Network design problem: structure of solutions and dominance relations

E. Burattini*, G. Marra**, A. Sforza**

Abstract

The network design problem (NDP), in its simplest form, is that of designing a connected subnetwork of an n node network, by selecting from the set of all edges a subset which minimizes the sum of user's shortest path costs between all node pairs of the networks, being subjected to a budget constraint which limits the number of edges that may be included in the optimal network.

In the present paper, remembering the complexity of the NDP (10) and the branch and backtrack (B&Bt) procedures applied to it, we point out the opportunity of reducing the total number of operations, required to solve it, using some dominance relations existing among its solutions. An algorithm which uses such relations is also proposed.

Introduction

The network design problem (NDP) is a well defined subject of transportation planning. In its general form it can be defined as follows: given a connected graph $G=(N, A)$ with n nodes and m edges; a subset of edges which can be invested in (improved or constructed); a set of investment costs on these edges; a set of user's costs on the edges with and without investments; a set of origin-destination (o/d) pairs on the graph; a set of demanded flows between o/d pairs: find the set of edges which minimizes the total user's cost with a budget constraint on the total investment cost. This problem is interesting since its solution may be relevant in the design of transportation networks. In all these applications the network design is obviously subjected to many more constraints than those considered in this paper,

* Istituto di Cibernetica — C.N.R. Via Toiano, 6 Arco Felice, Napoli, Italy.
** Facoltà di Ingegneria di Napoli, Cattedra di Ricerca Operativa Via Claudio, 21, Napoli, Italy.

but the solution of the NDP may be used as a measuring standard for the efficiency of applicative designs, and this justifies the study of the NDP.

Under the hypothesis that the demand flows are equal to 1 for all the *o/d* pairs on the graphs and that the subset of edges which can be invested in is equal to the set of all edges, it is possible to define a simplified version of the network design problem, i.e. to find the set of edges which minimizes the sum of the shortest path costs between all pairs of nodes with a budget constraint on the total investment cost.

This combinatorial problem is expressed by a binary programming model whose variables are associated to the edges which can be included in the network.

It belongs to the NP-complete class and then it requires exponential computation time (10).

Branch and bound techniques are generally used for its exact or approximate solution (3, 4, 5, 6, 7). The structure of these algorithms are substantially the same. It is based on two tests, lower bound test and feasibility test, applied to the partial problem P_i, generated during the procedure of separation and progressive evaluation of the B&B.

The lower bound test generally used is not powerful enough to exploit the aspects of the problem structures which are useful to improve the computation efficiency of the algorithms. It is possible to define a dominance relation $\boldsymbol{D}$ among the subproblems P_i of the NDP such that if $P_i \boldsymbol{D} P_j$ then the objective function value $f(P_i)$ is not greater than $f(P_j)$. If it occurs, and P_i has already been evaluated, then we can exclude from consideration the subproblem P_j.

The use of suitable parameters associated to the nodes of the B&B arborescence, allows us to evaluate the "goodness" of a solution and to define a bounding strategy useful to reduce the objective function evaluations number.

An algorithm which uses the dominance relation and this bounding strategy is proposed.

Mathematical formulation

Let $G=(N, A)$ be a connected undirected graph, with

$N \equiv \{v_1, \ldots, v_n\}$	set of nodes,
$A \equiv \{\alpha_1, \ldots, \alpha_m\}$	set of all possible edges,
$T \equiv \{\langle\omega, \delta\rangle \in N \times N,\ \omega \neq \delta\}$	set of origin-destination (*o/d*) pairs.

We define the following functions:

$C\colon A \to R^+;\ C \equiv \{c_1, \ldots, c_m\}$	set of user's costs on the edges,
$H\colon P \to R^+;\ H \equiv \{h_1, \ldots, h_m\}$	set of the investment costs on the edges.

Let $B \in R^+$ be the budget.

We can state the following problem:

Find a subgraph

$$G' = (N, A') \quad (A' \subseteq A)$$

such that
$$U(X) = \sum_{\langle \omega, \delta \rangle \in T} l_{\omega,\delta}(X) \text{ min!}$$

$$\sum_{i=1}^{m} x_i h_i \leq B$$

where $l_{\omega,\delta}(X)$ is the shortest path cost between ω and δ with $X \equiv \{x_1, ..., x_m\}$ i.e.:

$$x_i = \begin{cases} 1 \text{ if the edge } i \text{ is included in the network,} \\ 0 \text{ otherwise,} \end{cases}$$

and
$$l_{\omega,\delta}(X) = \sum_{i: \alpha_i \in \varrho} c_i x_i$$

where ϱ is the shortest path between ω and δ.

This problem was proved by Johnson et al. (10) to be NP-complete and by Wong (15) to be NP-hard i.e. it is quite unlikely to find an algorithm for it such that its running time be a polynomial function of the input size. Then, if we want to solve such a problem in an optimal way using the classical B&Bt techniques, it may happen that it would take too much time. For this reason, it is useful to apply branch and backtrack algorithms with some functions able to reduce at the minimum the branching on the B&Bt arborescence.

Branch and Backtrack Algorithm and Dominance Relations

The B&Bt is a computational principle which has been proved useful in solving various combinatorial optimization problems encountered in operations research and combinatorial mathematics. The underlying idea of a B&Bt procedure is to decompose a given problem into smaller and smaller partial problem. Two types of tests are applied to each partial problem to see if it can be solved or, on the contrary, be concluded that no optimal solution is obtainable from it; in both cases, the partial problem is terminated and not decomposed any further. These tests are called lower bound (for minimization problem) and feasibility tests. The computation terminates when all nodes are either decomposed or terminated. In the NDP the feasibility test is done verifying if the best solution of the subproblem P_i satisfy the constraint of the budget B. The lower bound $g(P_i)$ of the optimal value $f(P_i)$ of a partial problem P_i is generally found evaluating the objective function value of the best solution of P_i, obtained setting to 1 the free variables of P_i.

If $g(P_i) \geq z$ where z is the current optimum, that is the value of the best feasible solution of P_0 (the given minimization problem) obtained so far, we conclude that P_i does not provide an optimal solution of P_0, and P_i is terminated. A generalization of this lower bound test can be done using a binary relation $\boldsymbol{D}$, called dominance relation.

Let $\boldsymbol{P} \equiv \{P_1, ..., P_p\}$ be the set of partial problems generated by a B&Bt procedure. A dominance relation $\boldsymbol{D}$ is a partial order relation over $\boldsymbol{P}$ which satisfies the following conditions (7).

1) $\boldsymbol{D}$ is:
transitive $P_i \boldsymbol{D} P_j$ & $P_j \boldsymbol{D} P_k \rightarrow P_i \boldsymbol{D} P_k$,
reflexive $P_i \boldsymbol{D} P_i$,
antisymmetric $P_i \boldsymbol{D} P_j$ & $P_j \boldsymbol{D} P_i \rightarrow P_i = P_j$;

2) $P_i D P_j$ & $P_i \neq P_j \rightarrow f(P_i) \leqq f(P_j)$ and
P_i is not a proper descendant of P_j;

3) $P_i D P_j$ & $P_i \neq P_j$ imply that some descendant
$P_{i'}$ of P_i satisfies $P_{i'} D P_{j'}$ & $P_{i'} \neq P_{j'}$ for any descendant $P_{j'}$ of P_j.

It is obvious that P_j need to be solved if P_i is already generated and $P_i D P_j$ holds; thus P_j can be terminated. A dominance relation D may be interpreted as an embodiment of the information on optimal solutions of partial problems obtainable without actually solving them (that is computing $f(P_j)$), and can be regarded as a generalization of the lower bound test.

For the NDP we can assert that the total cost function $U(X)$ is a monotone non-increasing function of the decisional variables $\{x_i\}$. Consider two solutions X^J and X^K, best solution of the partial problems P_j and P_k.

We say that

$$X^J \geqq X^K \quad \text{if} \quad \forall i x_i^J \geqq x_i^K.$$

The following statements hold:

i) $X^J \geqq X^K$ & $X^K \geqq X^M \rightarrow X^J \geqq X^M$,

$X^K \geqq X^K$,

$X^J \geqq X^K$ & $X^K \geqq X^J \rightarrow X^K = X^J$,

ii) $X^J \geqq X^K \rightarrow U(X^J) \leqq U(X^K)$.

Consider an edge i such that $x_i^J \geqq x_i^K$ (that is $x_i^J = 1$ and $x_i^K = 0$);

the flow unit between each pair $\langle \omega, \delta \rangle$ using through X^J a shortest path containing α_i, will use through X^K another path of cost $l_{\omega,\delta}(X^K) \geqq l_{\omega,\delta}(X^J)$,

iii) $X^J \geqq X^K$ & $X^J \neq X^K \rightarrow$

$\forall K'$: $X^{K'} \leqq X^K \exists J'$: $X^{J'} \leqq X^J$ & $X^{J'} \geqq X^{K'}$.

If we consider the solutions X^J and X^K as best solutions of the problems P_j and P_k we can state that

$$X^J \geqq X^K \leftrightarrow P_j \mathbf{D} P_k. \tag{a}$$

In Fig. 1 the complete arborescence of an NDP with 4 variables is represented. In Fig. 2 a sequential graph which takes care of all dominance relations among the solution is shown. To each level of this graph the solutions with the same number of variables set to 1 belong. Generally, the known B&B algorithms use only partially the dominance relation. The thick line represents the dominance relations implicitly considered in B&Bt algorithms, the sharp line, the other dominance relations.

A more readable version of the graph of Fig. 2 is reported in Fig. 3. It can be remarked that:

a) none of the solution is dominated by any other of the successive level,
b) none of the solutions dominated by at least one solution belonging to the previous level,

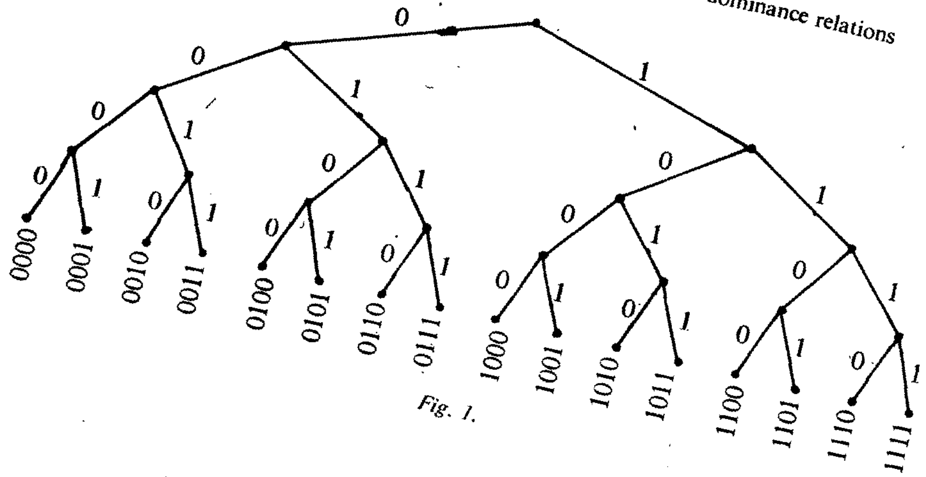

Fig. 1.

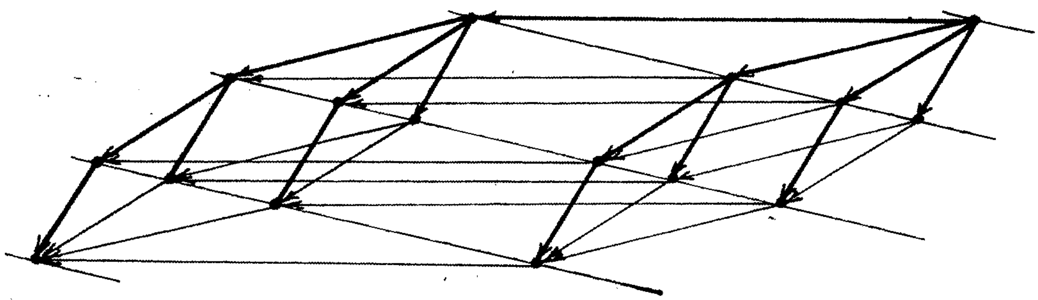

Fig. 2.

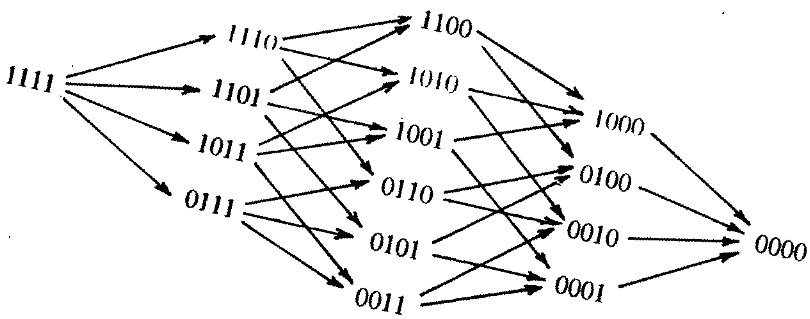

Fig. 3.

c) solutions belonging to the same level are not comparable using the $\boldsymbol{D}$ relations. Obviously the number of level is equal to $m+1$ and for each level there are $\binom{m}{l}$ different solutions, $(l=0, 1, \ldots, m)$.

If a problem P_i is dominated by another P_j already solved with $U(X^J) \geqq z$ (current upper bound) we do not need to solve P_i, that is to compute the relative objective function value.

We underline that applying the dominance relation does not reduce the number of the generated partial problems but only the number of the solved problems.

Some remarks on the structure of the solutions

The graphical representation of Fig. 3 enables us to identify quickly the subsets of feasible and infeasible solutions. In fact the "border" between these two subsets can be defined as the subset of those feasible solutions, dominated by the infeasible ones.

If the investment costs h_i $(i=1, \ldots, m)$ are all equal to $\bar{h}$ the border is represented by the level $\lfloor B/\bar{h} \rfloor$. If the h_i costs are not equal, the border can be represented by solutions belonging to different levels. In this case, arranging the h_i in non-decreasing order, the level l satisfying the conditions

$$\sum_{i=1}^{m+1-l} h_i \leqq B \quad \text{and} \quad \sum_{i=1}^{m+2-l} h_i \geqq B$$

is the first level with border solutions, that is the level with border solutions having the highest number of 1. The remaining feasible solutions of the border belong to successive levels $(l+1, \ldots, m)$ and can be identified starting from the infeasible solutions of level l.

The optimal solutions of the problem belong to the border, on which all solutions are not comparable and altogether dominate all the other feasible ones.

Using the dominance relation without solving the problems P_i, whose best solutions are infeasible, we must solve all and only those belonging to the border. If we do not use the dominance relation, the classical criteria applied in B&Bt algorithm does not prevent us from analyzing solutions not belonging to the border and then obviously not optimal.

The proposed algorithm

The algorithm we propose here is a classical B&Bt algorithm with the addition of the dominance test and of some heuristic devices for the bounding strategy. A more general scheme of our algorithm is reported in Fig. 4.

Some detailed steps are the following

— The algorithm realizes a preliminary arrangement of the variables to speed-up the procedure.

— A feasibility test verifies if the available budget allows us to set up an other variable to 1.

```
     begin
1 —      𝒜={P₀}; z←∞; 𝒪←Φ
     end
     begin
2 —      while 𝒜≠Φ do
3 —      Pᵢ=s(𝒜)
4 —      if Pⱼ𝒟Pᵢ for some Pⱼ≠Pᵢ belonging to the set of nodes currently generated then
             begin
5 —              𝒜=𝒜−{Pᵢ}
             end
         else
6 —          if Pᵢ can be solved or proved to be infeasible then
                 begin
7 —                  if z>f(Pᵢ) then
                         begin
8 —                          𝒪=0(Pᵢ); z=f(Pᵢ)
                             go to 11
                         end
                     else
9 —                      if z=f(Pᵢ) then
                             begin
10 —                             𝒪=𝒪∪0(Pᵢ)
                             end
11 —                 𝒜=𝒜−{Pᵢ}
                 end
             else
12 —             if g(Pᵢ)>z then
                     begin
13 —                     𝒜=𝒜−{Pᵢ}
                     end
                 else
14 —                 generate sons Pᵢ₁, ..., Pᵢₖ of Pᵢ; 𝒜←𝒜∪{Pᵢ₁, ..., Pᵢₖ}−{Pᵢ}
     end
     begin
15 —     if z=∞ then
             begin
16 —             print "P₀ is infeasible"
             end
         else
17 —         0(P₀)←𝒪; f(P₀)←z
     end
```

Fig. 4.

- A first test on budget left compares the minimum value of the free variables investment costs with the remaining available budget.
- The dominance test compare the best solution of the current subproblem with the list of the not-yet dominated solution.
- A second test on budget left compares the sum of the free variables investment costs with the remaining available budget.
- A solution "goodness" test compares the value of a solution "parameter" with the parameter value of the current upper bound solution.

The algorithm has been tested on small dimension networks (6 nodes, 13 edges) varying user's and investment costs.

The results showed that the dominance relation effect is relevant when many $U(X)$ values must be evaluated, that is when the budget is approximately the 50%

of the $\sum_{h_i \in H} h_i$. It is useful to define some conditions about the number of requested operations.

Let T_A be the number of solved problems, t_A the operation number to solve each partial problem, t_D the average operation number for the dominance test, T_D the number of generated and dominated problems. With good approximation we can state the total operations number T', using the dominance test, is

$$T' = t_D(T_A+T_D)+T_A t_A$$

and the total operations number T' without using the dominance test, is

$$T'' = (T_A+T_D)t_A.$$

Introducing dominance test is useful until $T' < T''$, i.e.

$$T_D > \frac{t_D}{t_A}(T_A+T_D).$$

Finally, it seems useful to introduce the dominance relation in B&Bt procedure for NDP, also if the running time to conclude the arborescence remain relevant in some cases. The optimal solution, with a good arrangement of the variables is fastly found, before the end of the procedure. This considerations allows us to use the algorithm as heuristic, stopping it before the end, and getting the last feasible solution (current optimum).

Conclusions

In this paper we developed some considerations on the structure of the solution set of the problem, identifying a border of feasible solutions between the two subsets of feasible and infeasible solutions.

We have also shown the opportunity of introducing a dominance test, regarded as a generalization of the lower bound test, in the basic structure of the B&Bt algorithm for the NDP.

The principal effect of this introduction is to improve computational efficiency of the B&Bt procedure. In fact in this way a larger number of partial problems can be terminated without evaluating the objective function value.

The first results suggested the fitness of using the dominance test in a quite well defined range of the investment costs. Moreover the considerations on the solutions set structure, suggested to construct an "ad hoc" algorithm which examines directly the set of feasible solutions of the border and the set of the infeasible solutions which dominate the border.

References

[1] Beale, E. M. L., M. G. Kendall, D. W. Mann (1967), The discarding of variables in multivariate analysis, *Biometrika*, 54, 357—366.

[2] Boyce, D. E., A. Farhi, R. Weischedel (1973), Optimal network problem: a branch and bound algorithm, *Environment and Planning*, 5, 529—533.

[3] BOYCE, D. E. eds. (1979), Transportation Research 13B, Special Issue on Network Design Problem.
[4] FLORIAN, M., R. DIONNE (1977), Exact and approximate algorithms for optimal network design, *Publication 41, Centre de recherche sur les transports,* Montreal.
[5] GALLO G. (1981), A new branch and bound algorithm for the network design problem, *"Optimization Days"* Montreal.
[6] GALLO, G. (1981), "Lower planes for the Network Design Problem", S—81—22 — ISI — Pisa.
[7] HOANG, H. H. (1973), A computational approach to the selection of an optimal network, *Management Science,* 19, 5, 488—498.
[8] IBRAKI, T. (1977)a, On the computational efficiency of the branch and bound algorithms, *Journal of the Operations Research of Japan,* 20, 1, 16—35.
[9] IBRAKI, T. (1977)b, The power of the dominance relation in branch and bouhd algorithm, *Journal of the Association for Computing Machinery,* 24, 2, 264—279.
[10] JOHNSON, D. S., J. K. LENSTRA, A. H. G. RINNOY KAN (1978), The complexity of the network design problem, *Networks,* 8, 279—285.
[11] LE BLANC, L. J. (1975), An algorithm for the discrete network design problem, *Transportation Science,* 9, 183—199.
[12] OCHOA-ROSSO, F., A. SILVA (1968), Optimum project addition in urban transportation network via descriptive traffic assignement models, *Research Report R68—44,* Transportation System Division, MIT, Cambridge, Massachusets.
[13] RIDLEY (1968), An investment policy to reduce the travel time in a transportation network, *Transportation Research,* 2, 409—424.
[14] STEENBRINK, P. A. (1974), *Optimization of transport network,* Wiley, New York, *Transportation Research* (1979) 13B, Special Issue on network design problem.
[15] WONG R. T. (1980), "Worst-case analysis of Network Design Problem Heuristics", Siam J. Alg. Discr. Math., Vol. 1, N. 1, March, 51—63.

Received June 2, 1983