Primal Transportation and Transshipment Algorithms

By J.H. Ahrens, Kiel, and G. Finke, Halifax, N.S.

Received September 1979

Summary: Capacitated transshipment problems constitute the most general class of single commodity network flow problems. During the last ten years the emphasis has shifted away from the primal-dual solution methods back to specializations of the primal simplex algorithm as the more efficient approach. In this survey numerous variations in data structures, selection rules and implementations of individual steps are extracted from the relevant literature. The exposition is centered around one basic algorithm which is explained in complete detail with particular attention to the intricacies of the basis exchange part. Specialized solution methods for transportation and assignment problems are taken into account.


1. Introduction

The capacitated transshipment problem is one of the most general (linear) network flow problems. Its specializations include capacitated and uncapacitated transportation problems, assignment problems, maximum flow problems, and problems of shortest paths. These models are used in a large variety of applications which are described for instance in the textbooks of Charnes/Cooper [1961], Dantzig [1963], and Ford/
Fulkerson [1962]. The bibliographies [Hausmann; Kastning] contain a comprehensive list of the available literature. It is estimated in Charnes/Karney/Klingman/Stutz/Glover [1975] that about 70% of all real world mathematical programming problems are 'network-related'. In the past, a number of different solution methods for solving transportation and transshipment problems have been proposed, e.g. the primal simplex method [Dantzig, 1951], the dual simplex method [Balas/Hammer], the primal-dual and the out-of-kilter method [Ford/Fulkerson, 1957 and Fulkerson], the negative cycle method [Bennington], and a shortest path method [Busacker/Gowen].

Primal computer codes for transportation problems were developed as early as 1952. However, the experiments in the late 1950's and early 1960's seemed to indicate a superiority of the primal-dual method [see Dantzig, 1963; Ford/Fulkerson, 1962; Flood]. Therefore, the emphasis shifted away from the simplex specializations to the out-of-kilter approach.

The basic ideas for the contemporary primal codes were suggested during the 1960's: the predecessor index for storing the basis by Glicksman/Johnson/Eselson [1960], and the tripel label by Johnson [1966]. However, these proposals were slow in attracting general attention. The predecessor index was adopted in 1966 by Müller-Merbach [1966], and the tripel label was used the same year by the Hungarians Domolky/Frivaldszky [1966]. The full implementation began in 1970 with Glover, Klingman et al. [Glover/Karney/Klingman, 1972; Glover/Karney/Klingman/Napier; Glover/Klingman; Glover/Klingman/Stutz] and by Srinivasan/Thompson [1972, 1973].

In the following years a great deal of further algorithm development and computational testing took place. The primal code was extended to non-dense [Glover/Karney/Klingman/Napier] and capacitated problems [Langley/Kennington/Shetty] and finally to general transshipment problems [Glover/Karney/Klingman]. Much of the attention focused on the comparison of the primal and the out-of-kilter algorithms. By now, the findings of Glover/Karney/Klingman [1974] have been verified independently by a number of researchers, and it seems well established that the primal code is superior (about 30% faster) to the out-of-kilter method and, above all, requires far less storage space.

Initially the new primal computer programs were proprietary. Only very recently specific primal codes up to the state-of-the-art have been described [Ali/Helgason/Kennington/Lall; Barr/Glover/Klingman, 1979; Bradley/Brown/Graves] with promises of computer documentations on request. The available expositions of the modern network codes tend to be either formal and hard to read, or they concentrate on the underlying ideas but hide some of the technical difficulties in the computer programs. In Sections 5—10 we shall wrap our descriptions around one basic implementation explaining alternatives as we proceed through the various steps of the method. The basic version is supported by small blocks of Fortran code using identifiers which are closely related to the text. Taken together, these blocks constitute a complete subroutine NET which can solve all single commodity network flow problems quite efficiently. However, the main purpose of NET and the illustrating diagrams in Figures 1—3 is to take the 'secret' out of the technicalities in the primal network codes. Therefore the subroutine is designed to ensure that the reader cannot misinterpret the text, for example in places where tedium is avoided by discussing in full detail only one of two