Evaluation and Program Planning 33 (2010) 128–137

Contents lists available at ScienceDirect

Evaluation and Program Planning

journal homepage: www.elsevier.com/locate/evalprogplan

Multiple Choice Knapsack Problem: Example of planning choice in transportation

Tao Zhong, Rhonda Young *

Department of Civil and Architectural Engineering, University of Wyoming, 1000 E. University Avenue, Laramie, WY 82071, USA

A R T I C L E I N F O

Article history:
Received 21 July 2008
Received in revised form 2 June 2009
Accepted 12 June 2009

Keywords:
Transportation programming
Integer programming
Optimization
Branch and bound

A B S T R A C T

Transportation programming, a process of selecting projects for funding given budget and other constraints, is becoming more complex as a result of new federal laws, local planning regulations, and increased public involvement. This article describes the use of an integer programming tool, Multiple Choice Knapsack Problem (MCKP), to provide optimal solutions to transportation programming problems in cases where alternative versions of projects are under consideration. In this paper, optimization methods for use in the transportation programming process are compared and then the process of building and solving the optimization problems is discussed. The concepts about the use of MCKP are presented and a real-world transportation programming example at various budget levels is provided. This article illustrates how the use of MCKP addresses the modern complexities and provides timely solutions in transportation programming practice. While the article uses transportation programming as a case study, MCKP can be useful in other fields where a similar decision among a subset of the alternatives is required.

1. Optimization methods in transportation programming

Transportation programming is the process of selecting a final set of projects to be funded by a transportation agency (Niemeier, Zabinsky, Zeng, & Rutherford, 1995). Two common difficulties associated with transportation programming include a competition for funding among transportation modes that may have differing objectives and an often lack of clear relationships between project implementation and policy goals (Humphrey, 1974). This process is increasingly complicated by requirements in Federal transportation legislation starting with the Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA) and continuing through the current legislation, the Safe, Accountable, Flexible, Efficient Transportation Equity Act—A Legacy for Users (SAFETEA-LU) adopted in 2005. Other legislation including the Federal Clean Air Act and state and local regulations has also added to the complexity. In general, the trend in transportation programming is towards consideration of more factors in the planning process (Gage & McDowell, 1995; Greenstone, 2002; United States Congress, 2005; US DOT & FHWA, 1992). At the same time, increased public involvement is being expected and often required.

The transportation programming process can be completed with or without complex modeling tools. Programming without modeling tools is typically based on rules of thumb, heuristic methods, or decision makers' experience (Turshen & Wester, 1986). These methods are common in current practice because they are easy to apply and do not require quantitative measurements. However, the increasing number of factors to be considered and an increased emphasis on system performance make it difficult to make efficient decisions using these simple methods. For this reason, transportation programming should incorporate advanced modeling tools to help decision makers arrive at final decisions using optimization methods.

Decision making with modeling tools can be considered a decision support system (DSS). Based on the research by Keen and Scott Morton (1978), DSS was developed from the Carnegie Institute of Technology's research on the theoretical studies of organizational decision making during the late 1950s and 1960s and the Massachusetts Institute of Technology's technical work on interactive computer systems in the 1960s. DSS has been applied in such diverse areas as environmental preservation, agriculture, and medical services (Eom, Lee, Kim, & Somarajan, 1998; Hunt, Haynes, & Smith, 1998; Jintrawet, 1995; McCall & Minang, 2005). In the transportation field, DSS has been used for airline flight scheduling and project location choice problems (Jankowski & Nyerges, 2003; Jarrah, Yu, Rishnamurthy, & Rakshit, 1993). DSS has not commonly been applied to transportation programming but the complexity of the process and need for public input make it valuable tool.

Different applications define DSS differently. Finlay (1994) defines it as a computer-based system that aids the process of decision making. Turban (1995) defines it as an interactive, flexible, and adaptable computer-based information system developed for a non-structured management problem that utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights. Little's (1970) definition, which may by most applicable to the transportation programming problem, is a model-based set of

* Corresponding author. Tel.: +1 307 766 2184; fax: +1 307 766 2221.
E-mail address: rkyoung@uwyo.edu (R. Young).

0149-7189/$ – see front matter
doi:10.1016/j.evalprogplan.2009.06.007

procedures for processing data and judgments to assist a manager in his decision-making.

For a model-based system, the choice of model for use in a DSS is very important. Because many factors will be considered during the transportation programming process, Multicriteria Decision Making (MCDM) is a natural choice for this decision situation. Note that Multicriteria Decision Making is sometimes called Multi-criteria Decision Analysis (MCDA). MCDM models can be divided into two types: Multiobjective Decision Making (MODM) models and Multiattribute Decision Making (MADM) models.

In transportation programming, decision criteria are measured by project attributes such as travel time savings and construction costs, the projects and their attributes are defined explicitly and, the number of projects under consideration is limited. Because of these traits, transportation programming should use MADM as the MCDM model.

There are many options to consider in building the MADM model. Niemeier et al. (1995) categorizes three types of transportation programming methods:

- Cost-effectiveness evaluation
- Point score ranking
- Cost–benefit analysis

Each of these three models has advantages and disadvantages. Evaluations using cost-effectiveness are simple, but it is difficult to rank projects in terms of desirability (Niemeier et al., 1995). Point score ranking, though simple to perform, usually results in tight clustering of projects because of the subjective category method labels projects as high, medium, or low categories. Cost–benefit analysis is the recommended method of ranking projects (McFarland & Memmott, 1987). However, most of its applications only include those benefits measures that have been used extensively, such as travel time savings. Some measurements (e.g., environmental impacts) or objectives (e.g., increased travel options, economic development potential) that cannot be easily quantified are often not considered. This limitation is unacceptable when regulations require these factors be taken into account.

Hwang and Yoon (1981) provided a method to overcome this issue. All quantitative and qualitative factors with their associated weights are used to produce a composite index (or ranking index). The composite index can then be used in traditional Linear Programming (LP) and Goal Programming (GP) methods. Linear Programming, defined as a method to maximize or minimize a linear objective function of the decision variables (Vanderbei, 2001), has been widely used in the transportation field since the 1970's for such tasks as planning, construction scheduling, and network analysis (Aneja & Nair, 1979; Ben-Ayed, Boyce, & Blair, 1988; Moreb, 1996). Goal Programming, a multiobjective extension of Linear Programming (Ignizio, 1978), has been used in project selection for highway construction and preservation (Muthusubramanyam & Sinha, 1982). Niemeier et al. (1995) points out that one of GP's problems is that its solution cannot meet strict cost constraints. Given recent reductions in federal funding (Turshen & Wester, 1986), funding increases to accommodate GP solutions are not likely. On the other hand, LP solutions can produce an optimized project list while meeting strict cost constraints. In the past, the problem with LP was a lack of computation power necessary to solve the problems quickly. Computation technology has developed so that this is no longer a problem in traditional transportation programming processes.

In addition to LP and GP, the transportation programming problem can also be considered an Integer Programming (IP) problem. The Knapsack Problem (KP) has been well studied as a specific form of IP. In fact, Mathews (1897) performed the early research on this topic in the late 1890's. The Multiple Choice Knapsack Problem (MCKP) is a variant of KP, first used in the late 1970's (Sinha & Zoltners, 1979), that adds additional constraints that prohibit the inclusion of an object in the solution set if another object is selected. For the transportation programming problem, this allows for projects that are solutions to the same transportation issue to be considered for funding. For example, the project list could include a transit and a non-transit alternative to the same congestion problem. The additional MCKP constraints are necessary to ensure that both projects are not funded since they inherently serve the same function.

In the traditional transportation planning process, the decision on which competing project alternative was preferred was made earlier in the planning process, prior to the programming step so all projects under consideration for funding were considered independent choices. As the programming and planning processes become more complex, it has become necessary to advance competing project alternatives into the programming decision process. Increased public involvement in the programming process has also necessitated this change since the choice of a particular project alternatives is often viewed by the public as just as important as the choice to spend funding resources to solve the particular transportation issue.

Like other KP variants, MCKP can be solved using two approaches. One approach is the use of the branch-bound method and the other is dynamic programming. The branch-bound method is an enumeration approach, which reduces its search space by excluding impossible solutions. Several branch-bound algorithms have been developed (Dyer, Kayal, & Walker, 1984; Nauss, 1978; Sinha & Zoltners, 1979). Among them the DyerKayalWalker algorithm is particularly well designed and easily understood. Dudziniski and Walukiewicz (1987) and Pisinger (1995) provided a dynamic programming algorithm approach. To improve on solution efficiency, Dyer, Riha, and Walker (1995) and Pisinger (1995) presented hybrid algorithms that combine dynamic programming and upper boundary tests. Though the hybrid algorithms perform better in computation tests, its integer requirement (either the project cost or project profit must be an integer) limits its application in transportation programming (Kellerer, Pferschy, & Pisinger, 2004). For this reason, the branch-bound method is a good choice for solving the MCKP in transportation programming, though there is a need for improved solution performance for larger scale problems. It should be noted that in practice the integer requirement in dynamic programming can be overcome by scaling the decision and constraint variable by a large constant value (e.g., 1000) to obtain integer values with enough practical precision.

2. Building and solving the optimization problem

Transportation programming aims to find an optimal package of projects for funding, given a budget level, funding priorities (expressed as criteria weights), and other agency constraints such as requirements for regional distribution of projects, that can be translated into an optimization problem. Atallah (1999) defines an optimization problem as "a computational problem in which the object is to find the best of all possible solutions." More formally, to find a solution within the feasible region that has the minimum (or maximum) value of the objective function. The feasible region for the transportation programming problem is a pool of candidate projects funded that is bounded by the decision constraints, such as the total available funds. The objective function is to maximize the sum of the weighted project scores (assuming all project attributes are expressed so that larger scores represent greater desirability). Building the optimization problem requires collecting information about the decision situation that defines the feasible region and the objective function. The general process for