Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/ 703-993-1530

A Decision Guidance System for Optimal Infrastructure Investments

Bedor Alyahya
Department of Computer Science,
University George Mason
Fairfax, VA 22030, U.S.A.
balyahya@gmu.edu

Alexander Brodsky
Department of Computer Science,
University George Mason
Fairfax, VA 22030, U.S.A.
brodsky@gmu.edu

Technical Report GMU-CS-TR-2021-3

Abstract

We design and develop an extensible model and a decision guidance system for making actionable recommendations on investments in heterogeneous infrastructure service networks. The model expresses the cash flows, as well as performance indicators, such as total cost of ownership and carbon emissions, as a function of both investment and operational controls within physical constraints of heterogeneous infrastructures and of balancing resource flows. Uniquely, it is designed to make Pareto-optimal investment decisions under the assumption of optimal operational controls over the time horizon. We also develop an extensible library of domain-specific operational analytic models for infrastructure components, initially for desalination and water systems, including pumps, renewable energy sources, water and power storage, and Revers Osmosis desalination units. Finally, we conduct and report on a feasibility study for this domain to demonstrate the ability to solve realistic size problems.

1 Introduction

Investing in interlarded infrastructure services, such as power grid, gas pipelines and water systems, is essential to meet the growing demand. However, these types of investments are associated with high costs and risks that must be studied carefully. In many cases, however, investments in infrastructure do not yield the maximum benefit, which results in a serious waste of resources that could be avoided with proper planning. Some challenges in investment planning are due to (1) the complexity of the heterogeneous infrastructure and its components that interact with each other during operation in a non-trivial way; (2) many trade-off choices to consider between objectives and performance measures; (3) investment performance depends on efficiency of operation (e.g., over hourly operational intervals) which is typically not steady-state, but is complex due to stochastic patterns of supply and demand; and, (4) rapid changes in infrastructure technologies and the challenge in managing hybrid systems mixing old and new technologies.

There has been extensive work to prioritize and optimize infrastructure investment. One line of this research, e.g., see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11], is the development of domain-specific models that focus on increasing the value for money, enhancing system design, or aiding in protecting the environment. These models are designed to solve domain-specific investment problems in infrastructure silos, such as water supply [1, 2, 12], power [7, 8], transportation [6],telecommunications [5] and supply chain [9, 10, 11]. However, they lack the flexibility to consider a mix of interrelated infrastructures holistically, across their silos. This inhibits model re-usability and, therefore, leads to high complexity and cost of investment models and systems.

The second line of work, e.g., see [13, 14, 15, 16, 17], focuses on studying inter-dependencies of a broad range of infrastructures. However, its focus is around the protection of these interrelated infrastructures to stand against any disruption with less emphasis on financial benefits of infrastructure investment across silos.

There is also a considerable work, such as [18, 19], to provide a way to optimize the cash flow over the time horizon by utilizing the *network flow* and determining the best time to liquidate. In these works, however, the authors focus only on modeling high-level financial terms, without modeling the underlying engineering and physical aspects of infrastructures and their effect on cash flows. In these works, however, the authors focus only on modeling high-level financial terms, without modeling the underlying engineering and physical aspects of infrastructures and their effect on cash flows. Instead, they assume that the high-level financial input is provided to their models as input, which is often either unrealistic or inaccurate.

In our previous work [20], we developed a model and decision guidance system for investments in interdependent infrastructure service networks. The model allows for the representation of an arbitrary hierarchical composition of interrelated infrastructures over the investment time horizon. The model represents mathematically how these components generated their metrics of interest (such as cost and CO_2) over the investment periods using an extensible repository of infrastructure-specific component models. However, the system does not track the physical infrastructure's operational control over the investment periods, but rather assumes optimal steady-state operation controls over the investment periods. Therefore, an enhancement can be made to refine the investment decision.

In[12], we proposed an analytical model that optimizes the operation of complex interconnected infrastructures using the service networks and an expandable library of infrastructure. However, this model only focuses on the operational part and was limited to desalination system designs.

This paper closes the gaps of [20] and [12] by proposing and developing an extensible model and decision guidance system for making actionable recommendations on investments in heterogeneous infrastructure service networks. Critically, it makes investment decisions under the assumption of optimal operational controls of the extended infrastructure over the time horizon. This is necessary in order to leverage the operational interaction of interrelated infrastructures (such as smart grids, renewable energy, power storage, schedulable loads,...) over short (e.g. 30 minutes) operational intervals.

More specifically, the contributions of this paper is as follows. First, we develop a formal investment analytic model for optimizing both operational and investment controls to maximize/minimize the performance metrics generated as a result of running the general investment analytical model. This model is unique in its ability to deal with a huge volume of operational variables by breaking the investment periods into operational windows that represent similar patterns of operation(e.g., weekend vs weekday pattern of operation). Second, we develop an extensible repository of domain-specific operational analytic models for system components which initially include pumps, renewable energy sources, water and energy storage and Reverse Osmosis (RO) plants. Third, we developed a decision guidance system that uses the model above based on Decision Guidance Analytics Language (DGAL) [21] and the system architecture proposed in [22]. Finally, we test the feasibility of applying this proposed system in solving realistic size problems and we report our findings.

The paper is organized as follows. Section 2 uses an example to illustrate how the model works to support optimal infrastructure investments. Section 3 overviews a high-level architecture of the Investment Decision Guidance system and methodology. Section 4 formalizes the investment service network model. Section 5 discusses the feasibility of the model through experimental study. Finally, Section 6 presents the conclusion and provides directions for future work.

2 Investment model by example

The *Service Network Investment Model* (SNIM) is a general but precise model that guides the decision-maker in making long-term investment decisions. The accuracy of the model relies on optimizing domain-specific performance metrics (e.g., cost, CO_2,..) produced by the operation of interconnected and complex infrastructures over the time horizon. By utilizing a library of extendable analytical models that represent mathematically how these infrastructures produce their performance matrices and constraints, the main model can solve a variety of problems without it being modified.

To explain how the SNIM works, we first explain what we mean by the *Service Network* and how it can create a uniform and comprehensive input form, as in Figure 1, to represent complex infrastructure configurations and system flows over time. The SNIM output contains the decision variables as will as aggregated performance matrices (e.g.,cost, NPV,...) and constraints resulting from running the model over every operational window within each investment period in the time horizon. The output the can be optimized to determine the optimal operational controls and investment choices over the investment periods.

The *Service Network* is a generic hierarchy of nested services (such as the one in Figure 2), which express services' activities (e.g., pumping, storing , desalinating,...) by capturing the inflows and outflows of these activities over and across these services. The rectangles in Figure 2 show *composite service*s (e.g., Water Desalination System, Power Storage,...), while at the bottom of the hierarchy reside the infrastructures that we refer to as *atomic service*s (e.g., batteries, grid,...). Also, each arrow in the figure represents the flow of resources (e.g., water, power,...) between the services. Whether these atomic services are already owned or consider to be an invested opportunity, each atomic service defines some fixed and controlled parameter (e.g., number of pumps, pump efficiency,...) to be used by the analytical models of their type (e.g., pump station analytical model) in the library. Every atomic analytical model in the library represents mathematically how this service operates and calculates the performance matrices and constraints as a result it been called by the investment analytical model. For ex-

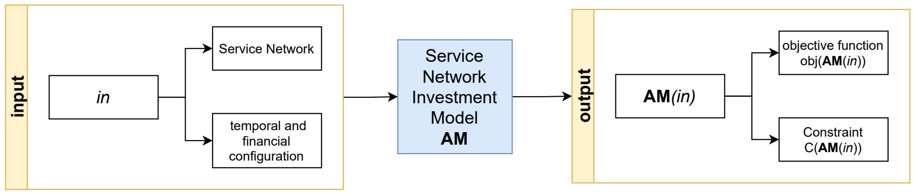

Figure 1: Service Network Investment Model

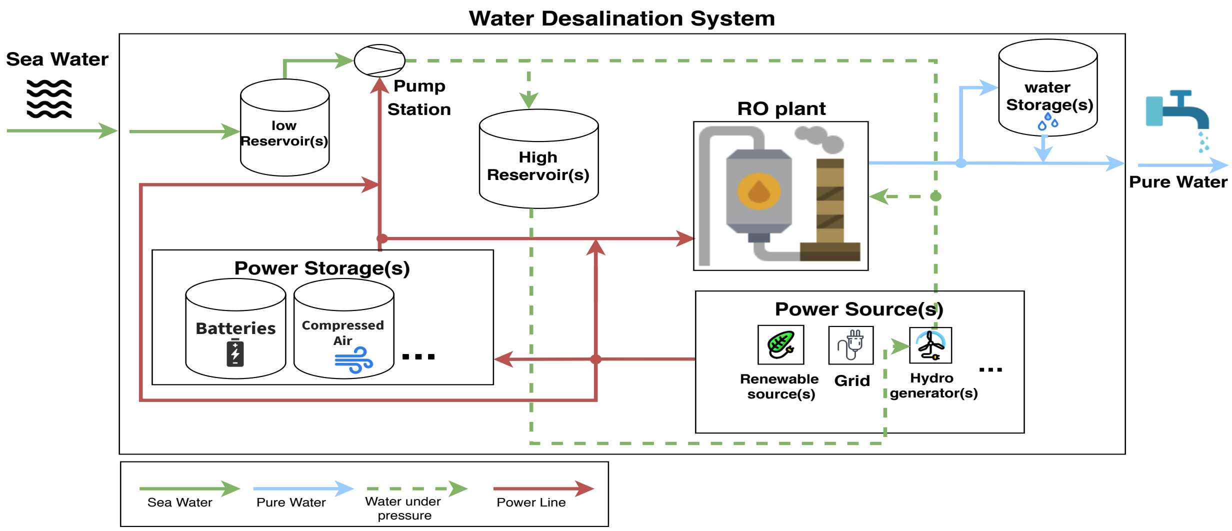

Figure 2: Service Network-water desalination system

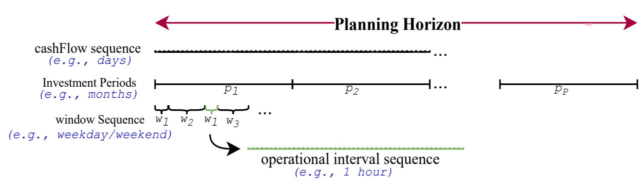

Figure 3: Time Horizon

ample, the pump analytical model balances the right amount of power needed to push x square feet of water given some fixed and controlled parameters (e.g., pump diminution, flow rate,...).

Now, the *Service Network* concept is used to create the SNIM input model which contains the system hierarchy including the services' fixed and controlled parameters and the configuration of the time horizon. By running the SNIM, the model calculates and aggregates the flows and the constraints bottom-up, according to the investment choices, for every operational interval over the time horizon and updates the state of the atomic services in each iteration. After the model generates all flows, the model then uses the result to calculate and aggregate bottom-up the performance metrics and constraints according to the atomic model numerical formula. The result then can be optimized by setting the objective function to minimize/maximize the aggregated performance metrics while satisfyingly the model-generated constraints. For example, in Figure 2 the optimizing problem is set to reduce the cost while satisfying fresh water demand for every operational interval and ensure not to exceed a certain level of carbon emission.

To overcome the limitation of handling massive decision variables of operational setting, the time horizon is broken down into operation windows (e.g., weekdays, weekend,...) within each investment period (e.g., months, session,...). Assuming that the investment decisions can be made at the beginning of each investment period, each window ,as shown in Figure 3, represents the operational intervals that specify how the infrastructures operates, while the cash-Flow sequence intervals is used to represent the flow of cash over the horizon. Assuming that the windows can have different lengths and the decision variables of system setting (e.g., flows and states) at the beginning and end of each window are constrained, so that the sequence of the windows can be ordered (with repetition) to represent the flow over the period without it being intermittent. In this way, we reduce a huge number of operational intervals to representative windows to account for different operational patterns.

In the next section, we introduce the architecture of the decision guidance system within which the proposed model operates.

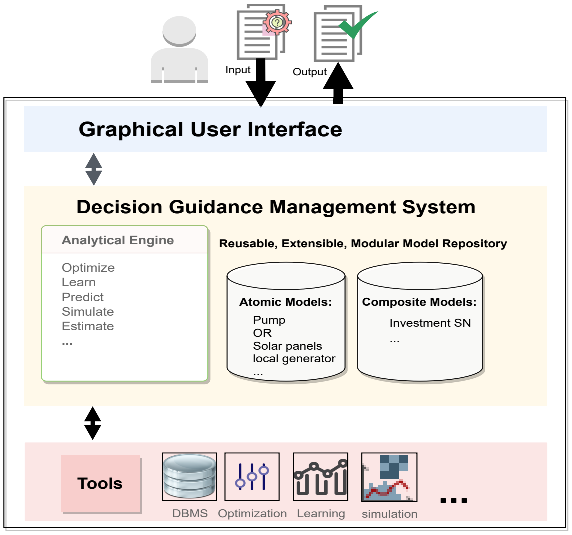

Figure 4: Decision Guidance System Architecture

3 The Decision Guidance System Architecture

Figure 4 shows the Decision Guidance System (DGS) architecture within which our proposed model operates. The middle tier of the architecture represents the decision guidance management system (DGMS), which was proposed in [23] and developed in [22]. The main ideas of the the middle tier are (1) to allow the user to perform different analytical tasks (e.g., optimizing, predicting, learning,...); (2) to manage the repository that includes reusable, modular and composable models; and, (3) to enable different external tools (bottom tier) to be used without hard wiring the model to a specific tool using the Decision Guidance Analytics Language (DGAL)[24].

The graphical user interface (GUI) at the top of the figure allows the user to construct the layout of the service networks and to select the atomic models that represent each atomic service from the library. The user can combine predefined composite services and modify existing models. After constructing the service networks, the user can then set temporal and financial parameters and leave the operational control and investment choices to be determined by the optimization tool. After the investment model constructed the problem, DGAL sends the constructed problem to the optimization tool and retrieves the result in the same form of the input, but with instantiated decision variables to its optimal values.

4 Formalization of Service Network Investment Model

4.1 High-level Optimization Problem

To obtain the best long-term investment plan, the investment optimization problem is formed as a result of the investment analytic performance model (AM), which computes performance metrics, such as cost and carbon emissions, as well as feasibility constraints, as a function of fixed and controlled operational and investment parameters of the service network over the investment time horizon. More formally, the analytic performance model AM is a function:

$$AM : IN \rightarrow OUT \tag{1}$$

which forms a valid output instance $out \in OUT$ of performance metrics, such as cost or waste, from a valid input $in \in IN$ of fixed and controlled operational and investment parameters. Therefore, the investment optimization problem is:

$$\begin{aligned} \min_{in \in IN} \quad & obj(AM(in)) \\ \text{s.t.} \quad & C(AM(in)) \end{aligned} \tag{2}$$

where

- $Obj : OUT \rightarrow \mathbb{R}$ is an objective function, which gives the real objective value in $\mathbb{R}$ given a valid output instance of the AM.
- $C : OUT \rightarrow \{T, F\}$ is a constraint function C, which gives *True* or *False* given a valid output instance of the AM.

This representation expresses the objectives and constraints as a function of analytical model AM output, which helps to formulate multiple investment optimization problems using different infrastructure configurations, objective functions and operation windows using the same AM. The re-usability of the investment model reduces the development life cycle, cost and allows the developer to use a library of domain-specific models, which reduce the dependence on subject matter experts by mathematically representing the mechanism and constraints of different operation units and infrastructures.

In the following, we will use the following notation to represent a set of *key-value* pairs:

$$\boxed{\mathtt{m} = \{key_1 : value_1, key_2 : value_2, \quad \ldots, key_n : value_n\}}$$

where the *keys* are unique identifiers. Note that this set represents a mapping

$$m : \{key_1, \ldots, key_\mathrm{n}\} \longrightarrow \cup_{i=1}^{n} D_\mathrm{i}$$

from the set $\{key_1, \ldots, key_n\}$ of keys to union of the domains $\cup_{i=1}^{n} D_\mathrm{i}$, where D_i is the domain of values associated with key_i, so that $m(key_\mathrm{i}) \in D_\mathrm{i}$ for all $i = 1, \ldots, n$.

We will use the notation $keys(m) = \{key_1, \ldots, key_n\}$ to denote the set of all keys associated with the set m of key-value pairs. We will also use the list notation of the form $l = [a_1, \ldots, a_n]$, which will be interpreted as a function $l : \{1, \ldots, n\} \to D$, where D is the domain of values in the list; thus $l(i) = a_i$ is the i's element of the list.

Now using the above notations we can describe all of the components above; starting with a valid service network investment model output instance *out* in section 4.2, followed by the input instance *in* in section 4.3, and finally, we describe the analytic model which is a function that computes an output instance from the input instance.

4.2 Service Network Instance: The Model Output

A valid *SN* output instance *out* is a set of *key:value* pairs of following form:

```
{config :<set of configurations>,
 rootServiceID :<root service ID>,
 services :<set of services>}
```

Form 1: out

where *config* value (copied from the input) is a set *key:value* pairs of the form:

```
{financial:{financialInterval:<value>,
            intRate:<value>,...},
 intervalsRatio:<value>,
 horizon: [$p_1, p_2, \ldots$]}
```

Form 2: config value

where *intRate* is the interest rate (e.g., 0.1%) per *financialInterval* (e.g., day); *intervalsRatio* (e.g., 24) represents the number of operation intervals (e.g., of 1 hour each) within the financial interval (e.g., 1 day). The underlying assumption is that the time is split into operational intervals per which operational controls (e.g., of equipment) are actuated. The *horizon* describes the list $[p_1, p_2, \ldots]$ of investment periods, each of the form:

```
{windows:{$w_1$:{length:$l_1$}, $w_2$:{length:$l_2$},...},
 winSeq:<sequence of window IDs>}
```

Form 3: period value

where each *period* is represented by a set $\{w_1, w_2, \ldots\}$ of windows (e.g., summer week) with varied lengths (e.g., 24*7 hourly intervals). The window sequence *winSeq* is the sequence of window IDs in the order in which these windows occur within a given investment period in the time horizon.

In Form 1, the **rootServiceID** represents the ID for the root service (e.g., desalination system). The knowledge about the service network structure and the parameters of each service are found under **services**. The **services** compose of a set of *key:value* pairs where each key uniquely identifies a service, while *value* represents the details of the corresponding service as described below. In addition, each service in the **services** is either a ***composite*** or an ***atomic*** service. The *composite* service, such as *powerService*, contains at least one subservice, which includes the IDs of these *services* under *subService*. So, each ***composite*** service has the following form:

```
{type:"composite",
 inFlow:{$f_1$:[$v_1, v_2, ..., v_P$],$f_2$:[$v_1, v_2, ..., , v_P$],...},
 outFlow:{$f_1$:[$v_1, v_2, ..., v_P$],$f_2$:[$v_1, v_2, ..., , v_P$],...},
 metrics:{NPV:<value as in form 6>,
          cashFlow:<value as in form 6>,
          $m_1$:<value as in form 6>,
          $m_2$:<value as in form 6>,...},
 constraints:True or False,
 subServices:<set of service ids>}
```

Form 4: composite service

where each **inFlow** and **outFlow** contains a set of flow $\{f_1, f_2, ...\}$ that represent the flow IDs going in and out each service, respectively. The value of each flow is a list of period flows$[v_1, ..., v_p]$, each of the form:

```
{ $w_1$:{qty: [$q_1, q_2$,...],total:<value>},
  $w_2$:{qty: [$q_1, q_2$,...],total:<value>},...}
```

Form 5: period flow value

where the **qty** is a list $[q_1, q_2, ...]$ that shows the quantities of operational flow at each window $\{w_1, w_2, \ldots\}$ of a given investment period p, while the **total** shows the total flow for the whole window within a given period.

In the same manner, the ***metrics*** value contains as set of additive metrics $\{m_1, m_2, \ldots\}$, such as operational cost or emissions. Each metric value list the quantities of a given metric for every investment period in the time horizon. Therefore, each metric value is of the form:

```
{perPeriod:[$v_1, v_2$,...],total:<numerical value> }
```

Form 6: metric value

The ***metrics*** also contain a special *key cashFlow* and other financial *metrics* which depend on the *cashFlow*, such as *NPV*. In the *cashFlow*, the value is a list of *payments* of the form in 7. Thus, each payment pair has the following form:

```
[{interval: $i_1$,amount: $m_1$},
 {interval: $i_2$,amount: $m_2$},...]
```

Form 7: payment value

where each interval i refers to payment interval over the entire time horizon appears at most once in the list. Note that negative (positive) *amount* m means cash inflow (outflow) occurred during the same time interval i. Negative interval means the number of intervals the amount has flowed before the beginning of the first period. The ***constraints*** value, in Form 4, indicates whether the composite service satisfies its constraints.

The ***atomic*** service has similar form, as in Form 4, except that:

- There is no **subServices** *key:value* pair.
- The *type* refers to one of the *atomic* analytical model in the library.
- Additional set of *key:value* pair:

```
numUnitInvested:[v1,...,vp],
avaliable:[v1,...,vp],
initAvaliable:<value>,
capacityPerUnit:<value>,
numUnitON:<value as in form 9>
```

Form 8: additional pairs

where each value in ***numUnitInvested*** and ***avaliable*** indicate the number of invested and available units of a given service, respectively, in each period $\in \{1 \dots P\}$ in the time horizon. While the ***initAvaliable*** value represent the number of unit available at the beginning of the first period and the ***capacityPerUnit*** value represent the capacity per unit. The ***numUnitON*** represent the number of unit running (ON) in each interval within each window within each period in the time horizon. Thus, the value of ***numUnitON*** has the following form:

```
{[w1:[v1,...,vp],w2:[v1,...,vp],...},
  w1:[v1,...,vp],w2:[v1,...,vp],...},...]}
```

Form 9: numUnitOn value

- An optional set of *key:value* pairs:

```
{state:<value as in form 9>}
```

Form 10: state key:value pair

where each value v_i, in Form 9, capture the state of the service in each operational interval within the periods' windows.

In the next section, we describe a valid input model needed to compute the output instance.

4.3 Service Network Instance: The Model input

The model input (*in*) follow the same structure as in the output, but with some modification:

- No *metrics* and *constraints key:value* pairs.
- Instead of having a list to describe the *qty* and *total* for each flow in the *composite* service, we replace it with a list of lower bounds (*LB*).
- For an *atomic* service:
  - An optional set of key value pair:

```
typeSpecific:{<set of key:value pairs>}
```

Form 11: type specific key:value pair

where **typeSpecific** value represents, using a *key:value* pairs , the parameters that are needed by the *atomic* analytic model **type** to calculate its metrics and constraints.

  - Instead of having a list to depict the *state*, we replace it with a single value that depicts the *state* at the beginning of the first operational window:

```
state:<initial value>
```

Form 12: initial state

  - Additional set of *key:value* pair:

```
payments:{invPayments:<value>,
          opPayments:<value>}
```

Form 13: payments

where the ***invPayments*** $\langle value \rangle$ represent a list of investment payments associated with each period, using the following form:

```
{1:{{due:<value>,amt:<value>},
    {due:<value>,amt:<value>},...},
 ...,
 P:{{due:<value>,amt:<value>},
   {due:<value>,amt:<value>},...}}
```

Form 14: invPayment

where each *due* determine the intervals in relative to the beginning of period p in which the amount *amt* must be paid, if the investment accrue at period p. On the other hand, the ***opPayments*** represents a list of operational payments and their relative billing intervals. Thus, the ***opPayments*** $\langle value \rangle$ is expressed using the following form:

```
[op1:{billAt:<list of intervals>,
      due:<value>},
 op2:{billAt:<list of intervals>,
      due:<value>},...]
```

Form 15: opPayments value

where the *billAt* value is a list of financial intervals that represents the beginning of every billing cycle and the *due* represent the number of financial intervals before these bills become due.

4.4 Analytic Model (AM)

Here we describe the investment analytic model (AM), i.e., the function that computes a valid output (*out*) from a valid input (*in*). Notes that parts of *out*, including *config* and *rootServiceID* are identical to those in *in*. In the following we describe the computation of all parts of *out* that are computed (and not identical to the input.) To compute $out(services)$, let $ID = keys(in(services))$ be the set of all services id's. Also, $out(rootServiceID) = in(rootServiceID$. The *services* part of the output, $out(services)$, is computed in two steps:

$$out(services) = \bigcup_{id \in ID} \boldsymbol{mOut}\Big(\boldsymbol{periodsOut}(in(services)(id))\Big)$$

where *periodsOut* is a recursive function that take a service input form and return the same service after it computes its *inFlow*, *outFlow*, *constraints* and updates the state for every interval in the time horizon. Whereas the *mOut* is a recursive function that takes a service, in the form returned by *periodsOut*, and computes the metrics for every period in the time horizon to construct the *out(services)* form. We describe the computation of *periodsOut* in Section 4.4.1 and *mOut* in Section 4.4.2 below.

4.4.1 periodOut

In this section, we show how *periodsOut* function calculates the *inFlow* and *outFlow* quantities for each window's interval over the investment periods as well as some other constraints depend on the type of the service.

The *composite* service(cs): Let CS be a set of all composite service IDs. Also, let the expression $wLength(p, w)$ denotes to the length of window w at period p, which is the short form of $in(config)(horizon)(p)(windows)(w)(length)$. The expression $qtyIn(id, x, p, w, i)$ denotes to the quantity of *inFlow* ID x of the service *id* at period p, window w and interval i, which is the short form of $out(services)(id)(inFlow)(f_x)(p)(w)(qty)(i)$. In the same way, we express the $qtyOut(id, x, p, w, i)$.

Therefore, for every *composite* service $cs \in CS$, and every flow $i \in keys(in(services)(cs)(inFlow))$ and $j \in keys(in(services)(cs)(outFlow))$ at period $p \in \{1, \ldots, P\}$, window $w \in keys(in(config)(horizon)(p)(windows))$ and interval $k \in \{1, \ldots, wLength(p, w)\}$, the *inFlow* quantity (*qty*), as in Form 4, is expressed recursively as:

$$\begin{aligned} &cs(inFlow)(f_{\mathrm{i}})(p)(w)(qty)(k) \\ &= \sum_{sub \in cs(subService)} \Big(qtyIn(sub, i, p, w, k) \\ &\quad - qtyOut(sub, j, p, w, k)\Big) \end{aligned} \tag{3}$$

Therefore, the (*total*) for every *inFlow* is expressed as:

$$\begin{aligned} &cs(inFlow)(f_i)(p)(w)(total) \\ &= \sum_{k=1}^{wLength(p,w)} services(cs)(inFlow)(f_i)(p)(w)(qty)(k) \end{aligned} \tag{4}$$

As with the *inFlow* above, the *outFlow* is expressed in a similar way.

Also, for every composite service $cs \in CS$, the (*constraints*) is expressed as a conjunction of constraints: to insure that the demand is satisfy (*demandConstraint*); to verify that the lower bound flows are met (*boundConstraint*) and to guarantee all sub services constraints are satisfied (*subServiceConstraints*). To expression each constraint, let the $inKeys(id)$ and $outKeys(id)$ denote to the keys in $keys(in(services)(id)(inFlow))$ and $keys(in(services)(id)(outFlow))$, respectively. Also, the $seq(p, x)$ denotes to x^{th} window at period p according to the window sequence at $in(config)(horizon)(p)winSeq(x)$.

Therefore, for every subservice $sub \in in(services)(cs)(subService)$, and flow key in $i \in \Big\{[outKeys(sub) \cup inKeys(sub)] - [inKeys(cs) \cup outKeys(cs)]\Big\}$, in period $p \in \{1, \ldots, P\}$, at window $w \in keys(in(config)(horizon)(p)(windows))$, and interval $k \in \{1, \ldots, wLength(p, w)\}$, the $domandConstraint(cs)$ is expressed as:

$$\boldsymbol{qtyIn(}sub, i, p, w, k\boldsymbol{)} \geq \boldsymbol{qtyOut(}sub, i, p, w, k\boldsymbol{)}$$

The *boundConstraint* for every $cs \in CS$ and flow key $i \in \{inKeys(cs)\}j \in outKeys(cs)\}$ for every interval $k \in \{1, \ldots, wLength(p, w)\}$ in window $w \in keys(in(config)(horizon)(p)(windows))$ at period $p \in \{1, \ldots, P\}$ is expressed:

$$\begin{aligned} &in(services)(cs)(inFlow)(i)(p)(w)(LB)(k) \\ &\qquad \leq \boldsymbol{qtyIn(}cs, i, p, w, k\boldsymbol{)} \\ &in(services)(cs)(outFlow)(j)(p)(w)(LB)(k) \\ &\qquad \leq \boldsymbol{qtyOut(}cs, j, p, w, k\boldsymbol{)} \end{aligned}$$

Similar to the lower bound (LB) constraint above, we can add the upper bound (UB) constrains.

Finally, the *subServiceConstraints* is expressed for every $cs \in CS$ as follow:

$$\begin{aligned} &\forall sub \in in(services)(cs)(subService) \\ &\qquad out(services)(sub)(constraints) \end{aligned}$$

The *atomic* service (as): Let AS be a set of all atomic services IDs. For every atomic service $as \in AS$, the *state* is expressed for every interval $k \in \{1, \ldots, wLength(p, w)\}$ within every window $w \in keys\Big(in(config)(horizon)(p)(windows)\Big)$ at period $p \in \{1, \ldots, P\}$:

$$\begin{aligned} &out(as)(state)(p)(w)(k) \\ &= \begin{cases} in(as)(state) & \text{,if } k = 1 \\ newState(as, out(as)(state)(p)(w)(k-1)), else \end{cases} \end{aligned} \tag{5}$$

where *newState* is a function that returns the new state from a given previous *state* for a given atomic service.

For every atomic service the quantity (***qty***) of every *inFlow* and *outFlow* is calculated by calling the analytical model of its *type* (see the appendix-Atomic Model Library). Additionally, every atomic ***constraints*** is expressed as a conjunction of constraints to:

- set a lower(upper) bound for every flow, using similar expression as in the composite service section 4.4.1 .
- guaranty that the number of (ON) units for every interval does not exceed the number of invested units:

$$in(services)(as)(numUnitON)(p)(w)(k) \leqslant in(services)(as)(initAvaliable) + \sum_{x \in \{1,\dots,p\}} in(services)(as)(numUnitInvested)(x)$$

- guaranty that the flows quantities do not exceed the capacity of (ON) units for every $as \in AS$, $i \in \{outKeys(as)$, $p \in \{1,\dots,P\}$, $w \in keys\Big(in(config)(horizon)(p)(windows)\Big)$, and $k \in \{1,\dots,wLength(p,w)\}$:

$$qtyOut(as,i,p,w,k) \leq \Big(in(services)(as)(numUnitON)(p)(w)(k) * in(services)(as)(capacityPerUnit)\Big)$$

- maintain the assumption that ensures uniform flow quantities and atomic services' states at the beginning and at the end of each window, so that the windows' order within the same period and between the periods do not cause the flows or the states to be inconsistent. Therefore, for every *inFlow* key $i \in \{keys(in(services)(as)(inFlow))\}$ and *outFlow* key $j \in \{keys(in(services)(as)(outFlow))\}$ and at each window $w \in keys\Big(in(config)(horizon)(p)(windows)\Big)$ at period $p \in \{1,\dots,P\}$, The following assumption must be fulfilled:

$$\begin{aligned} &qtyIn(as,i,p,w,1) \\ &\quad = qtyIn(as,i,p,wLength(p,w)) \\ &qtyOut(as,j,p,w,1) \\ &\quad = qtyOut(as,j,p,w,wLength(p,w)) \\ &out(as)(state)(p)(w)(1) \\ &\quad = out(as)(state)(p)(w)(wLength(p,w)) \end{aligned}$$

- Any domain specific constraints (see the appendix-Atomic Model Library)

4.4.2 metricOut (mOut)

In this section, we show how the metricOut **(*mOut*)** calculates the *metrics* for all services given the result of the *periodOut* function over all periods to form the out(service), as in form 1.

The *composite* service: Let $pOut(cs)$ be the short form for $periodsOut(in(services)(cs))$ and $windows(p)$ denote to the windows ids for a given period p, which is the short form for keys(in(config)(horizon)(p)(windows)). Then, for every composite service $cs \in CS$ and at every period $p \in \{1,\dots,P\}$, every ***metric*** m_i (such as cost and CO_2) is expressed recursively as:

$$out(services)(cs)(metrics)(m_i)(perPeriod)(p) = \sum_{\substack{sub \in in(services)\\(cs)(subServices)}} mOut(pOut(sub)(perPeriod)(p))$$

while the *total* metric value at a given composite service is expressed as:

$$out(services)(cs)(metrics)(m_i)(total) = \sum_{p \in \{1,\dots,P\}} out(services)(cs)(metrics)(m_i)(perPeriod)(p)$$

Note that the *cashFlow* value is a list, as in Form 7, that needs to be aggregated by combining the amounts that accrued within the same interval. The result is a list of the same form.

Other financial metrics that depend on the *cashFlow* (e.g., NPV) use the result of the *cashFlow* and some financial parameter located in $in(config)$ (e.g., intRate) to calculate its value.

The *atomic* service (as): For atomic services, the metric *perPeriod* is calculated by multiplying the metric value of a given window' intervals w with the number of times the window is repeated in the given period:

$$out(services)(as)(metrics)(m_i)(perPeriod)(p) = \sum_{w \in windows(p)} \Big(count(p,w) * \sum_{\substack{k \in \{1,\dots,\\wLength(p,w)\}}} mOut(services)(as)(m_i)(p)(w)(k)\Big)$$

where *count(p,w)* is a function that counts the number of windows of type w at period p, which can be found at $config(horizon)(p)(winSeq)$.

The *total* metric value is calculated by aggregating the value of a given metric over all periods as follow:

$$out(services)(as)(metrics)(m_i)(total) = \sum_{\substack{p \in\\ \{1\dots P\}}} out(services)(as)(metrics)(m_i)(perPeriod)(p)$$

To calculate the *cashFlow*, we need to append all the investment expenses as well as the operational expenses. Therefore, we need to calculate the financial intervals and the amounts (e.g., the days in which the investment amounts are due). Thus, for every $p \in \{1, \dots, P\}$ and payment $i \in in(services)(as)(payments)(invPayments)(p)$, the intervals, as in Form 7, are calculated as follow:

$$in(services)(as)(payments)(invPayments)(p)(i)(due) + \Big[\sum_{\substack{i \in \{1,\dots,p-1\} \\ w \in windows(p)}} wLength(p,w) * count(p,w)\Big] \div in(config)(intervalsRatio)$$

while the amounts associated with the above intervals are multiplied by the number of units invested at period p as follow:

$$in(services)(as)(payments)(invPayments)(p)(i)(amt) * in(services)(as)(numUnitInvested)(p)$$

On the other hand, the operational expenses use billing periods (e.g., at the beginning of every month) to calculate the expenses at that period (e.g., maintains expenses). Not that this expenses might be variable or fixed depend on the type of operation op and atomic service analytical model(AM). Thus, for every period $p \in \{1, \dots, P\}$, payment $op \in in(services)(as)(payments)(opPayments)(op)$, and billing cycle start at $b \in in(payments)(opPayments)(op)(billAt)$, the interval for every cashFlow appended is as follow:

$$in(services)(as)(payments)(opPayments)(op)(b) + in(services)(as)(payments)(opPayments)(op)(due)$$

Whereas the amount associated with the above interval are type specific and are calculated using the analytical model for its specific service type in the library. (see Section 4.4.3)

4.4.3 Atomic Model Library

Due to page limitation we select two atomic analytic models (AMs) from the library, to describe how they calculate the flows, constraints and the domain specific metrics.

Energy Contract (ec)

The Energy Contract AM takes no $inFlow$, but generate power, in the $outFlow$, for every service it supports. The amount of power for every $outFlow$ service can be represented as a decision variables for every windows' interval over the investment periods in the input. Therefore, for every $outFlow$ key $f_j \in \{keys(out(services)(ec)(outFlow))\}$ and for every interval $k \in \{1, \dots, wLength(p,w)\}$ within every window $w \in keys\Big(in(config)(horizon)(p)(windows)\Big)$ at period $p \in \{1, \dots, P\}$ the $outFlow$ qty is expressed as:

$$ec(outFlow)(f_j)(p)(w)(qty)(k) = in(services)(ec)(typeSpecific)(f_j)(w)(qty)(k)$$

For every billing period (bp) (e.g., every month), defined under the *opPayments*, the AM calculate the power consumption rate (pcr) by tracking the *inFlow* of power within the given billing period. Then, the average KWH can be used to control the power inflow to minimize the peak demand bound, which then reduces the fluctuation of power and reduces the power cost, as follow:

$$\begin{aligned} avgKWH &= pcr/bp \\ extraCost &= (pdb - avgKWH) \\ &\quad * peakPricePerExtraKWH \\ totalCost &= (pcr * costKWH) + extraCost \end{aligned}$$

The AM also calculate the actual emissions by multiplying the emission factor by the actual consumption, as follow:

$$emission = factor * pcr$$

The above cost is updated in the *cashFlow* using the structure in Form 7.
Some constraints are added to insure that pdb in the problem is bounded ,as follow:

$$pdb \geq avgKWH$$

Water Storage (ws)

The water storage AM use the quantities of water $inFlow$ and $outFlow$ at every operational interval k and the *state* of the water at the beginning of the time horizon to calculate new state as in Equation 5. The *newState* function can be expressed as follow:

$$\begin{aligned} & out(ws)(state)(water)(p)(w)(k) \\ &= out(ws)(state)(water)(p)(w)(k-1) \\ &\quad + out(ws)(inFlow)(water)(p)(w)(k) \\ &\quad - out(ws)(outFlow)(water)(p)(w)(k) \end{aligned}$$

The AM also calculate a constraint that balance the level of water over all interval:

$$\begin{aligned} & out(ws)(state)(water)(p)(w)(k) \\ &+ out(ws)(inFlow)(water)(p)(w)(k) \\ &\geq out(ws)(outFlow)(water)(p)(w)(k) \end{aligned}$$

Note that the "state" of water and water $inFlow$ and $outFlow$ have to be greater than zero.

5 Experimentation

In this section we evaluate the system capability in optimizing the investment of real size problem. The experimentation was performed on a batch-processing cluster using a single core of AMD Opteron Processor 6276. We used CPLEX 12 as an optimization tool.

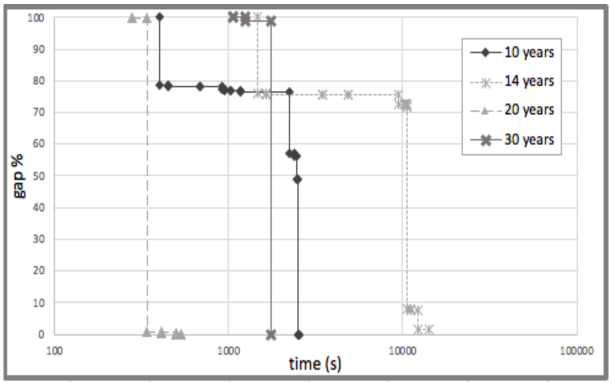

Figure 5: CPLEX solution progress

We created four different cases using the desalination system architecture shown in Figure 2 to set up the optimal operational control and to determine the optimal expansion in the number of infrastructures' units to meet the increasing demand rate of 1.8 over the years. The problem is constrained by balancing the flows within and across the desalination system components as well as unique business constraints such as lowering the peak power consumption each billing period to reduce the billing cost of power.

In the first case, we run the model to optimize 10 years of operation over two investment periods. Each period includes eight different windows of operations to represent the change in the daily demand of fresh water over the four seasons (spring, summer, fall, and winter). The first feasible solution for this problem was found after 7 minutes with gap of 78%, as shown in Figure 5 between the best known solution and the best possible one. The problem then gradually converge to reach the optimal solution after 43 minutes.

By increasing the time horizon to 14 years, the first feasible solution was found after 24 minutes. Then slowly converged to a gap of 71% after 3 hours. After that the gap dropped with a distance around 1% from the optimal solution. In the third case, we adjust the time horizon to 20 years and found the first solution at 5 minutes with a gap of of 99.87%, then drop to reach the optimal solution after 9 minutes.

In the last case, we increase the number of years to 30 and the number of investment period to 3 periods, which increase the number of decision variables. The first optimal solution was found after 18 minutes with a gap of 99.97%, then the solution sharply drop to the optimal solution at 29 minutes.

Our experiment represents an initial step to study the visibility of applying the model in solving realistic investment problems as most of the cases converge to the optimal solution within a reasonable time.

Years	Periods	Binary variables	Total decision variables	Total constraints
10	2	347,5	176,83	277,652,47
14	2	347,5	176,83	336,098,18
20	2	347,5	176,83	416,481,96
30	3	521,2	265,24	469,585,22

Table 1: data set

6 Conclusion

This paper is an enhancement for a previously proposed general investment model. We explained the moving parts of the model using the desalination system example, and we showed the architecture of our proposed decision guidance system. The initial experimentation showed encouraging results in employing the model to solve real problems. Further work is needed to expanding the repository. Moreover, we plan conduct a case study to explore the capability of implementing a preprocessing algorithm to decompose the problem resulted form the analytical model into pre-solved operational problems to reduce time complexity of the optimization problem.

Acknowledgment

These experiments were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University, VA.
(URL: http://orc.gmu.edu)

References

[1] B. Carvalho Trindade, "Advancing regional water supply portfolio management and cooperative infrastructure investment pathways under deep uncertainty," Ph.D. dissertation, Cornell University, 2019, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-06-04.

[2] M. Al-Nory, "Optimal decision guidance for the electricity supply chain integration with renewable energy: Aligning smart cities research with sustainable development goals," *IEEE Access*, vol. PP, pp. 1–1, 05 2019.

[3] A. B. Bedor Alyahya, "A decision guidance system for optimal operation of hybrid power desalination service network," Available at http://cs.gmu.edu, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030-4444 USA, Tech. Rep. GMU-CS-TR-2020-3, 2020.

[4] A. M. Abdelshafy, H. Hassan, and J. Jurasz, "Optimal design of a grid-connected desalination plant powered by renewable energy resources using a hybrid pso–gwo approach," *Energy Conversion and Management*, vol. 173, pp. 331–347, 10 2018.

[5] A. A. Gune, "Real options and networks," 2007.

[6] X. Xin, X. Wang, L. Ma, K. Chen, and M. Ye, "Shipping network design–infrastructure investment joint optimization model: a case study of west africa," *Maritime policy and management*, pp. 1–27, 2021.

[7] A. K. Hueffed, "Analysis and optmization of chp, cchp, chp-orc, and cchp-orc systems," Ph.D. dissertation, Mississippi State University, 2010, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-20.

[8] J. D. Osorio, "Modeling and optimization of a concentrated solar supercritical co2 power plant," 2016.

[9] F. E. Andersen, M. S. Díaz, and I. E. Grossmann, "Multiscale strategic planning model for the design of integrated ethanol and gasoline supply chain," *AIChE journal*, vol. 59, no. 12, pp. 4655–4672, 2013.

[10] O. Dey, "A fuzzy random integrated inventory model with imperfect production under optimal vendor investment," *Operational Research*, vol. 19, no. 1, pp. 101–115, Mar 2019.

[11] S. M. Neiro and J. M. Pinto, "A general modeling framework for the operational planning of petroleum supply chains," *Computers and Chemical Engineering*, vol. 28, no. 6, pp. 871 – 896, 2004, fOCAPO 2003 Special issue.

[12] B. Alyahya and A. Brodsky, "A decision guidance system for optimal operation of hybrid power desalination service network." in *ICORES*, 2021, pp. 416–424.

[13] P. Trucco, E. Cagno, and M. D. Ambroggi, "Dynamic functional modelling of vulnerability and interoperability of critical infrastructures," *Reliability Engineering & System Safety*, vol. 105, pp. 51 – 63, 2012, eSREL 2010.

[14] S. Thacker, R. Pant, and J. W. Hall, "System-of-systems formulation and disruption analysis for multi-scale critical national infrastructures," *Reliability Engineering & System Safety*, vol. 167, pp. 30 – 41, 2017, special Section: Applications of Probabilistic Graphical Models in Dependability, Diagnosis and Prognosis.

[15] P. Zhang and S. Peeta, "A generalized modeling framework to analyze interdependencies among infrastructure systems," *Transportation Research Part B: Methodological*, vol. 45, no. 3, pp. 553 – 579, 2011.

[16] N. Goldbeck, P. Angeloudis, and W. Y. Ochieng, "Resilience assessment for interdependent urban infrastructure systems using dynamic network flow models," *Reliability Engineering and System Safety*, vol. 188, pp. 62–79, 2019.

[17] J. Kong and S. P. Simonovic, "A model of interdependent infrastructure system resilience," *International journal of safety and security engineering*, vol. 8, no. 3, pp. 377–389, 2018.

[18] G. M. Righetto, R. Morabito, and D. Alem, "A robust optimization approach for cash flow management in stationery companies," *Computers & Industrial Engineering*, vol. 99, pp. 137 – 152, 2016.

[19] ——, "Cash flow management by risk-neutral and risk-averse stochastic approaches," *Journal of the Operational Research Society*, vol. 0, no. 0, pp. 1–14, 2019.

[20] B. Alyahya. and A. Brodsky., "Infrasmart: A decision guidance system for investment in infrastructure service networks," in *Proceedings of the 22nd International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, INSTICC. SciTePress, 2020, pp. 370–380.

[21] A. Brodsky and J. Luo, "Decision guidance analytics language (dgal)," in *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1*, ser. ICEIS 2015. Portugal: SCITEPRESS - Science and Technology Publications, Lda, 2015, pp. 67–78. [Online]. Available: http://dx.doi.org/10.5220/0005349600670078

[22] M. O. Nachawati, A. Brodsky, and J. Luo, "Unity decision guidance management system: Analytics engine and reusable model repository." in *ICEIS (1)*, 2017, pp. 312–323.

[23] A. Brodsky and X. S. Wang, "Decision-guidance management systems (dgms): Seamless integration of data acquisition, learning, prediction and optimization," in *Proceedings of the 41st annual Hawaii international conference on system sciences (HICSS 2008)*. IEEE, 2008, pp. 71–71.

[24] A. Brodsky and J. Luo, "Decision guidance analytics language (dgal)," in *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1*, ser. ICEIS 2015. Portugal: SCITEPRESS - Science and Technology Publications, Lda, 2015, pp. 67–78. [Online]. Available: http://dx.doi.org/10.5220/0005349600670078