Optimization and Engineering, 3, 109–127, 2002

An Interactive Goal Programming Procedure for Operational Recovery Problems

BOAZ GOLANY
Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Haifa 32000, Israel

YUSEN XIA
Department of Management Science and Information Systems, The University of Texas at Austin, Austin TX 78712, USA

JIAN YANG
Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology, Newark NJ 07012, USA

GANG YU
Department of Management Science and Information Systems, The University of Texas at Austin, Austin TX 78712, USA

Abstract. We propose an interactive Goal Programming (GP) for operational recovery problems that are present in diverse areas of application. After defining the problem we discuss its relevance to scenarios taken from airline scheduling, supply chain management and call centers operations. We then construct the GP procedure for operational recovery decision making and illustrate the mechanics of the proposed procedure through two examples: an abstract example based on a minimum spanning tree problem and a more practical example of a production-inventory problem.

Keywords: irregular operations, operational recovery, interactive goal programming, minimum spanning tree, economic production quantity

1. Introduction

This paper investigates situations in which there exists a prior operational plan (either optimal or sub-optimal) that is subjected to a stochastic stream of disruptions that affect its execution. The common terminology refers to systems under normal (undisrupted) conditions as "regular operations" while systems that are disrupted are called "irregular operations". Examples of such settings are now prevalent in transportation, manufacturing, military and other areas.

We develop an *operational recovery* procedure whose primary objective is to overcome the disruption while staying as close as possible to the original plan. This is in contrast to *adaptive* models that are designed for situations in which there does not exist a prior operational plan at the beginning of the planning horizon. Rather, the plan is being gradually constructed in reaction to external stimuli. For example, a "pull" production system that gradually adjusts its production rate as customers issue new orders. Thus, in adaptive systems, external interference are welcome since they actually serve as the building blocks for the operational plan. In the operational recovery settings, however, the ideal situation is to have no interferences at all.

Recent economical and technological developments have increased the motivation to provide robust and efficient techniques to handle irregular operations. Several factors contributed to this phenomenon. First, the nature of the marketplace has changed drastically as customers are becoming more and more sophisticated and are demanding more and more services and goods to be available immediately when they are needed. At the same time, technological breakthroughs are driving the life-cycle of many products downwards and timely decisions regarding their entrance and exit from the market are becoming essential. This is coupled with increased global competition that forces each of the competing agents in the market to react swiftly to market signals so as not to be left behind. Second, the advent of modern technologies, mainly in computer and communication systems, now enables real-time data collection, communication, and information processing at affordable prices. Third, methodological advancements in Operations Research (OR) enable the solution (in acceptable time and quality) of large-scale practical optimization problems that were considered intractable until recently. Pioneering applications in airlines operations recovery (see, e.g., Yu (1998)) have paved the way to in-depth investigation on how OR methodologies can be employed in a variety of irregular operations settings. Finally, as the size and proliferation of these systems has grown to its current levels, so have the economic consequences of their disruption. Untimely (or inappropriate) response to a disturbance may lead to propagation effects that may cause a complete break-down of the system. A major disruption to such a system (an airline schedule or a large production facility) can easily cost millions of dollars for every hour of idle time.

The purpose of this paper is twofold. The first objective is to develop a procedure for operations recovery decision making that will be useful to a diverse range of systems. The objective function of the plan recovery model is typically composed of several objectives that must be carefully balanced. As the relative importance of these objectives may change according to the specific circumstances in which the problem was detected, we propose a lexicographical scheme that allows the decision maker to interactively express preferences and select solutions accordingly.

The second objective of the paper is to demonstrate the execution of the proposed procedure through coherent, yet small-scale problems that will serve to illustrate the potential utility of the approach. We have selected two well-known problems as a vehicle for that illustration: the minimum spanning tree (MST) and the Economic Production Quantity (EPQ). The MST problem plays a pivotal role in many network design models in diverse areas such as logistics, communications and construction projects. Therefore, in addition to the MST value as the demonstration tool for our procedure, the results we prove for this problem may have immediate utility in some network planning circumstances. The EPQ is a basic problem in operational management. It has been extensively researched in the last few decades but rarely in the context of irregular operations. Recently, some interesting variations of the EPQ problem are found in the context of supply chain management models.

To set the stage for the proposed framework we provide the following definition. Given an operational plan (assume optimized or generated with the best effort during planning stage), *operational recovery* is the process of making an optimal (or near optimal) decision on revised execution of the plan and/or recovery of the plan.

The change to the plan might originate from various sources:

- Changes in system environment: e.g., inclement weather,
- Uncontrollable events: e.g., power failures, traffic accidents, strikes,
- Changes in system parameters: e.g., market price change for materials and/or for products, delivery time change from vendors,
- Changes in availability of resources: e.g., machine breakdowns, unavailability of workforce, supply shortages,
- New external restrictions: e.g., call-back of equipment for maintenance required by the vendor or a government agency, new union contracts,
- Uncertainties in system performance: e.g., unreliable estimates on various phases of a project,
- New considerations: e.g., new orders, changed customer priorities.

In general, we denote a change in the plan as "disruption" regardless of its cause and nature. We further clarify the characteristics of the problem through three concrete examples. For each example we list the essential components of the decision making procedure (the decision variables, the data parameters, the constraints and the objective for both the original problem and the operational recovery problem).

Airline irregular operations control:

- *The decision variables*: aircraft routes, crew pairings, maintenance schedules and workforce assignments, gate assignments, baggage routes, etc.
- *The parameters*: minimum connection time at various stations, crew required rest times, published flight schedules, maintenance base capacity, available workforce, gate availability, etc.
- *The constraints*: flight coverage, aircraft balance, station curfews, crew legality, maintenance deadlines, etc.
- *The original objective function*: minimizing total operational cost for servicing all offered flights.
- *The disruptions*:
  - Change in system environment: inclement weather that causes hub closures or reduction of landing and take-off events.
  - Uncontrollable events: union strikes that cause unavailability of crew.
  - Change in system parameters: maintenance that takes much longer than estimated.
  - Change in resource availability: aircraft grounding due to mechanical problems, crew sickness.
  - New external restrictions: aircraft call-back maintenance by FAA
  - Uncertainties in system performance: passenger or baggage loading/unloading time.
  - New considerations: newly added charter flights.

- *The schedule recovery objectives*: Minimize the total disruption impact including (1) customer ill-will; (2) lost revenue; (3) missed on-time performance; (4) increased maintenance cost due to overtime penalty, increased frequency etc.; (5) increase crew cost due to layover, deadheading, away from home, use of reserves, etc.; (6) increase baggage handling cost; (7) increased aircraft cost due to diversion, use of spares, etc.; and (8) operational cost for plan change.

Supply chain operations control:

- *The decision variables*: production plan; inventory replenishment policy; distribution channels, routes, and schedules; processing sequences, vendor selections, etc.
- *The parameters*: machine processing times, worker daily hours and maximum overtime hours, promised delivery dates, inventory obsolesence rate, transportation speed, etc.
- *The constraints*: meeting demands, warehouse capacity limitations, delivery deadlines, delivery vehicle availability, etc.
- *The objective function*: maximizing total profit.
- *The disruptions*:
  - Change in system environment: delivery routes or traffic conditions change.
  - Uncontrollable events: power failures.
  - Change in system parameters: production and inventory capacity.
  - Change in availability of resources: assembly line or machine breakdowns.
  - New external restrictions: government regulations.
  - Uncertainties in system performance: processing time of various parts by different workers.
  - New considerations: new customer orders.
- The operations recovery objectives are: Minimize the total disruption impact including (1) customer ill-will; (2) lost profit; (3) missed deliveries; (4) increased inventory cost; (5) increased over-time cost; (6) increased delivery cost; (7) increased production cost; and (8) operational cost for plan change.

Call centers control:

- *The decision variables*: routing of calls; maintenance schedule of servers and routers;
- *The parameters*: lines capacity, operators daily hours and maximum overtime hours, average service times, etc.
- *The constraints*: capacity limitations, meeting demand on time, etc.
- *The objective function*: minimizing cost.
- *The disruptions*:
  - Change in system environment: unpredicted burst in demand.
  - Uncontrollable events: system breakdown, power failures, system hit by virus or hackers.
  - Change in system parameters: longer than expected service maintenance.
  - Change in availability of resources: server breakdowns, operators sick.

- New external restrictions: new internal company guidelines on service quality.
- Uncertainties in system performance: variability in service time of various calls by different operators.
- New considerations: new services, new customers.

- The operations recovery objectives are: Minimize the total disruption impact including (1) customer ill-will; (2) lost calls; (3) changes to the original shift assignments; (4) over-time cost.

The paper is organized as follows. In Section 2 we provide a brief literature review focusing mainly on relevant GP developments. The review also discusses the limited literature that has appeared till now on the topic of recovery from irregular operations. In Section 3 we propose to use goal programming models as the platform for implementing operational recovery decision-making. In Section 4 we demonstrate the mechanisms of the proposed procedure through the minimum spanning tree problem and provide some analytical results that relate the complexity of the operational recovery problem with that of the original problem. Section 5 presents an EPQ scenario with some irregular operations and illustrate the use of our GP procedure to solve it. Section 6 concludes the paper by summing the main outcomes of this research and pointing out directions for further research.

2. Literature review

In this section we review some of the literature that we find relevant to the problem of recovery from irregular operations. We focus on interactive GP techniques which, as we discuss later, are most relevant to our developments. The GP methodology is a mature area and its literature contains literally thousands of publications concerning both theoretical and applied work. Within this vast field, the area of interactive GP is rather new and active (see the recent review by Sang and Olson (1999)). On the other hand, the area of recovery from irregular operations is considered a rather new research topic and specific literature on it is somewhat sketchy.

Goal programming has earned a reputation as one of the most practical multicriteria methods mainly due to its modeling flexibility and its capability to handle large-scale real problems. Although nonlinear GP and integer GP are sometimes used to handle specific problems, linear GP methods such as the weighted GP and the lexicographic GP are the ones that are most common in practice. In spite of its many advantages, GP has a number of weaknesses that must be addressed carefully before it is implemented (see Romero (1991)). One of the critical issues that Romero mentions is the difficulty in setting goals and in selecting weights for deviational variables. Thus, traditional GP models were sometimes considered inferior to other MCDM models as they provided a single solution that relied on questionable target setting and weight selection decisions.

To overcome the difficulties associated with the inherent uncertainty that characterizes human decision making processes, researchers have tried to integrate various interactive methods with GP (see the survey by Shin and Ravindran (1991)). The basic idea is to adjust targets for objectives or their associated preferences in light of initial model results.

Both weighted and lexicographical GP models were adapted for interactive use. Starting with weighted GP models, we note the work of Kalu (1999), who presented an interactive GP algorithm to adequately take into account the overall welfare of some systems. The algorithm combines features of GP with those of interactive methods and an economic efficiency index. Korhonen and Wallenius (1992) proposed a decision support system for multiple objective linear programming in which a visual interactive GP is used to facilitate a solution. Goedhart and Sproonk (1991) presented an interactive GP for a financial planning problem in which multiple goals are being considered. From the lexicographical GP models, we note the pioneering work by Franz and Lee (1981). In their work, each objective level corresponded to a single target. Mote et al. (1988) showed how multiple targets can be placed in the same priority level (as we do here). The interaction with the human decision maker in all these cases can be affected after each time a solution is generated through changes in the goal targets, revising preference weights etc. This process is repeated until a satisfactory solution is achieved.

Relatively little has been published until now on the problem of recovery from irregular operations. Most of these publications were recorded in the airline industry. Thengvall et al. (2000) provided airline schedule recovery from irregular operations. They formulated a network model with side constraints and an objective function that minimizes deviations from the original aircraft routings. Arguello et al. (1997) presented greedy randomized adaptive search procedure (GRASP) to reconstruct aircraft routings in response to groundings and delays experienced over the course of the day. Wei et al. (1997) presented a system-wide multi-commodity integer network flow model and a heuristic search algorithm to recover from irregular operations of airline crew management. Mathaisel (1996) designed a decision support system which integrated real-time flight following, aircraft routing, maintenance, crew management, gate assignment and flight planning with dynamic aircraft re-scheduling and fleet re-routing algorithms for irregular operations.

We summarize below some of the main concepts elicited from the literature. These concepts are integrated in the general framework that we present in the next section.

- *Partial solution*: If a solution can not be achieved in reasonable time one is satisfied with a partial solution in order to prevent undesirable (sometimes catastrophic) effects. Also, when a feasible recovery solution does not exist, a near feasible solution is desirable to ensure the continued execution of the plan. The minor infeasibility is to be fixed in the future.
- *Computer-human interaction*: The problems under investigation involve qualitative dimensions that are difficult to quantify and formulate. Therefore, a desirable characteristic of a solution procedure is to facilitate smooth interaction between a human decision-maker and a computerized system in ways that will best utilize the advantages that each of them possess.
- *Learning*: There is a need to equip the solution procedure with some learning capabilities since the complexity of the operational systems that are involved is such that many of the potential disturbances and the ways to tackle them are not known in advance. Time and energy can be saved in handling future disturbances by relying on information that was learnt and stored by the system.

- *Satisficing*[1]: In most cases, operational recovery procedures do not try to obtain a global optimum. Rather, they are ready to accept reasonably good solutions as long as they meet the time constraints.
- *Flexibility*: As the underline real-world environments are constantly undergoing changes, the solution procedure that are applied to them must exhibit flexibility (e.g., readiness to discard data parameters that are no longer valid and replace them with new information that was observed more recently).

3. Goal programming procedure

The implementation of the general decision making framework that is proposed in the previous section can be done in many ways depending on the environment that is modeled and the computational capabilities that are available to the decision maker. In this section we propose to base the implementation on the structure and concepts of goal programming (GP) models. We find GP to be a natural vehicle for this implementation because it offers the flexibility of defining the original plan as goals (where minimization of deviations from these goals will replace the original objective) and allowing partial solutions by enabling violations from the original constraints.

3.1. Original problem

To make our model developments and their subsequent discussions more concrete we shall describe the goal programming-based procedure with respect to linear programming (LP) formulation. Hence, we assume that the original problem is given by the LP model:

$$\begin{aligned} &\min c^T \cdot x \\ &A \cdot x = b \\ &\quad x \geq 0 \end{aligned} \tag{1}$$

where A, b, c are data parameters (A is a $m \cdot n$ matrix, b is a $m \cdot 1$ vector and c is a $1 \cdot n$ vector) and x is a vector of decision variables. Solving (1) yields the optimal vector x^*.

Using known LP results one can perform *sensitivity analysis* on the original problem to find out the ranges of permissible changes in the values of the parameters A, b, c. These ranges may later be instrumental in determining when to launch operational recovery procedures. Changes in our LP context will present themselves as new values to the data parameters (to be denoted below as $\hat{A}, \hat{b}, \hat{c}$). Thus, we may define any change in one or more data parameters that is larger than its relevant permissible range as a disruption (that requires some corrective action) while changes that are smaller than their respective ranges will be ignored. Disruptions can be further categorized as either major (requiring a return to the original problem and solving it again) or minor (requiring just the operational recovery procedure). This categorization requires us to define a threshold that will distinguish between minor and major disruptions. An example of such threshold could be a certain number of violated sensitivity ranges.

Additionally, running *robustness analysis* on the original problem may reveal trade-offs between the quality of the solution and its robustness. That is, we may prefer to replace the optimal solution with α-optimal solutions (solutions that are no larger than α times the original solution) for which we obtain larger ranges of insensitivity of the solution to changes in the parameter values. Again, we will need to define a threshold for the robustness analysis. For example, this threshold may be defined in terms of an aggregate measure of sensitivity that combines the effects of A, b, c. The robustness analysis will then be executed by running successive LP runs with a sequence of increasing α values until we meet the robustness threshold.

3.2. *Goal programming operational recovery*

Several alternative GP formulations are possible. We present these alternatives in an increasing order of complexity, starting with a very basic model and adding to it additional features to accommodate the requirements outlined in the previous section.

3.2.1. Complete solutions. Assuming that complete solutions are attainable we formulate a weighted sum GP model whose objective function focuses on finding new values for the x vector that will be as close as possible (according to given weights on deviations) to the optimal vector x^* that was found by the original LP model:

$$\begin{aligned} \min\{W^+ \cdot \alpha^+ + W^- \cdot \alpha^-\} \\ \hat{A} \cdot x = \hat{b} \\ x + \alpha^- - \alpha^+ = x^* \\ x \leq 0 \end{aligned} \tag{2}$$

Model (2) is totally dedicated to the objective of staying as close as possible to the original plan. In doing so it ignores the original objective. However, model (2) might have alternative optimal solutions in which case we will definitely prefer those that come closer to our original objective. To do so we employ lexicographic GP structure with two priority levels. The objective of model (2) will be at the first (and highest) priority level while the second priority level will drive the model to stay close to the original optimal value that was achieved in model (1). This is accomplished via:

$$\begin{aligned} \mathit{Min\ Lex} \quad P_1 : \{W^+ \cdot \alpha^+ + W^- \cdot \alpha^-\} \quad P_2\ \{\beta^+\} \\ \hat{A} \cdot x = \hat{b} \\ x + \alpha^- - \alpha^+ = x^* \\ \hat{c}^T \cdot x + \beta^- - \beta^+ = c^T \cdot x^* \\ x \geq 0 \end{aligned} \tag{3}$$

3.2.2. Partial solutions. The two GP models shown above assumed that a new vector x that meets the original constraints with the new values of the parameters may be found. This may not always be the case. In order to allow for partial solutions we extend the lexicographic formulation above by minimizing at the highest priority deviations from feasibility with respect to the original constraints.

$$\begin{aligned}
&\textit{Min Lex} \quad P_1 : \{\gamma^+ + \gamma^-\} \quad P_2 : \{W^+ \cdot \alpha^+ + W^- \cdot \alpha^-\} \quad P_3 : \{\beta^+\} \\
&\hat{A} \cdot x + \gamma^- - \gamma^+ = \hat{b} \\
&\quad x + \alpha^- - \alpha^+ = x^* \\
&\hat{c}^T \cdot x + \beta^- - \beta^+ = c^T \cdot x^* \\
&\qquad\qquad x \geq 0
\end{aligned} \tag{4}$$

3.2.3. Multiple solutions. Multiple solutions are preferred to single solutions. So, if time permits it, the procedure should try to generate several solutions. This can easily be done in our case by devising a certain pattern of changing the values of the weights (W^-, W^+) and then rerunning the appropriate GP model to generate alternative solutions.

4. Operational recovery for a spanning tree

In this section, we discuss a special case, the recovery of the spanning tree in the face of structural changes of a graph. Spanning trees and in particular minimum spanning trees (MST) are very useful tools in modeling real-life network-related applications. Therefore, studying the recovery of spanning trees may help us devise better reaction guides to changes in the real network. In the recovery model, what is to be optimized is not the cost of the finished spanning tree of the new network but the additional cost of adapting the original MST to a new spanning tree that is suitable for the new network.

To gain better insights on the MST recovery problem, let us consider the following example. Suppose an optical cable firm is laying optical cables for a communications network. The underlying network is denoted by $G \equiv (N, E)$, where N and E are the node and edge sets of the network, respectively. For instance, N may be {1, 2, 3, 4, 5} and E may be {12, 13, 14, 15, 23, 24, 25, 34, 35, 45}. For any given edge $e \in E$, the firm has to consider the following costs:

- Material cost for the cable covering e before it is laid into the ground: $C_M(e)$.
- Salvage value for the cable covering e once it is dug out of the ground: $V_s(e)$.
- Cost of laying into the ground the cable covering e: $C_L(e)$.
- Cost of digging out of the ground the cable covering e: $C_D(e)$.
- Implicit cost for adding the cable covering e when it was not planned to be laid out: $C_A(e)$.
- Implicit cost for removing the cable covering e when it was already laid out: $C_R(e)$.

To build a network that connects a set of nodes with the least cost, the firm should plan the network as a MST with each edge e costing

$$C(e) \equiv C_M(e) + C_L(e).$$

Let $x(e)$ be a binary variable indicating whether or not edge e is included in the original MST. Hence, the original MST problem can be formulated as:

$$
\begin{aligned}
\text{(MST)} \quad & \min \sum_{e\in E} C(e)x(e) \\
& \sum_{e\in E} x(e) = |N| - 1 \\
& \sum_{e\in K(I)} x(e) \geq 1 \quad \forall I \subset N \\
& x(e) \in \{0, 1\} \quad \forall e \in E,
\end{aligned}
\tag{5}
$$

where $| * |$ represents the cardinality of the set * and for any subset I of N, $K(I)$ stands for the subset of all edges in E with both end-points being in I. For the numerical example we have mentioned, suppose the $C(e)$'s form a vector {3.0, 6.0, 1.0, 10.0, 5.0, 8.0, 2.0, 7.0, 4.0, 9.0}, then the solution to the MST problem, described by its edges, is {12, 14, 25, 35}.

Now, suppose that after the firm has already done the physical laying work, it experiences some economic or topological changes that make the current network no longer feasible or create a potential to improve the network in terms of cost. Since the investment in the old network is already sunk (or committed), the firm should not try to build a new network that is optimal in cost for the new scenario. Rather, it should try to adapt the original network to a new feasible one that incurs the least adapting cost.

By using ∞-costing edges and using the underlying network which contains everything in both the original and the changed networks, we may model changes in both edge costs and in the network structure using only edge cost changes. In the following, any cost associated with the new network is denoted by the original notation with an additional. For every edge, we ascribe to it an edge-removal penalty

$$R(e) \equiv C_M(e) + C_L(e) - V_s'(e) + C_D'(e) + C_R'(e)$$

and an edge-addition penalty

$$A(e) \equiv C_A'(e).$$

Then, the cost of adding an additional edge e to the original spanning tree is

$$C^+(e) \equiv C'(e) + A(e)$$

instead of $C'(e)$ and the savings from removing one edge e from the original spanning tree is

$$-C^-(e) \equiv C'(e) - R(e)$$

instead of $C'(e)$.

We may use goal programming to formulate the problem of finding the new spanning tree that costs the least to be adapted from the original spanning tree (MAST—minimum adapted spanning tree). Let $y(e)$ be a binary variable defined as an indication of whether arc e is in the new spanning tree. The implementation of our goal programming procedure requires that $y(e)$ will be close to $x(e)$ and that deviations will be penalized. MAST is then:

$$\text{(MAST)} \quad \min \sum_{e \in E} [C^+(e)\delta^+(e) + C^-(e)\delta^-(e)]$$
$$\begin{aligned} y(e) + \delta^-(e) - \delta^+(e) &= x(e) \quad \forall_e \in E \\ \sum_{e \in E} y(e) &= |N| - 1 \\ \sum_{e \in K(I)} y(e) &\geq 1 \quad \forall I \subset N \\ y(e) &\in \{0, 1\} \quad \forall e \in E. \end{aligned} \tag{6}$$

The goal in model 6 stems from the fact that C_{ADP}, the cost of adapting from the original spanning tree indicated by $x(\cdot)$ to the new tree indicated by $y(\cdot)$, satisfies

$$\begin{aligned} C_{ADP} &= \sum_{e \in E} [C'(e)y(e) - C(e)x(e) + R(e)\delta^-(e) + A(e)\delta^+(e)] \\ &= \sum_{e \in E} [C'(e)(x(e) + \delta^+(e) - \delta^-(e)) - C(e)x(e) + R(e)\delta^-(e) + A(e)\delta^+(e)] \\ &= \sum_{e \in E} [(C'(e) + A(e))\delta^+(e) + (R(e) - C'(e))\delta^-(e) + (C'(e) - C(e))x(e)] \\ &= \sum_{e \in E} [C^+(e)\delta^+(e) + C^-(e)\delta^-(e)] + \sum_{e \in E} (C'(e) - C(e))x(e). \end{aligned}$$

Note that $\sum_{e \in E} (C'(e) - C(e))x(e)$ is independent of the decision on $y(\cdot)$.

In appearance, MAST is very different from MST. Actually, we can show that MAST is as difficult as MST. To show this, suppose the original spanning tree is T and the new tree after recovery is T'. Then, we also have

$$C_{ADP}(T', T) \equiv C'(T') - C(T) + \sum_{e \in E[T] \setminus E[T']} R(e) + \sum_{e \in E[T'] \setminus E[T]} A(e).$$

After a regrouping of terms, it follows that

$$\begin{aligned} C_{ADP}(T', T) = &\left[C'(T') - \sum_{e \in E[T'] \cap E[T]} R(e) + \sum_{e \in E[T'] \setminus ET} A(e) \right] \\ &+ \left[\sum_{e \in E[T]} (R(e) - A(e)) \right]. \end{aligned}$$

In the above formula, terms in the second bracket are independent of the decision on T' and terms in the first bracket sum up to be the total cost of T' under cost scheme $C_{-+}(\cdot)$ where

$$C_{-+}(e) = \begin{cases} -C^-(e) \equiv C'(e) - R(e) & \text{if } e \in E[T] \\ C^+(e) \equiv C'(e) + A(e) & \text{if } e \in E[G] \setminus E[T] \end{cases} \tag{7}$$

So, the T' that is optimal for MAST is an MST under a certain cost scheme. Hence, solving the general MAST is as difficult as solving the MST, which in turn can be solved in $O(|N|^2)$ time by Dijkstra's algorithm (Dijkstra, 1959) or Prim's algorithm (Prim, 1957), or in $O(|E|\log|E|)$ time by Kruskal's algorithm (Kruskal, 1956).

Now, returning to our numerical example, suppose after the laying of the original MST {12, 14, 25, 35} there is a topological change such that vertex 2 can no longer be used along with all edges connecting to it, and hence what remains of MST {14, 35}, is even no longer a feasible spanning tree for the remaining graph G'. Now, all the feasible spanning trees for G' are {13, 14, 15}, {15, 35, 45}, {14, 34, 45}, {13, 34, 35}, {13, 15, 45}, {15, 34, 45}, {13, 34, 45}, and {13, 15, 34}. The ones that are the closest to the remnant of MST are {14, 15, 35}, {13, 14, 35}, {14, 34, 35}, and {14, 35, 45}. The least costing such tree under the original cost is {13, 14, 35}. And, if $C^+(e)$'s for edges in G' form a vector {2.0, 2.0, 1.0, 0.5, 2.5, 2.0} and $C^-(e)$'s are the same for all edges, then the tree incurring the least change is actually {14, 34, 35}.

In a separate paper (Yang et al., 2000), we address three special cases where the original tree is already a MST. For these cases we develop special algorithms that tackle them individually. The first special case is when the cost of a single edge is changed. For this case, we find that it takes $O(|E|)$ time to find the MAST solution. The second special case is when multiple nodes or/and edges are removed. Suppose the spanning tree falls into K connected components after the node/edge removals. We find an $O(|E|+K^2)$ algorithm to obtain the MAST solution. The third case is when a single node along with some new edges incident to this new node are added. We find an $O(|N|^2)$ algorithm for MAST under this situation.

5. Operational recovery for an EPQ problem

5.1. Problem formulation

Consider an OEM (original equipment manufacturer) working with a finite production rate to satisfy constant continuous demand rate for his products.

Denote:

D = Annual demand rate
C = Unit production cost for the OEM
Q = Size of production run for the OEM
R = Production rate of the OEM per year
S = Setup cost per production run for the OEM
H = Holding cost of the OEM per unit per year.

It is well known that the optimal production and inventory schedule for the OEM is to adopt an economic production quantity (EPQ) policy (see figure 1).

Total annual cost = production cost + setup cost + holding cost,

$$TC = D \cdot C + \frac{D \cdot S}{Q} + Q \cdot \frac{(R-D)}{2 \cdot R} \cdot H$$

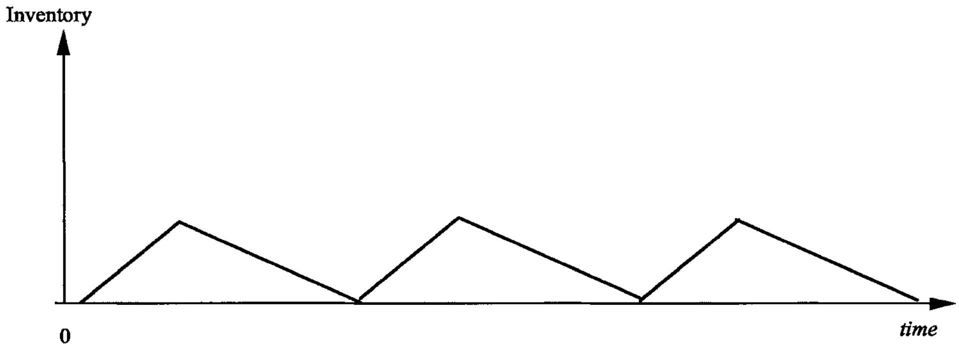

Figure 1. Economic production quantity model for the OEM without shortage.

Differentiating over Q and letting it equal to zero, we get the economic production quantity for the OEM as follows,

$$Q_r\sqrt{\frac{2SDR}{H(R-D)}}$$

The time length per cycle is $\frac{Q}{D}$ and the number of production runs within one year is $\frac{D}{Q}$. The OEM may face many possible disruptions, for example:

- Setup cost change
- Holding cost change
- Demand rate change
- Production rate change
- Production cost change
- Backlogging status change

We formulate the OEM disruption recovery problem as a goal programming problem for the case of increased demand over some disruption time.

Denote:

$\tilde{D}$ = Annual demand rate during the disruption time ($\tilde{D} > D$).
$\tilde{C}$ = Unit production cost when production occurs outside of its ordinary scheduled cycles.
$\tilde{S}$ = Setup cost per production run that is launched not according to its predefined schedule.

5.2. Solution space

When the OEM faces increased demand, he may choose one of the following recovery options:

- Extended production period: Extend the production period to produce more products at the same production rate as before.
- In advance production: Begin production before the original production schedule. Basically, the idea is to store some products to satisfy the extended demand.
- Outsourcing: Buy the extra products from an outside source (at a higher cost) instead of manufacturing them.

5.3. *Penalty function*

(a) Penalty for setups changes: Each time the OEM launches a production cycle (whether one that was planned or a new one) at a different time than originally planned, there is a larger than usual setup cost ($\tilde{S} > S$).
(b) Penalty for changes in the duration of production cycles: Each time the OEM extends a production cycle, he pays a larger than usual production cost per product during the extended production duration ($\tilde{C} > C$).

5.4. *A goal programming formulation*

We assume that the disruption lasts over some period t and the recovery time window is $T \geq t$. We formulate the OEM's disruption recovery problem as a three level goal programming model. The first level seeks to satisfy all the demand of the OEM's customers. The second level tries to keep the original production schedule (quantities and timing) as much as possible. The third level minimizes the cost of setup and inventory.

Let's assume that in the original schedule during T, the OEM was planning to run n production cycles. Assume that there are k cycles in the new schedule. Denote the time points for a setup in the original schedule as $t_1, \ldots, t_n$ and the time points for a setup in the new schedule as $t'_1, \ldots, t'_k$.

We formulate the following goal programming model to solve the OEM's problem,

$$
\begin{aligned}
&\textit{Min Lex}\\
&\quad P_1: \quad \{\gamma^- + \gamma^+\}\\
&\quad P_2: \quad \left\{\sum_{i=1}^{k}(\alpha_i^- + \alpha_i^+) + \sum_{i=1}^{k}(1 - \eta_i)\right\}\\
&\quad P_3: \quad \{TC'(T)\}\\
&\text{Subject to:}\\
&\quad \sum_{i=1}^{k} Q^i + \gamma^- - \gamma^+ = D \cdot T + (\tilde{D} - D) \cdot t\\
&\quad Q^i + \alpha_i^- - \alpha_i^+ = Q \quad \forall i = 1, \ldots, k
\end{aligned}
$$

$$\frac{1}{M} \cdot z_{ij} \leq (t_i' - t_j)^+ \leq M \cdot z_{ij}, \quad \forall i = 1, \ldots, k,\ j = 1, \ldots, n$$

$$\eta_i \leq n - \sum_{j=1}^{n} z_{ij} \leq M \cdot \eta_i, \quad \forall i = 1, \ldots, k$$

$$\sum_{i=1}^{k} Q^i \cdot C + \sum_{i=1}^{k} (Q^i - Q) \cdot (\tilde{C} - C) + k \cdot S + \sum_{i=1}^{k} (\tilde{S} - S) \cdot (1 - \eta_i) + \theta\gamma^-$$

$$+ \sum_{i=1}^{k} \frac{1}{2} H \left\{ \left(2I_i + \frac{Q^i}{R}(R - D_i)\right) \frac{Q^i}{R} + \left(I_i + I_{i+1} + \frac{Q^i}{R}(R - D_i)\right) \right.$$

$$\left. \times \left(t_{i+1}' - t_i' - \frac{Q^i}{R}\right) \right\} = TC'(T)$$

$$I_1 = 0$$

$$I_{i+1} = I_i + Q^i - (t_{i+1}' - t_i') \cdot D_i$$

$$Q^i, I_i \geq 0 \quad \forall i = 1, \ldots, k$$

$$\gamma^-, \gamma^+, \alpha_i^- \alpha_i^+ \geq 0, \quad \eta_i, z_{ij} \in \{0, 1\}$$

Q^i is the quantity produced by the OEM in cycle i for the new schedule. The first constraint attempts to satisfy all the demand. If there is unsatisfied demand, there is a penalty θ per unit unsatisfied which is included in $TC'(T)$ in the fifth constraint. The second constraint forces the program to stay close to the original plan from the quantities aspect. The idea in the third constraint is that if the time points of the new cycle coincide with time points of the original cycle, z_{ij} is forced to be 0, otherwise it is set to 1. The fourth constraint tries to force the program to stay close to the original plan from the timing aspect. Thus, $\eta_i = 1, i = 1, \ldots, k$ denotes that the setup time for cycle i in the new schedule is identical to one of the setup time points in the old schedule. In these cases, no penalty is incurred for setup change in the new schedule for cycle i. The fifth constraint defines the cost function which is minimized in the third objective level and the sixth and seventh constraints define boundary conditions on the inventory levels. The initial inventory is 0 in the original schedule and the end inventory equals initial inventory plus production minus demand.

Observe the fifth constraint that defines the total cost function of the OEM for the period T. The first two terms are the production cost and the penalty for production duration change, the second two terms are the setup cost and the penalty for setup time changes. The fifth term penalizes for unsatisfied demand and the seventh is the holding cost. For the holding cost, we didn't explicitly express the cycle in which part of the time, the demand rate is D and part of the time, the demand rate is $\tilde{D} = D + \delta$. The logic is the same except differentiation of small areas for calculating the whole area for those cycles (at most, there are two such cycles).

The third priority level in the above mathematical programming problem contains a nonlinear cost function. This required the application of the method suggested by Hwang and Masud (1979) to handle nonliniarities in GP.

5.5. *Numerical results*

Assume that the demand for the OEM products is 20,000 units per year, and there are 250 working days per year. The production rate is 100 units per day. The unit production cost is \$50.00, the holding cost is \$62.50 per unit per year, and the setup cost is \$200.00 per run.

$$Q = \left(\frac{2S\,DR}{H(R-D)}\right)^{\frac{1}{2}} = \left(\frac{2 \cdot 200 \cdot 20000 \cdot 25000}{62.5 \cdot (25000-20000)}\right)^{\frac{1}{2}} = 800.$$

The number of runs per year is $\frac{20000}{800} = 25$ and the OEM has a new setup for production every ten days. During the ten-day cycle, the total production time is eight days.

We assume that the demand rate for the OEM's products increases from $D = 20000$ to $D' = 20000 + \delta$ for a period with length = 10 working days (a cycle length); that the penalty for a setup time deviation from the original time point is half of the original setup cost = \$100.00; that the production cost is 10% higher than the original production cost for the extended production duration; that there is also a $10\% \cdot 50 = \$5.00$ fee associated with the reduced production duration. If there is unsatisfied demand, there is a penalty of \$100 per unit unsatisfied.

We fix the disruption recovery time window T for several cases and compare the relevant schedule of the OEM when δ changes from small to large values. We discuss four cases:

(1) $T = 30$ days, with one cycle before the demand rate increase and one cycle after the demand rate change.

Demand rate after disruption	Recovery solution	First level goal obj	Second level goal obj	Third level goal obj	Original cost during the period	Cost increase percentage
$D' = 22000$	A1	0.0	8	125532	121200	3.57%
$D' = 25000$	A1	0.0	20	132000	121200	8.91%
$D' = 30000$	A2	0.0	40	142800	121200	17.82%

Note: A1 = Keeps the original schedule for the first and third cycles and extends the production schedule of the second cycle.
A2 = Extends the production duration for the first two cycles and keeps the original third cycle schedule.

(2) $T = 20$ days, with one cycle before the demand rate change.

Demand rate after disruption	Recovery solution	First level goal obj	Second level goal obj	Third level goal obj	Original cost during the period	Cost increase percentage
$D' = 22000$	A3	0.0	8	85132	80800	5.36%
$D' = 25000$	A3	0.0	20	91600	80800	13.37%
$D' = 30000$	A4	0.0	40	102400	80800	26.73%

Note: A3 = Keeps the first cycle schedule and extends the production duration of the second cycle.
A4 = Extends the production duration for both cycles.

(3) $T = 20$ days, with one cycle after the demand rate change.

Demand rate after disruption	Recovery solution	First level goal obj	Second level goal obj	Third level goal obj	Original cost during the period	Cost increase percentage
$D' = 22000$	A5	0.0	80	85132	80800	5.36%
$D' = 25000$	A5	0.0	200	91600	80800	13.37%
$D' = 30000$	A5	200	200	111600	80800	38.12%

Note: A5 = Extends production duration of the first cycle and keeps the second cycle schedule as before.

(4) $T = 10$ days—the OEM must handle the disruption within its original production cycles.

Demand rate after disruption	Recovery solution	First level goal obj	Second level goal obj	Third level goal obj	Original cost during the period	Cost increase percentage
$D' = 22000$	A6	0.0	80	44732	40400	10.72%
$D' = 25000$	A6	0.0	200	51200	40400	26.73%
$D' = 30000$	A6	200	200	71200	40400	76.24%

Note: A6 = Extends the current production duration.

We list below some insights from the above numerical example:

1. The earlier the OEM knows about the disruption, the better. The longer the disruption recovery time window, the better, since the cost increasing percentage is a non-increasing function with respect to disruption recovery time window length.
2. Demand will always be met if the OEM is capable of doing so. This is because satisfying demand is our highest priority level in our goal programming procedure. Also, it could be seen that if there is unsatisfied demand, the cost increasing percentage is relatively large, one is 38.12% and the other is 76.24%.
3. The OEM may not want to affect other cycles schedule if it could handle the disruption within one setup cycle. This is due to the penalty for the number of setup changes and the penalty for the setup time point change.

6. Conclusions

This paper discusses the growing importance of recovery problems in various areas of application and provides detailed description of three areas that are now emerging as prominent candidates for recovery procedures for each of these areas we characterize the original problem, list its potential disruptions and state the possible objectives of the recovery problem. We then propose a general goal-programming formulation that serves as the core of the recovery engine. We illustrate the recovery decision process and the mechanics of the goal programming model through two well-known problems: the minimum spanning tree and the economic production quantity. In the first example, our illustration is a brief account of how to recover a spanning tree when the original network changes. For more detail, the reader is referred to Yang et al. (2000). The work in this area is preliminary and we present it here to show the relationship between an original optimization problem and its recovery problem version. In the second example we focus on the disruption case of increase in demand over a finite interval of time. Again, these are preliminary results which we intend to extend to more general supply chain scenarios.

The new topic of recovery problems has many directions for further research. One that is high on our agenda is to implement the proposed framework to a realworld cases. We anticipate that specific scenarios will require unique adaptations of the general procedure and the task of identifying such requirements and developing reasonable measures to test the validity of the adjusted procedures is certainly a challenge. Another direction is to explore the human interface with the decision support system (see, Trentesaux et al., 1998). This line of research may lead to a variety of interactive procedures where human intervention will be allowed (or required) at different levels of intensity.

Acknowledgments

Gang Yu's research was supported in part by the National Natural Science Foundation of China under grant number 79928001, and by the FADRC grant and a Special Research Grant from the University of Texas at Austin.

Note

1. This term, coined by H. Simon (1957), refers to satisfying (constraints) while sacrificing (objective function value).

References

M. F. Arguello, J. F. Bard, and G. Yu, "A GRASP for aircraft routing in response to groundings and delays," *Journal of Combinatorial Optimization* vol. 1, pp. 211–228, 1997.

E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik* vol. 1, pp. 269–271, 1959.

L. Franz and S. M. Lee, "A goal programming based interactive support system," in *Organizations: Multiple Agents with Multiple Criteria*, J. N. Morse, ed., Springer Lecture Notes on Economic Mathematical Systems, vol. 190, pp. 110–115, 1981.

M. Goedhart and J. Spronk, "Multifactor financial-planning—An outline and illustration," Lecture Notes in Economics and Mathematical Systems, vol. 351, pp. 176–199, 1991.

C. Hwang and A. Masud, *Multiple Objective Decision Making Methods and Applications*, Springer: Germany, 1979.

T. C. U. Kalu, "An algorithm for systems welfare interactive goal programming modeling," *European Journal of Operational Research* vol. 116, pp. 508–529, 1999.

P. Korhonen and J. Wallenius, "A computer graphics-based decision support system for multiple objective linear programming," *European Journal of Operational Research* vol. 60, pp. 280–286, 1992.

J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society* vol. 7, pp. 48–50, 1956.

D. F. X. Mathaisel, "Decision support for airline system operations control and irregular operations," *Computers & Operations Research* vol. 23, pp. 1083–1098, 1996.

J. Mote, M. Olson, and A. Venkataramanan, "A comparative multiobjective programming study," *Mathematical and Computer Modelling* vol. 10, pp. 719–729, 1988.

R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technological Journal* vol. 36, pp. 1389–1401, 1957.

C. Romero, *Handbook of Critical Issues in Goal Programming*, Pergamon Press: UK, 1991.

M. L. Sang and D. L. Olson, "Goal Programming," in *Multicriteria Decision Making, Advances in MCDM Models, Algorithms, Theory and Applications*, vol. 8, T. Gal, T. J. Stewart, and T. Hanne, eds., pp. 1–33, 1999.

W. Shin and A. Ravindran, "An Interactive MOMP Method: A survey," *Computers & Operations Research* vol. 18, pp. 97–114, 1991.

H. A. Simon, *Models of Man*, MacMillan: New York, 1957.

B. G. Thengvall, J. F. Bard, and G. Yu, "Balancing user preferences for aircraft schedule recovery during irregular operations," *IIE Transactions* vol. 32, pp. 181–193, 2000.

D. Trentesaux, N. Moray, and C. Tahon, "Integration of the human operator into responsive discrete production management systems," *European Journal of Operational Research* vol. 109, pp. 342–361, 1998.

G. Wei, G. Yu, and M. Song, "Optimization model and algorithm for crew management during airline irregular operations," *Journal of Combinatorial Optimization* vol. 1, pp. 305–321, 1997.

J. Yang, G. Yu, and Y. Xia, "Recovery of a spanning tree from disruptions," Working Paper, University of Texas at Austin, 2000.

G. Yu, *Operations Research in the Airline Industry*, Kluwer Academic Publishers: Dordrecht, 1998.