quad C_2(t) = C_{2min} + (C_{2max} - C_{2min})t/T; \\
    C_3(t) &= C_{3min} + (C_{3max} - C_{3min})t/T; \quad w(t) = w_{max} - (w_{max} - w_{min})t/T
\end{align*}
\] (21)

where \( T \) is the estimated or maximum number of steps until the algorithm’s convergence.

This work proposes an algorithm based on HTPSO endowed with the adaptation of the particle speed.

4.2. Implementation of Algorithm HTPSO_UV1

4.2.1. Algorithm’s General Description

Table 2 presents a pseudocode description of the algorithm HTPSO_UV1. This description is general enough that the algorithm can solve many other OPs (with a few changes for the problems that maximize the objective functions).

The algorithm is an iterative process that simulates the particles’ evolution in the search space. The variable \( t \) is the current step of the iterative process, and \( T \) is the estimated or maximum number of steps until convergence. The initialization of the algorithm presented in line #2 enumerates some parameters, defined in Appendix C.
Appl. Sci. 2021, 11, 2638

Table 2. Description of algorithm HTPSO_UV1.

| Line | Description |
|------|-------------|
| 1    | start HTPSO_UV1 |
| 2    | Initialization: \( N, n, v_{\text{max}}, x_{\text{min}}, x_{\text{max}}, C_{1\text{min}}, C_{1\text{max}}, C_{2\text{min}}, C_{2\text{max}}, \) \( C_{3\text{min}}, C_{3\text{max}}, w_{\text{min}}, w_{\text{max}}, T, NF, \text{MINNF} \) |
| 3    | Generate the initial particles' velocities as uniformly distributed values in the interval \([-v_{\text{max}}, v_{\text{max}}]\). |
| 4    | Generate the particles' initial positions; |
| 5    | for \( i = 1, \ldots, N \) |
| 6    | \( p_{\text{best}_i} \leftarrow X_i \) |
| 7    | \( \text{BEST}(i), \text{fitness}(i) \leftarrow \text{ffunction}(X_i) \) |
| 8    | end |
| 9    | \( i_0 \leftarrow \text{arg max} \{ \text{BEST}(i), i = 1, \ldots, N \} \), |
| 10   | \( p_{\text{gbest}} \leftarrow X_{i_0} \); GBEST \( \leftarrow \text{BEST}(i_0) \); /*Determine \( p_{\text{gbest}} \) and GBEST*/ |
| 11   | found \( \leftarrow 0 \) |
| 12   | \( t \leftarrow 1 \); |
| 13   | while \((t \leq T) \) \& \((\text{found} = 0)\) |
| 14   | Coefficients tuning: \( w, C_1, C_2, C_3 \) |
| 15   | for \( i = 1, \ldots, N \) /*move the particles swarm*/ |
| 16   | \( p_{\text{best}_i} \leftarrow \text{Generate_Plbest}(i); /*generate "local best"*/ |
| 17   | for \( d = 1, \ldots, n \) |
| 18   | Update the particles' speed using Equation (20) |
| 19   | \( \text{Speed\_limitation}(v^d_i(t + 1)) \) |
| 20   | \( x^d_i(t + 1) = x^d_i(t) + v^d_i(t + 1) \) |
| 21   | \( \text{Position\_limitation}(x^d_i(t + 1)) */using \text{xmin}, \text{xmax} */ \) |
| 22   | end |
| 23   | \( \text{fitness}(i) \leftarrow \text{ffunction}(X_i) \) |
| 24   | if \( \text{fitness}(i) < \text{BEST}(i) \) |
| 25   | \( p_{\text{best}_i} \leftarrow X_i; \text{BEST}(i) \leftarrow \text{fitness}(i); \text{NF}(i) = 0; \) |
| 26   | else \( \text{NF}(i) = \text{NF}(i) + 1; \) |
| 27   | end |
| 28   | if \( \text{fitness}(i) < \text{GBEST}(i) \) |
| 29   | \( p_{\text{gbest}} \leftarrow X_i; \text{GBEST} \leftarrow \text{fitness}(i) \) |
| 30   | end |
| 31   | if \( \text{min} \{ \text{NF}(i), i = 1, \ldots, N \} = \text{MINNF} \) then \( \text{found} \leftarrow 1; \) |
| 32   | \( t \leftarrow t + 1 \) |
| 33   | end /*while*/ |
| 34   | return \( p_{\text{gbest}} \) |

Line #4 calls the generation of the particles' initial positions. Generally speaking, these positions are randomly generated, but in our case, the particles are placed in a deterministic manner. These are uniformly spread inside the search domain denoted by SD, which in our case is the rectangle:

\[
S_D = [l_{\text{min}}, l_{\text{max}}] \times [Q_{\text{min}}, Q_{\text{max}}]
\]  

(22)

The action Coefficients tuning (in line #14) adjusts the coefficients of Equation (18) according to Equation (21). Action Speed\_limitation from line #19 restricts the speed to the interval \([-v_{\text{max}}, v_{\text{max}}]\). Similarly, the action Position\_limitation from line #21 keeps the particles inside the search domain.

The particles of the “social neighborhood” inform particle \( #i \) about their best experience, i.e., their best positions. The function Generate\_Plbest\( (i) \), called in line #16, selects the particle that belongs to the “social neighborhood” and has the minimum BEST value. This selected particle gives the best local position.

The variable \( \text{fitness}(i) \) stores the value of the objective function (ffunction) for the current position \( X_i \). Its value is involved in the BEST\( (i) \) and GBEST\( (i) \) updating (lines #25 and #29).

This implementation uses a simple and effective way to determine the algorithm’s convergence. While this state is not reached, the variable “found” takes the value 0. The
value $NF$. ($i$) is associated with particle $#i$ and counts the number of steps without improving the best personal experience (line #26). When all the particles have these values greater than the constant $MINNF$, one can assert the algorithm convergence (see line #31).

4.2.2. Specific Aspects of the Implementation

At the same time, the description given in Table 2 has a form very close to the implementation of the algorithm (see the scripts “HTPSO_UV1.m”, inside the folder “HTPSO_UV1_Study1”). There are details concerning UV_OCP1’s specific aspects. For example, our problem’s solution has two components. Hence, any position $X_i$ has two elements:

$$X_i = [I_0, Q_0]; \quad n = 2 \quad (23)$$

In our implementation, the social neighborhood of particle $#i$ has three particles randomly selected from the swarm. Particle $#i$ itself is also included in this neighborhood.

The OP under consideration’s main specificity is the objective function that has to be minimized: $J_{function}(X_i)$. A general description of this function is given in Table 3.

| 1 | start $J_{function}(X_i)$ |
|---|----------------------------|
| 2 | /* input argument: the position of a particle |
| 3 | $X_i = [x_1^i(t), x_2^i(t)]; */ |
| 4 | $I_0 \leftarrow x_1^i(t)$; |
| 5 | $Q_0 \leftarrow x_2^i(t)$; |
| 6 | $t_f \leftarrow V \sqrt{Q}$; /* $V$ is a global value*/ |
| 7 | $D \leftarrow I_0 \cdot t_f$ |
| 8 | $Ab \leftarrow A(I_0, Q_0, t_f)$ |
| 9 | $J \leftarrow D + \alpha \cdot |A_b - A_0|; /*\alpha, A_0$ are global values*/ |
| 10 | return $J$ |

Firstly, the two control inputs are extracted from the argument $X_i$. The computation of the dose $D$ and the bacterial abatement $Ab$ is achieved in lines #5–#7 of the objective function. Function $A(I_0, Q_0, t)$ described in Section 2.2 is essential for calculating the bacterial abatement.

5. Discussion

5.1. Searching Process and Results’ Analysis

Algorithm HTPSO_UV1 was implemented using the MATLAB language (and system) and executed for many imposed bacterial abatement $A_0$ values. An optimal solution has three components: the UV light intensity $I_0$, the flow rate $Q_0$, and the performance index.

Data concerning the UVDS under consideration are those from Appendices A and B. The constants and the implemented algorithm parameters are indicated in Appendix C. Some tests were carried out using the programs attached to this paper. Different tests could be conducted considering other data for UVDSs with a similar structure.

Figure 5 shows the variables’ typical evolution during the execution of algorithm HTPSO_UV1 for the imposed values of two bacterial abatements: $A_0 = 3$ and $A_0 = 3.5$. The searching process is traced for a few tens of steps because of the rapid algorithm’s convergence. One can see that the GBEST value (performance index) monotonically decreases during the searching process.
It should be noted that HTPSO_UV1 is a stochastic algorithm that tries to approach the UV_OCPI’s optimal solution as much as possible. To decide whether it works well needs a minimal statistical analysis. When considering the convergent variables’ evolution, each execution of the implemented algorithm generates stochastic processes. The evaluation of the algorithm was conducted using execution series because of its stochastic character. For example, our program was executed 30 times for each imposed value $A_0$.

Table 4 presents statistical measures regarding the number of steps until convergence and solution values, in the case $A_0 = 3$. The first five columns show the mean values of the number of steps until convergence ($t_{\text{conv}}$), the number of objective function calls ($N_{\text{calls}}$), $I_0$, $Q_0$, and the performance index $J$. The standard deviations of $t_{\text{conv}}$ and $N_{\text{calls}}$ are presented in the last two columns.

**Table 4. Description of the objective function.**

| Mean ($t_{\text{conv}}$) | Mean ($N_{\text{calls}}$) | Mean ($I_0$) | Mean ($Q_0$) | Mean ($J$) | Std ($t_{\text{conv}}$) | Std ($N_{\text{calls}}$) |
|--------------------------|---------------------------|--------------|--------------|------------|-------------------------|-------------------------|
| 149.3                    | 4480                      | 9.92         | 0.8          | 24.803     | 4.3                     | 129                     |

The optimal values for $I_0$, $Q_0$, and $J$ are practically constant at the end of the 30 executions. Therefore, their mean values given in Table 4 are identical to these constant values:

$$I_0 = 9.92 \text{ mW/cm}^2; \quad Q_0 = 0.8 \text{ L/s}; \quad J = 24.803.$$  

Consequently, the standard deviation of these values is practically zero because the optimal solution belongs to the search domain’s boundary ($Q_0 = 0.8$). That is, a position’s limitation stops the search. This is a general property, as shown in the next section.

One can see that the average number of steps until convergence is relatively small, namely 149 steps. All of the values are given in Appendix C. This relates to the complete convergence of the swarm, i.e., all the particles have the same SD position. There are two explanations for this fact. First, the hybrid topology and the adaptive particles’ speed involve a good convergence speed. Second, in our implementation, the swarm is comprised of 30 particles. That is, the swarm has an important dimension for a PSO algorithm. The smaller the number of particles, the greater the number of steps until convergence.

A very important aspect of such an algorithm’s analysis is the computational complexity as a measure of its efficiency. Practically, this can be evaluated through the number...
of objective function calls. This number can be considered a measure of the computational effort to yield the problem’s solutions. In our case, the function ffunction is the most time-consuming program unit because of its arithmetical calculations. The average value of the number of calls is 4480, which is acceptable if we compare it to an implementation using another metaheuristic (e.g., evolutionary algorithm).

5.2. Study of Optimal Solutions

Section 4 shows the development of a metaheuristic-based algorithm to solve the bi-criteria OP (UV-OCP1) stated in Section 3. We recall the practical statement of the UV-OCP1: the proposed algorithm has to yield the values I₀ and Q₀ that meet the performance index (16), with A₀ being the imposed bacterial abatement. That is, algorithm HTPSO_UV1 has to minimize the objective function (17) that aggregates both disinfection and energy criteria.

Using the script “Study1.m”, the implemented algorithm “HTPSO_UV1.m” was executed for different values A₀ (see Appendix D). First, we set the reference domain of parameter A₀, which for our study is:

\[ A₀ ∈ [2.5, 4.1]. \]  

The values characterizing the optimal solutions were stored and are represented in Figure 6. We recall that the energy consumption necessary to provide the UV dose is proportional to D(I₀, Q₀, t_f). The first term of the objective function (17) corresponds to the UV dose and is represented in Figure 6 by the black curve. In this work, the parameter α from Equation (17) is set as α = 100 after calibration tests. For each value A₀, an optimal solution is represented through four points corresponding to I₀, Q₀, D(I₀, Q₀, t_f), and J(I₀, Q₀). These points generate the four curves depicted in Figure 6.

![Figure 6](image)

**Figure 6.** Optimal solutions as functions of A₀.

The objective of our study is to answer the following questions:

1. Is there an optimal solution for any given A₀?
2. To what extent is the disinfection criterion met?
3. To what extent is the energy criterion met?
Questions #2 and #3 make sense because the objective function (17) aggregates both criteria.

We can remark that there is a unique optimal solution for every value $A_0$ of the reference domain (see (24)).

Let us note that the red and black curves from Figure 6 define two functions of argument $A_0$.

The performance index minimizes the sum of both terms of Equation (17) and is described by the red curve. Figure 6 shows that the red curve is slightly shifted up on purpose, such that the black curve can be distinctly seen. In fact, the two curves are identical, and the energy term is equal to the performance index. Therefore, it holds that:

$$\alpha \cdot |A(I_0, Q_0, t_f) - A_0| = 0. \quad (25)$$

We can now answer question #2: The optimal bacterial abatement always reaches the reference value $A_0$.

Considering the structure of the objective function, we can answer question #3: our algorithm searches for the solution $(I_0, Q_0)$, which minimizes the energy consumption on the condition that the constraint (25) is met.

The analysis of the four curves of Figure 6 shows us the particular placement of each optimal solution; the point $(I_0, Q_0)$ is placed on the boundary of the searching domain $SD$ (see (22)) as below:

- if $A_0 \in [2.5, 3]$, then $Q_0 = 0.8 \text{ L/s}$ and $8.5 \leq I_0 \leq 10 \text{ mW/cm}^2$;
- if $A_0 \in [3, 4.1]$, then $I_0 = 10 \text{ mW/cm}^2$ and $0.3 \leq Q_0 \leq 0.8 \text{ L/s}$;

In other words, the optimal solutions are placed on two sides of the rectangle $SD$.

5.3. Different Control Hypothesis

In this section, we briefly examine other control hypotheses that lead to a different objective function.

5.3.1. Imposed Lower Bound for the Bacterial Abatement

We have tried to relax the requirement concerning the bacterial abatement in addition to that used by UV-OCP1. Consequently, we replaced the constraint (11) with (12), and updated the objective function as below:

$$\begin{cases} A(I_0, Q_0, t_f) \geq A_0 \\ J(I_0, Q_0) = D(I_0, Q_0, t_f) \end{cases} \quad (26)$$

We can state a new OP called UV-OCP2, considering the constraints (26) and retaining the other constitutive elements defining UV-OCP1. Hence, the performance index only minimizes the UV dose, whereas the bacterial abatement should have a lower bound. Details concerning the implementation of the new algorithm and the test results are given in Appendix D.

The main conclusion based on the test results is that UV-OCP2 and UV-OCP1 have identical solutions for each lower bound $A_0$. The optimal solutions are graphically represented and generate a drawing identical to Figure 6. Consequently, the entire analysis conducted in Section 5.2 is also valid.

5.3.2. Flow Rate Imposed Owing to Technical Constraints

In this case, the flow rate is constant $(Q_0)$ and a priori known due to technical restrictions. The only control variable required to optimize the UVDS is $I_0$. Hence, the performance index aggregating the two criteria has the following expression:

$$\min_{I_0} J(I_0) = D(I_0, t_f) + \alpha \cdot |A(I_0, Q_0, t_f) - A_0| \quad (27)$$
The new objective function \( J(l_0) \) and the other elements defining UV-OCP1 describe a new OP, denoted UV-OCP3. Details concerning the implementation of the new algorithm are given in Appendix D.

Let us suppose that, for technical reasons, the flow rate is set to \( Q_0 = 0.2 \). Like in Section 5.2, the UV-OCP3’s optimal solutions yield the four curves presented in Figure 7. In this case, the green curve represents the final value of bacterial abatement that can be denoted \( A(l_0, Q_0) \) because the final time is considered implicitly. In this new conjuncture, this value may be different from \( A_0 \) because the UVSD does not follow the reference value.

![Figure 7](image_url)

**Figure 7.** Optimal solutions with an imposed flow rate: (a) \( Q_0 = 0.2 \); (b) \( Q_0 = 0.4 \).

Figure 7 depicts the placement of the optimal solutions in two individual cases. Figure 7a illustrates the case that arises when, for example, \( Q_0 = 0.2 \).

Analyzing this case, we can note that the four curves have two segments as defined below:

\[
2.5 \leq A_0 < 3.7 \Rightarrow \begin{cases} l_0 = l_{\min} = 5 \\ A(l_0, Q_0) = 3.7 \\ \text{UV dose} = 50 \end{cases}
\]

That is, the value \( A \) achieved by the UVDS is greater than the reference value, even if the intensity takes its minimum value. The UV dose also takes its minimum value for the given flow rate. At the same time, the performance index (the black curve) is greater than the UV dose because it holds that:

\[
\left| A(l_0, Q_0, l_f) - A_0 \right| > 0.
\]  
(28)

\[3.7 \leq A_0 < 4.1 \Rightarrow A(l_0, Q_0) = A_0;\]

The UVSD succeeds in reaching \( A_0 \). The greater the value of \( A_0 \), the greater the values of \( l_0 \) and UV dose.

Figure 7 also illustrates the case (b) that arises when, for example, \( Q_0 = 0.4 \). We can remark that the four curves also have two segments defined below:

\[
2.5 \leq A_0 < 3.748 \Rightarrow A(l_0, Q_0) = A_0;
\]

All curves increase linearly.

\[
3.748 \leq A_0 \leq 4.1 \Rightarrow \begin{cases} l_0 = l_{\max} = 10 \\ A(l_0, Q_0) = 3.748 \\ \text{UV dose} = 50 \end{cases}
\]
The value $A$ achieved by the UVDS is less than the reference value, even if the intensity takes its maximum value. The UV dose also takes its maximum value for the given flow rate. Simultaneously, the performance index is greater than the UV dose because the inequality (28) is valid.

5.3.3. UV Intensity Imposed by Technical Constraints

In this case, the UV intensity is constant ($I_0$) and a priori known due to technical restrictions. The only control variable required to optimize the UVDS is $Q_0$. The performance index has the new expression:

$$\min_{Q_0} f(Q_0) = D(Q_0, t_f) + \alpha | A(I_0, Q_0, t_f) - A_0 |. \quad (29)$$

Hence, we are faced with a new OP that can be called UV-OCP4.

Figure 8 shows the placement of the optimal solutions in two individual cases. Figure 8 illustrates the cases (a), which arises when, for example, $I_0 = 5$ mW/cm$^2$, and (b) when, for example, $I_0 = 7.9$ mW/cm$^2$.

![Figure 8](image_url)

**Figure 8.** Optimal solutions with an imposed UV intensity: (a) $I_0 = 5$; (b) $I_0 = 7.9$

In case (a), because the intensity $I_0$ is relatively small, the achieved bacterial abatement is limited to the value of 3.7. The optimal flow rate belongs to the interval $[0.2, 0.4]$.

Conversely, because the intensity $I_0$ has a relatively big value, the achieved bacterial abatement reaches any reference $A_0$.

6. Conclusions

This paper presented a study regarding the optimal control for an ultraviolet water disinfection process. As physical support of our work, we adopted a typical pilot disinfection water system described in the literature. The presented study covered the following mandatory aspects: the dynamic model of the UVDS, the optimization problem, the optimization algorithm, and analysis of the optimal solutions.

Despite the fact it is the result of an empirical identification method, the proposed dynamic model of the UVDS has restored its nonlinear characteristic. It has also provided an effective method to calculate bacterial abatement.

We defined a bi-criteria optimization problem that simultaneously considered the energy and disinfection criteria. It was proved that HTPSO is an effective framework to achieve an optimization algorithm. Three more OCPs were detailed to cover other practical situations.

The comprehensive analysis of the results established when an optimal solution exists and the extent to which the disinfection or energy criterion is fulfilled.
The method to calculate the bacterial reduction, OCP statement, optimization algorithm, and results analysis may be considered findings of this work.

Our desideratum is to develop our approach through the identification of the nonlinear dynamic model. A fruitful direction appears to be the use of a neuronal network that will “learn” the UVDS. Accordingly, the optimization algorithm has to be adapted.

Supplementary Materials: The following are available online at https://www.mdpi.com/2076-3417/11/6/2638/s1.

Author Contributions: Conceptualization, V.M. and E.R.; methodology, V.M.; software, S.R.; validation, V.M. and E.R.; formal analysis, V.M.; resources, S.R.; data curation, S.R.; writing—original draft preparation, V.M.; writing—review and editing, E.R.; visualization, S.R.; supervision, E.R.; project administration, E.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available within the article and Supplementary Material (UV_DS.zip).

Acknowledgments: This work was carried out in the framework of the research project DREAM (Dynamics of the REsources and technological Advance in harvesting Marine renewable energy), supported by the Romanian Executive Agency for Higher Education, Research, Development and Innovation Funding—UEFISCDI, grant number PN-III-P4-ID-PCE-2020-0008.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The parameters $k_I$, $k_Q$, $\xi$, $\tau$, $\omega_n$, $I_{\text{min}}$, $I_{\text{max}}$, $Q_{\text{min}}$, $Q_{\text{max}}$ used in Equations (9) and (10) have the usual definition and values given below:

\[
\omega_n = \sqrt{0.7} = 0.836 \text{ rad/ sec} \quad \xi = 0.998 \approx 1; \quad \tau = 1.78 \text{ sec}.
\]

$I_{\text{min}} = 5 \text{ mW/cm}^2$; $I_{\text{max}} = 10 \text{ mW/cm}^2$; $Q_{\text{min}} = 0.2 \text{ L/s}$; $Q_{\text{max}} = 0.8 \text{ L/s}$;

The values $k_I = 0.55/0.7 = 0.78$ and $k_Q = 2.79$ indicated in [5] were not used in this study. Equation (9) generates the following transfer functions:

\[
T_1(s) \approx \frac{0.7}{(s + 0.836)^2} \quad \text{and} \quad T_2(s) = \frac{1}{1 + 1.78 s}
\]

According to the hypothesis advanced in Section 2.2.2, the gain values $k_1(I_0, Q_0)$ and $k_2(I_0, Q_0)$ have parabolic expressions. The coefficients $a, b, c, d$ are linear functions in the argument $I_0$ that allow the curves’ parametrization. Using an empirical identification method based on some experimental, numerical values extracted from Figure A1a, we obtained the formulas shown in Table A1:

| Table A1. Equations resulting from numerical identification. |
|------------------------------------------------------------|
| $k_1(I_0, Q_0) = a(I_0) + b(I_0) \cdot (Q_0 - 0.9)^2$ | $k_2(I_0, Q_0) = c(I_0) + d(I_0) \cdot (Q_0 - 0.9)^2$ |
| $a(I_0) = 0.2873 \cdot I_0 - 0.4104$ | $c(I_0) = -0.06 \cdot I_0 + 0.085$ |
| $b(I_0) = -0.3283 \cdot I_0 + 5.7454$ | $d(I_0) = 0.068 \cdot I_0 - 1.19$ |

Appendix B

Procedure $A(I_0, Q_0, t)$ was implemented (see the scripts “abatement.m”, “val_k21.m”, “valk22.m”, inside the folder “UV_DS”) using the equations from Table A1. This procedure was used to generate the curves from Figure A1b (see the script “CURBE.m” inside the folder “UV_DS”).

Figure A1 shows a very good fit between the two sets of curves, (a) and (b).
Figure A1. Comparison between characteristic curves: (a) experimental data; (b) parametrized curves

Appendix C

Details concerning the implementation and use of algorithm HTPSO_UV1

The initialization phase of algorithm HTPSO_UV1 defines the following constants and variables:

- $N$: number of the swarm’s particles ($N = 30$, in our implementation)
- $n$: the solution dimension;
- $v_{\text{max}}$: the maximum speed of a particle in any direction;
- $C_{1\text{min}}, C_{1\text{max}}, C_{2\text{min}}, C_{2\text{max}}, C_{3\text{min}}, C_{3\text{max}}, w_{\text{min}}, w_{\text{max}}$: the minimum and respectively maximum values of the coefficients that appear in Equation (21);
- $x_{\text{min}}, x_{\text{max}}$: vectors, with $n$ elements, having the minimum and respectively maximum values of the particle’s position for each direction;
- $T$: the estimated or maximum number of steps until convergence;
- $N\text{F}$: vector of $N$ integer numbers, where $N\text{F}(i)$ is associated with particle #$i$ and represents the number of steps without improving the best personal experience;
- $\text{MINNF}$: the minimum value of $N\text{F}$ elements used to assert the algorithm convergence (see line #35, where $\text{found} = 1$ means convergence).

NB: The folder “UV_DS” contains all the other folders mentioned below.

The initialization of other constants and variables is shown in script “HTPSO_UV1.m”, inside the folder “HTPSO_UV1_Study1”.

The objective function is implemented by the function “Jfunction1.m”.

Generation of Figure 5: The two drawings of Figure 5 are generated by the script “DRAW_FIG5.m” that calls the script “HTPSO_UV1_evolution.m”.

Generation of Table 4: The script that helps to construct Table 4 is “Analyse.m”. It calls 30 times the script “HTPSO_UV1_converg.m”, yielding the final values of the following variables: steps’ number until convergence ($t_{\text{conv}}$), $I_0$, $Q_0$, performance index (GBEST), and the total number of calls until convergence.

Number of steps until convergence (example):

\[152, 154, 154, 151, 147, 152, 147, 144, 147, 154, 155, 144, 150, 142, 157, 147, 154, 150, 153, 153, 147, 144, 145, 148, 142, 145, 154, 147, 154, 154, 147, 144, 145, 148, 142, 145, 154].

Generation of Figure 6: The folder “HTPSO_UV1_study1” contains the script “Study1.m” that calls repeatedly the script “HTPSO_UV1.m” for $A_0 = 2.5, 2.6, \ldots, 4.1$. At the end, it calls the script that draws the Figure 6.
Appendix D

Appendix D.1. Imposed Lower Bound for the Bacterial Abatement

In this case, the basic algorithm is “HTPSO_UV2.m,” whose objective function is “Jfunction2.m,” which implements Equation (26).

The folder “HTPSO_UV1_study2” contains the script “Study2.m” that repeatedly calls the script “HTPSO_UV2.m” for \( A_0 = 2.5, 2.6, \ldots, 4.1 \).

Appendix D.2. Flow Rate Imposed Due to Technical Constraints

In this case, the basic algorithm is “HTPSO_UV3.m,” whose objective function is “Jfunction3.m,” which implements Equation (27). In this case, a particle has only one dimension.

The folder “HTPSO_UV1_study3” contains the script “Study3.m,” where \( Q_0 \) is a constant. This script repeatedly calls the script “HTPSO_UV3.m” for \( A_0 = 2.5, 2.6, \ldots, 4.1 \). Finally, it draws Figure 7a (for \( Q_0 = 0.2 \)) or Figure 7b (for \( Q_0 = 0.4 \)).

Appendix D.3. UV Intensity Imposed by Technical Constraints

In this case, the basic algorithm is “HTPSO_UV4.m”, whose objective function is “Jfunction4.m” that implements Equation (29). In this case, a particle has only one dimension.

The folder “HTPSO_UV1_study4” contains the script “Study4.m”, where \( I_0 \) is a constant. This script repeatedly calls the script “HTPSO_UV4.m” for \( A_0 = 2.5, 2.6, \ldots, 4.1 \). Finally, it draws Figure 8a (for \( I_0 = 5 \)) or Figure 8b (for \( I_0 = 7.9 \)).

References

1. Reckhow, D.; Linden, K.; Junsung, K.; Shemer, H.; Makdissy, G. Effect of UV treatment on DBP formation. J. AWWA 2010, 102, 100–113. [CrossRef]
2. Sharma, S.; Bhattacharya, A. Drinking water contamination and treatment techniques. Appl. Water Sci. 2017, 7, 1043–1067. [CrossRef]
3. Collivignarelli, M.C.; Abbà, A.; Benigna, I.; Sorlini, S.; Torretta, V. Overview of the Main Disinfection Processes for Wastewater and Drinking Water Treatment Plants. MDPI Sustain. 2018, 10, 86. [CrossRef]
4. Cristinel, B.; Remus, Z.; Lucian, T.; Eugen, A.; Dumitru, N. Elimination techniques of microbiological agents in water purification processes with UV radiation. J. Appl. Sci. Environ. Sanit. 2011, 6, 51–62.
5. Ouelhazi, K.; Chaabene, A.B.; Sellami, A.; Hassen, A. Multivariable model of an ultraviolet water disinfection system. Desalin. Water Treat. 2017, 67, 89–96. [CrossRef]
6. Johnson, K.M.; Kumar, M.R.A.; Ponmurugan, P.; Gananamangai, B.M. Ultraviolet radiation and its germicidal effect in drinking water purification. J. Phytol. 2010, 2, 12–19.
7. Artichowicz, W.; Luczkiewicz, A.; Sawicki, J.M. Analysis of the Radiation Dose in UV-Disinfection Flow Reactors. Water 2020, 12, 231. [CrossRef]
8. Chen, J.; Deng, B.; Kim, C.N. Computational fluid dynamics (CFD) modelling of UV disinfection in a closed-conduit reactor. Chem. Eng. Sci. 2011, 66, 4983–4990. [CrossRef]
9. Zitouni, N.; Andoulsi, R.; Selami, A.; Mami, A.; Abdennasser, H. Modelling of nonlinear pilot disinfection water system: A bond graph approach. Leonardo J. Sci. 2012, 20, 15–30.
10. Li, W.; Li, M.; Bolton, J.R.; Qiang, Z. Configuration optimization of UV reactors for water disinfection with computational fluid dynamics: Feasibility of using particle minimum UV dose as a performance indicator. Chem. Eng. Sci. 2016, 306, 1–8. [CrossRef]
11. Liu, D.; Wu, C.; Linden, K.; Ducoste, J. Numerical simulation of UV disinfection reactors: Evaluation of alternative turbulence models. Appl. Math. Model. 2007, 31, 1753–1769. [CrossRef]
12. Valadi, J.; Siarry, P. Applications of Metaheuristics in Process Engineering; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 1–39. [CrossRef]
13. Minzu, V.; Serbencu, A. Systematic Procedure for Optimal Controller Implementation Using Metaheuristic Algorithms. Intell. Autom. Soft Comput. 2020, 26, 663–677. [CrossRef]
14. Abraham, A.; Jain, L.; Goldberg, R. Evolutionary Multiobjective Optimization—Theoretical Advances and Applications; Springer: Berlin/Heidelberg, Germany, 2005; ISBN 1-85233-787-7.
15. Kruse, R.; Borgelt, C.; Braune, C.; Mostaghim, S.; Steinbrecher, M. Computational Intelligence—A Methodological Introduction, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2016. [CrossRef]
16. Talbi, E.G. Metaheuristics—From Design to Implementation; Wiley & Sons: Hoboken, NJ, USA, 2009; ISBN 978-0-470-27858-1.
17. Siarry, P. Metaheuristics; Springer: Berlin/Heidelberg, Germany, 2016; ISBN 978-3-319-45403-0.
18. Beheshti, Z.; Shamsuddin, S.M.; Hasan, S. Memetic binary particle swarm optimization for discrete optimization problems, ELSEVIER. *Inf. Sci.* 2015, 299, 58–84. [CrossRef]  
19. Minzu, V.; Barbu, M.; Nichita, C. A Binary Hybrid Topology Particle Swarm Optimization Algorithm for Sewer Network Discharge. In Proceedings of the 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 14–16 October 2015; pp. 627–634.  
20. Kennedy, J.; Eberhard, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; Volume 4, pp. 942–1948.  
21. Kennedy, J.; Eberhart, R.; Shi, Y. *Swarm Intelligence*; Morgan Kaufmann Academic Press: Cambridge, MA, USA, 2001.  
22. Hassen, A.; Mahrour, M.; Ouzari, H.; Cherif, M.; Boudabous, A.; Damelincourt, J.J. UV disinfection of treated wastewater in a large-scale pilot plant and inactivation of selected bacteria in a laboratory UV device. *Bioresour. Technol.* 2000, 74, 141–150. [CrossRef]  
23. Maurice, C. *L’optimisation par Essaims Particulaires-Versions Paramétriques et Adaptatives*; Hermes: Lavoisier, Paris, 2005.  
24. Chen, C.H.; Hwang, J.C.; Yeh, S.N. Personal Best Oriented Particle Swarm Optimizer. In *Particle Swarm Optimization*; Lazinica, A., Ed.; IntechOpen: Rijeka, Croatia, 2009; ISBN 978-953-7619-48-0.