InfraSmart: A Decision Guidance System for Investment in Infrastructure Service Networks

Bedor Alyahya and Alexander Brodsky[a]
Department of Computer Science, George Mason University Fairfax, VA 22030, U.S.A.

Keywords: Service Network, Investment Model, Infrastructure, Optimization, Cost-effective.

Abstract: Current approaches to infrastructure investment either (1) model the problem in high-level financial terms, but do not accurately express the underlying system behavior and non-financial performance indicators, or (2) are hard-wired to infrastructure silos, and do not take into account the complex interaction across these silos. This paper proposes to bridge the gap by modeling interrelated infrastructures as a hierarchical service network operating over a time horizon, as well as an extensible repository of infrastructure-specific component models. The paper reports on formal modeling, the development and an initial experimental study of *InfraSmart*, a decision guidance system for investment in interdependent infrastructure service networks.

1 INTRODUCTION

Capital investment in interrelated infrastructures, such as manufacturing, supply chain, renewable energy and smart grid, is vital for accomplishing organizational or societal long-term goals and enabling future growth. Infrastructures often require multi-billion dollar investments which are difficult to reverse or liquidate (Migliore and Mccracken, 2001).

Analyzing and making actionable recommendations on investment in infrastructures is challenging due to (1) complex interaction among different network components such as supply, manufacturing, transportation and energy; (2) trade-offs between multiple objectives and performance metrics; and, (3) uncertain patterns of supply and demand of resources. As outlined by Hsieh and Liu (2004) multi-objectivity in infrastructure decisions, numerous alternatives and temporal resource constraints make the problem even more complicated.

There has been extensive research in modeling and optimizing the investment in infrastructures, e.g., see (Breen et al., 2019; Dey, 2019; Hsieh and Liu, 2004; Manca et al., 2010). However, as we explain below, these models either (1) express the investment model in a very generalized way that fails to accurately express the underlying system behavior over the investment time horizon, or (2) are hard-wired to a silo domain-specific investment problem, which does not take into account the often complex interaction with interrelated infrastructures across the silos. These limitations inhibit the wide-spread adoption and reusability of these models.

The work (Hsieh and Liu, 2004) is an example of category (1), in which the authors approach the investment allocation problem as resource scheduling using genetic algorithms. However, this approach does not try to model the physical infrastructure systems and their operational controls which may effect investment performance indicators.

Modeling techniques around critical interrelated infrastructure protection and optimal performance under disruption have also been studied, e.g., (Trucco et al., 2012; Thacker et al., 2017; Zhang and Peeta, 2011). However, this work does not focus on infrastructure investment.

Works that attempted to model inter-dependencies among infrastructure systems were discussed in the review paper by Ouyang (2014). Among these models is the *network flow*, which represents a general structure that can depict how units are transferred between different infrastructures. As stated by (Holden et al., 2013)

> ”A major advantage of network flow models is that a single mathematical formulation can describe flows of commodities in *different* infrastructure systems”.

An initial step in this direction was made in developing a general financial optimization model by

[a] https://orcid.org/0000-0003-1694-0913

Alyahya, B. and Brodsky, A.
InfraSmart: A Decision Guidance System for Investment in Infrastructure Service Networks.
DOI: 10.5220/0009398003700380
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 370-380
ISBN: 978-989-758-423-7

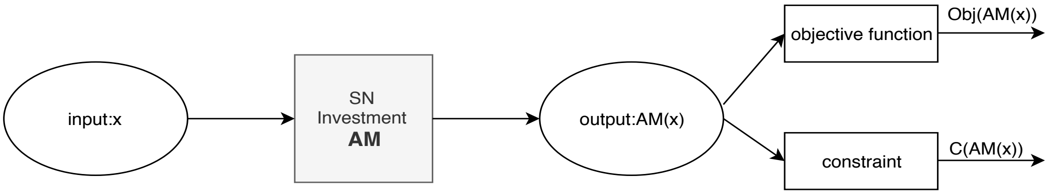

Figure 1: Service Network Investment AM.

Golden et al. (1979) based on network flows. By using network flows, the model increases its flexibility on handling input and output data dependencies between interdependent system components, which makes it very expressive compared to other approaches. Righetto et al. (2016, 2019) extends the above work by addressing uncertainty of the model. However, these models only optimize the financial management of the cash flow, but do not model underlying network components to enable express non-financial performance indicators, such as environmental and safety metrics.

There also has been work on optimizing cash flows in production networks using manually crafted mathematical programming models. However, these models are designed to optimize domain-specific problems, such as water supply (Manca et al., 2010), supply chain (Dey, 2019; Neiro and Pinto, 2004) and energy (Home-Ortiz et al., 2019). All of the above mentioned models are hard-wired to a specific domain which makes it difficult to generalize and take into account an often complex interaction among interdependant infrastructures and their components.

The focus of this paper is on overcoming the aforementioned limitations of investment decisions made in silos, as opposed to accounting for the synergistic value of strongly interdependent infrastructures. More specifically, the contributions of this paper are as follows. First, we develop a formal Analytic Model (AM) for Service Network (SN) investment over a time horizon, as well as an extensible repository of domain-specific component models, initially including supply, contract manufacturing and transportation.

The investment AM expresses, over the time horizon, (1) financial, environmental and quality-of-service metrics, and (2) capacity and demand constraints, as a function of investment and operation decision variables, such as investment choices and network planning and operational controls. The atomic model repository is designed to be extensible, so that additional component models can be added without the need to modify the SN model or previously defined components. For example, new component models may include unit manufacturing processes, elements of supply chain, transportation and logistics, and power network components from generation to transmission and distribution to renewable energy sources and power storage. The SN investment AM leverages our previous work on modeling the operation of (but not investment in) manufacturing service networks (Brodsky et al., 2019; Brodsky et al., 2017; Brodsky and Wang, 2008).

Second, we develop *InfraSmart*, a Decision Guidance System (DGS) to enable stakeholders to derive actionable recommendations on inter-dependent SNs investment, based on optimization of performance metrics under the assumption of optimal operation controls. To develop *InfraSmart*, we implement the formalized AM using Decision Guidance Analytics Language (DGAL) (Brodsky and Luo, 2015), and perform optimization and analysis using Unity Decision Guidance Management System (Unity DGMS) (Nachawati et al., 2017; Brodsky and Wang, 2008). The technical uniqueness of *InfraSmart* lies in modularity and composability of simulation-like AMs, yet without manually crafting mathematical programming (MP) models, which are instead machine-generated. This results in order-of-magnitude productivity gain, as well as quality of results and computational efficiency of the best available MP algorithms, which significantly outperform simulation black-box-based algorithms.

Third, we demonstrate the use of *InfraSmart* and its methodology by providing an example of a service network comprised of suppliers, transportation providers, and Tier 1 and 2 manufacturers. Finally, we conduct an initial experimental study on four problem instances of various computational complexity in terms of a number of atomic services and added combinatorial constraints. The initial results demonstrate computational feasibility of *InfraSmart*, at least on the tested examples, although more experimentation will be needed for different types and sizes of service networks and components.

The paper is organized as follows. Section 2 illustrates how the investment model works by using a simple supply chain example; Section 3 formalizes the investment model; Section 4 gives an overview of the *InfraSmart* architecture and methodology; and, Section 5 discusses the results of an initial experimental study. Finally, Section 6 presents concluding remarks and briefly outlines directions for future work.

2 INVESTMENT BY EXAMPLE

The purpose of the investment model is not to represent the domain-specific optimization problem directly, but instead to represent it using a general investment analytic model (AM), as shown in Figure 1. This AM uses a generic input structure that describes the domain-specific problem and defines the controlled parameters that need to be optimized. Using this input, the investment AM produces an output that contains (1) aggregated periodical performance metrics that are used to define the objective function of the optimization problem and (2) feasibility that serves as optimization constraints. This separation allows the AM to be reused to optimize other investment problems.

To demonstrate how the investment model works, we use a simple supply chain example that depicts, as in Figure 2, the delivery of raw materials from suppliers to manufacturing facilities through transportation lines. To manufacture the final products, Tier 2 facilities rely on supplying parts from Tier 1 facilities. As can be observed in Figure 2, there are multiple decision paths in which these products can be produced to meet the periodical demand. Some of these paths require investing in infrastructures. In this example, we consider transportation line 2 and manufacturing facilities A2 and B2 to be investment opportunities.

To analyze if it is worthwhile to invest in these infrastructures, the investment model should optimize performance metrics, such as cost, through these investment and operational decisions over a given time horizon. To enable the investment model to aggregate the performance metrics generated by these infrastructures over time, we use a generic structure called a *service network* (SN). The SN, as described in (Brodsky et al., 2017), represents a hierarchy of services that are linked together to depict flow of commodity over the network. The services at the bottom of the hierarchy, called atomic services, represent infrastructures that are either owned by the organization or considered to be an investment opportunity. In our example, the dotted boxes (e.g., combined supply) represent composite services which contain two or more nested services. Within a composite service, each nested service can be either composite (e.g., Tier 1 in combined manufacturer) or atomic (e.g., Supplier 1 and 2 in combined supplier). For each atomic service, the SN defines the initial status of these infrastructures as well as other fixed and controlled parameters that are needed to define the capacity and demand constraints and calculate the performance metrices. Also, each arrow in the figure represents the flow of items through these infrastructures to produce certain products.

The investment analytic model (AM) uses an input structure that contain of (1) temporal parameters that allow the user to define the time horizon in which these investments are evaluated; (2) Service Network which describes how these infrastructures are linked together as well as the status of these infrastructures; and, (3) repository of atomic analytic models (AMs) that define how each infrastructure type generated its performance metrics as well as the feasibility constraints at the infrastructure level (eg,. The level of inventory in the supplier).

The proposed model uses this input to optimize financial metrics that are generated bottom-up by invoking for each infrastructure the corresponding analytic model type. This model type calculates some financial metrics and determines feasibility to satisfy periodical demands at both the network level and the infrastructure level. The analytic model also updates the state of some infrastructures that are needed to calculate the next period performance metrics and constraints. For example, the supplier analytic model represents how Supplier 1 and 2 (1) calculated the supplying cost as a function of item quantity and; (2)generated a new state reflecting the inventory reduction after pulling items at a given period; and, (3) defines whether it can provide this quantity, given the available supplier stock at a certain point in time. In the next section, we formally define the service network investment model (SNIM).

3 FORMALIZATION OF SERVICE NETWORK INVESTMENT MODEL

3.1 High-level Optimization Problem

As mentioned in section 2, the optimization model for Service Network Investment is based on the concept of the analytic performance model (AM), which describes the performance metrics such as net present value (NPV) and internal rate of return (IRR) as well as feasibility as a function of fixed and controlled performance.

To formalize AM, the following notations are used for the Service Network Investment Model (SNIM):

d	A valid input instance to SNAM, which contains fixed and controlled parameters.
DD	The set of all valid inputs d.

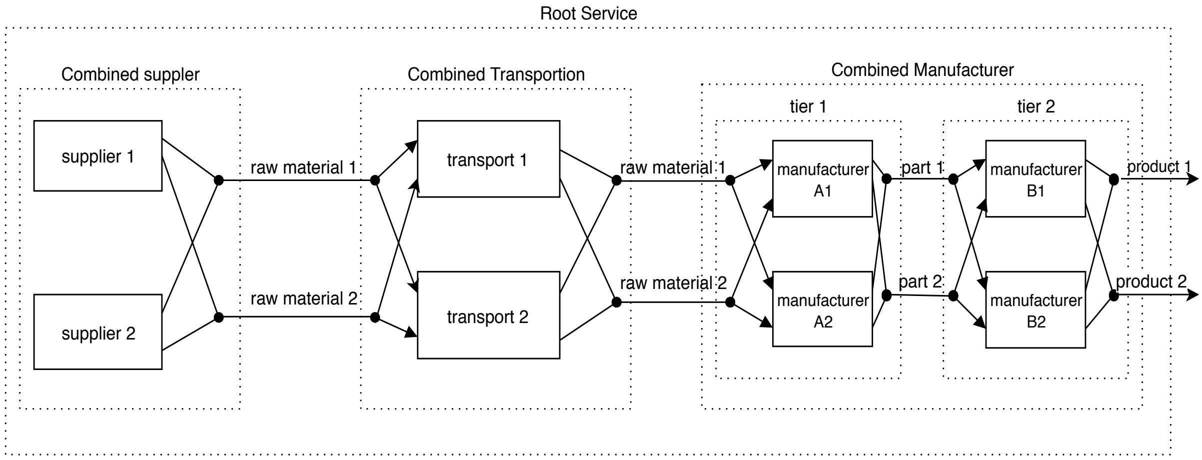

Figure 2: Service Network.

cd	A valid output from the SNAM, which contains performance metrics and feasibility.
CD	The set of all valid outputs cd.
$D \subseteq DD$	The set of all input alternatives, to be considered for optimization.

The analytic performance model AM (formalized in Appendix A) is a function:

$$AM : DD \to CD \tag{1}$$

which forms a valid output $cd \in CD$ of performance metrics, such as NPV and IRR, from a valid input $d \in D$ of fixed and controlled investment parameters. In the context of a particular investment optimization, we assume as given in (2) an objective function, which gives the real objective value in $\mathbb{R}$ given a valid output of the AM.

$$Obj : CD \to \mathbb{R} \tag{2}$$

Also, we assume C given a boolean constraint function:

$$C : CD \to \{T, F\} \tag{3}$$

which gives *True* or *False* given a valid output $cd \in CD$. Then, the investment optimization problem is:

$$\begin{aligned} \min_{x \in D} \quad & obj(AM(x)) \\ \text{s.t.} \quad & C(AM(x)) \end{aligned} \tag{4}$$

The reason we describe the objective and constraints using the analytic model AM, as opposed to describing them directly from $x \in D$ is modularity and flexibility, so that some AM can be used to formulate multiple investment related optimization problems. Now we need to describe all of the components above more formally, starting with a valid service network investment model output instance cd in section 3.2, followed by the input instance d in section 3.3, and finally, we describe the analytic model which is a function that computes an output instance from the input instance in APPENDIX A.

3.2 Service Network Investment Instance: The Model Output

A valid *SN* investment output instance cd is a tuple

$$\langle \textbf{\textit{config, services, rootServiceID}} \rangle$$

where:

- ***config*** is a tuple

  $$\langle \textbf{\textit{unitInterval, interestRate, noPeriods, periodDuration}} \rangle$$

  where:
  unitInterval represents the unit of temporal sequence (e.g. day); ***interestRate*** represents the zero-risk investment rate for net present value calculation; ***noPeriods*** is the number of periods, so that investment decisions can be made over a set of all of periods, $P = \{1, \ldots, noPeriods\}$, in the time horizon; ***periodDuration*** is a function

  $$periodDuration : P \to \mathbb{Z}^+$$

  which gives the duration (in ***unitInterval***, such as days) for every period $p \in P$.
- ***services*** is a set of services $\{s_1 \ldots s_n\}$, where each service s is either a *composite* or an *atomic* service instance. We begin describing common elements of these services in section 3.2.1 ; then in sections 3.2.2 and 3.2.3, we describe the rest that are unique for each service.
- ***rootService*** is the id of a service in *services* designated as a root service.

3.2.1 Common Service Instance

A common service instance is a tuple:

$$\langle \textbf{\textit{id, type, inFlow, outFlow, metrics, constraints}} \rangle.$$

where

- ***id***$\in \mathcal{S}$ is a unique identifier of a service, where $\mathcal{S}$ is the set of all service ids.
- ***type*** is either *composite* or one of the specific available atomic services (such as supplier, manufacturer) described in section 2.
- ***inFlow*** is a tuple:

  $$\langle \textbf{\textit{flowIDs, qtyPP ,totalQty}} \rangle$$

  where:

  – ***flowIDs*** is a set of inflow unique identifiers.

  – ***qtyPP:*** $P\times$ *flowIDs* $\rightarrow \mathcal{N}$, is a function that gives the quantity of flow *fid*$\in$*flowIDs* in period $p \in P$, where $\mathcal{N}$, is a numerical domain (either real numbers $\mathbb{R}$ or integers $\mathbb{Z}$)

  – ***totalQty***: *flowIDs* $\rightarrow \mathcal{N}$, is a function that aggregates all quantities of flow *fid*$\in$*flowIDs* over all periods.
- ***outFlow*** is a tuple that follows the same pattern as in *inFlow* to describe the *outFlow*s of a service.
- ***metrics*:** is a tuple

  $$\langle \textbf{\textit{costPP, totalCost, NPVPP, totalNPV , cashFlow}} \rangle$$

  where:

  – ***costPP:*** $P \rightarrow \mathbb{R}^+$, is a function that gives the cost of running this service at period $p \in P$.

  – ***totalCost:*** is the aggregated cost over all periods.

  – ***NPVPP:*** $P \rightarrow \mathbb{R}$, is a function that gives the net present value of cash flows at period $p \in P$.

  – ***totalNPV:*** $S \rightarrow \mathbb{R}$, is the total NPV over all periods.

  – ***cashFlow:***$\{firstPay, \ldots, lastPay\} \rightarrow \mathbb{R}$, is a function which gives the amount of payment for every pay interval $i \in \{firstPay, \ldots, lastPay\}$, where $\{firstPay, \ldots, lastPay\}$ is the set of all time intervals (e.g. days) with non-zero cash flows. Note that negative payment means cash inflow.
- ***constraints:*** true if the constraints of service investment are satisfied, and false otherwise (see section A).

3.2.2 Composite Service Instance

A composite service instance is a tuple of the form

$$\langle \textbf{\textit{..., subServices}} \rangle.$$

where:
... are composed of the common service instance components (see section 3.2.1) and ***subServices*** is a set $\{id_1 \ldots id_x\} \subseteq \mathcal{S}$, which defines the sub-service ids of this *composite* service.

3.2.3 Atomic Service Instance

An atomic service instance is a tuple of the form

$$\langle \textbf{\textit{..., onFlag, invested, investedAmt, [State]}} \rangle.$$

where:

- ... are composed of the common service instance components (see section 3.2.1).
- ***onFlag*:** $P \rightarrow \{0, 1\}$, is a boolean function that gives, for every period $p \in P$, "1" if the service is running (ON) and "0" otherwise.
- ***invested*:** $P \rightarrow \{0, 1\}$, is a boolean function that gives, for every period $p \in P$, "1" if investment occurs at period p, and "0" otherwise. By stating that the investment "occurs" at period *p*, we mean that the service being invested in becomes available at the beginning of period *p*.
- ***investedAmt*:** is the amount of the investment.

Some atomic services may have an optional element:

- ***state*:** $P \rightarrow dom(type)$, is a function that describes the temporal state of atomic service type in period $p \in P$, where *dom(type)* is the domain of this atomic service type.

In the next section, we describe a valid input model for the investment analytic model which contains fixed and controlled investment parameters, necessary to compute the output instance.

3.3 Service Network Investment Instance: The Model Input

A valid *SN* investment input instance *d* is a tuple

$$\langle \textbf{\textit{config, domainSpecific, inputServices, rootServiceID}} \rangle$$

where:

- ***config, rootServiceID*:** are defined previously in the model output (section 3.3)
- ***domainSpecific*:** defines shared elements that are needed for a domain specific atomic analytic model calculation.

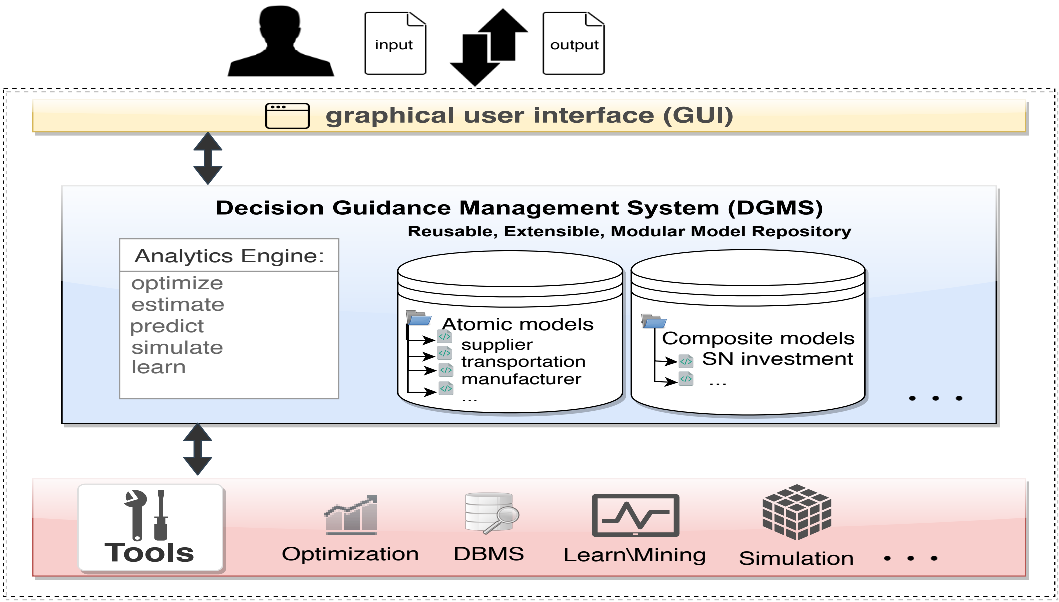

Figure 3: Decision Guidance System Architecture.

- ***inputServices:*** is a set of service input instances, where each service in model input follows the same structure as in the model output. In the following, we only describe differences.

Atomic **or** ***Composite*** **Service:**

- The service does not have *metrics* and *constraints*.
- Every *inFlow* and *outFlow* is tuple

$$\langle \textit{flowIDs, lbPerPeriod,ubPerPeriod} \rangle$$

  where:
  - ***flowIDs*****:** is a set of flow unique identifiers.
  - ***lbPP*****:** $P \times$ *flowIDs* $\rightarrow \mathcal{N}$ is a function that represents the lower bound of a given *flowIDs* for a given period $p \in P$.
  - ***ubPP*****:** $P \times$ *flowIDs* $\rightarrow \mathcal{N}$ is a function that represents the upper bound of a given *flowIDs* for a given period $p \in P$.

Atomic **Service:**

- ***investedAmt*****:** is replaced by ***investAmt*** which defined as:
  investAmt**:** $P \rightarrow \mathbb{R}^+$, is a function that gives a conditional investment amount which depends on the period $p \in P$ in which the investment occurs.
- additional elements are needed:
  - ***initAvailable****:* is a binary value $\{0,1\}$: "1" if the investment is made and the service is available at the beginning of the first period, and "0" otherwise.
  - ***typeSpecific*****:** defines parameters that are needed for a specific atomic-type analytic model calculation.
  - ***[initState]*****:** defines the initial state of this atomic service type.

The user can determine the investment controlled parameters in the model input by annotating (1) the ***invested*** at each infrastructure (i.e., atomic service) that can be considered as an investment opportunity and which periods $\subseteq P$ to look at in the time horizon. (2) the ***onFlag*** and other domain specific parameters, such as quantities of flow, that define how the services network can ideally operate while these investments take place.

4 DECISION GUIDANCE SYSTEM AND METHODOLOGY

A decision guidance system (DGS) is a system that provides recommendations to guide its users in making better decisions. Typical decision guidance systems are designed to solve domain-specific problems. Therefore, building such systems requires major efforts in modeling, designing and developing tightly integrated components which discourage any attempt to reuse or even to extend these systems to solve other problems (Brodsky and Wang, 2008).

A different distinctive approach in building DGS was proposed in (2008) and developed in (Nachawati

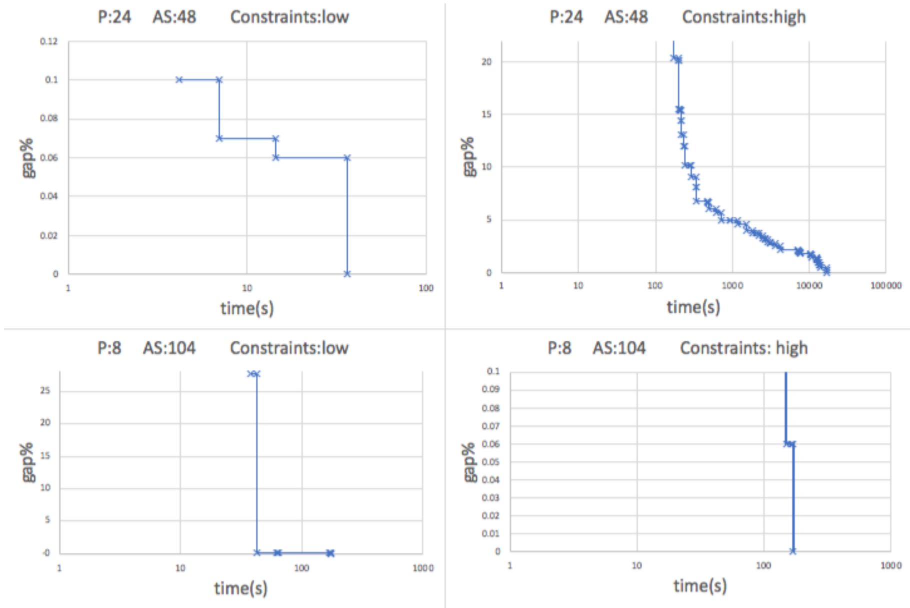

Figure 4: CPLEX solution progress.

et al., 2017). The key idea is to build a decision guidance management system (DGMS) that aids the user in executing different analytical tasks using a repository of reusable, modular and composable models. This architecture also provides an analytics language that hides the complexity in dealing with external tools to perform a variety of different tasks, such as simulation, optimization and learning (Brodsky and Luo, 2015).

In this paper, we develop *InfraSmart* - a decision guidance investment system based on DGMS. The architecture of this system is depicted in Figure 3. The middleware contains the decision guidance management system (DGMS) which in turn consists of a reusable, extensible and modular model repository as well as the analytics engine that symbolically executes and reduces the analytic models to perform analytical tasks. These tasks may require a variety of external tools that support the optimization and trade-off analysis as shown in the bottom of figure 3. The graphical user interface (GUI) in the top of the figure aids to the decision maker in creating and combining atomic models with the SN investment model, which is located in the repository to perform analytical tasks to solve the domain-specific problem.

We demonstrate how the user interacts with *InfraSmart* to solve a domain-specific problem as follows:

1. The user interacts with the GUI to create model input that defines the domain-specific problem through combining the atomic models in the repository.
2. The user annotates the controlled parameters in the model input that need to be optimized.
3. The user configures temporal and financial parameters that are needed to run the investment model.
4. The user defines the financial metric that needs to be optimized.
5. The user runs the investment model to produce the optimal investment and operational setting based on the metrics defined in the previous step.
6. By instantiating the annotated controlled parameters, the system provides a recommendation that aids the user in determining which infrastructures to invest in, as well as when these investment must occur.
7. The user can adjust the input to compare different investment alternatives.

5 INITIAL EXPERIMENTAL STUDY

For the purposes of evaluating *InfraSmart*, we conducted an experiment that aimed to assess the feasibility of this system in handling large-scale problems. We coded the SN investment model as well as the atomic analytic models described in section A.4 using Unity Decision Guidance Management System (DGMS). The experimentation was performed on a machine with a 1.8 GHz Intel Core i5 processor and

8 GB of DDR3 memory executed at 1600 MHz. We used CPLEX 12 as an optimization tool.

We generated four different instances of the supply chain example in section 2 by adjusting the number of periods in the time horizon, varying the number of atomic services, and altering the level of constraints in each problem. Table 1 summarizes the main parameters we used to generate each problem. For each instance as shown in Figure 4, we track the progress of the solver while converging to the optimal solution.

The first instance uses 48 atomic services, has 24 periods in the planning horizon, and is constrained by periodical demand of the final products and the flow of items along the supply chain. The first feasible solution for this problem was found after 4 seconds within the convergence bound of 0.01% and the optimal solution was identified after 36 seconds.

By restricting the total number of suppliers and transportation lines to deal with to 50% over the time horizon, we created combinatorial constraints $\binom{n}{\frac{n}{2}}$ that were added to the first instance to create the second. By adding these constraints, the time it took the solver to find the first feasible solution increased to 22 seconds within the convergence bound of 4.25% and the optimal solution was found after 2.8 minutes.

The third instance uses 104 atomic services, has 8 periods in the planning horizon and is constrained by periodical demand of the final products. The first feasible solution for was found after 38 seconds within the convergence bound of 27.62% and the optimal solution was found after 3 minutes.

The last problem was generated by adding additional combinatorial constraints to the third problem. The first feasible solution was found at 12 seconds with a proven gap of 99.96%. After 2.8 minutes, the gap had been reduced to 20%. After that the solution gradually improved until it reached the optimal solution at 4 hours and 40 minutes.

We can see that all problems converged optimally and all solutions except the last converged within seconds to a near optimal solution. As an initial step, the results look promising for solving realistic investment problems.

Table 1: Sample Dataset.

Number of periods (P)	Number of Atomic Services (AS)	Number of binary variables	Domain specific constraints
8	104	1664	low
8	104	1664	high
24	48	2304	low
24	48	2304	high

6 CONCLUSION

We present in this paper a new generic infrastructure investment model that is based on reusable analytic models. We described the model using a simple supply example where the model extracts and optimizes the domain-specific metrics to solve the investment problem. We also developed *InfraSmart*, a Decision Guidance System (DGS) that helped the investors meet their specified investment goals. The initial experiment shows a promising result that the model can be used to solve realistically-sized planning problems. Further work needs to be done in extending the investment model and expanding the repository by building more atomic analytic models for complex infrastructures.

REFERENCES

Breen, M., Upton, J., and Murphy, M. (2019). Development of a discrete infrastructure optimization model for economic assessment on dairy farms (diomond). *Computers and Electronics in Agriculture*, 156:508 – 522.

Brodsky, A., Krishnamoorthy, M., Nachawati, M. O., Bernstein, W. Z., and Menascé, D. A. (2017). Manufacturing and contract service networks: Composition, optimization and tradeoff analysis based on a reusable repository of performance models. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1716–1725.

Brodsky, A. and Luo, J. (2015). Decision guidance analytics language (dgal). In *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1*, ICEIS 2015, pages 67–78, Portugal. SCITEPRESS - Science and Technology Publications, Lda.

Brodsky, A., Nachawati, M. O., Krishnamoorthy, M., Bernstein, W. Z., and Menascé, D. A. (2019). Factory optima: a web-based system for composition and analysis of manufacturing service networks based on a reusable model repository. *International Journal of Computer Integrated Manufacturing*, 32(3):206–224.

Brodsky, A. and Wang, X. S. (2008). 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)*, pages 71–71. IEEE.

Dey, O. (2019). A fuzzy random integrated inventory model with imperfect production under optimal vendor investment. *Operational Research*, 19(1):101–115.

Golden, B., Liberatore, M., and Lieberman, C. (1979). Models and solution techniques for cash flow management. *Computers & Operations Research*, 6(1):13 – 20.

Holden, R., Val, D. V., Burkhard, R., and Nodwell, S. (2013). A network flow model for interdependent infrastructures at the local scale. *Safety Science*, 53:51 – 60.

Home-Ortiz, J. M., Melgar-Dominguez, O. D., Pourakbari-Kasmaei, M., and Mantovani, J. R. S. (2019). A stochastic mixed-integer convex programming model for long-term distribution system expansion planning considering greenhouse gas emission mitigation. *International Journal of Electrical Power and Energy Systems*, 108:86,95.

Hsieh, T.-Y. and Liu, H.-L. (2004). Genetic algorithm for optimization of infrastructure investment under time-resource constraints. *Computer-Aided Civil and Infrastructure Engineering*, 19(3):203–212.

Manca, A., Sechi, G. M., and Zuddas, P. (2010). Water supply network optimisation using equal flow algorithms. *Water Resources Management*, 24(13):3665–3678.

Migliore, R. and Mccracken, D. (2001). Tie your capital budget to your strategic plan. *Strategic Finance*, 82(12):38,43.

Nachawati, M. O., Brodsky, A., and Luo, J. (2017). Unity decision guidance management system: Analytics engine and reusable model repository. In *ICEIS (1)*, pages 312–323.

Neiro, S. M. and Pinto, J. M. (2004). A general modeling framework for the operational planning of petroleum supply chains. *Computers and Chemical Engineering*, 28(6):871 – 896. FOCAPO 2003 Special issue.

Ouyang, M. (2014). Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety*, 121:43,60.

Righetto, G. M., Morabito, R., and Alem, D. (2016). A robust optimization approach for cash flow management in stationery companies. *Computers & Industrial Engineering*, 99:137 – 152.

Righetto, G. M., Morabito, R., and Alem, D. (2019). Cash flow management by risk-neutral and risk-averse stochastic approaches. *Journal of the Operational Research Society*, 0(0):1–14.

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

Trucco, P., Cagno, E., and Ambroggi, M. D. (2012). Dynamic functional modelling of vulnerability and interoperability of critical infrastructures. *Reliability Engineering & System Safety*, 105:51 – 63. ESREL 2010.

Zhang, P. and Peeta, S. (2011). A generalized modeling framework to analyze interdependencies among infrastructure systems. *Transportation Research Part B: Methodological*, 45(3):553 – 579.

APPENDIX

A Analytic Model (AM)

A.1 Basic Notation

In this section, we use the following notations to formalize AM:

$\mathcal{S}$	a set of service ids in the inputServices model input.
$\mathcal{CS} \subseteq \mathcal{S}$	a set of composite service ids.
$\mathcal{AS} \subseteq \mathcal{S}$	a set of atomic service ids.
$\mathcal{SAS} \subseteq \mathcal{S}$	a set of atomic service ids with state.
$inFIds(s)$	a set of *infow* ids for service $s \in \mathcal{S}$.
$outFIds(s)$	a set of *outfow* ids for service $s \in \mathcal{S}$.
$sub(s)$	a set of *subService* ids for service $s \in \mathcal{CS}$.
$onFlag(s,p)$	is the *onFlag(p)* for a service with id $s \in \mathcal{AS}$ for period $p \in P$, which gives "1" if the service is running (ON) and "0" otherwise. (see section 3.2.3).
$invested(s,p)$	is the *invested(p)* for a service with id $s \in \mathcal{AS}$ for period $p \in P$, which gives "1" if investment occurs and "0" otherwise. (see section 3.2.3).
$investAmt(s,p)$	is the amount required to invest in a service with id $s \in \mathcal{AS}$ in period $p \in P$.
$investedAmt(s)$	is the invested amount for a service with id $s \in \mathcal{AS}$.
$lb(s,p,f)$	is a lower bound *lb* of a service with id $s \in \mathcal{S}$ for period $p \in P$ and flow id $f \in inFIDs(s)$.
$ub(s,p,f)$	is an upper bound *ub* of a service with id $s \in \mathcal{S}$ for period $p \in P$ and flow id $f \in inFIDs(s)$.
$inQtyPP(s,p,f)$	is an inFlow quantity per period *(qtyPP)* for a service with id $s \in \mathcal{S}$, for period $p \in P$ and flow id $f \in inFIDs(s)$.
$outQtyPP(s,p,f)$	is an outFlow quantity per period *(qtyPP)* for a service with id $s \in \mathcal{S}$, for period $p \in P$ and flow id $f \in outFIDs(s)$.
$totalQty(s,f)$	is the total quantity *(totalQty)* for inFlow with flow id $f \in inFIDs(s)$ or outFlow with flow id $f \in$

	$outFIDs(s)$ of a service with id $s \in \mathcal{S}$.
$costPP(s,p)$	is a cost per period $costPP$ of a service with id $s \in \mathcal{S}$ and for period $p \in P$.
$totalCost(s)$	is a total cost $totalCost$ of a service with id $s \in \mathcal{S}$.
$NPVPP(s,p)$	is a NPV per period $NPVPP$ of a service with id $s \in \mathcal{S}$ and for period $p \in P$.
$totalNPV(s)$	is a total NPV $totalNPV$ of a service with id $s \in \mathcal{S}$.
$cashFlow(i,s,p)$	is the amount of payment ($cashFlow$) for pay interval $i \in \{firstPay, \ldots, lastPay\}$ of a service with id $s \in \mathcal{S}$ and for period $p \in P$.
$constraints(s,p)$	true if the constraints of a service with id $s \in \mathcal{S}$ and for period $p \in P$ are satisfied, and false otherwise.
$state(s,p)$	is the state of a service with id $s \in \mathcal{AS}$ at period $p \in P$.
$initAvailable(s)$	"1" if the investment is made and the service is available at the beginning of the first period of a service with id $s \in \mathcal{AS}$.

A.2 Computation for Composite and Generic Atomic Services

Some of the notations above are associated with the model input while others are associated with the model output. We first describe how the ones that are associated with output notation are computed from those associated with model input. For composite service, $inQtyPP(s,p,f)$ for $\forall s \in \mathcal{CS}, \forall p \in P$ and $\forall f \in inFIds(s)$ is expressed, recursively as:

$$inQtyPP(s,p,f) = \sum_{s\prime \in sub(s)} outQtyPP(s\prime,p,f) - inQtyPP(s\prime,p,f)$$

Similarly, $outQtyPP(s,p,f)$ for $\forall s \in \mathcal{CS}, \forall p \in P$ and $\forall f \in outFIds(s)$ is expressed as:

$$outQtyPP(s,p,f) = \sum_{s\prime \in sub(s)} inQtyPP(s\prime,p,f) - outQtyPP(s\prime,p,f)$$

For atomic service, $inQtyPP(s,p,f)$ and $outQtyPP(s,p,f)$ are domain-specific. Section A.4 describes how some atomic analytical models (AMs) formulate the input and output quantities.

Then, the *totalQty(s,f)* for $\forall s \in \mathcal{S}$, and $\forall f \in (inFIds(s) \cup inFIds(s))$ is expressed as:

$$totalQty(s,f) = \begin{cases} \sum_{p \in P} inQtyPP(s,p,f), & \text{if } f \in inFIds(s) \\ \sum_{p \in P} outQtyPP(s,p,f), & \text{if } f \in outFIds(s) \end{cases}$$

The ***metrics*** $costPP(s,p)$ and $npvPP(s,p)$ for $\forall s \in \mathcal{CS}$ and $\forall p \in P$ are expressed as:

$$costPP(s,p) = \sum_{s\prime \in sub(s)} costPP(s\prime,p)$$

$$npvPP(s,p) = \sum_{x \in sub(id)} NPVPP(x,p)$$

Therefore, the *totalCost(s)* and *totalNPV(s)* for $\forall s \in \mathcal{S}$ are expressed as:

$$totalCost(s) = \sum_{p \in P} costPP(s,p)$$

$$totalNPV(s) = \sum_{p \in P} npvPP(s,p)$$

The ***cashFlow(i,s,p)*** for $\forall i \in \{firstPay, lastPay\}, \forall s \in \mathcal{CS})$ and $\forall p \in P$ is expressed recursively as:

$$cashFlow(i,s,p) = \sum_{s\prime \in sub(s)} cashFlow(i,s\prime,p)$$

For every composite service $s \in \mathcal{CS}$ and for every period $p \in P$, the ***constraints(s,p)*** is expressed as a conjunction of *demandConstraint(s,p)*, *boundConstraint(s,p)* and *subServiceConstraints(s,p)*. Each constraint is expressed recursively as follows:

- $demandConstraint(s,p) \equiv$

$$\begin{aligned} &(\forall s\prime \in sub(s)) \\ &\quad (\forall f \in [inFIds(s\prime) \cup outFIds(s\prime) \\ &\quad\quad - (inFIds(s) \cup outFIds(s))]) \\ &\quad inQtyPP(s\prime,p,f) \geqslant outQtyPP(s\prime,p,f) \end{aligned}$$

- $boundConstraint(s,p) \equiv$

$$\begin{aligned} &(\forall f \in (inFIds(s) \cup outFIds(s)) \\ &\quad lb(s,p,f) \leqslant inQtyPP(s,p,f) \leqslant ub(s,p,f) \end{aligned}$$

- $subServiceConstraints(s,p) \equiv$

$$(\forall s\prime \in sub(s))\ constraints(s\prime,p)$$

For every atomic service $s \in \mathcal{AS}$ and for every period $p \in P$, the ***constraints(s,p)*** is expressed as a conjunction of *boundConstraint(s,p)*, *qtyConstraint(s,p)*, *onFlagConstraint(s,p)* and *investedConstraint(s)*. The *boundConstrant(s,p)* is expressed as in the composite service *boundConstrant(s,p)* and the others is expressed as follows:

- $qtyConstraint(s, p) \equiv$

$$(\forall f \in inFIds(s)) \quad (onFlag(s, p) = 0) \rightarrow (outQtyPP(s, p, f) = 0)$$

- $onFlagConstraint(s, p) \equiv$

$$onFlag(s, p) \leqslant initAvailable(s) + \sum_{p' \in \{1...p\}} invested(s, p')$$

- $investedConstraint(s, p) \equiv$

$$0 \leqslant initAvailable(s) + \sum_{p \in P} invested(s, p) \leqslant 1$$

The *investedAmt(s)* for $\forall s \in \mathcal{AS}$ is expressed as:

$$investedAmt(s) = \sum_{p \in P} investAmt(s, p) * invested(s, p)$$

The *state(s,p)* for $\forall s \in \mathcal{SAS}$ and $\forall p \in P$ is expressed as:

$$state(s, p) = \begin{cases} newState(s, p, state(s, p-1)) & \text{if } p \geqslant 1 \\ initState & \text{if } p = 0 \end{cases}$$

where *newState* is a function that returns the new state from a given *state* for a service with id $s \in \mathcal{SAS}$, and period $p \in P$.

A.3 Computation of Output from Input

To formalize the analytic model $AM : DD \rightarrow CD$, where DD is the set of all valid inputs d and CD is the set of all valid outputs cd, we need to describe how a valid input instance $cd \in CD$ is computed from a valid output $d \in DD$.

$$cd = AM(d) = \langle \textbf{config, services, rootServiceID} \rangle$$

where *config* and *rootServiceID* are taken from the input *d*, and each service *srv* ∈***Services*** is computed as follows. For *composite service*:

$$srv = \langle \textbf{id, type, inFlow, outFlow, metrics, constraints, subServices} \rangle$$

where *id*, *type* and *subServices* are taken from the services in the *inputServices*. The *inFlow,outFlow* and *constraints* are computed using the expressions in section A.2. For *atomic service*:

$$srv = \langle \textbf{id, type, inFlow, outFlow, metrics, constraints, onFlag, invested, investedAmt, [state]} \rangle$$

where *id*, *type*, *onFlag*, *outFlow* and *invested* are taken from the atomic services in the *inputServices*. The *investedAmt*, *constraints*, and *state* are computed using the expressions in sectionA.2. The other elements are calculated by invoking the AM of the atomic service type which can be found in the repository of atomic analytic models as described in section 4.

A.4 Atomic Models Formulation

So far we have created three analytic models (AMs): supplier, transportation and manufacturer. Due to page limitation the formalization of the atomic analytic models (AMs) are omitted here.

We briefly describe how each atomic service $s \in \mathcal{AS}$ that belong to these AM types produces in every given period $p \in P$ its performance metrics using these atomic AMs.

The supplier AM generates no *inFlow*s but for every *flowID* in the supplier atomic service, the AM simply takes the *outflow* quantity given in *typeSpecific* elements under the same service in *inputServices*. By knowing the *inFlow* quantity from each raw material, the cost and NPV per period can be calculated using the cost per item of each *flowID* defined in the *typeSpecific* and the *interestRate* located in the *config* in the input model. The supplier analytic model also generate a *newState* that basically update atomic service inventory level after pulling some items.

In the transportation AM, the inflow and the outflow quantities are calculated using the orders that define the source, destination and the quantity from each row material. Some general information that are needed for calculating the cost, such as the distances between the suppliers and manufacturing facility as well as items' weight are shared globally in *domainSpacific*.

The manufacturer AM computes the *inFlow* quantities for each item (raw material or part) from the quantity of each outflow given in *typeSpecific* and the number of units needed from each inflow to produce one unit of each outflow. The cost is computed by using the quantity and the price per unit given in model input.

The cash flow for all these AMs is calculated using the cost payment due and the investment payment due located in atomic service under the *typeSpecific*. These values represent intervals(e.g., day) relative to the beginning of the periods where the cost and the investment are due. For example, 2 means that it is due on the second day of the period, while -3 means that it is due 3 days before the beginning of the period.