Multi-Stage Investment Decision under Contingent Demand for Networking Planning

Miguel F. Anjos * Michael Desroches † Anwar Haque ‡ Oleg Grodzevich§
Hua Wei ¶ Henry Wolkowicz ‖

Abstract

Telecommunication companies, such as Internet and cellular service providers, are seeing rapid and uncertain growth of traffic routed through their networks. It has become a challenge for these companies to make optimal decisions for equipment purchase that simultaneously satisfy the uncertain future demand while minimizing investment cost.

This paper presents a decision-making framework for installing the required equipment into the networks while in the uncertain environment. The framework is based on new multi-stage stochastic programming mathematical models that capture the complexity of the individual Central Office (CO) decision-making process. The models are solved using the on-line NEOS server. Two examples are presented to illustrate the procedure. The optimization model also addresses the equipment pricing problem, i.e., what premium is worth paying for shorter installation times.

1 Introduction

Rapid but uncertain growth in traffic that is routed through telecommunications networks has made it extremely difficult for the companies that own these networks to make optimal decisions for the provisioning of network equipment, i.e., to decide when to buy and what equipment to buy so that the uncertain future demand can be met at minimum investment cost. This paper proposes a decision-making framework for installing the required equipment in the networks in the uncertain environment. This work deals with the costs and timings of the installation of new equipment in a typical telecommunications service provider network. The motivation is to develop algorithms and models to help network planners make the right decision under the contingent demand that is driving the expansion of their network. The proposed framework is based on new multi-stage mathematical models that capture the complexity of the individual Central Office (CO) decision-making process. Stochastic programming is used in the formulation. The problem is solved using the on-line NEOS server (`http://www-neos.mcs.anl.gov/`).

The solution to the decision-making problem will enable network planners to provision, purchase, and install equipment such that the current and future demand can be met, while minimizing the purchasing costs.

There are several related methods that deal with this type of problem. Laguna [8] uses robust optimization to deal with the investment expansion problem for one location. The *real option approach* [4], [5] is popular to deal with capacity problems under uncertain future. Alternatively, one can use stochastic programming [1] for this problem. See also [10] and [9], for the computational issues in stochastic programming.

2 Problem Definition

The existing capacity details of a particular switch/router in a CO and the traffic demand history data (i.e., the growth of the customer base) are given. The objective is to find an optimal decision strategy for buying new equipment. This optimal strategy is based on possible future scenarios. The scenarios in-

*Department of Management Sciences, University of Waterloo, Canada. E-mail: anjos@stanfordalumni.org
†Bell Canada. E-mail: michael.desroches@bell.ca
‡Bell Canada. E-mail: anwar.haque@bell.ca
§Department of Management Sciences, University of Waterloo, Canada. E-mail: ogrodzev@uwaterloo.ca
¶Department of Combinatorics & Optimization, University of Waterloo, Canada. E-mail: h3wei@math.uwaterloo.ca
‖Department of Combinatorics & Optimization, University of Waterloo, Canada. E-mail: hwolkowicz@uwaterloo.ca

clude a time-line for buying necessary equipment to satisfy possible future demand at a minimum cost.

Definition 2.1. An *Option* is a decision that involves: ordering a piece of equipment with capacity C and associated total cost P, and installing it by time period T. (The money is paid at the end of the installation period T, but this assumption is not important and can be changed.)

Definition 2.2. The *scenario tree* consists of *Scenarios*, where each scenario contains a sequence of events (branches) and each event E_{ij} is the transition from node i to node j at the next time period. Each scenario starts at the root node (time 0) and ends at a leaf node.

The scenario tree reveals the uncertainties over the planning horizon; thus a deterministic distribution of the random variables is obtained at each time period. Each node represents a state with a certain demand.

Problem 2.3.

1. **Given:**
   (a) *several pieces of equipment with several options* O_i*;*
   (b) *a scenario tree that describes the future demand;*
   (c) *penalty cost factors for deficiency in satisfying the demand.*
2. **Decision to make:**
   - *Choose the best options at time* 0*, that minimize the penalty deficiency costs plus the total installation costs.*

3 Stochastic Programming (SP) Model

To model Problem 2.3, the following points need to be considered:

- Multistage nature of the problem: Since different options reflect different order/installation times, the problem has to be considered over the given planning horizon consisting of several time periods.
- Infinite vs finite planning horizon: Given a finite number of time periods, the solution can be affected by the singularity at the last time period. (Due to the unknown future beyond that point.) This issue is resolved by extending the horizon to the future simulating an infinite time series.
- Interest Rate: A discount (based on the interest rate) is applied to obtain the present value of future cash flows.

To address the above issues, a general problem formulation is developed that provides a strategy on *options* selection.

Each event E_{ij} representing the transition from node i to node j is assigned a probability π_{ij}. Each scenario S_s then has a probability π_s that is found from the product of probabilities assigned to each of the events constituting the scenario path. (The sum of all scenario probabilities is 1.) There is a family of constraints representing relation between the demand and capacities at each node of the scenario tree.

Defining a deterministic multi-stage SP problem requires the enumeration of all the nodes in the scenario tree. Each node is determined by two indices, time period t and scenario index s.

The deterministic multi-stage SP model for the problem described above follows.

$$\begin{aligned} \min \ & \sum_p \gamma^{-T_p} P_p\, x_p + \sum_s \pi_s \sum_{t=1}^{K} \sum_p \gamma^{-(t+T_p)} P_p\, y_{tp}^s \\ & + \sum_s \pi_s \sum_{t=0}^{K} \vec{\omega} \cdot \vec{z}_{ts} \\ s.t.\ & \vec{U} + \sum_{T_p \le t} \sum_p \vec{C}_p\, x_p + \sum_{\substack{q=1 \\ q+T_p \le t}}^{t} \sum_p \vec{C}_p\, y_{qp}^s \\ & + \vec{z}_{ts} \ge \vec{D}_{ts} \quad \forall\, s, t \\ & \vec{z}_{ts} \ge 0, \quad \text{and } x_p,\ y_{tp}^s \text{ nonnegative integer.} \end{aligned} \tag{1}$$

where

K – number of time periods
t – time period index
p – option index
s – scenario index
x_p – number of boxes (switches, routers, etc.) ordered right now (t=0) using the option p
y_{tp}^s – number of boxes ordered using the option p at time t under scenario s
$\vec{z}_{ts}$ – recourse variable, which represents deficiency
π_s – probability of scenario s to occur
$\vec{\omega}$ – deficiency penalty factor
$\vec{U}$ – initial total capacity

P_p – cost of option p
T_p – order/installation time of option p
$\vec{C}_p$ – capacity expansion provided by option p
$\vec{D}_{ts}$– demand at time period t under scenario s
γ – cash discount factor

Note that $\vec{z}_{ts}, \vec{\omega}, \vec{U}, \vec{C}_p, \vec{D}_{ts}$ are vector variables indexed by the capacity types.

The constraints ensure that the total equipment capacity will be no less than the future demand $\vec{D}_{ts}$ minus allowable deficiency $\vec{z}_{ts}$ at each time period t and under any scenario s. The total capacity includes the initial capacity $\vec{U}$, the capacity provided by the boxes acquired at time period zero ($\sum_{T_p \le t} \sum_p \vec{C}_p x_p$), and the capacity brought by the future time investments under scenario s ($\sum_{\substack{q=1 \\ q+T_p \le t}}^{t} \sum_p \vec{C}_p\, y_{qp}^s$).

The objective is to minimize the total expected cost which consists of three terms. The first term is the investment cost associated with the decision made at time period zero. The second term is the expected cost of future investments depended on scenarios. The last term is the expected penalty cost of allowing deficiencies.

3.1 Extension of the model

In the real network context the structure of a switch/router is quite complicated. For example, a typical access switch or a router has three level structure – box (switch or router chassis), shelf, and card. Each box is a chassis and can be initially bought with certain pre-installed shelves and cards. A box holds several shelves; a shelf holds several cards; and a card has several ports on it. Each port can be connected to a port on another box. Whenever ports on all of the cards are exhausted, a new card must be bought and installed into one of the shelves. If there exists no shelf that can accommodate the card, a new shelf needs to be mounted into the box. If the box can not carry additional shelves, then a new box needs to be installed. The model can be extended as follows to accurately reflect this situation.

$$\min \sum_{p=p_c,p_s,\text{ or } p_b} \gamma^{-T_p} P_p\, x_p + \sum_s \pi_s \sum_{t=1}^{K} \sum_{p=p_c,p_s,\text{ or } p_b} \gamma^{-(t+T_p)} P_p\, y_{tp}^s + \sum_s \pi_s \sum_{t=0}^{K} \omega \cdot \vec{z}_{ts}$$

such that

$$U + \sum_{T_p \le t} \sum_{p=p_c} C_p\, x_p + \sum_{\substack{q=1 \\ q+T_p \le t}}^{t} \sum_{p=p_c} C_p\, y_{qp}^s + z_{ts} \ge D_{ts} \quad \forall\, s,t \tag{2}$$

$$U_{p_s} + \sum_{T_{p_s} \le t} C_{p_s}\, x_{p_s} + \sum_{\substack{q=1 \\ q+T_{p_s} \le t}}^{t} C_{p_s}\, y_{qp_s}^s \ge \sum_{\substack{p_c \in p_s \\ T_{p_c} \le t}} x_{p_c} + \sum_{\substack{p=p_c \\ p_c \in p_s}} \sum_{\substack{q=1 \\ q+T_p \le t}}^{t} y_{qp}^s \quad \forall\, s,t,p_s \tag{3}$$

$$U_{p_b} + \sum_{T_{p_b} \le t} C_{p_b}\, x_{p_b} + \sum_{\substack{q=1 \\ q+T_{p_b} \le t}}^{t} C_{p_b}\, y_{qp_b}^s \ge \sum_{\substack{p_s \in p_b \\ T_{p_s} \le t}} x_{p_s} + \sum_{\substack{p=p_s \\ p_s \in p_b}} \sum_{\substack{q=1 \\ q+T_p \le t}}^{t} y_{qp}^s \quad \forall\, s,t,p_b \tag{4}$$

$$U_{p_b}^B + \sum_{T_{p_b} \le t} B_{p_b} x_{p_b} + \sum_{\substack{q=1 \\ q+T_{p_b} \le t}}^{t} B_{p_b} y_{qp_b}^s \ge \sum_{\substack{p_s \in p_b \\ p_c \in p_s \\ T_{p_s} \le t}} B_{p_c} x_{p_c} + \sum_{\substack{q=1 \\ p_s \in p_b, p_c \in p_s \\ q+T_p \le t}} B_{p_c} y_{qp_c}^s \quad \forall\, s,t,p_b \tag{5}$$

$z_{ts} \ge 0$,

x_p and y_{tp}^s are nonnegative integers.

where

K	– number of time periods
t	– time period index
$p_{(\cdot)}$	– option index for cards p_c, shelves p_s, and boxes p_b
s	– scenario index
π_s	– probability of scenario s.
x_p	– number of units ordered right now (t=0) using the option p
y_{tp}^s	– number of units ordered using the

		option p at time t under scenario s
z_{ts}	–	recourse variable, which represents deficiency
ω	–	deficiency penalty factor
U	–	initial total capacity in terms of ports
U_{p_s}, U_{p_b}	–	initial extra capacity, number of cards a certain kind of shelf can carry (U_{p_s}) and number of shelves a certain kind box can accommodate (U_{p_b})
$U^B_{p_b}$	–	extra bandwidth per box
P_p	–	cost of option p
T_p	–	order/installation time of option p
C_p	–	capacity expansion provided by option p
D_{ts}	–	demand at time period t under scenario s
γ	–	cash discount factor
B_{p_b}	–	bandwidth capacity of a box (p_b)
B_{p_c}	–	bandwidth a card (p_c) can consume, equals to the average bandwidth per port times the number of ports on the card

The main difference to the original model (1) is that the option p is split into three categories, cards p_c, shelves p_s, and boxes p_b. Two families of constraints (3) and (4) are also added to describe the relation between the newly added cards, shelves, and boxes. The constraints (2) ensure that the capacity provided by the cards should be no smaller than the demand minus allowable deficiency. The new family of constraints (3) says that the shelves capacity has to be enough to accommodate the cards acquired so far, and constraints (4) ensure that the boxes can carry all the shelves. Finally constraints (5) specify that boxes should have enough bandwidth to support all the ports on cards.

4 Implementation

4.1 Scenario Construction

It is generally hard to predict the future. It is even harder without solid analytical data to start with. This subsection explains how the historical data is analyzed and the future scenario tree is generated based on the historical growth rate.

The history demand data is first collected and then analyzed to estimate the distribution of the growth rate. For example, using a particular history data set, it can be found that the monthly growth rate always lies in the range $[1\%, 10\%]$. Assume now that each node of the scenario tree has three branches. For example, let us restrict the analysis to the two-stage case, when every branch represents a single scenario. The growth rate range can then be subdivided into three equal intervals, and the number of times the data falls into one of these intervals (i.e., the particular scenario is realized) can be computed. In an example with 100 historical records, scenario 1 (growth rate in the range of $[1\%, 4\%)$) happens 25 times, scenario 2 (growth rate in the range of $[4\%, 7\%)$) happens 45 times, and scenario 3 (growth rate in the range of $[7\%, 10\%]$) happens 30 times. These counts are used as estimates of the probability each given scenario is likely to occur. To generate the scenario tree, an average of the growth rate in the range is taken as the growth rate for the corresponding scenario. Hence, the future scenario tree has three branches: scenario 1 represents a growth rate of 2.5% and occurs with 0.25 probability, scenario 2 represents a growth rate of 5.5% and occurs with 0.45 probability, and scenario 3 represents a growth rate of 8.5% and occurs with a probability of 0.30.

4.2 SMPS File Generation

The problem is solved using a stochastic programming (SP) solver. The standard input format for SP solvers is the stochastic MPS (SMPS) format. The format specifications are outlined in [2, 7]. A tool is created to generate the SMPS format for the general model (1) as well as for the more detailed model discussed in Section 3.1. The tool is coded in C++ and does not rely on a modeling language like GAMS or AMPL. The design of the tool leverages the object-oriented concepts providing several model-independent classes that generate the output SMPS file and a model-dependent external interface that supplies details of a particular model to the model-independent classes.

5 Results

The SMPS file created by the tool described in Section 4.2 is supplied as an input to the FortSP solver on the NEOS server [3, 6]. The results are very satisfactory.

5.1 Examples Run By NEOS

A specific example is used to illustrate the application of the model (2-5). The list of equipment options is given in Table 1.

Equipment Option ($p_{(\cdot)}$)	Price P_p	Installation time T_p	Capacity C_p
Box1	253	1	23
Shelf1	52	2	42
Shelf2	42	1	52
Card1	1.1	0	11
Card2	2.1	1	21

Table 1: Equipment options.

Assume that the future demand is described by a scenario tree with three branches on each node as listed in Table 2. The initial demand $D_{0\cdot}$ is 1000. The initial capacity U is the same as the initial demand $D_{0\cdot}$, which means there are no extra ports available on the boxes.

Scenarios	Growth Rate (Demand)	Probability
Scenario 1	-0.2% (9998)	0.09
Scenario 2	1% (1010)	0.42
Scenario 3	2.4% (1024)	0.49

Table 2: Scenarios.

For simplicity, the number of time periods is fixed to $K = 2$. The deficiency penalty factor ω is 1. The initial extra capacities for both shelves and boxes are $U_{p_s} = 12$, $U_{p_b} = 13$, respectively. The initial extra bandwidth of a box $U^B_{p_b}$ is 503. The bandwidth capacity of a box B_{p_b} is 1500 and the bandwidth consumed by a card B_{p_c} is 5 for Card1 and 10 for Card2. Since $U^B_{p_b}$ is large compare to B_{p_c} (5 and 10), it can be seen that the bandwidth constraints are not active for this example, which means that the box has enough capacity to support the bandwidth growth attributed to adding of new cards.

Based on the above parameters, the tool generates the desired SMPS file. The SMPS file is then uploaded online using NEOS web service which redirects the problem to the chosen solver and prints out the solution.

The solution to the above example suggests to refrain from buying anything at the present time (stage one). This means that one can wait till the second stage and then re-run the optimization. If Scenario 1 realizes then the solution suggests buying nothing, therefore the cost of this outcome is 0. If Scenario 2 realizes one unit of Card1 has to be bought. And the cost of this outcome equals to price of the card, which is 1.1. Under Scenario 3 a higher demand is faced and three units of Card1 are needed leading to the cost of 3.3. The expected cost at the second stage is thus $0 \times 0.09 + 1.1 \times 0.42 + 3.3 \times 0.49 = 2.079$. Since the Card1's installation time is 0, it explains why it is possible to wait till the next stage. There is no need in planning ahead since a desired capacity level can be achieved instantaneously.

Suppose that the installation time of the Card1 is changed to 1, and that the program is re-run. The optimum strategy changes to ordering one unit of Card1 and one unit of Card2 at the stage one and no activities on the second stage. Since under this strategy there is no deficiency at the second stage the total cost is just the price of new cards, which is 3.2. To illustrate the optimality of this solution, consider another strategy where only one unit of Card2 is ordered. The cost at the first stage is 2.1, but at the second stage, if Scenario 3 occurs, there will be a deficiency of 3 ($1024 - 1000 - 21 = 3$). The penalty factor ω for deficiency is 1. So the cost of deficiency is 3 if the scenario 3 happens. The expected deficiency cost is $3 \times 0.49 = 1.47$. Therefore, the total cost is $2.1 + 1.47 = 3.57$, which is larger than 3.2 – the optimal cost.

The two examples above illustrate how the stochastic programming-based model can be used to make intelligent decisions based on forecasts of future demand.

6 Option Pricing

The decision-maker is often faced with the question of how much of a premium is worth paying for shorter installation times. This important practical question can be answered by changing the parameters of the above model.

The decision-maker can reduce the total cost by adding new options with a smaller installation time (and normally higher cost). The change in the total cost is caused by the decrease of the penalty due to less deficiency, and partially offset by the increase of the cost of the option. Hence, by changing the associated cost for the option so as to obtain a total cost that is the same as the original one, the premium people are willing to pay for a shorter installation time can be obtained. The detailed steps are as follows:

1. Start with a small set of options describing the longest desirable installation times. Solve the model with a given set of options obtaining total cost.

2. Add one or more new options with a smaller installation time than the existing options have, and higher associated cost.

3. Solve the model with the new set of options, obtaining an updated total cost.

4. By changing the associated cost for the added option, the ranges can be observed for which the new total cost is still smaller than the older one. In this way, the range of associated costs for the new added option without increasing the total cost is obtained. This range of associated cost minus the original cost is the premium we are willing to pay for the shorter installation time.

7 Conclusion

A stochastic programming model is created to handle a complex investment decision problem under uncertainty. Implementation issues are discussed. Two examples illustrate how the stochastic programming model is able to make intelligent decisions based on the predicted future demand. The model can also be used to answer the question of how much premium a decision maker should be willing to pay for a shorter installation time.

8 Acknowledgments

This work is funded by Bell Canada through Bell University Laboratories (BUL).

References

[1] S. Ahmed, A.J. King, and G. Parija. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *J. Global Optim.*, 26(1):3–24, 2003. Supply chain optimization.

[2] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer Verlag, New York, 1997.

[3] J. Czyzyk, M. Mesnier, and J. Moré. The NEOS Server. *IEEE J. on Comp. Sc. and Engineering*, 5, 1998.

[4] T. Dangl. Investment and capacity choice under uncertain demand. *European J. Oper. Res.*, 117:415–428, 1999.

[5] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Managing capacity for telecommunications networks under uncertainty. *IEEE/ACM Trans. Netw.*, 10(4):579–587, 2002.

[6] M.C. Ferris, M.P. Mesnier, and J.J. Moré. NEOS and Condor: Solving optimization problems over the Internet. *ACM Trans. on Math. Software*, 26(1):1–18, 2000.

[7] H.I. Gassmann and E. Schweitzer. A comprehensive input format for stochastic linear programs. *Ann. Oper. Res.*, 104:89–125, 2001.

[8] M. Laguna. Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. *Management Science*, 44(11):S101–S110, 1998.

[9] J. Linderoth and S.J. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Comput. Optim. Appl.*, 24(2-3):207–250, 2003. Stochastic programming.

[10] J. Linderoth and S.J. Wright. Computational grids for stochastic programming. In *Applications of stochastic programming*, volume 5 of *MPS/SIAM Ser. Optim.*, pages 61–77. SIAM, Philadelphia, PA, 2005.