An Unified Approach to Find an Optimal Parameterized Path in a Digraph with Multiple Features

Nikolai Guschinsky, Genrikh Levin, Jean-Marie Proth

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An Unified Approach to Find an Optimal Parameterized Path in a Digraph with Multiple Features

Nikolai Guschinsky - Genrikh Levin
Jean-Marie Proth

N° 3745
Juillet 1999

THÈME 4

Rapport de recherche

An Unified Approach to Find an Optimal Parameterized Path in a Digraph with Multiple Features

Nikolai GUSCHINSKY*, Genrikh LEVIN* and Jean-Marie PROTH** and ***

ABSTRACT

This paper is devoted to a class of problems which are modeled by a digraph, and such that each arc is characterized by several weights which depend on a parameter. This parameter takes its values in a set which characterizes the arc. The goal is to find a so-called parameterized path (i.e. a path and the values of the parameters corresponding to its arcs) that optimizes a criterion which is a combination of the weights of the arcs. The problem is introduced using transportation, communication and manufacturing problems. An unified formulation is provided, as well as an algorithm. A numerical example is proposed to illustrate this algorithm.

KEYWORDS

Network Programming, Optimal Path, Multiple Features

*Insitute of Engineering Cybernetics, National Academy of Sciences of Belarus, Surganova str., 6, 220012 Minsk, Belarus

**INRIA Lorraine, Tecnopôle Metz 2000, 4 rue Marconi, 57070 Metz, France

***Institute for System Research, University of Maryland, College Park, MD 20742, USA

This work has been supported by the INTAS Project 96-820.

Une Approche Unifiée pour le Calcul d'un Chemin Paramétré dans un Graphe Orienté à Caractéristiques Multiples

Nikolai GUSCHINSKY*, Genrikh LEVIN* et Jean-Marie PROTH** et ***

RESUME

Cette communication est consacrée à une classe de problèmes modélisés par un graphe orienté dans lequel chaque arc est caractérisé par plusieurs poids qui dépendent d'un paramètre. Le paramètre prend ses valeurs dans un ensemble qui caractérise l'arc. L'objectif est de trouver un chemin paramétré optimal (i.e. un chemin et les valeurs des paramétres correspondant à chaque arc) qui optimise un critère formé d'un combinaison des poids attachés aux arcs. Le problème est introduit à l'aide de problèmes de transport, de communication et de fabrication. Une formulation unique est proposée, ainsi que l'algorithme correspondant. Un exemple numérique illustre cet algorithme.

MOTS-CLEFS

Programmation de réseaux, Chemin optimal, Caractéristiques multiples

*Insitute of Engineering Cybernetics, National Academy of Sciences of Belarus, Surganova str., 6, 220012 Minsk, Belarus

**INRIA Lorraine, Tecnopôle Metz 2000, 4 rue Marconi, 57070 Metz, France

***Institute for System Research, University of Maryland, College Park, MD 20742, USA

Ce travail a été financé par le projet INTAS 96-820.

1 INTRODUCTION

The shortest path problem has been introduced in [1] and has attracted many researchers since then (see, for instance, [12, 18]). The solutions to this problem have been widely used to solve real life problems as in ([9, 17]). In discrete optimization, the shortest path problem often appears as a subproblem in effective algorithms (see [3, 5, 10, 14]). The complexity of the shortest path problem has been studied when the graph under consideration contains circuits of negative lengths. In this case, finding the shortest path which does not contain such a circuit is an NP-hard problem (see [13]). Otherwise, algorithms of polynomial complexity have been proposed as, for instance, in [4, 12].

Freeze [11] generalized the shortest path problem to the case when the function to be optimized is recurrent-monotone (i.e. when it can be given by recurrent relationships and when non-decreasing real-valued functions are the lengths of the arcs).

In [15, 16] the constrained shortest path problem is considered. It is showed in [13] that this problem is NP-hard but, when all weights or all lengths of arcs are the same, an algorithm of complexity $O(|V|^3)$ is developed, where $|V|$ is the number of vertices of the graph. In these papers, pseudo-polynomial algorithms and ϵ-approximate algorithms are proposed.

In [2] the problem of finding the path with maximal value of the product of reliability and capacity is studied. The problem of finding path of minimal cost/capacity ratio is considered in [19]. The proposed algorithms consist in solving a sequence of problems of longest or shortest path in graphs which are derived from the initial graph by removing some arcs.

This paper is devoted to a class of problems which are modeled by a digraph, and in which each arc is characterized by several parameters (time and cost, or reliability and capacity, for instance).

In section 2, we introduce three basic problems which can be solved using the same approach. Section 3 is devoted to the transformation of the basic problems in order to reach formulations having the same properties. In sections 4, the properties are used to develop a common algorithm for these problems; the discrete and continuous cases are considered. Section 5 contains a numerical example concerning the design of a flow shop. Some simplifications of the general algorithm are introduced in section 6. Section 7 is the conclusion.

2 THREE BASIC PROBLEMS

This section is devoted to the presentation of three basic problems which are at the root of the general problem that is solved in this paper. The first problem concerns the reliability of a communication network, the second is a transportation problem, and the last one concerns the selection of a manufacturing process in linear production systems.

2.1 Reliability of a communication network

A communication network can be represented by a graph $G = (V, E)$, where V is the set of vertices and E the set of arcs. The following notations are used to formulate the problem which has been introduced in [2] :

$c_{(i,j)}$ is the reliability of arc $(i, j) \in E$. It is the probability that arc (i, j) be able to transfer information when needed.

$b_{(i,j)}$ is the capacity of $(i, j) \in E$, that is the quantity of information which can be transfered simultaneously.

We consider two vertices s and t of the graph, and we want to select the optimal path to transfer information from s to t. Let $P_{s,t}$ be the set of paths from s to t. The goal is to find $p^* \in P_{s,t}$ such that:

$$\prod_{(i,j)\in p^*} c_{(i,j)} \cdot \min_{(i,j)\in p^*} b_{(i,j)} = \max_{p\in P_{s,t}} \left\{ \prod_{(i,j)\in p} c_{(i,j)} \cdot \min_{(i,j)\in p} b_{(i,j)} \right\} \tag{1}$$

2.2 Transportation optimization

This problem, presented in [19], is modeled by $G = (V, E)$, where each vertex represents a city and each arc the connection between two cities. The following notations are the same as those introduced in subsection 2.1, but the meaning of $c_{(i,j)}$ is quite different:

$c_{(i,j)}$ is the cost incurred for transporting a unit of good from i to j.

In this case, the optimal path p^* is such that:

$$\sum_{(i,j)\in p^*} c_{(i,j)} / \min_{(i,j)\in p^*} b_{(i,j)} = \min_{p\in P_{s,t}} \left\{ \sum_{(i,j)\in p} c_{(i,j)} / \min_{(i,j)\in p} b_{(i,j)} \right\} \tag{2}$$

In this problem, we select the path which minimizes the total cost for transporting a good from s to t divided by the capacity of the path. In other words, we select the path such that the cost per unit of capacity is minimum.

2.3 Selection of a manufacturing process in a transfer line

In this problem (see [8]), each arc of $G = (V, E)$ represents an operation, and each vertex represents the beginning or the end of operation. The following notations are used:

$b_{(i,j)}$ is the manufacturing time of operation (i, j),

$c_{(i,j)}$ is the cost per unit of time to perform (i, j),

$d_{(i,j)}$ is the cost incurred to prepare the resource to perform (i, j). In other words, $d_{(i,j)}$ is the set-up cost related to operation (i, j).

p^* is the optimal manufacturing process to transform a product from state s to state t:

$$\sum_{(i,j)\in p^*} c_{(i,j)} \cdot \max_{(i,j)\in p^*} b_{(i,j)} + \sum_{(i,j)\in p^*} d_{(i,j)} =$$

$$\min_{p\in P_{s,t}}\left\{\sum_{(i,j)\in p} c_{(i,j)}\cdot \max_{(i,j)\in p} b_{(i,j)} + \sum_{(i,j)\in p} d_{(i,j)}\right\} \tag{3}$$

In this formulation, $\max_{(i,j)\in p} b_{(i,j)}$ reflects the time required to manufacture one unit of product, since the system is a transfer line. Thus, the first term of each side of (3) is the production cost, while the second term is the total set-up cost.

In the next section, we show how to generalize these problems and to propose a unified formulation.

3 GENERALIZATION AND UNIFICATION OF THE FORMULATIONS

3.1 Generalization of the formulations

A generalization can be introduced in the problems presented in subsections 2.2 and 2.3.

In the transportation problem, $c_{(i,j)}$ and $b_{(i,j)}$ may depend upon the transportation system used. In other words, we may have the choice amongst several elements of a set of transportation resources denoted by $\Gamma_{(i,j)}$, and this set may be different from one connection (i.e. arc) to the other.

Let us introduce a parameterized path $x = (p, \gamma)$.The parameter $\gamma_{(i,j)} \in \Gamma_{(i,j)}$, $(i,j) \in p$ is associated with the arc (i,j). If $X_{s,t}$ denotes the set of all parameterized paths from s to t, then relation (2) can be rewritten as follows:

$$\sum_{(i,j)\in p^*} c_{(i,j)}(\gamma^*_{(i,j)}) / \min_{(i,j)\in p^*} b_{(i,j)}(\gamma^*_{(i,j)}) =$$

$$\min_{x\in X_{s,t}}\left\{\sum_{(i,j)\in p} c_{(i,j)}(\gamma_{(i,j)}) / \min_{(i,j)\in p} b_{(i,j)}(\gamma_{(i,j)})\right\} \tag{4}$$

where p^* and $\gamma^*_{(i,j)}$ are the parameters to be defined to optimize the criterion.

Similarly, if we assume that an operation (i,j) can be performed using different machines, the criterion (3) can be rewritten as:

$$\sum_{(i,j)\in p^*} c_{(i,j)}(\gamma^*_{(i,j)}) \cdot \max_{(i,j)\in p^*} b_{(i,j)}(\gamma^*_{(i,j)}) + \sum_{(i,j)\in p^*} d_{(i,j)}(\gamma^*_{(i,j)}) =$$

$$\min_{x\in X_{s,t}}\left\{\sum_{(i,j)\in p} c_{(i,j)}(\gamma_{(i,j)}) \cdot \max_{(i,j)\in p} b_{(i,j)}(\gamma_{(i,j)}) + \sum_{(i,j)\in p} d_{(i,j)}(\gamma_{(i,j)})\right\} \tag{5}$$

where $\Gamma_{(i,j)}$ is the set of machines available to perform (i,j).

Note that the problem introduced in 2.1 (reliability of a communication network) can be generalized in the same way. We have only to assume that $\Gamma_{(i,j)}$ contains only one element.

Sets $\Gamma_{(i,j)}$ can also be derived from the following transformation. If, in graph G, there exist paths the vertices of which (except start and end vertices) have only one predecessor and one successor, then each such a path can be replaced by a unique arc. The reliability and the capacity of this unique arc are equal, respectively, to the product of the reliabilities of the arcs of the initial path and to the minimal capacity of these arcs. In this case, the set $\Gamma_{(i,j)}$ represents paths amongst which the choice has to be made, and thus, criterion (1) can be rewritten as:

$$\prod_{(i,j)\in p^*} c_{(i,j)}(\gamma^*_{(i,j)}) \cdot \min_{(i,j)\in p^*} b_{(i,j)}(\gamma^*_{(i,j)}) = \min_{x\in X_{s,t}} \left\{ \prod_{(i,j)\in p} c_{(i,j)}(\gamma_{(i,j)}) \cdot \min_{(i,j)\in p} b_{(i,j)}(\gamma_{(i,j)}) \right\} \tag{6}$$

3.2 General formulation and properties

In this subsection, we propose a unified approach to formulate these problems. The objective of this formulation is twofold: (i) provide an unified formulation, and (ii) provide a formulation which permits to develop a unique algorithm. We will consider the three problems introduced in section 2 and show how to re-write them as an unique formulation. All the components of the unified formulation have the same properties, as explained in the next subsections.

3.2.1 Reliability of a communication network

Criterion (6) can be rewritten as:

$$\max_{x\in X_{s,t}} \left\{ \prod_{(i,j)\in p} c_{(i,j)} \cdot \min_{(i,j)\in p} b_{(i,j)} \right\} = \min_{x\in X_{s,t}} \left[\ln \left\{ \prod_{(i,j)\in p} \frac{1}{c_{(i,j)}} \cdot \max_{(i,j)\in p} \frac{1}{b_{(i,j)}} \right\} \right] \tag{7}$$

Criterion (7) is equivalent to:

$$\min_{p\in P_{s,t}} \left[\ln \left\{ \prod_{(i,j)\in p} \frac{1}{c_{(i,j)}(\gamma_{(i,j)})} \cdot \max_{(i,j)\in p} \frac{1}{(b_{i_{k-1},i_k})(\gamma_{(i,j)})} \right\} \right] =$$

$$\min_{p\in P_{s,t}} \left[\sum_{(i,j)\in p} (-\ln c_{(i,j)}(\gamma_{(i,j)})) + \max_{(i,j)\in p} (-\ln b_{(i,j)}(\gamma_{(i,j)})) \right] \tag{8}$$

By setting $\hat{c}_{i_{k-1},i_k}(\gamma_{(i,j)}) = -\ln c_{i_{k-1},i_k}(\gamma_{(i,j)})$ and $\hat{b}_{i_{k-1},i_k}(\gamma_{(i,j)}) = -\ln b_{i_{k-1},i_k}(\gamma_{(i,j)})$, the criterion becomes:

$$\min_{x\in X_{s,t}} \left\{ \sum_{(i,j)\in p} \hat{c}_{(i,j)}(\gamma_{(i,j)}) + \max_{(i,j)\in p} \hat{b}_{(i,j)}(\gamma_{(i,j)}) \right\} \tag{9}$$

If we set:

$$\begin{cases} f^1(x) = \max_{(i,j)\in p} \hat{b}_{(i,j)}(\gamma_{(i,j)}) \\ f^2(x) = \sum_{(i,j)\in p} \hat{c}_{(i,j)}(\gamma_{(i,j)}) \end{cases}$$

then the criterion is:

$$\min_{x\in X_{s,t}} (f^1(x) + f^2(x)) = \min_{x\in X_{s,t}} \Phi(f^1(x), f^2(x)) \tag{10}$$

where Φ is increasing with regard to $f^1(x)$ and $f^2(x)$.

3.2.2 Transportation optimization

Using the same approach as in subsection 3.2.1, criterion (4) can be rewritten as:

$$\min_{X \in X_{s,t}} \sum_{(i,j) \in p} c_{(i,j)}(\gamma_{(i,j)}) / \min_{(i,j) \in p} b_{(i,j)}(\gamma_{(i,j)}) =$$

$$\min_{x \in X_{s,t}} \left\{ \sum_{(i,j) \in p} c_{(i,j)}(\gamma_{(i,j)}) \cdot \max_{(i,j) \in p} \frac{1}{b_{(i,j)}(\gamma_{(i,j)})} \right\} \tag{11}$$

If we set:

$$\begin{cases} f^1(x) = \max_{(i,j) \in p} \frac{1}{b_{(i,j)}(\gamma_{(i,j)})} \\ f^2(x) = \sum_{(i,j) \in p} c_{(i,j)}(\gamma_{(i,j)}) \end{cases}$$

then the criterion is:

$$\min_{x \in X_{s,t}} (f^1(x) \cdot f^2(x)) = \min_{x \in X_{s,t}} \Phi(f^1(x), f^2(x)) \tag{12}$$

where Φ is increasing with regard to $f^1(x)$ and $f^2(x)$.

3.2.3 Selection of the manufacturing process in a transfer line

Consider criterion (5) and set

$$\begin{cases} f^1(x) = \max_{(i,j) \in p} b_{(i,j)}(\gamma_{(i,j)}) \\ f^2(x) = \sum_{(i,j) \in p} c_{(i,j)}(\gamma_{(i,j)}) \\ f^3(x) = \sum_{(i,j) \in p} d_{(i,j)}(\gamma_{(i,j)}) \end{cases}$$

With these notations, the problem is:

$$\min_{x \in X_{s,t}} (f^1(x) \cdot f^2(x) + f^3(x)) = \min_{x \in X_{s,t}} \Phi(f^1(x), f^2(x), f^3(x)) \tag{13}$$

where Φ is increasing with regard to $f^1(x)$, $f^2(x)$ and $f^3(x)$.

3.2.4 General formulation

In this subsection, we introduce a formulation which encapsulates formulations (10), (12) and (13).

If we set

$$\begin{cases} f^1(x) = \max_{(i,j) \in p} b^1_{(i,j)}(\gamma_{(i,j)}) \\ f^2(x) = \sum_{(i,j) \in p} b^2_{(i,j)}(\gamma_{(i,j)}) \\ \vdots \\ f^m(x) = \sum_{(i,j) \in p} b^m_{(i,j)}(\gamma_{(i,j)}) \end{cases}$$

then this formulation is:

$$\min_{x \in X_{s,t}} g(x) = \min_{x \in X_{s,t}} \Phi(f^1(x), f^2(x), \ldots, f^m(x)) =$$

$$\min_{x \in X_{s,t}} ((g^1 \circ f^1)(x) + (g^2 \circ f^1)(x) \cdot f^2(x) + \ldots + (g^m \circ f^1)(x) \cdot f^m(x)) \tag{14}$$

where g^r is an increasing non-negative function for $r = 1, \ldots, m$.

We also assume that $b^r_{(i,j)}(\gamma_{(i,j)}) \geq 0$ for $r = 1, \ldots, m$.

It should be remembered that x represents two parameters, that is:

- a path p;

- the value of $\gamma_{(i,j)}$ for each $(i,j) \in p$.

If we set $m = 2$, $g^1 \equiv$ identical function and $g^2 \equiv 1$, then we obtain (10).

If we set $m = 2$, $g^1 \equiv 0$ and $g^2 \equiv$ identical function, then we obtain criterion (12).

If we set $m = 3$, $g^1 \equiv 0$, $g^2 \equiv$ identical function and $g^3 \equiv 1$, then we obtain criterion (13).

We denote by **A** the problem which consists in minimizing criterion (14).

4 METHOD FOR SOLVING PROBLEM A

4.1 Basic principle

Function $g(x)$ is discrete: each path between s and t will provide one point of this function.

To solve this problem, we introduce function $\tilde{g}(x,y)$ defined as follows:

$$\tilde{g}(x,y) = g^1(y) + g^2(y) \cdot f^2(x) + \ldots + g^m(y) \cdot f^m(x)$$

where $y \in Y_0 \supset \{f^1(x) | x \in X_{s,t}\}$.

Function $\tilde{g}(x,y)$ is a weighted sum of functions $f^1(x), \ldots, f^m(x)$ which are discrete functions related to the paths between s to t.

Let us define $X_{s,t}(y) = \{x | x \in X_{s,t} \text{ and } f^1(x) \leq y\}$. This subset of $X_{s,t}$ is easy to define since, according to the definition f^1, $X_{s,t}(y)$ is the set of paths which include only arcs (i,j) whose "length" $b^1_{(i,j)}(\gamma_{(i,j)})$ is less than or equal to y. These paths are obtained by removing from the set of initial arcs the ones whose length is greater than y.

Let $x^*(y)$ be the optimal solution related to $\tilde{g}(x,y)$, that is

$$\tilde{g}(x^*(y), y) = \min_{x \in X_{s,t}(y)} \tilde{g}(x,y)$$

We know that $f^r(x) = \sum_{(i,j) \in p} b^r_{(i,j)}(\gamma_{(i,j)})$ for $r = 2, \ldots, m$. As a consequence,

$$\tilde{g}(x,y) = g^1(y) + g^2(y) \cdot f^2(x) + \ldots + g^m(y) \cdot f^m(x) =$$

$$g^1(y) + \sum_{(i,j) \in p} [g^2(y) \cdot b^2_{(i,j)}(\gamma_{(i,j)}) + \ldots + g^m(y) \cdot b^m_{(i,j)}(\gamma_{(i,j)})]$$

and the minimization of $\tilde{g}(x,y)$ for $x \in X_{s,t}(y)$ is obtained by solving the shortest path problem (see annex 1).

Obviously:

$$\tilde{g}(x^*(y), y) \geq g(x^*(y)) \tag{15}$$

and the equality is reached if and only if $f^1(x^*(y)) = y$.

Assume that $\tilde{g}(x,y)$ is minimized for a sequence $y_1, \ldots, y_k$ of values belonging to Y_0. The optimal solutions will be $x^*(y_1), \ldots, x^*(y_k)$ respectively. The minimal value among those solutions is an approximation of the optimal value of $g(x)$. The greater k, that is the greater the number of trials, the better the approximation.

4.2 Refinement in the case of finite sets of resources

We consider the case when $\Gamma_{(i,j)}$ is finite, whatever the arc belonging to a path joining s to t.

Instead of choosing a sequence $y_1, \ldots, y_k$ at random in Y_0, we propose to build the sequence of values really taken by $f(x)$ for $x \in X_{s,t}$. To build this sequence in the decreasing order of the values, we define $f^1_M(y) = \max\{f^1(x) | x \in X_{s,t} \text{ and } f^1(x) < y\}$ and compute

$$\begin{cases} y_1 = f^1_M(\infty) \\ y_k = f^1_M(y_{k-1}) \text{ for } k = 2, 3, \ldots, K, \end{cases} \tag{16}$$

where K is the cardinality of $\{f^1(x) | x \in X_{s,t}(y)\}$.

The way to find y_k knowing y_{k-1} is explained in annex 2.

Doing so, we are sure to reach the value of y which meets the optimal value of $f^1(x)$. For each y_k, we solve the problem which consists in optimizing $\tilde{g}(x, y_k)$, as explained in 4.1. The value $x^*(y_k)$ is composed with the optimal path p^* and the values $\gamma^*_{(i,j)}$, corresponding to each $(i,j) \in p^*$.

It is easy to prove the following theorem:

Theorem 1 *If $x^* = x^*(y^*)$ such that $\tilde{g}(x^*(y^*), y^*) = \min_{1,2,\ldots,K} \tilde{g}(x^*(y_k), y_k)$, then $g(x^*) = \min_{x \in X_{s,t}} g(x)$.*

Proof

Assume that there exists $x' \in X_{s,t}$ such that $g(x') < g(x^*)$. According to the process given in (16), all the possible values of y are generated. Thus, one of the values generated, say y_k, is such that $y_k = f^1(x')$ an x' belongs to $X_{s,t}(y_k)$. As a consequence:

$$g(x') = (g^1 \circ f^1)(x) + (g^2 \circ f^1)(x) \cdot f^2(x) + \ldots + (g^m \circ f^1)(x) \cdot f^m(x) \geq$$

$$\min_{x \in X_{s,t}(y_k)} \{g^1(y_k) + g^2(y_k) \cdot f^2(x) + \ldots + g^m(y_k) \cdot f^m(x)\} =$$

$$\tilde{g}(x^*(y_k), y_k) \geq \tilde{g}(x^*(y^*), y^*) = g(x^*).$$

This contradicts the assumption.

Q.E.D

It should be noticed that $K \leq \sum_{(i,j)\in E} n_{(i,j)}$ where $n_{(i,j)}$ is the cardinality of $\Gamma_{(i,j)}$.

4.3 The case of infinite sets of resources

In this case, the approach presented in 4.2 does not apply. We generate a sequence $y_1, y_2, \ldots,$ at random in Y_0 and compute the optimal $x^*(y_k)$ for $k = 1, 2, \ldots, L$. We keep $x^*(y^*)$ such that $\tilde{g}(x^*(y^*), y^*) = \min_{1,2,\ldots,L} \tilde{g}(x^*(y_k), y_k)$.

This value is not an optimal value to $g(x)$, but hopefully a near optimal value if the number of trials is large enough.

5 NUMERICAL EXAMPLE: SELECTION OF THE MACHINES IN A TRANSFER LINE

We consider a manufacturing system dedicated to a unique type of product. A set of operations is applied to each product, in the same order. Let N be the number of operations to be performed on each product.

Operation $j \in \{1, 2, \ldots, N\}$ can be performed using a machine of the set $\{m_1^j, m_2^j, \ldots, m_{n_j}^j\}$. Performing j on m_k^j requires a time $b(m_k^j)$. Furthermore, $c(m_k^j)$ is the cost per unit of time when using m_k^j, and $d(m_k^j)$ is the set-up cost of machine m_k^j. We assume that the set-up cost results from the set-up time required at the beginning of each operation: it is, for instance, the time required to unload a product and to load the next one.

The goal is to select one machine for each operation in order to solve:

$$\min_{k_1,\ldots,k_N \in \mathcal{K}} \quad \sum_{j=1}^{N} c(m_{k_j}^j) \cdot \max_{1\leq j\leq N} b(m_{k_j}^j) + \sum_{j=1}^{N} d(m_{k_j}^j)$$

where $\mathcal{K}$ is the set of indices which define a manufacturing process.

This criterion is the criterion (5) rewritten using manufacturing terms. It takes into account the fact that, in steady state, a transfer line works at the speed defined by the bottleneck machine. It provides the minimal cost incurred when manufacturing one product.

The problem arises at the design level, when the goal is to select one machine for each operation in order to minimize the production cost.

Let us consider a four stage manufacturing process. In Table 1, we provide, for each machine:

- the manufacturing time,

- the manufacturing cost per time unit,

- the set-up cost.

5 machines are candidates for operation 1, 7 for operation 2, 6 for operation 3 and 4 for operation 4.

Table 1. Machines and their characteristics.

Operation 1				Operation 2				Operation 3				Operation 4			
M	C	T	S	M	C	T	S	M	C	T	S	M	C	T	S
1	8	6	2	1	2	6	4	1	6	9	1	1	9	3	2
2	2	2	3	2	11	2	1	2	7	2	2	2	2	2	1
3	7	1	1	3	8	4	1	3	9	3	1	3	4	6	3
4	4	7	3	4	3	1	2	4	2	8	3	4	5	4	2
5	1	3	2	5	7	3	2	5	11	6	2				
				6	2	7	3	6	5	1	4				
				7	5	2	1								

The following abbreviations are used in table 1: M - machine, C - manufacturing cost of time unit, T - manufacturing time, S - set-up cost.

For this problem, graph G is shown in Fig. 1 and set $X_{s,t}$ consists of only one path.

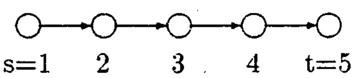

Fig. 1. Graph G.

When applying the approach proposed in 3.2, we obtain:

$y_1 = 9, y_2 = 8, y_3 = 7, y_4 = 6, y_5 = 4, y_6 = 3$ and $y_7 = 2$.

For $y = 9$ we have:

$\tilde{g}(x, 9) = 9[c(m^1_{k_1}) + c(m^2_{k_2}) + c(m^3_{k_3}) + c(m^4_{k_1})] + d(m^1_{k_1}) + d(m^2_{k_2}) + d(m^3_{k_3}) + d(m^4_{k_4}) =$
$9c(m^1_{k_1}) + d(m^1_{k_1}) + 9c(m^2_{k_2}) + d(m^2_{k_2}) + 9c(m^3_{k_3}) + d(m^3_{k_3}) + 9c(m^4_{k_1}) + d(m^4_{k_4})$,

where

$m^1_{k_1} = \text{argmin}\{8 \cdot 9 + 2, 2 \cdot 9 + 3, 7 \cdot 9 + 1, 4 \cdot 9 + 3, 1 \cdot 9 + 2\} = \text{argmin}\{74, 21, 64, 39, 11\} = 5$,

$m^2_{k_2} = \text{argmin}\{2 \cdot 9 + 4, 11 \cdot 9 + 1, 8 \cdot 9 + 1, 3 \cdot 9 + 2, 7 \cdot 9 + 2, 2 \cdot 9 + 3, 5 \cdot 9 + 1\} =$
$\text{argmin}\{22, 100, 73, 29, 65, 21, 46\} = 6$,

$m^3_{k_3} = \text{argmin}\{6 \cdot 9+1, 7 \cdot 9+2, 9 \cdot 9+1, 2 \cdot 9+3, 11 \cdot 9+2, 5 \cdot 9+4\} = \text{argmin}\{55, 65, 82, 21, 101, 49\} = 4$,

$m^4_{k_4} = \text{argmin}\{9 \cdot 9 + 2, 2 \cdot 9 + 1, 4 \cdot 9 + 3, 5 \cdot 9 + 2\} = \min\{93, 19, 39, 47\} = 2$.

Thus, $\tilde{g}(x, 9) = 11 + 21 + 21 + 19 = 72$.

In the same way, we obtain:

$\tilde{g}(x, 8) = 10 + 19 + 19 + 17 = 65$

$\tilde{g}(x, 7) = 9 + 17 + 39 + 15 = 80$

$\tilde{g}(x, 6) = 8 + 16 + 34 + 13 = 71$

$\tilde{g}(x, 5) = 7 + 17 + 29 + 11 = 64$

$\tilde{g}(x, 4) = 6 + 14 + 24 + 9 = 53$

$\tilde{g}(x, 3) = 5 + 11 + 19 + 7 = 42$

$\tilde{g}(x, 2) = 7 + 8 + 14 + 5 = 34$

Then the solution is $y^* = 2$, $m^1_{k_1} = 2$, $m^2_{k_2} = 4$, $m^3_{k_3} = 6$, $m^4_{k_4} = 2$.

An improved approach is proposed hereafter to solve problem **A**. This approach allows us to avoid the computation of $\tilde{g}(x, y)$ for some $y' < y$.

Let us consider $y \in Y_0$ and $\bar{f}^2(y), \ldots, \bar{f}^m(y)$ the minimal values of $f^2(x), \ldots, f^m(x)$ for $x \in X_{s,t}(y)$. We define

$$z = \min\{y'' | g^1(y'') + g^2(y'') \cdot \bar{f}^2(y) + \ldots + g^m(y'') \cdot \bar{f}^m(y) \geq a\} \tag{17}$$

where a is an arbitrary value.

We claim that $g^1(y') + g^2(y') \cdot f^2(x) + \ldots + g^m(y') \cdot f^m(x)$ is greater than a if $z < y$, $y' \in [z, y]$ and $x \in X_{s,t}(y')$. The reason is that

$g^1(y') + g^2(y') \cdot f^2(x) + \ldots + g^m(y') \cdot f^m(x) \geq$
$g^1(y') + (g^2(y') \cdot \bar{f}^2(y) + \ldots + g^m(y') \cdot \bar{f}^m(y)$

since $\bar{f}^r(y') \geq \bar{f}^r(y)$ if $y' < y$ (i.e. $X_{s,t}(y') \subset X_{s,t}(y)$).

The following algorithm is derived from the above remarks.

Algorithm 1

1. Compute Y_0, set of all possible values of $f^1(x)$ for $x \in X_{s,t}$. Since we do not want to consider all the paths and, for each arc (i, j) of each path p all the elements of $\Gamma_{(i,j)}$, we just compute the minimal and the maximal values in Y_0, and consider that Y_0 is the interval defined by those values. The minimal value is:

$y_m = \min\{f^1(x) | x \in X_{s,t}\}$

and it can be found by applying Dijkstra algorithm (see annex 1) by setting $c_{i,j} = b^1_{(i,j)}(\gamma^1_{(i,j)})$ and using the maximum operation instead of the sum operation.

The maximal value is:

$y_M = \max\{b^1_{(i,j)}(\gamma^{n(i,j)}_{(i,j)}) | (i, j) \in E\} = f^1_M(\infty)$.

Thus, we consider that $Y_0 = [y_m, y_M]$.

2. We optimize $\tilde{g}(x, y)$ for $y = y_m$ and $y = y_M$. The optimal paths are respectively $x^*(y_m)$ and $x^*(y_M)$. According to (15), we then consider $g(x^*(y_m))$ and $g(x^*(y_M))$ and keep the path x^* which leads to the minimal value. In other words:

$x^* | g(x^*) = \min\{g(x^*(y_m)), g(x^*(y_M))\}$.

3. Set $y = y_M$.

4. Compute $\bar{f}^r(y)$ for $r = 2, 3, \ldots, m$.

5. Compute z as defined in (17) replacing a by $g(x^*)$.

6. Compute $y = f^1_M(z)$.

7. If $y = y_m$, then stop the computation since, in this case, all the values of y have been explored, at least implicitly.

8. If $y > y_m$, compute $\tilde{g}(x, y)$ and $g(x^*(y))$.

9. If $g(x^*(y)) < g(x^*)$, then set $x^* = x^*(y)$ and $g(x^*) = g(x^*(y))$.

10. Go to 4.

Remarks

1. In general case, computation of z in point 5 can be performed using dichotomy method since $g^1(y'') + g^2(y'') \cdot \bar{f}^2(y) + \ldots + g^m(y'') \cdot \bar{f}^m(y)$ is increasing with regard to y''. But, for special cases, it is better to use the procedure defined by user. For instance, for the problems presented in 3.2.1 - 3.2.3 these procedures can be defined respectively as follows:

$$z = a - \bar{f}^2(y)$$

$$z = a/\bar{f}^2(y)$$

$$z = \frac{a - \bar{f}^3(y)}{\bar{f}^2(y)}$$

2. When computing $\bar{f}^r(y)$ for $r = 2, 3, \ldots, m$, we can take into account the fact that $\bar{f}^r(y') = \bar{f}^r(y)$ for $y' \leq f^1(x^r(y))$ where $x^r(y)$ is the path such that $f^r(x^r(y)) = \bar{f}^r(y)$.

6 FURTHER SIMPLIFICATIONS

a. Let assume that we compute the solution of $\tilde{g}(x, y)$ for a y chosen at random in $[y_m, y_M]$. The optimal solution is a path $x^*(y)$, and this path depends on y, since it has been computed on $X_{s,t}(y) \subset X_{s,t}$.

Let us now consider that our graph is restricted to this unique path $x^*(y)$, and that we try to find the optimal solution on $X_{s,t}$ which represents path $x^*(y)$ with all the possible values of $\Gamma_{(i,j)}$ for each arc (i, j) of $x^*(y)$. Since we relax the constraint on $\Gamma_{(i,j)}$, we may improve the solution $x^*(y)$.

b. Another improvement is possible when $m = 2$. In this case, point 4 of Algorithm 1 can be drastically simplified since it is no more necessary to compute $\bar{f}^2(y)$: it is equal to $f^2(x^*(y))$. This is the consequence of the following inequality:

$g^1(y) + g^2(y) \cdot f^2(x^*(y')) \geq g^1(y) + g^2(y) \cdot f^2(x^*(y))$

which holds for all $y' \leq y$ since $x^*(y') \in X_{s,t}(y)$ and $x^*(y)$ is the optimal solution.

c. Using this inequality for $m > 2$, we can also obtain an expression for $f^r(x^*(y'))$ in the following way:

$$f^r(x^*(y')) \geq \frac{\tilde{g}(x, y)}{g^r(y)} - \sum_{j=2, j \neq r}^{m} \frac{g^j(y)}{g^r(y)} \cdot f^j(x^*(y')).$$

As a consequence,

$\tilde{g}(x, y') \geq g^1(y') + \frac{g^r(y')}{g^r(y)} \cdot \tilde{g}(x, y) + \sum_{j=2, j \neq r}^{m} (g^j(y') - g^j(y) \cdot \frac{g^r(y')}{g^r(y)}) \cdot f^j(x^*(y'))$

If $r \in \{2, \ldots, m\}$ is such that $(g^j(y') - g^j(y) \cdot \frac{g^r(y')}{g^r(y)}) > 0$ for all $j \in \{2, \ldots, m\}$, $j \neq r$, then (17) can be rewritten as:

$$z = \min\{y'' | g^1(y'') + \frac{g^r(y'')}{g^r(y)} \cdot \tilde{g}(x, y) + \sum_{j=2, j \neq r}^{m} (g^j(y'') - g^j(y) \cdot \frac{g^r(y'')}{g^r(y)}) \cdot \bar{f}^j(y) \geq a\} \quad (18)$$

For instance, for problem **A** with criterion (13), relation (18) becomes:

$$z = \min\{y'' | \frac{y''}{y} \cdot \tilde{g}(x,y) + (1 - \frac{y''}{y}) \cdot \bar{f}^3(y) \geq a\} \tag{19}$$

and therefore

$$z = y \cdot \frac{a - \bar{f}^3(y)}{\tilde{g}(x,y) - \bar{f}^3(y)} \tag{20}$$

Now we apply this approach to the numerical example.

We have $y_m = 2$, $y_M = 9$, $\tilde{g}(x, y_m) = g(x^*(y_m)) = 34$, and $\tilde{g}(x, y_M) = 72$, $g(x^*(y_M)) = 65$. Therefore, $g(x^*) = 34$.

$$\bar{f}^3(9) = 4, z = 9 \cdot \frac{34 - 4}{72 - 4} = 3.97$$

$y = f_M^1(3.97) = 3$, $\tilde{g}(x,3) = 42$, $g(x^*(3)) = 42$

$$\bar{f}^3(3) = 4, z = 3 \cdot \frac{34 - 4}{42 - 4} = 2.36$$

$y = f_M^1(2.36) = 2$

Algorithm stops since $y = y_m$.

7 CONCLUSION

In this paper, we considered problems which can be expressed in a unified form given by relation (14). The properties of the components of this formulation are precisely fixed. We showed that the optimal solution can be reached if the problem is discrete, and that we can converge in probability to the optimal solution if the problem is continuous.

The basic algorithm can be drastically improved using the results of section 6.

Further research will consist in listing real life problems which can be written using the unified formulation, so we can use the proposed algorithm to obtain their solution.

References

[1] Bellman, R.E., On a routing problem, *Quarterly Applied Mathematics* (1958), 16, 87-90.

[2] Christofides, N., *Graph Theory: an Algorithmic Approach*, Academic Press, New York, (1975).

[3] Christofides, N., Mingozzia, A., and Toth, P., Exact algorithms for the vehicle routing problem, based on spanning trees and shortest path relaxations, *Mathematical Programming* (1981), 20, 255-282.

[4] Denardo, E.V., and Bennet, L., Shortest-route methods. 2. Group knapsacks, expanded networks and branch-and-bound, *Operations Research* (1979), 27, 3, 545-566.

[5] Derigs, U., A shortest augmenting path method for solving minimal perfect matching problems, *Networks* (1981), 1, 379-390.

[6] Dial, R., Glover, F., and Karney, D., A computational analysis of alternative algorithms for finding shortest path trees, *Networks* (1979), 9, 215-248.

[7] Dijkstra, E.W., A note on two problems in connection with graphs, *Numeric Mathematics* (1959), 1, 269-271.

[8] Dolgui, A., Guschinsky, N.N., and Levin, G.M., Optimal design of transfer lines and multi-position machines, *Proceedings of the 5-th International Conference on Control and Automation (MED'99),* 1999, 1962-1973.

[9] Dress, A.W.H., and Havel, T.E., Shortest path problems and molecular conformation, *Discrete Applied Mathematics* (1988), 19, 1-3, 129-144.

[10] Frieze, A., Shortest path algorithms for knapsack type problems, *Mathematical Programming* (1976), 1, 150-157.

[11] Frieze, A., Minimum paths in directed graphs, *Operations Research Quaterly* (1977), 28, 2, i, 339-346.

[12] Gallo, G., and Pallottino, S., Shortest path algorithms, *Annals of Operations Research* (1988), 13, 1-4, 3-79.

[13] Garey, M.R., and Johnson, D.S., *Computer and Intractability: a Guide to the Theory of NP-Completeness* , W.H.Freeman & Company, San Francisco, (1979).

[14] Hung, M., and Room W., Solving the assignment problem by relaxation, *Operations Research* (1980), 28, 969-982.

[15] Jaffe, J.M., Algorithms for finding path with multiple constraints,*Networks* (1984), 14, 1, 95-116.

[16] Kovaljov, M.Y., Interval ϵ-approximate algorithms for problems of finding optimal path in graph, *Vesti AN BSSR, ser.ph.-mat.n* (1988) 2, 15-20, (in Russian).

[17] Mandl, C., Evaluation and optimization of urban public transportation networks, *European Journal of Operations Research* (1980), 5, 396-404.

[18] Marsingh, D., and Chruin, P., Shortest-path algorithms: taxonomy and annotation, *Networks* (1984), 14, 2, 275-329.

[19] Martins, E.Q.V., An algorithm to determine a path with minimal cost/capacity ratio, *Discrete Applied Mathematics* (1984), 8, 189-194.

Annex 1 Minimizing the function $\tilde{g}(x, y)$ on $X_{s,t}(y)$ (Dijkstra algorithm)

The problem is modeled by a graph $G = (V, E)$, where V is the set of vertices and E the set of arcs. Furthermore, s is the origin and t the extremity of the path under consideration.

1. For each arc $(i, j) \in E$ compute $c_{(i,j)}(y)$ and $\gamma^*_{(i,j)}(y)$ as follows:

$$c_{(i,j)}(y) = g^2(y) \cdot b^2_{(i,j)}(\gamma^*_{(i,j)}(y)) + \ldots + g^m(y)) \cdot b^m_{(i,j)}(\gamma^*_{(i,j)}(y)) =$$

$$\min\{g^2(y) \cdot b^2_{(i,j)}(\gamma_{(i,j)}) + \ldots + g^m(y)) \cdot b^m_{(i,j)}(\gamma_{(i,j)}) | \gamma_{(i,j)} \in \Gamma_{(i,j)} \text{ and } b^1_{(i,j)}(\gamma_{(i,j)}) \leq y\}.$$

Set $c_{(i,j)}(y) = \infty$ if there is no $\gamma_{(i,j)} \in \Gamma_{(i,j)}$ such that $b^1_{(i,j)}(\gamma_{(i,j)} \leq y$.

2. Set $l(s) = g^1(y)$, $pr(s) = 0$, and $l(i) = \infty$, $pr(i) = -1$ for other $i \in V$.

3. Set $V_1 = \{s\}$.

4. Choose a vertex i in V_1 such that $l(i) = \min_{j \in V_1} l(j)$. If $i = t$ then go to step 6, otherwise set $V_1 = V_1 \backslash \{i\}$.

5. For any $(i, j) \in E$ whose origin is i and such that $c_{(i,j)}(y) \neq \infty$ compute $lt(j) = l(i) + c_{(i,j)}(y)$. If $l(j) > lt(j)$ then set $l(j) = lt(j)$, $pr(j) = i$ and $V_1 = V_1 \cup j$. Go to step 4.

6. Set $k = 0$ and $j = t$.

7. Set $i = pr(j)$.

8. If i is equal to 0 then stop. Otherwise set $k = k + 1$, $p(k) = (i, j)$, and $\gamma(k) = \gamma^*_{(i,j)}(y)$. Go to step 7.

Annex 2 Finding $f_M^1(y)$

1. For each arc $(i,j) \in E$ compute $b_{(i,j)}(y)$ as follows:

$$b_{(i,j)}(y) = \max\{b^1_{(i,j)}(\gamma_{(i,j)}) | \gamma_{(i,j)} \in \Gamma_{(i,j)} \text{ and } b^1_{(i,j)}(\gamma_{(i,j)}) < y\}.$$

Set $b_{(i,j)}(y) = \infty$ if there is no $\gamma_{(i,j)} \in \Gamma_{(i,j)}$ such that $b^1_{(i,j)}(\gamma_{(i,j)}) < y$.

2. Set $ps(i) = 0$ for all $i \in V$.

3. Set $V_1 = \{s\}$.

4. Choose a vertex i in V_1 such that $ps(i) = 0$. If there is no such vertex then go to step 6. Otherwise set $ps(i) = 1$ and $V_1 = V_1 \backslash \{i\}$.

5. For any $(i,j) \in E$ such that $b_{(i,j)}(y) \neq \infty$ and $ps(j) = 0$ set $ps(j) = 1$ and $V_1 = V_1 \cup \{j\}$. Go to step 4.

6. Set $pt(i) = 0$ for all $i \in V$.

7. Set $V_1 = \{t\}$.

8. Choose a vertex j in V_1 such that $pr(j) = 0$. If there is no such vertex then go to step 10. Otherwise set $pt(j) = 1$ and $V_1 = V_1 \backslash \{j\}$.

9. For any $(i,j) \in E$ such that $b_{(i,j)}(y) \neq \infty$ and $ps(i) = 0$ set $pt(i) = 1$ and $V_1 = V_1 \cup i$. Go to step 8.

10. Compute $f_M^1(y) = \max\limits_{(i,j)\in E}\{b_{(i,j)}(y) | ps(i) = 1 \text{ and } pt(j) = 1\}$.

In steps 2-5 we determine all the vertices $i \in V$ for which there is a path between s to i, and in steps 6-9 we determine all the vertices $j \in V$ for which there is a path between j to t.

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

RR-3745