



## Article

# A New Approach to the Optimization of Looped Water Distribution Networks with Velocity Constraints

Antonis Manolis  and Epaminondas Sidiropoulos \* 

Department of Rural and Surveying Engineering, Faculty of Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; antonismanolis@hotmail.com

\* Correspondence: nontas@topo.auth.gr; Tel.: +30-2310-996143

**Abstract:** The optimal design of a looped water distribution system is a problem that is addressed frequently in the literature. Usually, the flow velocity in the pipes is not taken into account. Nevertheless, in real-life applications, there are velocity restrictions that must be considered for the proper function of water distribution systems. An algorithm has been presented recently for the optimal design of such systems, relying entirely on the hydraulic characteristics of the system, and not involving any parameters to be adjusted. This paper presents a new suitably designed algorithm that retained the features of the original algorithm and handled the problem of velocity restrictions without recourse to penalty functions. The new algorithm was tested and compared with others that used penalty functions to handle the velocity constraints. The results demonstrated its efficiency, reliability, and better performance.

**Keywords:** optimization; water distribution networks; least-cost design; velocity constraints



**Citation:** Manolis, A.; Sidiropoulos, E. A New Approach to the Optimization of Looped Water Distribution Networks with Velocity Constraints. *Hydrology* **2022**, *9*, 73. <https://doi.org/10.3390/hydrology9050073>

Academic Editor: Vasilis Kanakoudis

Received: 30 March 2022

Accepted: 24 April 2022

Published: 27 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Looped water distribution networks are commonly used to supply water in urban areas. The purpose of these networks is to provide a specific amount of water in specific nodes of the network at a specific pressure. They consist of several components, such as pipes, pumps, reservoirs, valves, fittings, and other hydraulic appurtenances, the characteristics of which must be chosen judiciously. Out of the decisions that have to be made, the selection of pipe diameters is of crucial importance. The optimal selection of diameters is the one that leads to the most economical design, while the network also meets the preset targets and satisfies environmental, geometric, or functional restrictions.

The problem of optimal network design has attracted great interest from the engineering community over the past decades, with hundreds, and possibly thousands of research publications on the subject. In the vast majority of these publications, no restrictions on flow velocity were considered. However, in real-world applications and in technical specifications of many countries, flow velocity restrictions are imposed or at least recommended. The reason is that low velocities lead to sedimentation and high velocities to erosion of the pipes.

In the present research, the problem of velocity restrictions in a water distribution system was addressed, in relation to a recently presented targeted path search algorithm [1] based on graph-theoretic concepts and the hydraulic characteristics of the flow, without including any parameters to be adjusted. A new algorithm, named targeted path search with velocity restrictions (TPSVR), is introduced that retains the features of TPS and handles the velocity constraints without the use of penalty functions or any adjustment parameters.

## 2. Problem Formulation

The general mathematical formulation of the single-objective diameter selection optimization in a water distribution network (WDN) is well established, and can be found in the literature [2]. The WDN layout is dictated by the structure of the area to be served [3],

and is considered as predefined in the present research. The demands at the nodes are also assumed to have been specified. The objective is to find an optimal combination of pipe diameters that leads to the most economical design while providing specific amounts of water, under a specific minimum pressure, from the supply to the demand junctions of the network. The diameters must be chosen from a given set of commercially available diameters, and the flow velocity in the pipes must be within certain limits. Thus, the optimization problem is summarized as follows:

Let

$$C = \sum_{i=1}^{np} c_i(D_i) \times L_i \quad (1)$$

be the objective function to be minimized, where:

- $C$  is the total cost of the network pipes;
- $D_i$  is the internal diameter of pipe  $i$ ;
- $c_i$  is the cost per unit length of pipe  $i$  as a function of  $D_i$ ;
- $L_i$  is the length of pipe  $i$ ;
- $np$  is the number of pipes in the network.

The constraints of the problem are:

$$\sum_{i \in in(j)} Q_i - \sum_{i \in out(j)} Q_i = q_j, \quad j = 1, 2, \dots, nj \quad (2)$$

$$\sum_{i \in l} hf_i = 0, \quad l = 1, 2, \dots, nl \quad (3)$$

$$H_j > H_{j \min}, \quad j = 1, 2, \dots, nj \quad (4)$$

$$V_{\min} \leq V_i \leq V_{\max}, \quad i = 1, 2, \dots, np \quad (5)$$

$$D_i \in A, \quad i = 1, 2, \dots, np \quad (6)$$

where:

- $Q_i$  is the flow rate in pipe  $i$ ;
- $in(j)$  is the set of pipes with flow entering junction  $j$ ;
- $out(j)$  is the set of pipes with flow leaving junction  $j$ ;
- $q_j$  is the demand at junction  $j$ ;
- $hf_i$  is the head loss in pipe  $i$ ;
- $H_j$  is the hydraulic head at junction  $j$ ;
- $H_{j \min}$  is the minimum allowable head at junction  $j$ ;
- $V_i$  is the flow velocity in pipe  $i$ ;
- $V_{\min}$  is the minimum and  $V_{\max}$  the maximum allowable velocity;
- $A$  is a set of commercially available diameters;
- $nj$  is the number of the network junctions;
- $nl$  is the number of loops in the network.

Equation (2) expresses the conservation of mass at junction  $j$ , and Equation (3) gives the conservation of energy in loop  $l$ . Head loss is commonly given by the Hazen–Williams (7) or Darcy–Weisbach (8) equations:

$$hf_i = \omega \frac{L_i Q_i^\alpha}{C_i^\alpha D_i^\beta}, \quad i = 1, 2, \dots, np \quad (7)$$

$$hf_i = f_i \frac{L_i V_i^2}{D_i 2g}, \quad i = 1, 2, \dots, np \quad (8)$$

where:

- $\omega$ ,  $\alpha$ , and  $\beta$  are constants and  $C_i$  is the roughness coefficient of the Hazen–Williams equation;
- $f_i$  is the friction factor of the Darcy–Weisbach equation;

- $g$  the acceleration of gravity.

The system of Equations (2), (3) and (7) or (8) was solved by means of the simulation package EPANET [4] in most of the published articles in the relevant literature. In the present study, the gradient algorithm of Todini and Pilati [5] was applied for the same purpose. This algorithm is also used by EPANET, thus allowing direct comparison with the results of related papers.

For the head-loss Equations (7) and (8), the same values of the parameters were used, as in EPANET 2.0. Specifically,  $\omega = 10.667$  for SI units or  $\omega = 4.727$  for imperial units, and  $\alpha = 1.852$  and  $\beta = 4.871$  for the Hazen–Williams Equation (7). The friction factor of the Darcy–Weisbach equation was calculated using a Hagen–Poiseuille formula for  $Re$  (Reynolds number)  $< 2000$ , with a cubic interpolation from a Moody diagram for  $2000 < Re < 4000$ , and by the Swamee–Jain approximation to the Colebrook–White equation for  $Re > 4000$  (Rossman 2000 [4]).

Equation (5) expresses the flow velocity restrictions in their simplest form, in which the velocity limits are independent of the diameter of the pipe. In some national technical specifications or guidelines, the velocity limits depend on the diameter of the pipe, and in that case, Equation (9) replaces Equation (2):

$$V_{min}(D_i) \leq V_i \leq V_{max}(D_i), \quad i = 1, 2, \dots, np \quad (9)$$

### 3. Previous Research

Velocity restrictions were taken into account in very few published reports on the subject. Penalty functions are the most obvious and common way to tackle velocity restrictions. A typical penalty function formulation is given in the following:

$$PF = C \left( c_1 \sum_{i=1}^{np} \max[(V_i - V_{max}), 0] + c_2 \sum_{i=1}^{np} \max[(V_{min} - V_i), 0] \right) \quad (10)$$

where  $c_1$  and  $c_2$  are coefficients that determine the “weight” attached to maximum and minimum violation degrees, respectively.

Penalty functions have been presented by Jabbari et al. [6] and Fallah et al. [7]. In the first of these papers, the central force optimization (CFO, [8]) was employed, while the second one used the crow search algorithm (CS, [9]), as well as a new variant of the original CS. In both cases, the velocity limits were not reported. It is, therefore, uncertain whether they affected the final solution and to what extent. However, among the results of the first paper, a solution seems to have been accepted in which the flow velocity in a pipe was 0.11 m/s, a velocity with a very low value. In the second paper, water velocities were not reported.

Another way of dealing with velocity restrictions is to incorporate the velocity constraints in a linear programming model. This approach was presented by Samani and Mottaghi [10], Samani and Zangeneh [11], and Haghghi et al. [12]. In the first case, integer linear programming (ILP) was used; the second one used mixed integer linear programming (MILP); and in the third, a hybrid of genetic algorithms (GAs) with ILP were used. One disadvantage of the method is that a new inequality must be imported into the model for every pipe in the network. The aforementioned methods were applied with such “loose” boundaries that the velocity restrictions did not seem to affect the final solution.

### 4. Solution Method

As in TPS, the general idea of TPSVR consists of finding optimal paths while considering hydraulic gradients and suitable diameter reductions through an iterated search carried out in three consecutive subroutines, called path finding, universal reduction, and consecutive reduction. However, embedding mechanisms for ensuring the satisfaction of velocity constraints involve a considerable additional treatment. The universal reduction subroutine is more strongly engaged in satisfying these additional restrictions. More specif-

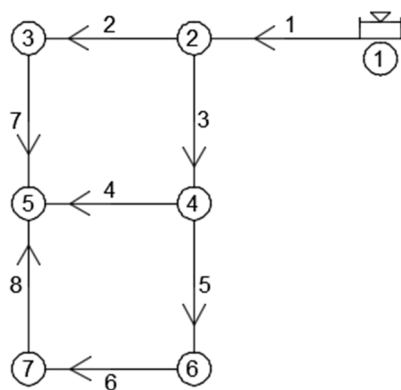
ically, the velocity in an individual pipe is regulated based on heuristic rules involving the pipe in question, as well as the pipes connected to it. These rules are based on the hydraulics of flow behavior in a network, as described below in Section 5.

The exact implementation of these mechanisms is described in detail in Section 6, both in the form of pseudocodes and in flow charts, while also showing the algorithmic steps of coupling the satisfaction of the velocity constraints in the pipes with the pressure constraints in the junctions. It can be noted here that the overall TPSVR scheme retained the advantage of not involving adjustment parameters or penalty functions.

### 5. Flow Velocity in the Looped Network

When the flow rate in the pipes of a network is constant and known, as in a branched network, the change in flow velocity in a pipe as a function of the change in its diameter can be easily found using the equation  $V = 4Q/\pi D^2$ . It is obvious that an increase in the pipe's diameter will lead to a decrease in the flow velocity and vice versa. However, the same is not true in a looped network, in which the flow rates are a priori unknown. In the looped network generally, if the diameter of a pipe increases, its flow rate also increases. Therefore, in the looped network, it is uncertain whether the flow velocity will increase or decrease, with an increase of the pipe's diameter.

Let us examine the two-loop network that was introduced in the literature by Alperovits, E. and U. Shamir [13]. They designed it for a problem without velocity restrictions and attempted to solve it using a variation of linear programming. Since then, it has been used in countless studies to test and evaluate new algorithms. Its data were provided by Alperovits and Shamir, and its layout is presented in Figure 1. The optimum solution to this problem has already been found by many algorithms in the literature. It is given by the selection of diameters in mm: (457.2, 254, 406.4, 101.6, 406.4, 254, 254, and 25.4). In the course of a numerical experiment, let us vary the diameter of the last pipe over the whole range of available diameters, while keeping all the other diameters of the network constant. Then, a velocity versus diameter diagram can be created for the last pipe, such as the one in Figure 2.



**Figure 1.** Layout of the two-loop network.

The shape of the curve in Figure 2 above also appears when examining any other individual pipe in any looped network. The shape of the curve confirms that, generally, it is uncertain whether a pipe's diameter should be increased or decreased in order to increase the flow velocity in the pipe. When trying to manipulate the flow velocities in the looped network, it should also be considered that changing the diameter of a pipe in a looped network will affect the velocities in all the other pipes in the network. Figure 3 shows the change in the flow rate of the same pipe. Again, the shape of the curve is similar to any pipe in any looped network.

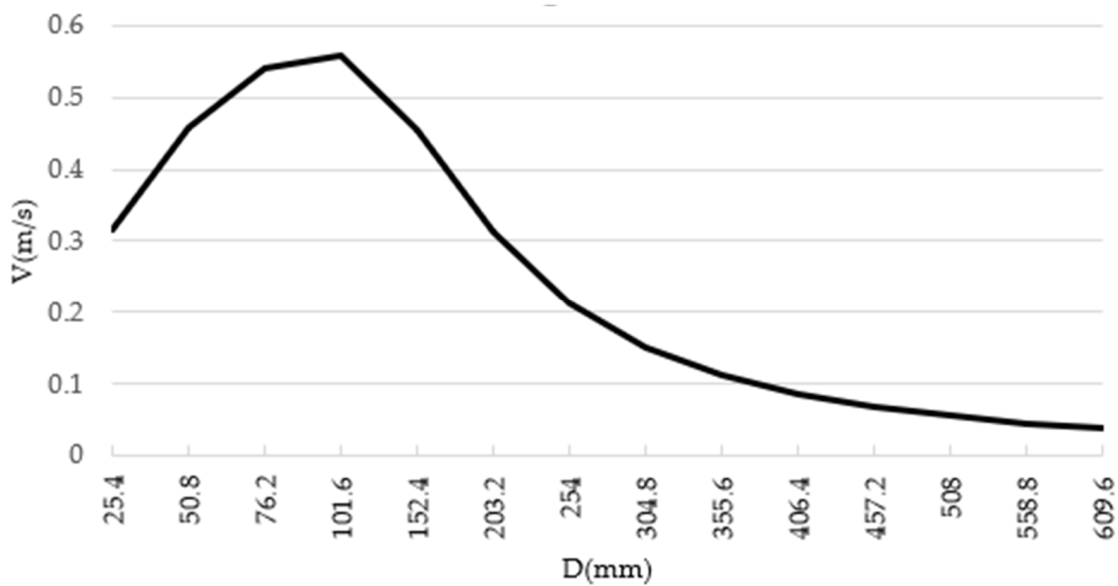


Figure 2. Change in velocity of pipe 8 in the two-loop network.

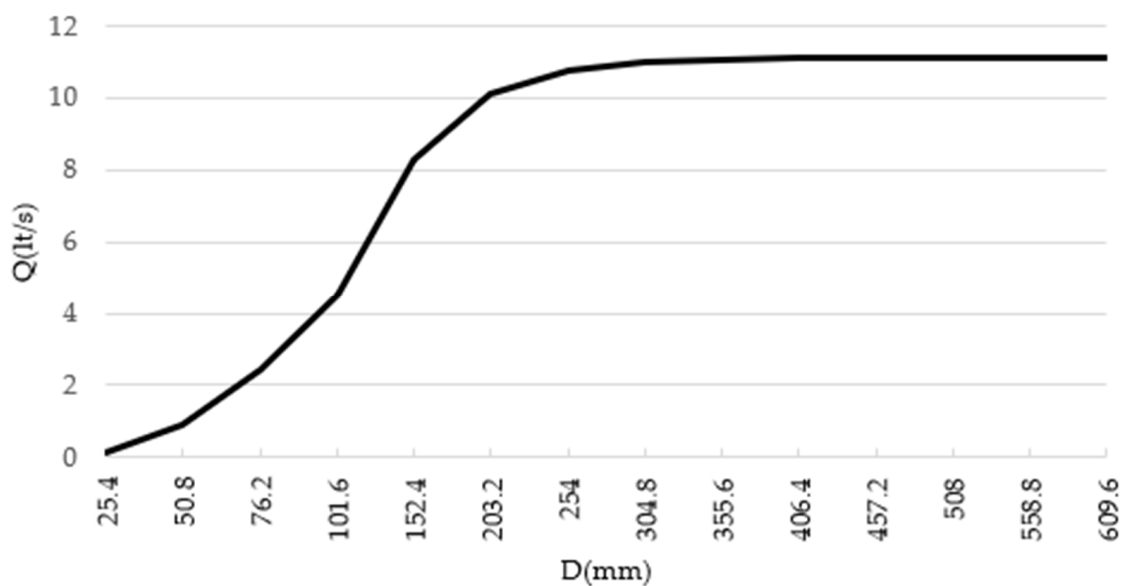
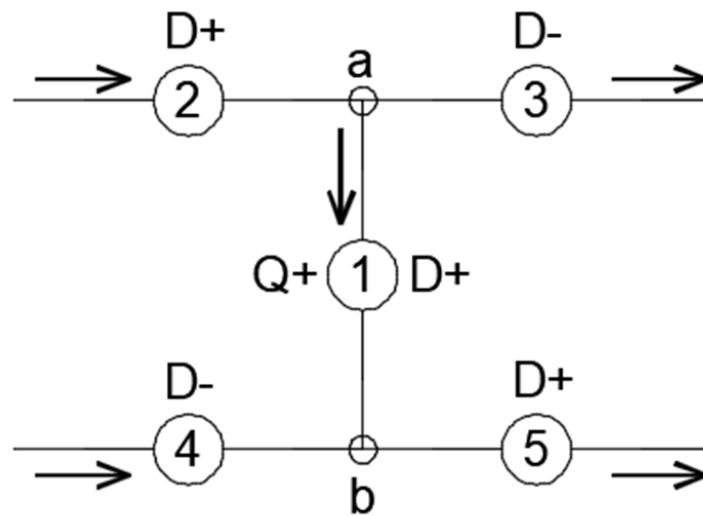


Figure 3. Change in flow rate of pipe 8 in the two-loop network.

It can therefore be concluded that if the flow velocity in a pipe is out of allowable bounds, it is not certain whether decreasing or increasing its diameter will fix the problem. Furthermore, in some cases in which the pipe's velocity is lower than the minimum, the problem may be impossible to solve with any diameter selection. This is because the maximum of the curve in Figure 2 may be below the minimum velocity bound. A way to correct the velocity is to adjust the flow rate of the pipe by changing the diameters of the adjacent pipes. This will certainly lead to a desirable result. For example, if the flow rate of a pipe increases with an appropriate change in the diameter of an adjacent pipe, then it is certain that its flow velocity will also increase.

Stipanić et al. [14] presented an interesting study showing the behavior of flow velocity in a network. Their research was not about optimizing the network from an economic point of view, but some of their conclusions were used in the present analysis. A section of a hypothetical water distribution network is presented in Figure 4.



**Figure 4.** Different options for the increase in the flow velocity in a pipe.

The cases analyzed below for adjusting flow rates and velocities can be applied respectively to any part of any looped network. Figure 4 shows how the flow rate in pipe 1 can be increased. This can be done in five ways. The most obvious one is the increase in the diameter of pipe 1 itself, as can be seen in Figure 3. Two others, according to the conservation of mass (Equation (2)) at node a, include:

- Increasing the diameter of pipe 2;
- Decreasing the diameter of pipe 3.

In addition, the remaining two, according to the conservation of mass equation at node b, include:

- Decreasing the diameter of pipe 4;
- Increasing the diameter of pipe 5.

These changes can also be applied to the sections of pipes connected to the pipes described above. For example, it has been mentioned that to increase the flow rate of pipe 1, the diameter of pipe 2 can be increased. Respectively, the diameter of any pipe upstream of pipe 2 can be increased with the same effect. The reverse changes have the opposite effect.

## 6. Targeted Path Search with Velocity Restrictions

The conclusions described above were incorporated in the original TPS algorithm in order to formulate a new, augmented targeted path search with velocity restrictions (TPSVR) algorithm. Figure 5 presents the flowchart of the TPSVR. The algorithm was guided by three subroutines, specifically the path finding, universal reduction, and consecutive reduction subroutines. The path finding subroutine remained the same as it is in the TPS algorithm. In the consecutive reduction subroutine, the only difference was that new solutions were only accepted if the velocity restrictions were not violated. Therefore, only the modified universal reduction subroutine was reanalyzed in the next subsection.

In Process 2 of the flowchart, the flow rate that corresponded to the minimum allowable flow velocity  $V_{min}$  was calculated, since it was used in the universal reduction subroutine. These flow rates were calculated for every commercially available diameter. Thus, the set  $Q_{min} = \{Q_{min_i}, i = 1, 2, \dots, nav\}$  was obtained:

$$Q_{min_i} = V_{min} \times \pi D_i^2 / 4, \quad i = 1, 2, \dots, nav \quad (11)$$

where  $nav$  is the number of available diameters.

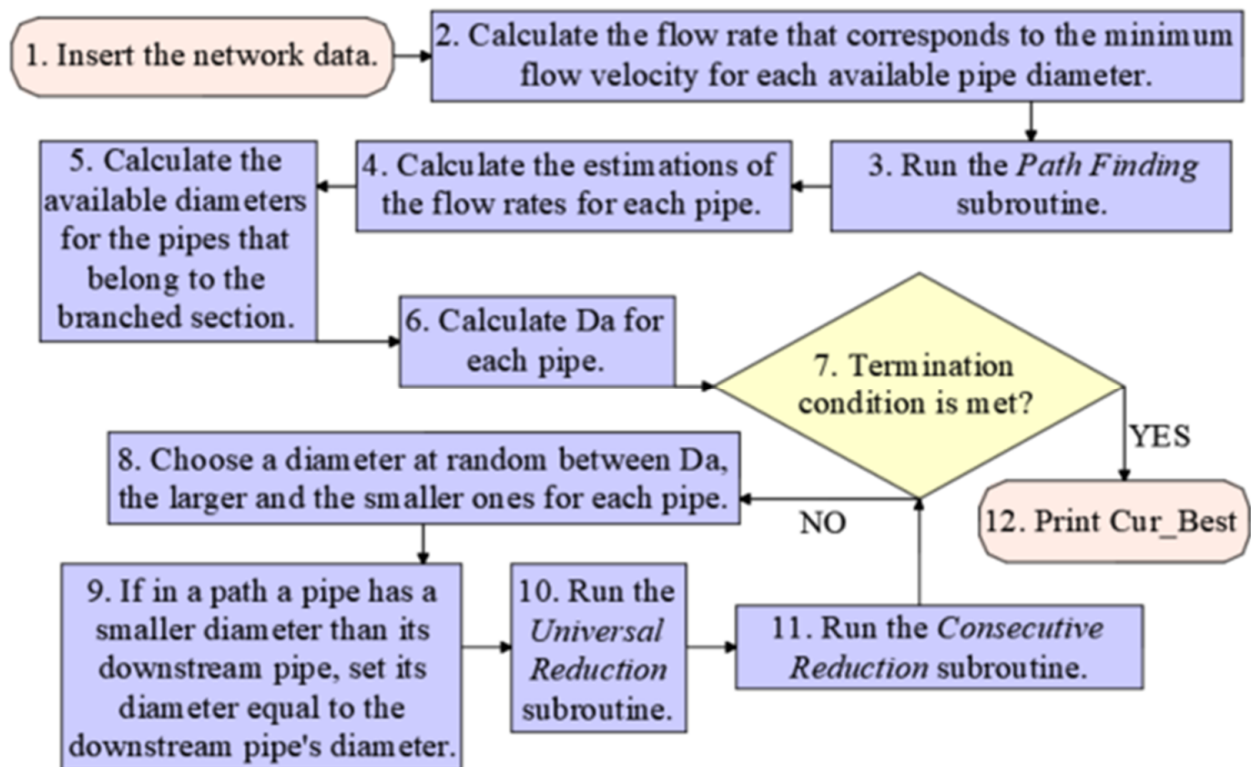


Figure 5. Flowchart of the TPSVR algorithm.

In Process 3 of the algorithm, the path finding subroutine was executed. Its purpose was to identify the most probable paths of water in the looped network. With that knowledge, the flow rates of the pipes could be estimated in Process 4. These were calculated by adding the flow rate of the downstream pipe to the demand of the downstream junction. If the network had branched sections, the flow rates of the pipes of these sections were the final ones. In the looped sections of the network, the flow rate of each pipe was estimated using the water flow paths that were derived from the path finding subroutine. When there were paths that ended in the same junction, the slope  $SL$  was used to distribute the demand of the junction to each path:

$$SL_{sj} = \frac{H_s - H_{jmin}}{L_{sj}} \quad (12)$$

Where  $H_s$  the hydraulic head in the supply junction  $s$ , and  $L_{sj}$  is the total length of the path that connects  $s$  to the demand junction  $j$ . If, in the junction  $j$  with demand  $q_j$ , end  $n$  paths with slopes  $SL_{sj,i}$ , where  $i = 1, 2, \dots, n$ , then the pipes of each path  $p_{sj,i}$  will also carry the following demand:

$$q_j \frac{SL_{sj,i}}{\sum_{i=1}^n SL_{sj,i}} \quad (13)$$

The addition of the velocity restriction constraint (5) to the classical optimization problem made it more difficult to solve it, as with any additional constraint. Nevertheless, there was an advantage that could be utilized. If the network had branched sections, it was possible to reduce the available diameters for the pipes of these sections. Since the flow rate of these pipes was known in advance from Process 4, it was easy to select as available only the diameters for which the water velocity would be within the predetermined limits. Solving the equation  $V = 4Q/\pi D^2$  for  $D$ , substituting into Inequalities (5) and considering Equation (6), the minimum and maximum diameter of these pipes could be found. This advantage was not offered in the looped sections of the network, since the actual flow rates of the pipes were unknown. This was performed in Process 5 of the algorithm.



In Process 6, the initial diameters  $Da$  for all the pipes were assigned. Since the estimates of the flow rates of the pipes were calculated in Process 4, if a desired flow velocity was set, estimates of the diameters of the pipes could also be calculated. This flow velocity was set to 1 m/s, since this value is generally accepted as desirable in new networks. This value was chosen because it was about halfway through the velocity limits, and because it led to relatively small losses, since the head losses increased according to a power law as velocity increased. Therefore, if an initial velocity  $Va = 1$  m/s is assigned and if  $Q_{ai}$ ,  $i = 1, 2, \dots, np$  are the flow rate estimations that were calculated in Process 4, then the initial diameters will be:

$$Da_i = \sqrt{\frac{4 \times Q_{ai}}{\pi}}, \quad i = 1, 2, \dots, np \quad (14)$$

Since continuous values for the diameters were calculated using Equation (14), these values were rounded to the closest available discrete value, according to the constraint (6). For each pipe, the available diameter that had the smallest difference from the diameter calculated by Equation (14) was selected.

The iterations of the algorithm began at Process 7 and continued until a user-defined termination criterion was met. At the beginning of each iteration, a “quasi-random” solution was generated in Process 8. For each pipe, a diameter was chosen randomly from  $Da$ , the smaller and the larger one, if of course, they existed in those available. At this point, there was a key difference compared to TPS. In the TPSVR, each iteration began with a new quasi-random solution that was optimized by the other subroutines; while in the TPS, the initial solution of the iteration was the currently best-found solution. This change was introduced because it was ascertained that the TPSVR showed an inability to escape from local optima in the previous design. By starting each iteration from a new quasi-random solution, the search expanded. At the same time, initializing from the diameters that were calculated by Equation (14), or diameters that were close to them, prevented a pointless search in areas of the search space where the objective function was highly infeasible.

This initial solution was “corrected” in Process 9, as is done in TPS. The difference here was that the solution was corrected within the iteration, as a new initial solution was created in each iteration. This solution was then optimized within the universal reduction and consecutive reduction subroutines. In the first iteration of the algorithm, the solution after its optimization was registered as *Cur\_Best*. In each subsequent iteration, it was checked to determine if the resulting solution was better than *Cur\_Best*, and if it was, it replaced it.

In the TPSVR, the knowledge of the paths of flow, or equivalently the direction of flow in each pipe in the network, was crucial. Based on this information, the flow velocities were regulated in the universal reduction subroutine. It was noted that the direction of flow could change with a different selection of diameters. For this reason, in each solution of the network, a new directed graph was created that showed the directions of flow resulting from the respective selection of diameters.

It should be noted that in the use of the TPSVR, the velocity constraints were treated as hard constraints, meaning that they were required to be absolutely satisfied. If the intention was to treat them as soft constraints, then the use of a penalty function would probably have been more efficient.

#### *Universal Reduction Subroutine of the TPSVR*

The mechanism that regulated the flow velocities in the network was incorporated in the universal reduction subroutine. The solution that was produced by Processes 8 and 9 given in Figure 5 was initially imported into the subroutine.

Then, the procedure that followed can be described as follows in the form of a pseudocode and in Figure 6 in the form of a flowchart.



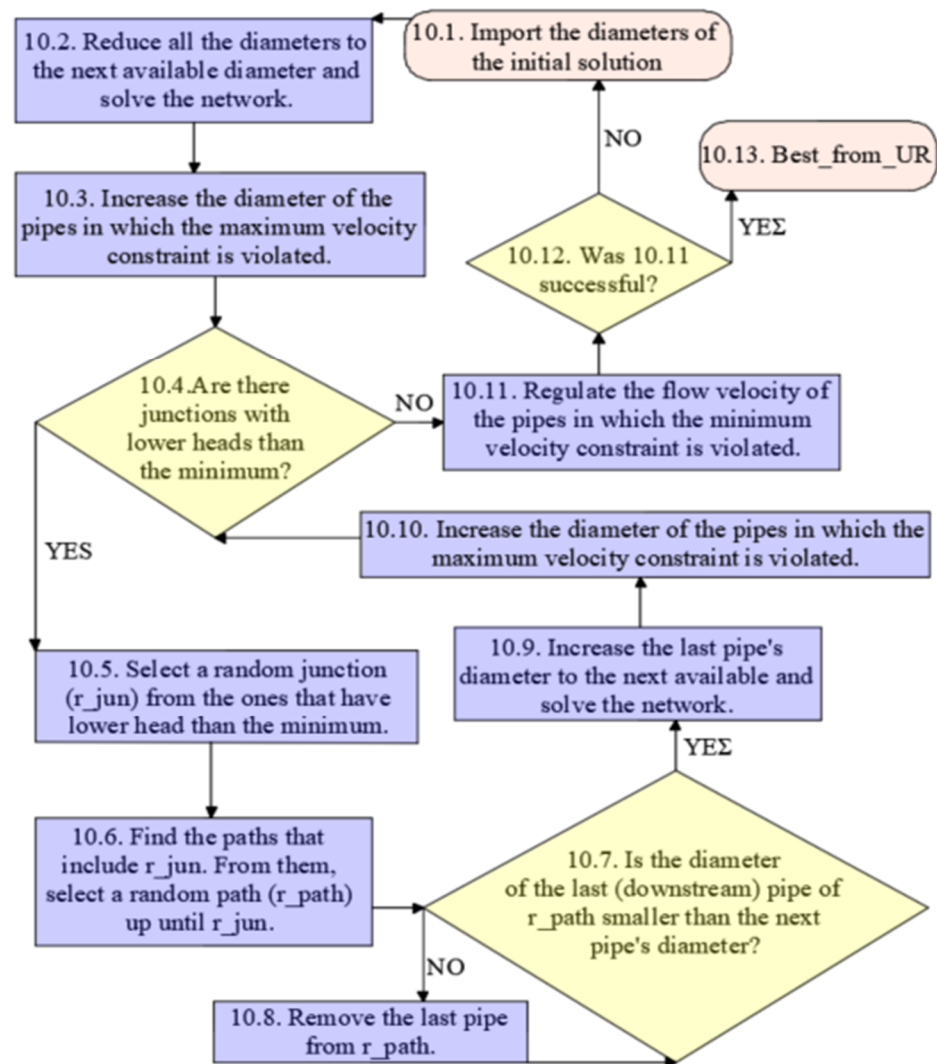


Figure 6. Flowchart of the universal reduction subroutine.

1. **For** count = 1 to number of available diameters:
  - 1.1 The diameters from the initial solution are reduced synchronously to the next available ones (Process 10.2).
  - 1.2 The network is solved via EPANET.
  - 1.3 An increase is given to the diameters of the pipes with maximum velocity violation.
  - 1.4 A list of junctions  $J_{vw}$  results the hydraulic head of which is lower than allowed:  $J_{vw} = \{J_{vw_1}, J_{vw_2}, \dots, J_{vw_{nh}}\}$  where  $nh$  is the number of the junctions with minimum head violation.
  - 1.5 **While**  $J_{vw}$  is not empty:
    - 1.5.1 Randomly select one of the elements of  $J_{vw}$ .
    - 1.5.2 Find the paths that include this random junction and select one of them at random.
    - 1.5.3 If the diameter of the last pipe of this random path is smaller than the next pipe's diameter, increases that diameter.
    - 1.5.4 Solve the network via EPANET.
    - 1.5.5 Increase the diameters of the pipes with violation of the maximum velocity constraint.
    - 1.5.6 **End While**
  - 1.6 Regulate the flow velocity of the pipes with minimum velocity constraint violation.

- 1.7 **If** the above is successful:
  - 1.7.1 If the current solution is better than the best solution found until now from Universal Reduction, it replaces it.
- 1.8 **Else:**
  - 1.8.1 Import the initial solution from Processes 8 and 9.
- 1.9 count = count + 1
2. **Return** the best solution from Universal Reduction.

The main purpose of universal reduction was to find an acceptable solution, and was not necessarily the best. Increasing the diameter at this stage helps the subsequent subroutine Processes to locate a solution that does not violate the minimum pressure constraints besides the velocity constraints. Let PH be the set of pipes in which the maximum velocity limit is violated. Process 10.3 is described by the following simple pseudocode.

1. **While** PH  $\neq \emptyset$ :
  - 1.1 Chose randomly a pipe  $p, p \in \text{PH}$ .
  - 1.2 Increase the diameter of pipe  $p$  to the next available one.
  - 1.3 Solve the network and update the set PH.

In processes 10.4–10.9, the minimum hydraulic head constraint was addressed, just as in TPS. In these, and specifically in 10.9, the diameters of the pipes always increased. However, there was a small chance that this would lead to a violation of the maximum velocity limit. This was explained by the shape of the curve in Figure 2. Hence, Process 10.10 was added, which was similar to 10.3. When the programming loop 10.4–10.10 was completed, it was certain that the minimum pressure and the maximum velocity constraints were not violated. There were, however, pipes in which the minimum velocity constraint was violated.

The minimum velocity constraint was addressed in Process 10.11, which was presented due to its size in the separate flowchart of Figure 7. In Process 10.11.1, whether there were pipes in which the flow velocity was less than the minimum was checked. Of these, the pipe in which the flow velocity was the smallest is defined as the pipe with low velocity (Pwlv). In the other processes given in Figure 7, we attempted to increase the flow velocity of only this specific pipe.

The process was iterative, and there was a limit to the number of iterations. This limit was set as equal to the number of pipes in the network ( $n_p$ ), as it was proportional to the number of pipes in which the minimum velocity constraint could be violated. If this limit was exceeded, then the process had failed, and the algorithm returned to Process 10.1. It was also considered a failure not to find a solution that increased the velocity of Pwlv in one iteration, even slightly. This last feature was used as an escape route in case the algorithm stalled in an unfeasible solution from which a feasible one could not be obtained. The failure rate for all problems examined was less than 1%.

Figure 2 shows that there was a maximum flow velocity in one pipe while its diameter changed and the diameters of the other pipes were kept constant. This velocity may have been less than the minimum. Therefore, there may have been a situation in which the diameter of a pipe would fluctuate endlessly, without leading to a velocity higher than the allowed one. For this reason, a mechanism for monitoring each diameter change was introduced into the algorithm. If, in one iteration, the diameter of one pipe was reduced, in the next one it was not allowed to increase, and vice versa.

The  $Q_{\min}$  was already calculated in Process 2 of the TPSVR, and consisted of the flow rates for each available diameter with which the minimum allowable velocity was achieved. In Process 10.11.2, it is checked whether the flow rate of Pwlv,  $|Q(\text{Pwlv})|$ , was greater than the required flow rate to achieve the minimum velocity with the smallest available diameter,  $Q_{\min}$  ( $D_{\min}$ ). If this was the case, reducing the diameter of Pwlv would possibly increase its velocity. This was done in Process 10.11.3, and the selected diameter was the largest available diameter for which the minimum flow rate was less than the current flow

rate of Pwlv. Process 10.11.3 is also described by the following pseudocode. Let  $D_{av}$  be the set of the available diameters, sorted in ascending order:

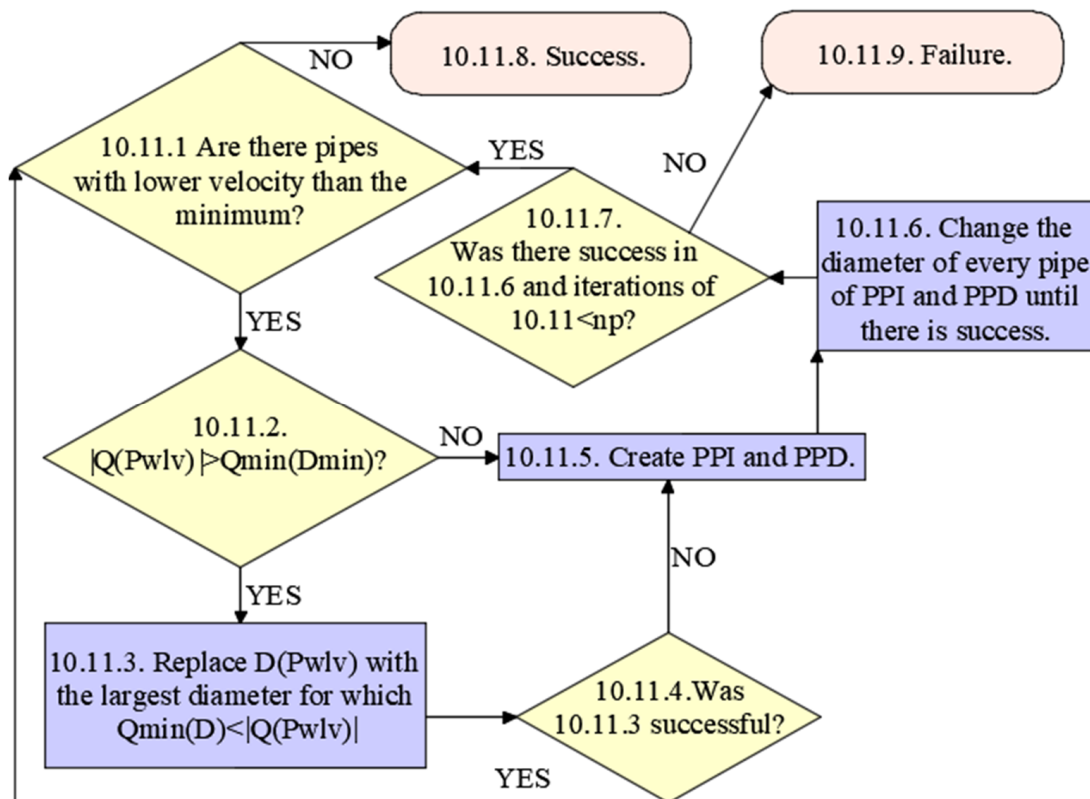


Figure 7. Minimum velocity constraint handling.

1. **For every**  $d$  of  $D_{av}$ :
  - 1.1 **If**  $Q_{min}(d) < |Q(Pwlv)|$ :
    - 1.1.1  $D(Pwlv) = d$
  - 1.2 **Else**:
    - 1.2.1 **Break**.

Figure 7 and the above pseudocode show the absolute value of the flow rate  $|Q(Pwlv)|$ , considering that the flow rate had a negative value if the flow was in the opposite direction from what was initially set in the network. Process 10.11.3 was considered to have failed if one or more of the following applied after it:

- The velocity of the Pwlv pipe had decreased;
- There were junctions with lower hydraulic heads than the minimum;
- There were pipes with a higher velocity than the maximum.

If none of the above was true, the process was considered successful, and the minimum velocity bound was rechecked. If the process failed, then the algorithm continued in Processes 10.11.5–10.11.8, with the aim to increase the flow in Pwlv.

In Process 10.11.5, two sets of pipes were created: the possible pipes increase (PPI) and the possible pipes decrease (PPD) sets. In these, the pipes that were candidates for a diameter increase or decrease were respectively stored. The PPI and PPD sets were created following the thoughts developed in Section 5 and presented in Figure 4. According to these, the following were placed in the PPI set:

- The pipes belonging to a path that ended at the upstream junction of Pwlv;
- The pipes belonging to a path that started from the downstream junction of Pwlv.

The following were placed in the PPD set:

- The pipes belonging to a path that ended at the downstream junction of Pwlv;
- The pipes belonging to a path that started from the upstream junction of Pwlv.

In all the looped networks, some pipes belonged to different paths. Therefore, these pipes may have been placed in both sets. However, the effects of changing the diameter of these pipes was uncertain. For this reason, these pipes were removed from both sets. Let us define the upstream and the downstream junction as up and down, respectively. The above can also be described by the following pseudocode.

1. **For every path** in paths that end up in junction *up* and paths that begin from junction *down*:
  - 1.1 **For every pipe** in *path*:
    - 1.1.1 **If pipe** not already in *PPI*:
      - 1.1.1.1 Insert *pipe* in *PPI*.
2. **For every path** in paths that begin from junction *up* and paths that end in junction *down*:
  - 2.1 **For every pipe** in *path*:
    - 2.1.1 **If pipe** not already in *PPD* and if *pipe*  $\neq$  *Pwlv*
      - 2.1.1.1 **If pipe** in *PPI*:
        - 2.1.1.1.1 Remove *pipe* from *PPI*.
      - 2.1.1.2 **Else if pipe** not in *PPI*:
        - 2.1.1.2.1 Insert *pipe* in *PPD*.

For the Pwlv pipe itself, its flow rate was considered to determine in which set it would be placed. Thus, it was placed in the PPD set, if its flow rate was greater than the  $Q_{min}$  corresponding to the immediately smaller available diameter. Otherwise, it was placed in the PPI set in order to increase its flow rate by increasing its diameter.

The PPI and PPD sets were sorted so that the changes in diameters of specific pipes were tested first. The PPD set was sorted based on the proximity of each pipe to the Pwlv. Proximity here refers to how many junctions there were between the two pipes, and not their distance. The PPI set was sorted first based on the proximity of each pipe to its supply junction, and then based on its diameter. The reason was that generally, it was desirable to first increase the diameters of smaller pipes. If two pipes had the same diameter, then initially the diameter increase for the pipe that was upstream in its path would be tested. In that way, the general hydraulic requirement that the upstream pipes should have larger diameters than the downstream ones was met.

In Process 10.11.6. the changes in the diameters of the PPD and PPI sets were attempted until there was a success. As previously established, success meant increasing Pwlv's flow velocity with a solution that did not violate the minimum pressure and maximum velocity constraints. However, the changes that were made to the pipes outside Pwlv now would certainly lead to an increase in the velocity of Pwlv. The process began by decreasing the diameter of the first pipe in the PPD set to the next available one. If the result was a failure, the diameter increase for the first pipe in the PPI set was attempted, then the diameter decrease for the second pipe in the PPD set, and so forth. Naturally, if a change in a diameter failed, the initial diameter was restored. If a change was successful, then the process stopped, and the algorithm returned to the evaluation of the velocity of the pipes (Process 10.11.1). If all possible changes failed, then Process 10.11 was considered to have failed, and the algorithm returned to the input of a new initial solution.

## 7. Application Example

For a better understanding of Process 10.11, an example based on the two-loop network was used. The layout of the network is presented in Figure 1, in which the arrows show the flow direction for the diameters applied in the example. These flow directions were kept constant during the procedure. Obviously, this did not mean that these directions could not change with a different selection of diameters. In all the iterations of the example, pipe 8 was the Pwlv.

In the example in Table 1, the flow velocity correction of pipe 8 was attempted, and was initially lower than the minimum (0.5 m/s). The process was successful since with the final selection of diameters, the flow velocity in pipe 8 was 0.56 m/s. In addition, the flow velocity bounds in the other pipes and the minimum pressure bound in any junction were not violated. In each iteration, it was checked whether the flow rate of pipe 8 was greater than the minimum flow results for the minimum available diameter. The minimum flow rates for each available diameter are presented in Table 2.

Table 1. Application example of Process 10.11.

Pipe	1	2	3	4	5	6	7	8
<b>Iteration</b>	1							
Diameter (mm)	558.8	203.2	457.2	203.2	355.6	254	152.4	152.4
Velocity (m/s)	1.27	1.4	1.45	1.6	1.54	1.2	0.96	0.3
Flow rate (m <sup>3</sup> /h)	1120	163.25	856.75	187.02	549.73	219.73	63.25	19.73
$ Q(Pwlv)  > Qmin(Dmin)$	YES							
Note	The diameter 101.6 was chosen for pipe 8.							
<b>Success</b>								
<b>Iteration</b>	2							
Diameter (mm)	558.8	203.2	457.2	203.2	355.6	254	152.4	101.6
Velocity (m/s)	1.27	1.42	1.45	1.65	1.51	1.16	1	0.39
Flow rate (m <sup>3</sup> /h)	1120	165.55	854.45	192.95	541.49	211.49	65.55	11.49
$ Q(Pwlv)  > Qmin(Dmin)$	YES							
Note	The diameter 76.2 mm was chosen for pipe 8.							
<b>Success</b>								
<b>Iteration</b>	3							
Diameter (mm)	558.8	203.2	457.2	203.2	355.6	254	152.4	76.2
Velocity (m/s)	1.27	1.43	1.44	1.68	1.5	1.13	1.02	0.39
Flow rate (m <sup>3</sup> /h)	1120	166.98	853.02	196.6	536.43	206.43	66.98	6.43
$ Q(Pwlv)  > Qmin(Dmin)$	YES							
Note	The diameter 50.8 mm was chosen for pipe 8.							
<b>Failure</b>	The flow velocity in pipe 8 was decreased to 0.34 m/s.							
PPD	{4. 7. 2}							
PPI	{6. 5}							
Try	The diameter of pipe 4 was reduced to 152.4 mm							
<b>Failure</b>	The flow velocity in pipe 4 was higher than the maximum, and the pressure in junction 5 was lower than the minimum.							
Try	The diameter of pipe 6 was increased to 304.8 mm.							
<b>Success of the Process</b>								
Diameter (mm)	558.8	203.2	457.2	203.2	355.6	304.8	152.4	76.2
Velocity (m/s)	1.27	1.42	1.44	1.67	1.51	0.8	1.01	0.56
Flow rate (m <sup>3</sup> /h)	1120.0	166.22	853.78	194.66	539.12	209.12	66.22	9.12

Since, in this example, the flow rate of pipe 8 was greater in every iteration than the minimum results for the minimum diameter, in each iteration, the reduction in pipe's 8 diameter was tested according to Process 10.11.3. If this failed, the PPD and PPI sets were

created. According to Figure 1, and because the flow direction does not change, in all iterations, it was  $PPD = \{4, 7, 2\}$  and  $PPI = \{6, 5\}$ . In the PPD set, pipes 4 and 7 were inserted before pipe 2, as they were closer to pipe 8. Pipe 4 was inserted before pipe 7 randomly, as their proximities to pipe 8 were the same. In the PPI set, pipe 6 was inserted before pipe 5 because it had a smaller diameter. Pipes 1 and 3 could be inserted in both sets as long as they led to both the upstream and downstream junction of pipe 8, and thus, they were not inserted in either.

**Table 2.** Minimum flow rate for each available diameter.

<b>Dav (mm)</b>	<b>Qmin (m<sup>3</sup>/h)</b>
25.4	0.91
50.8	3.65
76.2	8.21
101.6	14.59
152.4	32.83
203.2	58.37
254	91.21
304.8	131.34
355.6	178.77
406.4	233.49
457.2	295.51
508	364.83
558.8	441.44
609.6	525.35

If Process 10.11.3 failed, as in iteration 3, the following actions were performed with the following specific order until there was a success:

1. Reducing the diameter of pipe 4;
2. Increasing the diameter of pipe 6;
3. Reducing the diameter of pipe 7;
4. Increasing the diameter of pipe 5;
5. Reducing the diameter of pipe 2.

If an action failed, then the next diameter change was examined. If they all failed, then the entire process was considered to have failed, and the algorithm returned to the initial solution. The same was true if the number of iterations exceeded the number of the network pipes, which, in this example, was 8. It should be noted that the pipe with the lowest velocity ( $Pwlv$ ) and the PPD and PPI sets were recalculated at each iteration, as they generally changed.

## 8. Applications of the TPSVR

The TPSVR was implemented in the same networks as the original TPS algorithm for better comparison. These were probably the most popular benchmark networks in the literature, specifically the two-loop, the Hanoi, and the Balerna networks. To test the efficiency of the TPSVR, a selection of the most popular optimization algorithms, along with TPS, were coded and applied to these networks with the velocity constraints of Equation (5). All the algorithms were coded in Python V3.6 and were executed on a personal HP laptop with an AMD 2.9 GHz processor. In all the applications, the minimum and maximum velocity were set as  $V_{min} = 0.5$  m/s and  $V_{max} = 2.0$  m/s. The penalty function that was used was that of Equation (10), with  $c_1 = c_2 = 10$ . These algorithms are presented in Table 3. They were coded according to the guidelines given by the authors of Table 3 when they applied them to the problem without velocity restrictions. The parameters that were used were the ones that produced the best results according to the respective authors. Note that the TPSVR, as well as the TPS, did not require any parameters.



**Table 3.** Algorithms used in the applications alongside TPSVR.

Algorithm	Abbr.	Parameters
Differential evolution [15]	DE	$F = 0.7, CR = 0.8$
Genetic algorithms [2]	GA	$p_{cross} = 0.9, p_{mut} = 0.05, r_{cross} = 0.5$
Harmony search [16]	HS	$HMCR = 0.9, PAR = 0.1$
Particle swarm optimization [3]	PSO	$C_1 = 1, C_2 = 1$
Simulated annealing [17]	SA	$a = 0.3, r = 0.2, n_1 = \text{varies}, n_2 = 10$
Shuffled complex evolution [18]	SCE	$p = \text{varies}, m = 20, q = 10, a = 1, b = 20$
Targeted path search [1]	TPS	-

With the intention of testing the algorithms' performances in different stages of their executions, three termination criteria were used for the first two networks, based on the maximum allowable function evaluation number (FEN). The FEN was set equal to 1000, 10,000 and 40,000. In the Balerma network, the FEN was set equal to 45,400, on one hand because smaller numbers were meaningless for such a large network, and on the other hand because this number was used extensively in the literature ([19,20] and others). The population sizes involved in the algorithms in Table 1 were set at 50 for the first stopping criterion, and 100 for the second and third ones and for the Balerma network. The parameter  $n_1$  of the SA algorithm was set equal to 70, 700, 200, and 3000 for the three criteria and for the Balerma network, respectively. Regarding the parameter  $p$  of the SCE algorithm, it was set equal to 2, 4, 5, and 5, respectively, for the same criteria and for the Balerma network.

Due to the stochastic nature of all the algorithms that were used, 20 independent runs of the programs were executed for each criterion and each algorithm. Nevertheless, independent runs did not make sense for the TPSVR, since in each iteration, the algorithm began from an independent random solution. In essence, whether there were five independent runs of 10 iterations or one of 50 iterations, it made no difference.

### 8.1. The Two-Loop Network

The layout of the network is presented in Figure 1. The best value of the objective function to the problem without velocity restrictions was found to be 419,000 arbitrary monetary units. This was found by many different algorithms, so it was very probable that this was in fact the optimum. In this solution, the flow velocity in pipe 8 was 0.31 m/s, which was less than the minimum allowable (0.5 m/s), and therefore was infeasible for the problem with velocity restrictions.

The results of the 20 independent runs are presented in Table 4, and they were very promising for the new algorithm. The best value of the objective function that was found was 426,000 arbitrary monetary units, which is presented in Table 5. Interestingly, this solution had different diameters for several pipes compared to the corresponding best solution without velocity restrictions. That value was found by the TPSVR 2 times when the FEN was set equal to 1000, 12 times when it was equal to 10,000, and every time when it was set equal to 40,000. The fastest execution of the TPSVR in the ones for which the best value was found lasted 0.05 s and required 98 evaluations of the objective function. The slowest lasted for 18s and required 24,171 evaluations. When the results of the other algorithms were compared, the best performance found was that of the DE for the FEN of 10,000 and the FEN of 40,000. Regarding the performance of the TPS algorithm, it was evident that it provided good results for low values of the FEN, but it was unable to locate the best-found solution for higher values of the FEN. It is noted here that, as already stated, the TPS was enhanced by the inclusion of a penalty function.



**Table 4.** Results for the two-loop network.

FEN		TPSVR	DE	GA	HS	PSO	SA	SCE	TPS
1000	Minimum	4.26	5.81	4.55	4.58	6.05	4.42	4.98	4.40
	Mean	4.43	8.19	5.34	5.44	11.85	5.34	28.16	4.46
	Maximum	4.53	14.72	7.09	6.42	18.48	6.83	67.72	4.61
10,000	Minimum	4.26	4.29	4.26	4.27	4.33	4.27	4.71	4.40
	Mean	4.32	4.37	4.79	4.80	7.40	5.01	5.32	4.41
	Maximum	4.44	4.58	5.86	5.52	14.36	6.11	6.95	4.44
40,000	Minimum	4.26	4.26	4.41	4.26	4.26	4.27	4.39	4.40
	Mean	4.26	4.32	4.66	4.58	6.43	4.96	4.69	4.40
	Maximum	4.26	4.42	5.48	5.36	12.58	6.76	5.44	4.40

Numerical results are presented in ( $\times 10^5$ ) arbitrary monetary units.

**Table 5.** Best solution for the two-loop network.

ID	Pipes			Junctions	
	Diameter (in)	Diameter (mm)	Velocity (m/s)	ID	Pressure (m)
1	20	508	1.53	2	55.95
2	10	254	1.97	3	31.56
3	16	406.4	1.42	4	46.45
4	1	25.4	0.52	5	33.69
5	14	355.6	1.51	6	30.50
6	10	254	1.15	7	30.18
7	10	254	1.42		
8	3	76.2	0.60		

### 8.2. The Hanoi Network

The Hanoi network was introduced by Fujiwara and Khang [21], and also has been used extensively for the evaluation of new algorithms. The value of the objective function in the best-found solution without velocity restrictions was USD 6,081,128. It has also been found by many algorithms such as GA [2], NLP-DE [15], and others. In this solution, the flow velocity in pipes 1, 2, 3, 4, 17, 19, and 20 was higher than allowed; while in pipes 16, 28, and 31, it was lower. In fact, a solution with the specific velocity restrictions and the available diameters was impossible. This was due to the lack of available diameter to serve the flow rate of pipes 1 and 2 concerning the maximum velocity bounds. The flow rate of pipes 1 and 2 was constant and equal to 5.539 and 5.292 m<sup>3</sup>/s, respectively. If the maximum available diameter of 1016 mm was selected for these pipes, the resulting velocities were 6.83 and 6.53 m/s, respectively. These velocities were much higher than the maximum allowed, which was 2.0 m/s. The layout of the network is presented in Figure 8.

To enable the network to be solved with the specified velocity restrictions while keeping its other data identical, two new available diameters were added. These are 55 and 75 inches (in) or 1397 and 1905 millimeters (mm). In the Hanoi network, the cost of pipes in USD/m was set equal to  $1.1 \times D^{1.5}$ , where  $D$  is the diameter in inches. So, the corresponding costs of the two new diameters were USD 448.68 and USD 714.47/m, respectively. Practically, only the diameter of 75 inches was required for the network to be solvable. The 55-inch diameter was introduced into the problem for the sake of uniformity with the other available diameters. With a diameter of 75 inches, the flow velocities of pipes 1 and 2 were 1.94 and 1.86 m/s, respectively, so they were within the velocity bounds. The results of the tested algorithms are presented in Table 6.

The good performance of the TPSVR was shown in this network as well, since it generally managed to find better solutions than the other algorithms. It was the only algorithm that managed to find reliable solutions, even when the FEN was set equal to 1000, while the other algorithms produced highly unfeasible ones. The SA and TPS, enhanced by

a penalty function, also showed good performance in this problem. The best solution that was found is presented in Table 7. The value of the objective function was USD 7,209,104.24. In this solution, the flow velocities varied between 0.58 and 2.00 m/s.

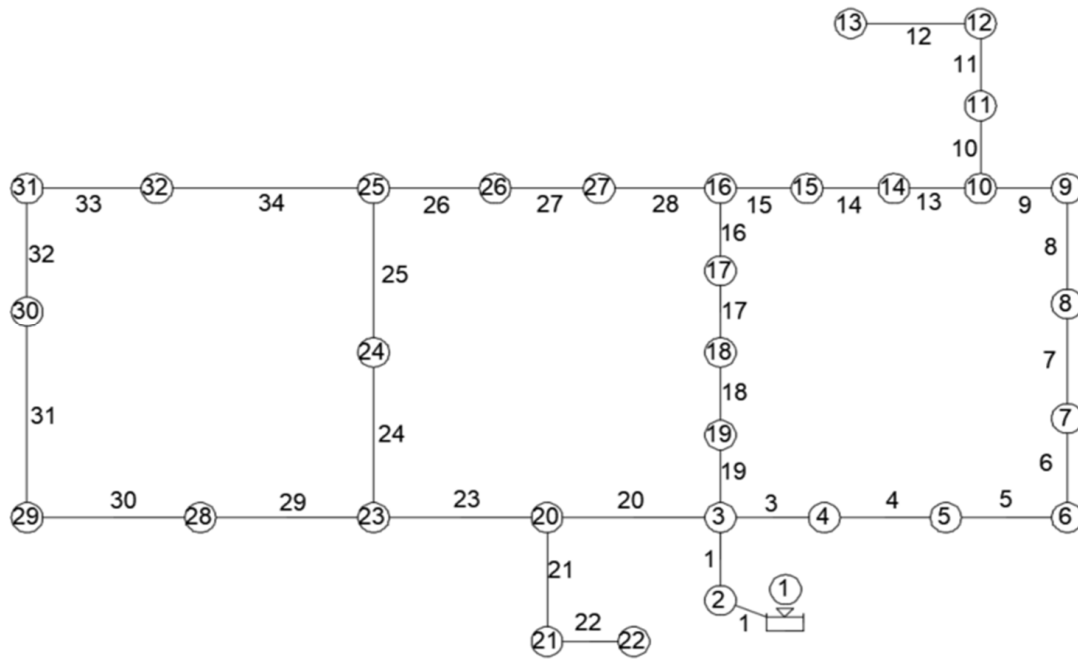


Figure 8. Layout of the Hanoi network.

Table 6. Results for the Hanoi network.

FEN		TPSVR	DE	GA	HS	PSO	SA	SCE	TPS
1000	Minimum	7.643	326.396	80.343	145.006	191.357	7.457	527.672	7.467
	Mean	7.885	594.692	228.928	240.377	445.831	9.176	1135.127	26.584
	Maximum	8.139	881.175	402.602	364.354	736.968	28.100	1510.300	133.666
10,000	Minimum	7.385	26.367	7.374	8.798	77.144	7.678	84.440	7.404
	Mean	7.726	123.746	8.150	9.638	345.074	8.072	269.165	7.704
	Maximum	7.858	213.143	8.998	12.025	553.235	8.773	495.141	7.914
40,000	Minimum	7.209	9.841	7.337	7.856	121.186	7.560	13.767	7.337
	Mean	7.533	15.957	7.922	8.173	323.295	8.039	66.838	7.563
	Maximum	7.815	56.863	9.280	8.686	510.339	9.746	193.329	7.858

Numerical results are presented in ( $\times 10^6$ ) USD.

Table 7. Best solution for the Hanoi network.

ID	Diameter (in)	ID	Diameter (in)	ID	Diameter (in)	ID	Diameter (in)
1	75	11	24	21	20	31	12
2	75	12	20	22	12	32	12
3	40	13	20	23	30	33	16
4	40	14	24	24	20	34	20
5	40	15	24	25	12		
6	40	16	40	26	20		
7	30	17	40	27	30		
8	24	18	55	28	30		
9	20	19	55	29	12		
10	24	20	40	30	12		

### 8.3. The Balerma Network

This network is an adaptation of a real water distribution network in the city of Balerma, located in the Andalusian region of Spain. Its layout is shown in Figure 9. It was first presented by Reca and Martinez [2], and various algorithms have been applied to it ever since. The authors also published the .inp file from EPANET, in which the network is presented in detail. In the best solution found by the TPS without velocity restrictions, the flow velocities of 48 pipes were higher than the maximum, and the flow velocities of 145 pipes were lower than the minimum. Since there were 454 pipes in the network, the flow velocities in 43% of the pipes were out of bounds.



**Figure 9.** Layout of the Balerma network.

As in the Hanoi network, the solution with velocity constraints and with the available diameters was impossible. In this network, diameters smaller than the available ones were also required. In order to solve the problem with velocity restrictions, we chose to include available diameters from 113 mm, which was the previous smallest available, down to 20 mm. The cost of each newly available diameter was chosen randomly, but followed the pricing logic of the originally available diameters. Additionally, a larger diameter than the existing ones was added. These are presented in Table 8.

**Table 8.** Additional available diameters for the Balerma network.

Diameter (mm)	Cost (EUR/m)
20	1.28
30	1.92
40	2.56
50	3.00
60	3.83
70	4.48
80	5.11
90	5.76
100	6.39
700	312.15

In a large network such as the Balerma network, it is possible to end up with small diameters for some pipes, as it resulted here, in order to enforce velocity restrictions. If these diameters are deemed unrealistic, there are some options available to the design engineer. One of them is to accept velocities lower than the minimum, in which case special attention should be paid to the consequent deposition of debris through regular supervision and maintenance. Another option is to locally modify the architecture of the network, and the proposed method is again suitable for this purpose. However, in this paper, adherence to the given architecture of the Balerma network was necessary in order to carry out the proper comparisons within the same framework employed for the previous unrestricted version.

The results of the application of the selected algorithms are presented in Table 9. The good performance of the TPSVR in this difficult problem was obvious. Almost all the other algorithms, except for the SA, produced highly infeasible solutions. Another characteristic of the TPSVR that was evident in the results was its reliability, since the best and worst solutions had only a small difference. A table containing the optimal diameters resulting from TPSVR is given as Table S1 in the Supplementary Material.

**Table 9.** Results for the Balerma network.

FEN		TPSVR	DE	GA	HS	PSO	SA	SCE	TPS
45,400	Minimum	<b>1.984</b>	21,528	106,599	67,692	32,733	2.148	18,854	297
	Mean	<b>2.052</b>	46,220	126,649	83,234	88,267	12	29,424	348
	Maximum	<b>2.176</b>	105,581	149,817	104,118	318,424	94	45,631	390

Numerical results are presented in ( $\times 10^6$ ) EUR.

## 9. Conclusions

In this research, an algorithm was presented that could handle velocity restrictions in the problem of water distribution network optimization. The TPSVR does not have any parameters that need to be adjusted, and therefore it can be used directly in any new network. Additionally, it does not require a penalty function that also has to be adjusted. It provides an innovative way of dealing with velocity restrictions, guiding a single solution toward the optimum in each iteration.

The new algorithm emerged from the modification of the TPS algorithm. The TPSVR and the TPS algorithms, with the use of a penalty function, were compared with a selection of the most popular algorithms. They were used to solve three benchmark problems, with differences in their size and the rest of their characteristics. The TPSVR showed an advanced performance due to the way it handled velocity restrictions, while most of the other algorithms could not reach a feasible solution in the same time.

The TPSVR was used in this research with a single objective, namely the determination of a set of pipe diameters that led to the most economical design. However, in real-world applications, more objectives must be taken into account, such as the system's robustness and the quality of the water. The latter is in fact directly related to the flow velocity in the pipes of a network. These extra objectives lead to a multiobjective optimization problem, for which the solution of by the TPSVR may be suitably extended. Moreover, the manner in which the TPSVR handles the velocity restrictions may be used in combination with other methods besides TPS.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/hydrology9050073/s1>, Table S1: Optimal diameters resulting from TPSVR.

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

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in this manuscript are available from the cited sources.

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

## References

1. Manolis, A.; Sidiropoulos, E.; Evangelides, C. Targeted path search algorithm for optimization of water distribution networks. *Urban Water J.* **2021**, *18*, 195–207. [[CrossRef](#)]
2. Reza, J.; Martínez, J. Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resour. Res.* **2006**, *42*, W05416. [[CrossRef](#)]
3. Suribabu, C.R.; Neelakantan, T.R. Design of water distribution networks using particle swarm optimization. *Urban Water J.* **2006**, *3*, 111–120. [[CrossRef](#)]
4. Rossman, L. *Epanet 2 Users Manual*; Water Supply and Water Resources Division, National Risk Management Research Laboratory: Cincinnati, OH, USA, 2000.
5. Todini, E.; Pilati, S. A gradient algorithm for the analysis of pipe networks. *Comput. Appl. Water Supply* **1988**, *1*, 1–20.
6. Jabbar, A.; Poteh, H.T.; Younesi, H.; Haghiabi, A.H. Development of central force optimization for pipe-sizing of water distribution networks. *Water Supply* **2016**, *16*, 1398–1409. [[CrossRef](#)]
7. Fallah, H.; Kisi, O.; Kim, S.; Rezaie-Balf, M. A New Optimization Approach for the Least-Cost Design of Water Distribution Networks: Improved Crow Search Algorithm. *Water Resour. Manag.* **2019**, *33*, 3595–3613. [[CrossRef](#)]
8. Formato, R.A. Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*; Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 129, pp. 221–238. [[CrossRef](#)]
9. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [[CrossRef](#)]
10. Samani, H.M.V.; Mottaghi, A. Optimization of Water Distribution Networks Using Integer Linear Programming. *J. Hydraul. Eng.* **2006**, *132*, 501–509. [[CrossRef](#)]
11. Samani, H.M.V.; Zanganeh, A. Optimisation of water networks using linear programming. *Proc. Inst. Civ. Eng.-Water Manag.* **2010**, *163*, 475–485. [[CrossRef](#)]
12. Haghighi, A.; Samani, H.M.V.; Samani, Z.M.V. GA-ILP Method for Optimization of Water Distribution Networks. *Water Resour. Manag.* **2011**, *25*, 1791–1808. [[CrossRef](#)]
13. Alperovits, E.; Shamir, U. Design of optimal water distribution systems. *Water Resour. Res.* **1977**, *13*, 885–900. [[CrossRef](#)]
14. Stipanić, D.; Travaš, V.; Kranjčević, L.; Holjević, D. An Iterative Algorithm for Optimizing Pipe Diameter in Pressurized System. *Teh. Vjesn. Tech. Gaz.* **2020**, *27*, 1284–1292. [[CrossRef](#)]
15. Zheng, F.; Simpson, A.R.; Zecchin, A.C. A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems: Combined NLP-DE for optimizing water networks. *Water Resour. Res.* **2011**, *47*. [[CrossRef](#)]
16. Geem, Z.W. Optimal cost design of water distribution networks using harmony search. *Eng. Optim.* **2006**, *38*, 259–277. [[CrossRef](#)]
17. Cunha, M.d.C.; Sousa, J. Water Distribution Network Design Optimization: Simulated Annealing Approach. *J. Water Resour. Plan. Manag.* **1999**, *125*, 215–221. [[CrossRef](#)]
18. Liong, S.; Atiquzzaman, M. Optimal Design of Water Distribution Network Using Shuffled Complex Evolution. *J. Inst. Eng. Singap.* **2004**, *44*, 93–107.
19. Geem, Z.W. Particle-swarm harmony search for water network design. *Eng. Optim.* **2009**, *41*, 297–311. [[CrossRef](#)]
20. Qi, X.; Li, K.; Potter, W.D. Estimation of distribution algorithm enhanced particle swarm optimization for water distribution network optimization. *Front. Environ. Sci. Eng.* **2016**, *10*, 341–351. [[CrossRef](#)]
21. Fujiwara, O.; Khang, D.B. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resour. Res.* **1990**, *26*, 539–549. [[CrossRef](#)]