LA-UR-11-06242

Approved for public release; distribution is unlimited.

Title: Water Distribution Expansion Planning with Decomposition

Author(s): Russell Bent
Carleton Coffrin
David Judi
Tim McPherson
Pascal van Hentenryck

Intended for: WDSA 2012

WATER DISTRIBUTION EXPANSION PLANNING WITH DECOMPOSITION

Russell Bent[1], Carleton Coffrin[2], David Judi[1], Tim McPherson[1], and Pascal van Hentenryck[2]
[1]Los Alamos National Laboratory, Los Alamos, New Mexico
[2]NICTA, Australia

ABSTRACT
In recent years, the water distribution expansion planning (WDEP) problem has become increasingly complex as the demands on water distribution systems have evolved to meet modern requirements. This paper describes an algorithm that combines the strengths of local search with global search to flexibly handle these difficult problems. The algorithm decomposes a WDEP problem into sub problems using a local search procedure called Large Neighborhood Search (LNS). Each sub problem is solved exhaustively (or partially) using global search techniques such as branch-and-bound search. The utility of the approach is demonstrated on the problem described in the Battle of the Water Networks II competition.

INTRODUCTION
The past few decades have seen stress placed on water distribution networks around the world due to aging infrastructures, water quality concerns, increasing demand for water, diminishing supplies, and a desire to reduce the carbon footprint of water systems. To address this challenge, the 14th Water Distribution Systems Analysis Symposium has issued a Battle of the Water Networks II competition. In this paper, we describe our approach to solving the problem of expanding and controlling closed pipe water distribution systems to meet future demand, satisfy multiple objective functions, and meet robustness criteria. We generalize algorithms developed for the electric power systems domains of transmission expansion planning (Bent et al. (2012)), integrated resource planning (Bent et al. (2011)), restoration scheduling (Coffrin et al. (2012)), and vehicle routing (Bent and Van Hentenryck (2004)).

In this paper we use a decomposition-based approach that separates the problem into distinct sub problems that are solved iteratively using a procedure referred to as Large Neighborhood Search (LNS) (Shaw (1998)). Each sub problem is solved within a (potentially truncated) branch-and-bound search procedure that intuitively is not unlike some of the ideas developed by the simulation optimization community where each partial solution explored in the branch-and-bound search is evaluated using a water network simulation package. The key contribution of the algorithm is a technique for combining global search techniques with local search to find high quality solutions. The local search iteratively determines the sub problems to consider and the global search procedure is executed on each sub problem.

Review The WDEP problem is NP-Hard due to the non-linearities present in the head-loss models of water systems (Rossman (2000)) and the discrete variables (e.g., pipe diameters, etc.) in water network expansion (Yates et al. (1984); Gupta et al. (1993)). As a result, there is a large body of solution approaches for solving the WDEP in the literature. These approaches include linear programming and non-linear programming (Kessler and Shamir (1989); Fujiwara and Khang (1990); Sherali et al. (2001); Bragalli et al. (2008)) and evolutionary algorithms (EA). The EA algorithms include genetic algorithms (Dandy et al. (1996); Wu and Simpson (2001); Reca and Martinez

(2006); Ewald et al. (2008); Kadu et al. (2008)), simulated annealing (Tospornsampan et al. (2007)), ant colony optimization (Maier et al. (2003); Zecchin et al. (2005); Tong et al. (2011)), Harmony Search (Geem (2009)) and particle swarm optimization (Montalvo et al. (2008)). While these approaches have made significant contributions to the field, it is clear that there are oppurtunities to expand and enhance this literature to further the state-of-the-art.

The rest of the paper is organized as follows. The next section formally defines the WDEP problem. The second section describes the algorithm used to find solutions to the WDEP problem. The third section describes to the solution the problem provided by the Battle of the Water Networks II. The fourth section discusses the results. The final section concludes the paper.

PROBLEM DEFINITION

The model of a water distribution network described in this paper follows the definitions provided in the EPANET software package (Rossman (2000)). The minimal set of features of a water distribution network that are required to fully define our approach are discussed in this section.

Nodes The problem is described in terms of a set of nodes, $\mathcal{N}$, that represent geographically located points in a water network e.g., reservoirs (RESERVOIRS), tanks (TANKS), and junctions (JUNCTIONS), such that $\mathcal{N} = \text{RESERVOIRS} \cup \text{TANKS} \cup \text{JUNCTIONS}$. For each junction $i \in$ JUNCTIONS, the function $d_{i,\tau}$ is used to define the demand for water at time τ. For each reservoir $r \in$ RESERVOIRS, the function $h_{r,\tau}$ is used to define the hydraulic head at time τ. For each tank $t \in$ TANKS, e^+ and e^- define the maximum and minimum water storage elevation respectively. The decision variable v_t is used to define the volume of the tank. v_t has discrete domain $[{v_t}^0, {v_t}^1, \ldots, {v_t}^n]$.

Edges The problem is also described in terms of a set of edges, $\mathcal{E}$. For an edge $i, j \in \mathcal{E}$ between nodes i and j, the decision variable $p^d_{i,j}$ is used to denote the number of pipes with diameter d, where $p^{d-}_{i,j} \leq p^d_{i,j} \leq p^{d+}_{i,j}$. The set of possible diameters is denoted by D. The decision variable $u^\pi_{i,j}$ is used to denote the number of pumps of type π between i and j, where $u^{\pi-}_{i,j} \leq u^\pi_{i,j} \leq u^{\pi+}_{i,j}$. The set of possible pump types are defined by I. The decision variable $v^d_{i,j}$ is used to denote the number of valves with diameter d between i and j, where $v^{d-}_{i,j} \leq v^d_{i,j} \leq v^{d+}_{i,j}$. The Boolean decision variable $g_{i,j}$ is used to denote the existence of a backup diesel generator for components (pumps) on edge i, j.

Controls Finally, the problem is also defined by control statements that determine the status of edges. The decision variables $k^+_{i,j}$ and $k^-_{i,j}$ denote the times when components (pumps and valves) on edge i, j are activated and deactivated respectively. The coupled decision variable $\kappa^+_{i,j}$ and $\kappa^-_{i,j}$ denotes the node attribute and value that causes edge i, j to be activated and deactivated respectively (for example, tank water levels).

Solution A solution, σ, is defined as a set of variable assignments to the variables of the WDEP problem, i.e.[1].

$$\bigcup_{\tau\in T}[v_t \leftarrow \chi_{v_t}] \cup \bigcup_{i,j\in\mathcal{E}}[p^d_{i,j} \leftarrow \chi_{p^d_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[u^\pi_{i,j} \leftarrow \chi_{u^\pi_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[v^d_{i,j} \leftarrow \chi_{v^d_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[k^+_{i,j} \leftarrow \chi_{k^+_{i,j}}] \cup$$

$$\bigcup_{i,j\in\mathcal{E}}[k^-_{i,j} \leftarrow \chi_{k^-_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[\kappa^+_{i,j} \leftarrow \chi_{\kappa^+_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[\kappa^-_{i,j} \leftarrow \chi_{\kappa^-_{i,j}}] \cup \bigcup_{i,j\in\mathcal{E}}[g_{i,j} \leftarrow \chi_{g_{i,j}}] \quad (1)$$

[1]The notation $[a \leftarrow b]$ is used to denote the assignment of a value, b, to a variable a

where χ is drawn from the respective domains of each variable. By convention, unassigned variables are assumed to be the "no change" assignment, i.e. the assignment in the initial network model. For convenience, we use the notation y to denote a generic variable drawn from equation (1).

Simulation Our algorithm has at its disposal a simulator[2], $\mathcal{S}$, for determining the flow of water in σ. $\mathcal{S}_{p_{i,\tau}}(\sigma)$ is used to denote the pressure at node i at time τ as calculated by $\mathcal{S}$. Similarly, $\mathcal{S}_{e_{t,\tau}}(\sigma)$ is used to denote the water elevation at tank t at time τ, $\mathcal{S}_{\zeta_{i,j,\tau}}(\sigma)$ is used to denote the cost of energy used by edge (pump) i,j, $\mathcal{S}_{\eta_{i,j,\tau}}(\sigma)$ is used to denote the carbon usage of edge (pump) i,j, $\mathcal{S}_{\nu_{i,j,\tau}}(\sigma)$ is used to denote the velocity of flow through edge i,j at time τ, and $\mathcal{S}_{\omega_{i,\tau}}(\sigma)$ is used to denote the age of water at node i at time τ. The notations are shortened to $p_{i,\tau}, e_{i,\tau}, \zeta_{i,j,\tau}, \eta_{i,j,\tau}, \nu_{i,j,\tau}$ and $\omega_{i,\tau}$ when $\mathcal{S}(\sigma)$ is understood from context.

A solution σ is feasible when the following constraints are satisfied:

$$
\begin{aligned}
p_{i,\tau} &\geq p_i^- && \forall_{i\in\mathcal{N}}\forall_{\tau\in T} && (2)\\
e_{t,\tau} + e_{t,\tau+\Delta} &> 2e^- && \forall_{t\in\text{TANKS}}\forall_{\tau\in T} && (3)\\
e_{t,T} &\geq l_t && \forall_{t\in\text{TANKS}} && (4)\\
\sum_{d\in D} p_{i,j}^d &\geq p_{i,j}^{d-} && \forall_{i,j\in\mathcal{E}} && (5)
\end{aligned}
$$

Constraint (2) states that the pressure of a node i, must be larger than some threshold p_i^- for all times in a time period T. Constraint (3) states that a tank cannot be at its minimal level for Δ time (30 minutes in this case). Constraint (4) states that a tank must have at least a specified level (l_t) of water in the last time period T. Constraint (5) states that there must be a minimal number of pipes between two nodes i and j. This ensures that existing pipe locations cannot be decommissioned.

In addition, there is a robustness criterion where a sequence of problems is created with the following control statements to model power outages.

$$
\begin{aligned}
k_{i,j}^- &= \tau && \forall_{d\in D}\forall_{i,j\in\mathcal{E}} : \neg g_{i,j} \ \wedge \ u_{i,j}^\pi > 0 && (6)\\
k_{i,j}^+ &= \tau + \delta && \forall_{d\in D}\forall_{i,j\in\mathcal{E}} : \neg g_{i,j} \ \wedge \ u_{i,j}^\pi > 0 && (7)
\end{aligned}
$$

The control statements (equations 6 and 7) turn off pumps that do not have backup generators between times τ and $\tau+\delta$ ($\delta = 2$ hours in this case). Constraint (2) must be satisfied for these problems. Problems are created for each possible τ.

As the initial network models often contain constraint violations, the physical constraints (equations 2-4) are relaxed and added to the objective function. The pressure violation of σ is calculated as the sum of pressures the fall below thresholds, i.e.

$$\mu(\sigma) = \sum_{i\in\mathcal{N}} \sum_{\tau\in T} \max(0, p_i^- - p_{i,\tau}). \tag{8}$$

[2] EPANET

The tank minimum elevation violation of σ is calculated as the sum of the number of times a tank is at its minimal level for Δ time, i.e.

$$\rho(\sigma) = \sum_{t \in \text{TANKS}} \sum_{\tau \in T} 1 - \left\lceil \frac{e_{t,\tau} + e_{t,\tau+\Delta} - 2e^-}{2e^+} \right\rceil. \tag{9}$$

The final tank elevation violation of σ is calculated as the sum of elevations that do not meet minimal elevation requirements at the end of the simulation, i.e.

$$\psi(\sigma) = \sum_{t \in \text{TANKS}} \max(0, l_t - e_{t,T}). \tag{10}$$

The cost of σ is calculated as the sum of construction and operating costs, i.e.

$$\zeta(\sigma) = \sum_{i,j \in \mathcal{E}} \sum_{d \in D} \zeta(p^d_{i,j}) + \sum_{i,j \in \mathcal{E}} \sum_{d \in D} \zeta(v^d_{i,j}) + \sum_{i,j \in \mathcal{E}} \sum_{\pi \in I} \zeta(u^\pi_{i,j}) + \sum_{t \in \text{TANKS}} \zeta(v_t) + \sum_{i,j \in \mathcal{E}} \sum_{\tau \in T} \zeta_{i,j,\tau}, \tag{11}$$

where $\zeta()$ is a function that computes the cost of building the specified components. The carbon output of σ is calculated as the sum of carbon output from construction and operations, i.e.

$$\eta(\sigma) = \sum_{i,j \in \mathcal{E}} \sum_{d \in D} \eta(p^d_{i,j}) + \sum_{i,j \in \mathcal{E}} \sum_{\tau \in T} \eta_{i,j,\tau}, \tag{12}$$

where $\eta()$ is a function that calculates the carbon output when the specified components are constructed. The water age of σ is calculated according to the formula provided in (Salomons et al. (2012)), i.e.

$$\omega(\sigma) = \frac{\sum_{i \in \mathcal{N}} \sum_{\tau \in T} w_{i,\tau} d_{i,\tau}(i,\tau) \omega_{i,\tau}}{\sum_{i \in \mathcal{N}} \sum_{\tau \in T} d_{i,\tau}(i,\tau)}, \tag{13}$$

where $w_{i,\tau} = 1$ if $\omega_{i,\tau}$ is greater than some threshold W [3], and 0 otherwise. The objective function, $f(\sigma)$, is a lexicographic multi-objective function[4] of the form

$$\min f(\sigma) = \langle \mu(\sigma), \rho(\sigma), \psi(\sigma), \zeta(\sigma), \eta(\sigma), \omega(\sigma) \rangle. \tag{14}$$

A lexicographic ordering of the objectives is a natural model in this case as constraint satisfaction clearly has primacy over cost, water age, and carbon. Cost and carbon tend be correlated, where decreasing cost decreases carbon output, and vice versa. Water age is slightly more problematic, however, as we observed that reducing water age tends to require significant increases in cost. As the competition expects equal weighting for each of the objectives, this suggested cost reduction dominates water age. More generally, the last three objectives should be treated as purely multi-objective, however in this competition a lexicographic ordering appears to be reasonable.

METHODOLOGY

The core of our algorithm relies on a Discrepancy Bounded Local Search (DBLS) developed for expansion planning of power grids (Bent et al. (2012)). DBLS builds on simulation optimization ideas by encapsulating the simulation of infrastructure networks into a "black box" that is queried by DBLS for information about how a solution operates (i.e., $\mathcal{S}(\sigma)$). The intuition behind DBLS is

[3] 2 days in this case

[4] Lexicographic objective functions define objective functions in order of primacy. The first objective is used to compare two solutions. In the case of ties, the second objective is used, and so forth.

to generalize constructive heuristics that make good decisions on how to build solutions, but make a few bad decisions from time to time. DBLS embeds the heuristic in a branch and bound search tree as the branching heuristic and explores those solutions that are within δ violations (discrepancies) of the heuristic, where δ is a user-specified parameter. The formal algorithm of DBLS is presented in Figure 1.

DBLS$(\sigma, \Upsilon, \delta)$
1 **if** $\delta = 0$
2   **then return** σ;
3 $\sigma^* \leftarrow \sigma$;
4 $y \leftarrow \textsc{chooseVariable}(\Upsilon, \sigma)$;
5 $\langle \chi_1, \chi_2, \ldots, \chi_k \rangle \leftarrow \textsc{orderDomain}(y)$;
6 $\sigma \leftarrow \sigma \setminus [y \leftarrow \sigma(y)]$;
7 **for** $i \leftarrow 1 \ldots k$
8 **do** $\sigma_i \leftarrow \sigma \cup [y \leftarrow \chi_i]$;
9   **if** $f(\sigma_i) \leq f(\sigma^*)$ **and** $\mathcal{S}(\sigma_i)$
10   **then** $\sigma^* \leftarrow \sigma_i$;
11   DBLS$(\sigma_i, \Upsilon \setminus y, \delta - i)$;
12 **return** σ^*;

Figure 1: Discrepancy-Bounded Local Search

DBLS takes as arguments a solution σ, (often the current state of the network); a set of variables, Υ, drawn from

$$\bigcup_{\tau \in T} v_t \cup \bigcup_{i,j \in \mathcal{E}, d \in D} p^d_{i,j} \cup \bigcup_{i,j \in \mathcal{E}, \pi \in I} u^{\pi}_{i,j} \cup \bigcup_{i,j \in \mathcal{E}, d \in D} v^d_{i,j} \cup \bigcup_{i,j \in \mathcal{E}} k^+_{i,j} \cup$$
$$\bigcup_{i,j \in \mathcal{E}} k^-_{i,j} \cup \bigcup_{i,j \in \mathcal{E}} \kappa^+_{i,j} \cup \bigcup_{i,j \in \mathcal{E}} \kappa^-_{i,j} \cup \bigcup_{i,j \in \mathcal{E}} g_{i,j}; \tag{15}$$

and a discrepancy parameter, δ. The first two lines of Figure 1 check if the number of discrepancies has dropped to 0. Line 3 initializes the best solution discovered with the current solution. Line 4 chooses a variable y to explore. More formally the function $\textsc{chooseVariable}$ is defined by:

$$y = \arg \min_{y \in Varset} f(\sigma \cup [\sigma(y) \leftarrow \textsc{orderDomain}(y)_1]) \tag{16}$$

Line 5 executes the heuristic for ordering the domain of y. The domain is ordered by the following function

$$\langle \chi_1, \chi_2, \ldots, \chi_n \rangle : f(\sigma \cup [\sigma(y) \leftarrow \chi_i]) \leq f(\sigma \cup [\sigma(y) \leftarrow \chi_{i+1}]) \tag{17}$$

Line 6 unassigns the current variable assignment of y (if any) and lines 7–11 iterate over the ordered domain of the variable. δ is decremented by violations in the ordering heuristic. Line 9 implicitly updates attributes associated with the new σ by executing $\mathcal{S}$. Line 12 returns the best solution discovered. From a search tree perspective, Figure 2 provides an illustration of DBLS's search on a binary tree for $\delta = 0, 1, 2$ and 3. As is seen in the figure, the running time of DBLS is exponential in δ and $|\Upsilon|$ (the number of solutions considered is $\sum_{i=1 \ldots \delta} \binom{|\Upsilon| \times k}{i}$, where k is the maximum size of a variable's domain) (Korf (1996)).

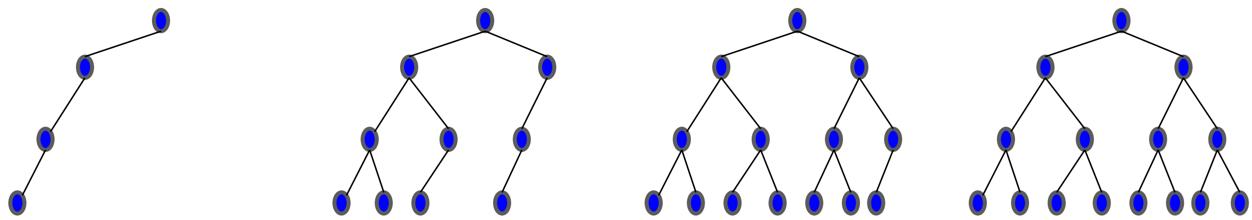

Figure 2: An example of the DBLS algorithm on a simplified binary search tree from (Bent et al. (2012)). On the left is the portion of the search tree explored when $\delta = 0$, i.e., only expansion decisions suggested by the constructive heuristic are considered. The subsequent pictures show the portions of the search tree explored when $\delta = 1, 2$ and 3.

DBLS$(\sigma, \Upsilon, \delta, \alpha)$
if $\delta = 0$ **or** $\alpha = 0$
  then return σ;
$\sigma^* \leftarrow \sigma$;
$y \leftarrow$ CHOOSEVARIABLE(Υ, σ);
$\langle \chi_1, \chi_2, \ldots, \chi_k \rangle \leftarrow$ ORDERDOMAIN(y);
$\sigma \leftarrow \sigma \setminus [y \leftarrow \sigma(y)]$;
for $i \leftarrow 1 \ldots k$
do $\sigma_i \leftarrow \sigma \cup [y \leftarrow \chi_i]$;
  if $f(\sigma_i) \leq f(\sigma^*)$ **and** $\mathcal{S}(\sigma_i)$
    then $\sigma^* \leftarrow \sigma_i$;
  if $f(\sigma_i) \leq f(\sigma)$
    then $\alpha \leftarrow 0$;
    else $\alpha \leftarrow \alpha - 1$;
  DBLS$(\sigma_i, \Upsilon \setminus y, \delta - i, \alpha)$;
return σ^*;

Figure 3: Discrepancy-Bounded Local Search with α

As in (Bent et al. (2012)) DBLS includes a key generalization that improves its performance based on the observation that $f(\sigma)$ is non-monotonic. In other words, adding components can cause $\mu(\sigma), \rho(\sigma)$ and $\psi(\sigma)$ to rise or fall (sometimes referred to as Braess's paradox (Bienstock and Verma (2010))). To control this behavior, a parameter, α, is used to limit the number of times in a row that $f(\sigma)$ may worsen. The modified algorithm is presented in Figure 3.

Large Neighborhood Search The computational requirements of DBLS can be significant, in particular the execution of ORDERDOMAIN and CHOOSEVARIABLE requires $\mathcal{S}$ to be executed $O(|\chi||\Upsilon|)$ times at each node in the search tree.[5] Thus, it was computationally beneficial to iteratively decompose the problem into smaller subproblems, and solve each subproblem via DBLS using a technique called Large Neighborhood Search (LNS) (Shaw (1998)). The overall algorithm is described in Figure 4. In this algorithm, line 3 selects a subset of the variables to consider based on the current solution σ, the set of variables Υ, and the size of the subset i. Once the set

[5] We reduce the computational requirements by executing the ordering once at the beginning of the search, trading computational efficiency for heuristic accuracy.

of variables is selected, the neighborhood of σ, consisting of assignments of the variables in $\hat{\Upsilon}$, is explored using DBLS. This is repeated for subsets of up to size *maxSize* with *maxIterations* (lines 1-2) decompositions created for each size. In this paper we considered four implementations of SELECTVARIABLES and choose an implementation at random when executing line 3. The first implementation selects a subset of variables at random:

```
SELECTVARIABLES-R(σ, n, Υ)
 1  Υ̂ ← ∅;
 2  for i ← 1...n
 3      do y ← RANDOM(Υ);
 4         Υ̂ ← Υ̂ ∪ y;
 5         Υ ← Υ \ y;
 6  return Υ̂;
```

where lines 2-5 iteratively add a randomly selected variable until n are selected. The second implementation selects a class of variables at random (for example, all pump variables), and selects from that class at random, i.e.

```
SELECTVARIABLES-RT(σ, n, Υ)
 1  Υ̂ ← RANDOM(Υ);
 2  return SELECTVARIABLES-R(σ, n, Υ̂);
```

where line 1 selects a random type of variable and line 2 selects a subset of those variables randomly. The third implementation selects a graphically connected set of variables at random. The intent of this implementation is to exploit graphical structure and discover connected regions of the network that require augmentation, i.e.

```
SELECTVARIABLES-C(σ, n, Υ)
  1  Υ̂ ← ∅;
  2  Γ ← RANDOM(𝒩 ∪ ℰ);
  3  while |Υ̂| < n
  4      do γ ← RANDOM(Γ);
  5         y ← RANDOM(γ, Υ);
  6         Γ ← Γ ∪ GRAPHNEIGHBOR(𝒩 ∪ ℰ, γ);
  7         if y ≠ ∅
  8            then Υ̂ ← Υ̂ ∪ y;
  9                 Υ ← Υ \ y;
 10            else Γ ← Γ \ γ;
 11  return Υ̂;
```

where line 2 selects a random component in the network to initialize the search. Lines 3-10 iteratively adds variables that are graphically connected to the selected component. Line 4 selects a component γ from Γ. Line 5 selects a random variable associated with γ. Line 6 adds the components connected to γ. These include edges connected to a node and vice versa, as well as components connected through control statements. If there is a variable associated with γ, it is added to $\hat{\Upsilon}$ (line 8) and removed from future consideration (line 9). Otherwise, γ is removed from future consideration (line 10).

The final implementation selects a random set of pipe variables, weighted towards those pipes

```
LNS(σ, ϒ, δ, α)
for i ← 1 ... maxSize
do for j ← 1 ... maxIterations
    do ϒ̂ ← SELECTVARIABLES(σ, i, ϒ);
        σ ← DBLS(σ, ϒ̂, δ, α);
return σ;
```

Figure 4: Large Neighborhood Search

with high velocity flows. High velocities in pipes cause increased pressure drops due to increased friction. Thus, high velocities are an indicator for areas with large pressure drops, providing a heuristic for targeted pipe augmentation, i.e.

```
SELECTVARIABLES-V(σ, n, ϒ)
1  ϒ̂ ← ∅;
2  ϒ ← ORDERPIPEVARIABLES(σ, ϒ);
3  for i ← 1 ... n
4      do y ← ϒ_⌊RANDOM([0,1])^β × |ϒ|⌋;
5          ϒ̂ ← ϒ̂ ∪ y;
6          ϒ ← ϒ \ y;
7  return ϒ̂;
```

where line 2 orders the pipe variables according to the velocity of flow in the pipes, i.e.

$$\langle p^d_{i,j}(1), p^d_{i,j}(2), \dots p^d_{i,j}(n) \rangle : \sum_{\tau \in T} \nu_{i,j,\tau}(k) \geq \sum_{\tau \in T} \nu_{i,j,\tau}(k+1) \tag{18}$$

Line 4 selects a random pipe variable, weighted towards those pipes with high velocity. The user parameter β controls the degree of bias in the weight; as $\beta \to \infty$, the selection becomes more biased.

As mentioned early, our approach uses all four approaches by choosing one uniformly at random, i.e.

```
SELECTVARIABLES(σ, n, ϒ)
return RANDOM(SELECTVARIABLES-R(σ, nϒ) ∪
    SELECTVARIABLES-RT(σ, nϒ) ∪ SELECTVARIABLES-C(σ, nϒ) ∪
    SELECTVARIABLES-V(σ, nϒ));
```

Finally, including the robustness calculation during the search was computationally burdensome. Instead, once the best candidate solution was discovered, we enumerated the possible generator options and selected the minimal set that achieved the necessary robustness requirements. This was a reasonable approach as the cost of backup generators compared to the rest of the costs in the network (operations, tanks, etc.) was considerably less expensive.

SOLUTION

In order to evaluate our approach, we used the problem defined by the Battle of Water Networks II (Salomons et al. (2012)). For brevity, the details of the problem are omitted in this paper. However, it is important to note that we modified the model file in a couple of ways in order to reconcile the model file with the documentation of Salomons et al. (2012)). These modifications include:

1. The provided model file set the pump efficiency of existing pumps as 70%, however the documentation indicated existing pumps had an efficiency of 65%. The model file was changed so that the efficiency of existing pumps is 65% and new pumps have an efficiency of 75%.

2. A pattern for the energy costs was not included in the model file. The pattern in (Salomons et al. (2012)) was added.

3. The documentation of (Salomons et al. (2012)) indicated that the initial end water level of tanks is equivalent to half the volume of the tanks. The initial water levels in the model file were not consistent with this requirement and the model file was adjusted accordingly.

4. It is our understanding that the new pipes numbered 3-14 are place holders for pipes that must be built whereas pipes 1 and 2 do not have to be included. All pipes in these areas were incorporated into the construction cost and carbon calculations, even if their diameter is the diameter provided in the initial model file.

The initial model contained numerous constraint violations: 92 nodes violate pressure constraints, 305 times a tank is at its minimum level for two or more consecutive time steps, and 6 tanks are not half full at the end of the simulation. As a result, we explored two approaches for obtaining a solution. In the first approach, we adopted a "hot start" where a subject matter expert spent an eight hour day adjusting the model manually to eliminate as many constraint violations as possible. In the hot start model only 11 nodes exhibit violations of the pressure constraints and all tank constraints are satisfied. This model was then the input to LNS. In the second approach we adopted a "cold start" where LNS started with the model initially provided by the competition. In both cases the algorithm was allowed to run 24 hours on a Intel Xeon 2.67 Ghz processor Windows 7 machine with 12 GB of RAM and the best result of each approach is presented here. Tables 1-5 provide the hot start results and Tables 6-10 the cold start results. In all cases the pipe expansions replaced existing pipes (as allowed on the competition web site). In both cases, the following algorithm parameters were used: $maxRemove = 30$, $maxIterations = 10$, $\delta = 5$, and $\alpha = 3$. The hot start solution is the solution submitted to the Battle of the Water Networks II competition as it is substantially better than the cold start solution.

DISCUSSION OF RESULTS

There are a number of interesting features of the solutions described in this paper. First, the hot start solution resulted in replacement of 3 existing pipes in the network. These pipes were replaced with pipes of smaller diameter. Similarly, the cold start solution resulted in the replacement of 5 existing pipes with smaller diameter pipes. Pipe replacement was allowed in the FAQ discussion-and no rules were posted that disallowed replacing pipes with smaller diameters, thus we allowed diameter reduction. In all cases, the reduced pipe diameters were upstream of pumps. The smaller pipe diameters mimic the behavior of a valve-reducing flow through a constriction. While this creates a larger head drop over the length of the pipe, pressures upstream of the reduced pipes are slightly higher as a result of the decreased flow capacity. Although the change in pressure upstream was small, it was enough to eliminate pressure violations during peak demands. This type of a result provides an indication to the utility that considering the addition of valves will improve the performance of the system with less cost.

Second, it was noted that the cost of adding tanks and operating pumps more frequently (primary mechanisms for allowing the system to ride through power outages other than backup generation) was considerably more than the cost of back up generation. This observation, combined with

Pipe ID	Pipe Diameter mm	Pipe Length m	Pipe Cost/m $/m/year	Pipe Cost $/year	GHG/m $kgCO_2$-e/m/year	GHG $kgCO_2$-e/year
1	152	328.74	10.10	3320.27	9.71	3192.06
3	102	113.05	8.31	939.45	5.90	667.00
4	102	310.40	8.31	2579.42	5.90	1831.36
5	102	231.10	8.31	1920.44	5.90	1363.49
6	102	218.93	8.31	1819.31	5.90	1291.69
7	102	259.37	8.31	2155.36	5.90	1530.28
8	102	470.52	8.31	3910.02	5.90	2776.07
9	102	244.52	8.31	2031.96	5.90	1442.67
10	102	393.42	8.31	3269.32	5.90	2321.18
11	102	314.33	8.31	2612.08	5.90	1854.55
12	102	240.65	8.31	1999.80	5.90	1419.84
13	102	293.73	8.31	2440.89	5.90	1733.01
14	102	221.76	8.31	1842.82	5.90	1308.38
P398	152	6.75	12.10	81.68	9.71	65.54
P468	203	31.29	14.49	453.39	13.94	436.18
P96	203	53.72	14.49	778.40	13.94	748.85
P787	203	127.08	14.49	1841.39	13.94	1771.50
P992	254	69.15	15.55	1075.29	18.43	1274.43
P287	305	45.12	18.28	824.79	23.16	1044.98

Table 1: Hot Start Solution: Pipes Replaced

Tank ID	Volume mm^3	Cost $
No tanks added		

Pump ID	Pump Curve	Cost $/year
PU2	8b	4554

Valve ID	Diameter mm	Cost $/year
No valves added		

Table 2: Hot Start Solution: Tanks, Pumps, and Valves Added to the Network

<table>
<tr><th>Diesel Generator ID</th><th>Max Power</th><th>Pump ID</th><th>Max Pump Power kw</th><th>Diesel Generator Cost $year</th></tr>
<tr><td rowspan="4">1</td><td rowspan="4">200</td><td>PU1</td><td>45.24</td><td rowspan="4">11630</td></tr>
<tr><td>PU2</td><td>45.24</td></tr>
<tr><td>PU3</td><td>45.24</td></tr>
<tr><td>16</td><td>54.28</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">100</td><td>PU6</td><td>49.76</td><td rowspan="2">10560</td></tr>
<tr><td>PU7</td><td>49.76</td></tr>
<tr><td>4</td><td>50</td><td>PU9</td><td>31.67</td><td>9450</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">50</td><td>PU10</td><td>22.62</td><td rowspan="2">9450</td></tr>
<tr><td>PU11</td><td>22.62</td></tr>
</table>

Table 3: Hot Start Solution: Diesel Generators Added to the Network

Pump/Valve ID	Tank	Activation Height	Deactivation Height
PU1	T1	0	6.5
PU2	T1	0	6.5
PU3	T1	0	6.5
PU16	T1	0	6.5
PU4	T3	0	4.0
PU5	T3	4.0	5.4
PU6	T4	2.82	4.7
PU7	T4	3.0	4.5
PU8	T5	0.0	0.9
PU9	T5	3.6	4.5
PU10	T7	3.0	4.0
PU11	T7	4.0	5.0
V2	T2	3.5	5.5

Table 4: Hot Start Solution: Control Designs

	Cost	Water Age	Carbon
Operations	305184	25536	2510897
Construction	81540	-	28073
Total	386725	25536	2538970

Table 5: Hot Start Solution: Objective Function Summary

the computational requirements of computing the robustness of a network relative to its ability to absorb power failures, led us to treat the robustness criteria as a post-processing full enumeration step. In the hot start model, it was noted that 2 pumps were assigned new controls that did not allow them to operate (so did not require back up), and 1 pump could be shut off for two hours without detrimental pressure drops.

Third, for a number of pumping stations, the pumps essentially continuously operate in order to maintain pressure in portions of the system. This led to a solution were pump operations represent a significant fraction of the cost portion of the objective function and a substantial portion of the carbon output of the solution. We view this observation as the best opportunity for improving the quality of our solutions in future work, where a few additional components added to the system may allow us to decrease the amount of time some pumps run.

Fourth, the hot start solution did not result in any tank upgrades. Generally speaking, the algorithm found that tank upgrades resulted in substantial system performance degradation, including increased water age, increase in pump operation time to satisfy tank level constraints, and pressure drops during tank filling periods. Additionally, tank upgrades were more expensive than back up generation. Thus, the addition of tanks did not appear to bring much benefit.

Finally, while the algorithm found feasible solutions for both the cold start and hot start initializations in the specified time period, the hot start solution is considerably better after 24 hours of computation. The solution to the cold start model was still improving after 24 hours, but at a very slow rate. This provides some evidence of the value of starting from a near feasible solution to get high quality solutions quicker.

Pipe ID	Pipe Diameter mm	Pipe Length m	Pipe Cost/m $/m/year	Pipe Cost $/year	GHG/m $kgCO_2$-e/m/year	GHG $kgCO_2$-e/year
1	102	328.74	8.31	2731.83	5.9	1939.57
3	102	113.05	8.31	939.45	5.9	667.00
4	102	310.4	8.31	2579.42	5.9	1831.36
5	102	231.1	8.31	1920.44	5.9	1363.49
6	102	218.93	8.31	1819.31	5.9	1291.69
7	102	259.37	8.31	2155.36	5.9	1530.28
8	102	470.52	8.31	3910.02	5.9	2776.07
9	102	244.52	8.31	2031.96	5.9	1442.67
10	254	393.42	12.96	5098.72	18.43	7250.73
11	102	314.33	8.31	2612.08	5.9	1854.55
12	102	240.65	8.31	1999.80	5.9	1419.84
13	102	293.73	8.31	2440.90	5.9	1733.01
14	102	221.76	8.31	1842.83	5.9	1308.38
P1027	254	82.32	15.55	1280.08	18.43	1517.16
P103	203	35.98	14.49	521.35	13.94	501.56
P134	254	68.79	15.55	1069.68	18.43	1267.80
P144	102	130.79	9.97	1303.98	5.9	771.66
P154	102	170.64	9.97	1701.28	5.9	1006.78
P20	610	579.67	42.8	24809.88	54.99	31876.05
P270	102	88.36	9.97	880.95	5.9	521.32
P282	152	131.93	12.1	1596.35	9.71	1281.04
P398	102	6.75	9.97	67.30	5.9	39.83
P424	254	27.8	15.55	432.29	18.43	512.35
P468	203	31.29	14.49	453.39	13.94	436.18
P752	305	70.36	18.28	1286.18	23.16	1629.54
P787	152	127.08	12.1	1537.67	9.71	1233.95
P815	203	50.52	14.49	732.03	13.94	704.25
P99	406	452.37	23.26	10522.13	33.09	14968.92
P996	406	10.22	23.26	237.72	33.09	338.18

Table 6: Cold Start Solution: Pipes Replaced

Tank ID	Volume mm^3	Cost $
T3	1000	30640
T1	5000	122420
T5	500	14020
T2	5000	122420

Pump ID	Pump Curve	Cost $/year
PU1 (1)	8a	3225
PU1 (2)	8a	3225
PU2 (1)	8a	3225
PU3 (1)	8a	3225
PU4 (1)	8a	3225
PU5 (1)	8a	3225

Valve ID	Diameter mm	Cost $/year
No valves added		

Table 7: Cold Start Solution: Tanks, Pumps, and Valves Added to the Network

Diesel Generator ID	Max Power	Pump ID	Max Pump Power kw	Diesel Generator Cost $year
1	200	PU1	45.24	11630
1	200	PU2	45.24	11630
1	200	PU3	45.24	11630
1	200	PU1 (1)	22.62	11630
1	200	PU2 (1)	22.62	11630
1	200	PU1 (2)	22.62	11630
3	100	PU6	49.76	10560
3	100	PU7	49.76	10560
4	50	PU9	31.67	9450
5	50	PU10	22.62	9450
5	50	PU11	22.62	9450

Table 8: Cold Start Solution: Diesel Generators Added to the Network

Pump/Valve ID	Tank	Activation Height	Deactivation Height
PU1	T1	3.9	5.2
PU1 (1)	T1	3.9	5.2
PU1 (2)	T1	3.9	5.2
PU2	T1	0	6.5
PU2 (1)	T1	0	6.5
PU3	T1	3.9	5.2
PU3 (1)	T1	3.9	5.2
PU4	T3	2.7	4.0
PU4 (1)	T3	2.7	4.0
PU5	T3	0	4.0
PU5 (1)	T3	0	4.0
PU6	T4	3.75	4.7
PU7	T4	3.0	4.5
PU8	T5	0.0	0.9
PU9	T5	3.6	4.5
PU10	T7	2.5	4.8
PU11	T7	4.0	5.0
V2	T2	3.5	5.5

Table 9: Cold Start Solution: Control Designs

	Cost	Water Age	Carbon
Operations	329301	24789	2710126
Construction	430454	-	85015
Total	759756	24789	2795141

Table 10: Cold Start Solution: Objective Function Summary

CONCLUSION

Water distribution systems are increasingly being subjected to new requirements and stress. As a result, new algorithms are needed to expand existing networks and modify their operational points. This paper describes a basic algorithm framework (LNS) for decomposing the WDEP problems into subproblems, where each sub problem is solved exhaustively (or partially) using global search techniques such as branch-and-bound. The algorithm successfully found solutions to the problem that meet all design requirements (constraints), while at the same time minimizing economic costs, carbon emissions, and water age. The algorithm was tested on the model provided by the competition and a hot start solution based on subject matter expertise.

There are a number of important future directions for LNS on the WDEP. First, an in depth study of different decomposition algorithms is needed. In this paper four different implementations were considered, however, the relative performance of each implementation was not considered. More importantly, perhaps, other types of decompositions may yield better results than what was achieved here (for example decomposition targeting existing expansions—in particular operations—which makes up the bulk of the carbon emissions and costs, or constraint violations). These need to be explored. Second, the robustness criteria was treated separately from the main search algorithm as a post processing enumeration algorithm. Better results might be achieved if the robustness criteria are included in the main search procedure. However, new techniques are needed to address the computational requirements for assessing the robustness criteria. Third, it will be interesting to embed existing global search techniques such as (Kessler and Shamir (1989); Fujiwara and Khang (1990); Sherali et al. (2001); Bragalli et al. (2008)) as opposed to our DBLS to determine if these approaches improve performance. Finally, fast approximations of CHOOSEVARIABLE and ORDERDOMAIN need to be developed to improve computational efficiency.

REFERENCES

Bent, Russell, Alan Berscheid, and G. L. Toole. "Generation and Transmission Expansion Planning for Renewable Energy Integration." In *Power Systems Computation Conference (PSCC)*. 2011.

Bent, Russell, G. L. Toole, and Alan Berscheid. "Transmission Network Expansion Planning with Complex Power Flow Models." *IEEE Transactions on Power Systems* 27, 2: (2012) 904–912.

Bent, Russell, and Pascal Van Hentenryck. "A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows." *Transportation Science* 38, 4: (2004) 515–530.

Bienstock, Daniel, and Abhinav Verma. "The N k Problem in Power Grids: New Models Formulations, and Numerical Experiments." *SIAM Journal of Optimization* 20, 5: (2010) 2352–2380.

Bragalli, C., C. DAmbrosio, J. Lee, A. Lodi, and P Toth. "Water Network Design by MINLP." Technical report, 2008.

Coffrin, Carleton, Pascal Van Hentenryck, and Russell Bent. "Smart Load and Generation Scheduling for Power System Restoration." In *Power Engineering Society General Meeting*. 2012.

Dandy, G. C., A. R. Simpson, and L. J. Murphy. "An improved genetic algorithm for pipe network optimization." *Water Resources Research* 32, 2: (1996) 449–458.

Ewald, G., W. Kurek, and M. A. Brdys. "Grid implementation of a parallel multiobjective genetic algorithm for optimized allocation of chlorination stations in drinking water distribution systems: Chojnice case study." *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews* 38, 4: (2008) 497–509.

Fujiwara, O., and D. B. Khang. “A 2-Phase Decomposition Method for Optimal-Design of Looped Water Distribution Networks.” *Water Resources Research* 26, 4: (1990) 539–549.

Geem, Z.W. “Particle-Swarm Harmony Search for Water Networks.” *Engineering Optimization* 41, 4: (2009) 297–311.

Gupta, I., J.K. Bassin, A. Gupta, and P Khanna. “Optimization of Water Distribution Systems.” *Environmental Software* 8, 4: (1993) 101–113.

Kadu, M. S., R. Gupta, and P. R. Bhave. “Optimal design of water networks using a modified genetic algorithm with reduction in search space.” *Journal of Water Resources Planning and Management-Asce* 134, 2: (2008) 147–160.

Kessler, A., and U. Shamir. “Analysis of the Linear-Programming Gradient-Method for Optimal-Design of Water-Supply Networks.” *Water Resources Research* 25, 7: (1989) 1469–1480.

Korf, Richard. “Improved Limited Discrepancy Search.” In *Thirteenth National Conference on Artificial Intelligence*. Portland, Oregon, 1996, 286–291.

Maier, H. R., A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan. “Ant colony optimization distribution for design of water systems.” *Journal of Water Resources Planning and Management-Asce* 129, 3: (2003) 200–209.

Montalvo, I., J. Izquierdo, R. Perez, and M. M. Tung. “Particle Swarm Optimization applied to the design of water supply systems.” *Computers and Mathematics with Applications* 56, 3: (2008) 769–776.

Reca, J., and J. Martinez. “Genetic algorithms for the design of looped irrigation water distribution networks.” *Water Resources Research* 42, 5.

Rossman, Lewis. “EPANET 2 User Manual.” Technical report, National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, 2000.

Salomons, Elad, Avi Ostfeld, Zoran Kapelan, Aaron Zecchin, Angela Marchi, and Angus R. Simpson. “The Battle of the Water Networks II: Detailed Problem Description and Rules.” Technical report, 2012.

Shaw, Paul. “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems.” In *Proceedings of the Fourth International Conference on the Principles and Practice of Constraint Programming (CP)*. Pisa, Italy, 1998, 417–431.

Sherali, H. D., S. Subramanian, and G. V. Loganathan. “Effective relaxations and partitioning schemes for solving water distribution network design problems to global optimality.” *Journal of Global Optimization* 19, 1: (2001) 1–26.

Tong, L., G. Han, and J. Qiao. “Design of water Distribution Network via Ant Colony Optimization.” In *2nd International Conference on Intelligent Control and Information Processing*. 2011, 366–370.

Tospornsampan, J., I. Kita, M. Ishii, and Y Kitamura. “Split-Pipe Design of Water Distribution Network Using Simulated Annealing.” *International Journal of Computer and Information Engineering* 1, 3: (2007) 154–164.

Wu, Z.Y., and A.R Simpson. "Competent Genetic-Evolutionary Optimization of Water Distribution Networks." *Journal of Water Resources Planning and Management-Asce* 132, 2: (2001) 67–77.

Yates, D. F., A. B. Templeman, and T. B. Boffey. "The Computational-Complexity of the Problem of Determining Least Capital-Cost Designs for Water-Supply Networks." *Engineering Optimization* 7, 2: (1984) 143–155.

Zecchin, A. C., A. R. Simpson, H. R. Maier, and J. B. Nixon. "Parametric study for an ant algorithm applied to water distribution system optimization." *IEEE Transactions on Evolutionary Computation* 9, 2: (2005) 175–191.