

**AN EFFICIENT TRANSFORMATION OF THE
GENERALIZED TRAVELING SALESMAN PROBLEM**

Charles E. Noon
Management Science Program
The University of Tennessee
Knoxville, TN 37996

James C. Bean
Department of Industrial & Operations Eng.
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 89-36

December 1989

An Efficient Transformation of the Generalized Traveling Salesman Problem

Charles E. Noon

Management Science Program

The University of Tennessee, Knoxville 37996.

James C. Bean

Department of Industrial and Operations Engineering

The University of Michigan, Ann Arbor 48109.

December 17, 1989

ABSTRACT

This paper discusses the Generalized Traveling Salesman Problem (GTSP). The GTSP is defined on a directed graph in which the nodes are members of predefined *nodesets*. Each node is a member of some but not all nodesets. The GTSP seeks a minimum cost cycle which passes through each nodeset exactly once. Many discrete decisions which complicate routing problems can be modeled in this form. In this paper, we describe a class of problems that can be formulated as GTSP's and show how any problem in this class can be efficiently transformed into a standard traveling salesman problem.

Keywords: Traveling salesman problem, vehicle routing, integer programming.

The Traveling Salesman Problem (TSP) is one of the oldest and most widely studied optimization problems in the field of operations research (see [8, 13]). For the standard TSP, we assume that a single salesman, based at a home city, must visit a number of cities (or customers). The problem is to find the salesman's minimum cost cycle (or *tour*) which leaves the home city, passes through all the customer cities and returns to the home city.

Real routing or sequencing problems often require additional decisions or considerations. Often, the standard TSP is used only after a number of discrete decisions have been made. For example, in vehicle routing, the standard TSP is often applied after the customer to vehicle assignments have been made. Preliminary decisions may also include choice of fleet vs. contract carrier, depot locations, backhauling, and alternative vehicles.

This paper provides a framework for including many of these discrete complications into a single, structured model, the *Generalized Traveling Salesman Problem* (GTSP). The GTSP is defined on a directed graph in which the nodes are members of predefined *nodesets*. Each node is a member of some, but not all, nodesets. For the GTSP, we seek a minimum cost cycle which passes through each nodeset exactly once. Since a feasible cycle need not include every node, the decision regarding which nodes to pass through is imbedded within the GTSP. The model requires that we consider both the sequence and selection decisions simultaneously.

Early applications using GTSP formulations are given for sequencing computer files [3] and for routing welfare clients through governmental agencies [14]. More recently, Laporte, Nobert and Mercure [7] discuss GTSP applications in the routing of mail vans. Several potential applications are given in [9] including warehouse order-picking with multiple stock locations, airport selection/routing for courier planes, and certain types of flexible manufacturing scheduling. Recent GTSP formulations for the Multi-Depot Multiple Traveling Salesman Problem (MTSP) and the Heterogeneous MTSP can be found in [11].

Certain other discrete decision/routing problems from the literature may be reformulated as GTSP's. One recent example is the Covering Salesman Problem (CSP) studied by Current and Schilling [2]. For the CSP, we are given a directed graph and a "covering radius", S . Unlike the standard TSP, the CSP does not require that each city be visited. A feasible tour for the CSP need only include a visit within radius S of each city. In this respect the CSP seeks a

minimum cost tour that “covers” all cities. The CSP can be modeled as a GTSP by replacing arc costs with shortest path costs (as discussed in [2]) and then defining nodesets. For each city, we define a nodeset with members consisting of the city and all surrounding cities within radius S .

In this paper, we show how any problem formulated as a GTSP can be transformed into a standard TSP. One benefit of this type of transformation is that it allows existing TSP approaches, both heuristic and optimal, to be applied to the transformed GTSP problems. Such an approach may not necessarily outperform specialized algorithms for the GTSP type problems, however, it would provide researchers with a tool for pursuing or verifying optimality on smaller problems. Another benefit is that the transformation may allow us to relate results for the well-studied TSP to the more realistic complex routing problems. As an intermediate step, the GTSP is transformed to a canonical GTSP. An alternative solution approach is to employ specialized algorithms available for this problem, e.g. [7, 10].

1.0 Problem Statement

The asymmetric GTSP is defined on a directed graph with nodes \mathcal{N} and connecting arcs \mathcal{A} . Each arc, $(i, j) \in \mathcal{A}$, has a corresponding cost, c_{ij} , with $|c_{ij}| < +\infty$. The nodes of the GTSP belong to one or more predefined *nodesets*, $\mathcal{S}_i, i = 1, 2 \dots, m$. Any feasible solution must pass through at least one member of each nodeset. We assume that each node is a member of some nodesets, but not all. A further assumption is that there does not exist a nodeset which overlaps with all other nodesets, that is, \nexists some nodeset \mathcal{S}_i such that for each $j, \mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$.

A feasible GTSP *tour* is defined as a directed cycle (not necessarily simple) which enters and exits each nodeset exactly once. A *subtour* is a directed cycle which passes through each nodeset at most once and is not a tour. Figures 1(a) and 1(b) display example feasible GTSP tours. Figure 1(a) displays a feasible tour which passes through a node twice. Figure 1(b) displays a feasible tour which passes through an arc more than once. Both are acceptable as long as the tour is uninterrupted and each nodeset is entered and left exactly once. The GTSP can be stated as *the problem of finding a minimum cost feasible GTSP tour*. The standard TSP is a special case of the GTSP with mutually exclusive nodesets of cardinality one.

We can formulate the GTSP as an integer program with variable x_{ij} equal to the number of times arc (i, j) is traversed in a solution. The GTSP can be stated:

$$\begin{aligned}
& \text{Minimize } \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} \\
& \text{Subject to,} \\
& \left. \begin{aligned}
& \sum_{i \in \mathcal{S}_k} \sum_{\substack{j \in \mathcal{S}_k \\ (i,j) \in \mathcal{A}}} x_{ij} = 1 \\
& \sum_{i \notin \mathcal{S}_k} \sum_{\substack{j \in \mathcal{S}_k \\ (i,j) \in \mathcal{A}}} x_{ij} = 1
\end{aligned} \right\} \text{for all nodesets, } \mathcal{S}_k, k = 1, \dots, m. \quad (i) \\
& \sum_{\substack{i \in \mathcal{N} \\ (i,j) \in \mathcal{A}}} x_{ij} - \sum_{\substack{k \in \mathcal{N} \\ (j,k) \in \mathcal{A}}} x_{jk} = 0 \quad \text{for all nodes } j \in \mathcal{N}. \quad (ii) \\
& \sum_{(i,j) \in \mathcal{A}_{\mathcal{T}}} x_{ij} - \sum_{k \in \mathcal{N}_{\mathcal{T}}} \sum_{\substack{l \in \mathcal{N}_{\mathcal{T}} \\ (k,l) \in \mathcal{A}}} x_{kl} \leq |\mathcal{A}_{\mathcal{T}}| - 1 \quad \text{for all possible subtours } \mathcal{T}, \\
& \text{where } \mathcal{A}_{\mathcal{T}} \text{ represents the arcs of } \mathcal{T} \quad (iii) \\
& \text{and } \mathcal{N}_{\mathcal{T}} \text{ represents the nodes of } \mathcal{T}. \\
& x_{ij} \geq 0, \text{ integer} \quad \text{for all } (i, j) \in \mathcal{A}.
\end{aligned}$$

Group (i) contains two constraints for each nodeset to ensure that exactly one arc entering a nodeset and one arc leaving a nodeset are included in a solution tour. Constraint group (ii) contains one constraint for each node to ensure a solution tour is uninterrupted. Constraint group (iii) prevents a *disjoint* subtour from appearing. As seen in Figures 1(a) and 1(b), small cycles are allowed if they are part of a complete tour.

The GTSP form presented in this paper requires that a feasible solution pass through each nodeset *exactly once*. In [9], a more general form is presented which allows nodesets to be specified as requiring either *exactly one* visit or *at least one* visit. It is shown that any problem in the more general form can be transformed into the form presented in this paper.

2. Problem Transformation

We now want to show that any problem formulated as a GTSP can be transformed to a standard TSP. The fact that this is possible is not surprising since complexity theory assures us that any NP-hard problem can be polynomially transformed to any other NP-hard problem. Such transformations, however, are often accompanied by a significant increase in the size of the problem instance. For the transformation we present, the number of nodes in the resulting TSP is given by $\sum_{i=1}^m |\mathcal{S}_i|$. That is, the number of nodes is only increased according to nodes appearing in multiple nodesets.

The entire transformation is divided into three main stages. In the first stage, the general problem is transformed to a GTSP having mutually exclusive nodesets. In the second stage, the intraset arcs are eliminated leading to the *canonical form*. In the third stage, the canonical form of the problem is transformed to a clustered TSP and then finally into a standard TSP.

We define the following notation. Let $\mathcal{M}(i)$ be the collection of nodesets of which node i is a member, i.e, if $i \in \mathcal{S}_K$, then $K \in \mathcal{M}(i)$. Let $(P0)$ be a restatement of the general problem. See Figure 2 for an example.

Definition : Problem $(P0)$ is a GTSP defined over nodes \mathcal{N}^0 and arcset \mathcal{A}^0 with corresponding cost vector c^0 . For a given node i , let $\mathcal{M}(i)$ index the nodesets containing i .

2.1 Mutually Exclusive Nodesets

Given that we have a problem in the form of $(P0)$, this section describes a transformation to make the nodesets mutually exclusive. We begin by defining a problem $(P1)$ which has nodes identical to $(P0)$ but a different arcset. Define an arc in $(P1)$ from i to j only if node j belongs to a nodeset to which i does not. This ensures that each arc crosses into *at least one new nodeset*. This characteristic will allow us to change all arc costs without disturbing the order of solutions.

Definition: Problem $(P1)$ is a GTSP defined over nodes $\mathcal{N}^1 = \mathcal{N}^0$. The nodes are the same with respect to nodeset membership as well. The arc set \mathcal{A}^1 is defined as follows.

Let i, j be two nodes of \mathcal{N}^0 such that $\mathcal{M}(i) \not\supseteq \mathcal{M}(j)$. Calculate the shortest path cost from i to j over the arcset $\mathcal{A}_{ij} \subseteq \mathcal{A}^0$, where arc $(k, l) \in \mathcal{A}_{ij}$ if the following hold,

$$\mathcal{M}(k) \subseteq \mathcal{M}(i) \cup \mathcal{M}(j),$$

$$\mathcal{M}(l) \subseteq \mathcal{M}(i) \cup \mathcal{M}(j),$$

l does not belong to a nodeset $\mathcal{S}_P, P \in \mathcal{M}(i)$, such that $k \notin \mathcal{S}_P$,

k does not belong to a set $\mathcal{S}_Q, Q \in \mathcal{M}(j)$, such that $l \notin \mathcal{S}_Q$.

If the shortest path has finite cost, create arc $(i', j') \in \mathcal{A}^1$ with cost $c_{i',j'}^1$, equal to the shortest path cost. If no feasible path exists, then no arc (i', j') is created for \mathcal{A}^1 .

Each shortest path is calculated over a subnetwork which allows nodeset boundaries to be crossed only once. Each arc $(i', j') \in \mathcal{A}^1$ corresponds to the shortest path from i to j which does not pass through any other nodesets. Figure 3 shows an example. If the shortest path contains a negative cycle, the negative cycle can only occur over intraset arcs without crossing nodeset boundaries. In this case, a corresponding arc is created and given a large negative cost. If there exists a feasible solution to $(P1)$ which contains the arc with the large negative cost, then the original problem is unbounded below.

Lemma 1 : *Given an optimal solution, y^* , to $(P1)$, we can construct an optimal solution x^* to $(P0)$ with $c^0 x^* = c^1 y^*$.*

Proof : See Appendix A.

Now that we have redefined the arcs and arc costs of the problem so that each arc enters at least one new nodeset, we can then add a constant to the cost of all arcs entering a nodeset without changing the order of solutions. Let us create a problem $(P2)$ which differs from $(P1)$ only by arc costs.

Definition: **Problem $(P2)$** is a GTSP over the nodeset $\mathcal{N}^2 = \mathcal{N}^1$ with $\mathcal{A}^2 = \mathcal{A}^1$. The nodes are the same with respect to nodeset membership as well. The arc costs for $(P2)$ are computed below. First, however, we must make all costs of $(P1)$ greater than or equal to zero. Let $c_{min} = \text{Minimum}\{c_{ij}^1 \text{ over all } (i, j) \in \mathcal{A}^1, 0\}$. For each nodeset, subtract c_{min} from all arcs entering that nodeset. Since each arc of $(P1)$ enters at least

one new nodeset, all arc costs will become nonnegative.

Assuming now $c^1 \geq 0$, let α be defined as a positive number strictly greater than the sum of all arc costs, i.e., $+\infty > \alpha > \sum_{(i,j) \in \mathcal{A}^1} c_{ij}^1$. Given an arc $(i, j) \in \mathcal{A}^1$, let the cost of arc $(i', j') \in \mathcal{A}^2$ be defined as $c_{i'j'}^2 = (|\mathcal{M}(j) - \{\mathcal{M}(i) \cap \mathcal{M}(j)\}|)(\alpha) + c_{ij}^1$. This is equivalent to adding α to the arcs entering each nodeset. Since each arc crosses into at least one new nodeset, $|\mathcal{M}(j) - \{\mathcal{M}(i) \cap \mathcal{M}(j)\}| \geq 1$.

Lemma 2 : *Problem (P2) is equivalent to Problem (P1).*

Proof : Both cost adjusting operations are permitted since each nodeset must be entered exactly once in a solution tour. ■

Figure 4 shows an example. After the transformations to (P1) and (P2), every arc enters at least one new nodeset and has a cost of at least α . Any feasible solution to (P2) will have a total cost strictly less than $(m + 1)\alpha$ and greater than or equal to $(m)\alpha$. These characteristics ensure equivalence is preserved throughout the next step of the transformation. We now create a problem (P3) which has mutually exclusive nodesets and is equivalent to (P2).

Definition: Problem (P3) is a GTSP over $\mathcal{N}^3, \mathcal{A}^3, c^3$, defined below. For every node i in (P2), we create a group of $|\mathcal{M}(i)|$ nodes for (P3), each with membership in only one nodeset. More formally, for $i \in \mathcal{N}^2$, define $i^P \in \mathcal{N}^3$ for all $P \in \mathcal{M}(i)$ with i^P belonging to only one nodeset, S_P . Hence, $|\mathcal{M}(i^P)| = 1$.

Let the arcset \mathcal{A}^3 be defined as follows.

- For an arc $(i, j) \in \mathcal{A}^2$, create $(i^P, j^Q) \in \mathcal{A}^3$ with cost $c_{i^P j^Q}^3 = c_{ij}^2$ for all $P \in \mathcal{M}(i), Q \in \mathcal{M}(j)$.
- For a node $i \in \mathcal{N}^2$ such that $|\mathcal{M}(i)| > 1$, create arcs $(i^P, i^Q) \in \mathcal{A}^3$ with cost $c_{i^P i^Q}^3 = 0$ for all $P \in \mathcal{M}(i), Q \in \mathcal{M}(i), P \neq Q$.

Problem (P3) now has mutually exclusive nodesets. Each arc cost of c^3 is either greater than or equal to some nonzero multiple of α or is equal to zero. Figure 5 shows an example.

Lemma 3 : *Given an optimal solution, y^* , to (P3) such that $c^3y^* < (m + 1)\alpha$, we can construct an optimal solution, x^* , to (P2) with $c^2x^* = c^3y^*$.*

Proof : See Appendix B.

The importance of the cost adjusting (P2) transformation towards ensuring equivalence can best be illustrated by an example. Figure 6 shows a problem *without* the (P2) transformation and the same problem *with* the (P2) transformation. Both problems are then put in the form of (P3). The first problem allows a feasible solution with cost zero when, in fact, we cannot construct such a solution for the original problem. The problem that was transformed through (P2) allows only feasible solutions with cost greater than $(m+1)\alpha$. Thus, it correctly recognizes that the original problem is infeasible. In general, α represents a large penalty for visiting *some*, rather than *all*, of the (P3) nodes spawned from a particular multiple membership (P2) node. It forces the spawned nodes to be visited contiguously if visited at all.

A feasible solution to (P3) passes through each nodeset exactly once. The arc costs ensure that if a nodeset is entered, a cost of at least α corresponding to that nodeset is incurred in the preceding arcs. The only way the α cost for entering a nodeset could not be incurred is if the nodes of a solution to (P3) all belonged to some particular nodeset. However, in the original problem, we assumed there existed no nodeset which overlapped with all other nodesets. Therefore, any feasible solution to (P3) must have cost greater than or equal to $(m)\alpha$.

In this section, we began with a problem with overlapping nodesets and transformed it to a problem in which the nodesets are mutually exclusive. We may now summarize the results of this section.

Theorem 4 : *Given a GTSP in the form of (P0), we can transform the problem to a problem of the form of (P3). Given an optimal solution to (P3) with cost less than $(m + 1)\alpha$, we can construct an optimal solution to (P0). If an optimal solution to (P3) has cost greater than or equal to $(m + 1)\alpha$, problem (P0) is infeasible.*

The next stage begins with a problem of the form (P3) and transforms it to a problem which contains no intraset arcs.

2.2 Intraset Arcs

The second stage of the problem transformation eliminates intraset arcs. This procedure is applied after the problem has mutually exclusive nodesets. Given a problem of the form (P3), we can define a problem (P4) which differs only in the arcs and arc costs.

Definition: Problem (P4) is a GTSP defined over nodes $\mathcal{N}^4 = \mathcal{N}^3$. The nodes are the same with respect to nodeset membership as well. The arc set \mathcal{A}^4 is defined as follows. Let i, j be any two nodes of \mathcal{N}^3 such that $\mathcal{M}(i) \neq \mathcal{M}(j)$. Calculate the shortest path cost from i to j over the arcset $\mathcal{A}_{ij} \subseteq \mathcal{A}^3$, where arc $(k, l) \in \mathcal{A}_{ij}$ if the following hold,

$$\mathcal{M}(k) \subseteq \mathcal{M}(i) \cup \mathcal{M}(j),$$

$$\mathcal{M}(l) \subseteq \mathcal{M}(i) \cup \mathcal{M}(j),$$

if $\mathcal{M}(l) = \mathcal{M}(i)$ then $\mathcal{M}(k)$ must also equal $\mathcal{M}(i)$,

if $\mathcal{M}(k) = \mathcal{M}(j)$ then $\mathcal{M}(l)$ must also equal $\mathcal{M}(j)$.

If the shortest path has finite cost, create arc $(i', j') \in \mathcal{A}^4$ with cost $c_{i', j'}^4$, equal to the shortest path cost. If no feasible shortest path exists, then no arc (i', j') is created for \mathcal{A}^4 .

Theorem 5 : *Given an optimal solution, y^* , to (P4), we can construct an optimal solution, x^* , to (P3).*

Proof : See Appendix C.

Problem (P4) represents the *canonical form* of the GTSP. The form has mutually exclusive nodesets each requiring exactly one visit and containing no intraset arcs (see Figure 7). The canonical GTSP offers considerable problem structure. Various approaches for solving canonical form GTSP's can be found in [3, 5, 6, 7, 10, 14, and 15].

2.3 Transformation to the Standard TSP

The last stage transforms a canonical form GTSP to a clustered TSP. The clustered TSP is similar to the standard TSP except the nodes are grouped into predefined mutually exclusive and exhaustive node *clusters*. A feasible tour for the clustered TSP must visit all nodes of a cluster before visiting the nodes of any other cluster.

Definition: **Problem (P5)** is a clustered TSP defined over nodes $\mathcal{N}^5 = \mathcal{N}^4$. A set of nodes in \mathcal{N}^4 , \mathcal{S}_i , corresponds to a cluster of nodes in \mathcal{N}^5 , \mathcal{C}_i . For each nodeset (or cluster), we assume its member nodes have a given ordering. Let i^1, i^2, \dots, i^r be the nodes of nodeset \mathcal{S}_i (or cluster \mathcal{C}_i) with $r = |\mathcal{S}_i| = |\mathcal{C}_i|$. The arcset \mathcal{A}^5 is defined as follows.

Within each cluster i with $r = |\mathcal{C}_i| > 1$, create intracluster arcs forming a single directed cycle according to its given ordering. Hence, create arcs $(i^1, i^2), (i^2, i^3), (i^3, i^4), \dots, (i^{r-1}, i^r), (i^r, i^1)$ in \mathcal{A}^5 . Let each of these intracluster arcs have cost equal to zero, that is, $c_{i^1 i^2}^5 = \dots = c_{i^r i^1}^5 = 0$.

For each interset arc $(i^j, k^l) \in \mathcal{A}^4$, with $j > 1$, create an arc $(i^{j-1}, k^l) \in \mathcal{A}^5$ with identical cost, hence, $c_{i^{j-1} k^l}^5 = c_{i^j k^l}^4$. For each interset arc $(i^1, k^l) \in \mathcal{A}^4$ with $r = |\mathcal{C}_i|$, create an arc $(i^r, k^l) \in \mathcal{A}^5$ with cost $c_{i^r k^l}^5 = c_{i^1 k^l}^4$.

Problem (P5) is a clustered TSP with a special structure. Since the only intracluster arcs are the zero cost arcs which form a directed cycle, it is easy to see that any feasible clustered TSP solution will traverse $|\mathcal{C}_i| - 1$ of the intracluster arcs of \mathcal{C}_i . See Figure 8.

Lemma 6 : *Problem (P5) is equivalent to problem (P4).*

Proof : See Appendix D.

Once the problem is in the form of (P5), it can be solved as a clustered TSP as in [4] or we can use an approach equivalent to that of Chisman [1] to transform a clustered TSP into a standard TSP by simply adding a large cost to all the intercluster arcs, as follows. See Figure 9.

Definition: **Problem (P6)** is a standard TSP over nodes $\mathcal{N}^6 = \mathcal{N}^5$ and arcs $\mathcal{A}^6 = \mathcal{A}^5$.

The arc costs for (P6) are computed as follows. Let

$$c_{ij}^6 = c_{ij}^5 + \beta \text{ if } i \text{ and } j \text{ belong to different clusters}$$

$$c_{ij}^6 = c_{ij}^5 \text{ if } i \text{ and } j \text{ belong to the same cluster.}$$

$$\text{where } +\infty > \beta > \sum_{(i,j) \in \mathcal{A}^5} c_{ij}^5.$$

Our main result for transforming the canonical GTSP to a standard TSP easily follows.

Theorem 7 : *Given a canonical GTSP in the form of (P4) with m nodesets, we can transform the problem into a standard TSP in the form of (P6). Given an optimal solution y^* to (P6) with $c^6 y^* < (m + 1)\beta$, we can construct an optimal solution x^* to (P4).*

Combining Theorems 4, 5 and 7, we can transform a GTSP into a TSP, solve, and transform the TSP solution to a solution for the original GTSP.

3.0 Summary and Conclusion

The results in Section 2 show how to transform any problem modeled as a GTSP into a standard TSP. The number of digraph nodes is increased according to the number of multiple membership nodes in the original problem. Given a problem with m nodesets defined over an n -node digraph, the transformation can yield a TSP with at most mn nodes. Along with the increased number of nodes, there is a corresponding increase in the number of arcs. The effort required throughout the transformation is worst-case polynomially bounded by $O(m^2 n^4)$. But since the worst case bound is based on complete nodeset membership overlap, the transformation effort for real problems will likely be much less. One practical complication with these transformations is that arc costs can become very large. This is of no theoretical importance, but may cause stability problems in some solution techniques.

The resulting TSP has an arc and arc cost structure that may cause difficulties for traditional assignment based methods for solving asymmetric TSP's. Specifically, the directed zero cost intracluster cycle will cause a simple subtour elimination algorithm to branch m levels before the first non-zero lower bound might be reached. This, along with the large arc cost values and the potential for degeneracy, suggests that approaches for the TSP based on integer linear programming [12] may be more appropriate for optimally solving the transformed problem.

An area for further research is to understand the extent to which the facet results for the TSP polytopes ([8], chapter 8) can be related to the canonical GTSP or even to the general form of the problem. This may allow us to define valid inequalities for complex routing problems which involve both sequencing and selection decisions.

APPENDIX A

The proof of Lemma 1 follows the development of Lemmas 8, 9, and 10.

Lemma 8 : *Given a feasible solution, y , to problem (P1), there exists a feasible solution to (P0), x , with cost $c^0 x = c^1 y$.*

Proof : We know that for an arc (i', j') in y , there exists a path with cost $c_{i', j'}^1$ between i and j of \mathcal{N}^0 which passes through only the nodesets to which i and j belong. Therefore, given y , we can construct a feasible solution x as a collection of paths corresponding to the arcs in y . ■

Lemma 9 : *Given a feasible solution, x , to (P0) expressed as a sequence of nodes from i to i as $\{i, \dots, j, \dots, k, \dots, i\}$, there exists a corresponding subsequence, $y = \{j', k', \dots, j'\}$, such that for any two consecutive nodes, j' and k' , $\mathcal{M}(j') \not\supseteq \mathcal{M}(k')$.*

Proof : The subsequence can be constructed as follows. Start checking x in sequence until a node j is found such that $\mathcal{M}(i) \not\supseteq \mathcal{M}(j)$. We are guaranteed that such a node will be found since node i cannot belong to all nodesets. Designate j' to begin the subsequence y . From j , check in a similar manner until some node k (or i) is found such that $\mathcal{M}(j) \not\supseteq \mathcal{M}(k)$. Again, we are guaranteed that such a node will be found since j cannot belong to all nodesets. Let k' follow j' in the subsequence. Continue checking x in sequence. Whenever a node is checked which belongs to at least one new nodeset, add its equivalent node to y . Continue until finally node i is checked. Since the nodeset memberships of \mathcal{N}^0 and \mathcal{N}^1 are the same, $\mathcal{M}(j) \not\supseteq \mathcal{M}(k)$ implies $\mathcal{M}(j') \not\supseteq \mathcal{M}(k')$. ■

Lemma 10 : *Given a feasible solution, x , to (P0), there exists a feasible solution, y , to (P1) with cost $c^1 y \leq c^0 x$.*

Proof : We can write out the solution x as a sequence of nodes. From Lemma 9, we know there exists a subsequence y such that for any two consecutive nodes, i' and j' , $\mathcal{M}(i') \not\supseteq \mathcal{M}(j')$ and, by definition, there exists an arc $(i', j') \in \mathcal{A}^1$ with cost $c_{i', j'}^1$. If nodes i and j are not consecutive within the sequence x , then the nodes between them

belong to, at most, the nodesets defined by $\mathcal{M}(i)$. The cost of the path from i to j over the sequence x will be greater than or equal to c_{ij}^1 . ■

Proof of Lemma 1 : From Lemma 8, we know we can construct x^* which is feasible and $c^1 y^* = c^0 x^*$. Suppose x^* was not optimal and there was another tour x such that $c^0 x < c^0 x^*$. From Lemma 10 we know there would exist a (P1) tour y such that $c^1 y \leq c^0 x$. This would imply $c^1 y \leq c^0 x < c^0 x^* = c^1 y^*$ and, hence, y^* would not be optimal. By contradiction, x^* is optimal for (P0). ■

APPENDIX B

The proof of Lemma 3 follows the development of Lemmas 11 and 12.

Lemma 11 : *Given a feasible solution to (P2), x , there exists a feasible solution to (P3), y , with $c^2x \geq c^3y$.*

Proof : We can write out the solution x as a sequence of nodes from node i to itself as

$\{i, j, \dots, k, l, \dots, p, \dots, q, \dots, i\}$. Construct a sequence y as follows. Choose some $P \in \mathcal{M}(i)$, and start y at i^P , i.e. $y = \{i^P, \dots\}$. Continue building the sequence y by adding nodes i^Q , for all $Q \in \mathcal{M}(i)$, $Q \neq P$, i.e. $\{i^P, i^Q, \dots, i^R, \dots\}$, $Q \in \mathcal{M}(i)$, $R \in \mathcal{M}(i)$. By definition $\mathcal{M}(i) \not\supseteq \mathcal{M}(j)$ so there exists some nodesets to which j belongs but i does not. Continue the sequence by adding j^T , for all nodesets T such that $T \in \{[\mathcal{M}(j) \cup \mathcal{M}(i)] - \mathcal{M}(i)\}$, i.e. $\{i^P, i^Q, \dots, i^R, j^T, \dots, j^U, \dots\}$, $T, \dots, U \in \{[\mathcal{M}(j) \cup \mathcal{M}(i)] - \mathcal{M}(i)\}$. Continue building the sequence, in general, by adding nodes l^V for all V such that $V \in \{[\mathcal{M}(l) \cup \mathcal{M}(i) \cup \mathcal{M}(j) \dots \cup \mathcal{M}(k)] - \mathcal{M}(i) - \mathcal{M}(j) \dots - \mathcal{M}(k)\}$. The construction will end at some node p^W such that $W \in \mathcal{M}(p)$, and W is the last nodeset not already visited, i.e., $\{i^P, i^Q, \dots, i^R, j^T, \dots, j^U, \dots, l^V, \dots, p^W, i^P\}$. The sequence cannot include more than m nodes since each node belongs to only one set and the construction prevents visiting nodesets already visited.

All arcs of the sequence y of the form (i^P, i^Q) , exist and have a cost of zero. The arcs between nodes created from different nodes, such as (i^R, j^T) , exist and have cost $c_{i^R, j^T}^3 = c_{i, j}^2$. Hence, the arc existence and arc costs are the same as that for all arcs of x until we reach the string of nodes $\{p, \dots, i\}$. If p immediately precedes i , the equivalence holds since $c_{p^W, i^P}^3 = c_{p, i}^2$. If p does not immediately precede i , then a node q between p and i can, at most, belong to the union of the nodesets of p and i , i.e., $\mathcal{M}(q) \subset \{\mathcal{M}(p) \cup \mathcal{M}(i)\}$. If q belonged to a nodeset to which neither p nor i belonged, then the nodeset would be separately visited twice in the sequence x and would contradict our assumption that x is feasible. Since all nodes between p and i belong to the nodesets of p and i , there will exist an arc $(p^W, i^P) \in \mathcal{A}^3$ with cost c_{p^W, i^P}^3 equal to the shortest path over nodes which will include those of the sequence between p and i . The shortest

path cost will be less than or equal to the path taken between p and i of the sequence x .

Hence, $c^2x \geq c^3y$. ■

Lemma 12 *For any feasible solution, y , to (P3) with $c^3y < (m + 1)\alpha$, there exists a feasible solution, x , to (P2) with $c^2x = c^3y$.*

Proof : The solution y enters each nodeset exactly once and consists of arcs with cost zero or some amount greater than or equal to α . Since c^3y is less than $(m + 1)\alpha$, any nodes of the form i^P and i^Q , with $P, Q \in \mathcal{M}(i)$, must be adjacent in the tour y , otherwise an extra cost of α would be incurred and cause c^3y to be greater than or equal to $(m + 1)\alpha$.

We can construct a solution x by including arcs corresponding only to those arcs of y having the costs greater than zero (exactly the reverse of the construction in the proof of Lemma 11). ■

Proof of Lemma 3 : From Lemma 12, we know that we can construct x^* which is feasible and $c^3y^* = c^2x^*$. Suppose x^* was not optimal and there was another tour x such that $c^2x < c^2x^*$. From Lemma 11, we know there would exist a (P3) tour y such that $c^3y \leq c^2x$. This would imply $c^3y \leq c^2x < c^2x^* = c^3y^*$ and y^* would not be optimal. By contradiction, x^* is optimal for (P2). ■

APPENDIX C

The proof of Theorem 5 follows the development of Lemmas 13, 14, 15, and 16.

Lemma 13 : *Given a feasible solution, x , to (P3) expressed as a finite sequence of nodes $\{i, \dots, j, \dots, k, \dots, i\}$, there exists a corresponding subsequence $y = \{i', \dots, j', k', \dots, i'\}$ in (P4) which visits exactly one node in each nodeset.*

Proof : We can construct the subsequence with the following rule. For every nodeset, choose one member node from x and include it in y . ■

Lemma 14 : *Given two consecutive nodes of the subsequence y , $j' \in S_J$ and $k' \in S_K$, if the same nodes j and k are not consecutive in the sequence x , then any node between j and k must belong to either S_J or S_K .*

Proof : Suppose the path between j and k includes some node $l \in S_L$. Since l' was not between j' and k' in the subsequence, then S_L is visited in some other part of the sequence x . This would imply that S_L is visited twice in the sequence x . By contradiction, l must belong to either S_J or S_K . ■

Lemma 15 : *Given a feasible solution to (P3), x , we can construct a feasible solution to (P4), y , such that $c^4 y \leq c^3 x$.*

Proof : From Lemma 13, we know there exists a subsequence y which visits exactly one node in each nodeset. Given two consecutive nodes of y , j' and k' , Lemma 14 establishes that there exists a feasible path between j and k which passes through only the nodesets of j and k . Since $c_{j',k'}^4$ is equal to the shortest path through the nodes between j and k , then the total costs incurred on the sequence x from j to k must be greater than or equal to $c_{j',k'}^4$. Hence, in total, $c^4 y \leq c^3 x$. ■

Lemma 16 : *Given a feasible solution to (P4), y , we can construct a feasible solution, x , to (P3) with $c^4 y = c^3 x$.*

Proof : We can write out the feasible tour y as a finite sequence of nodes $\{i', \dots, j', k', \dots, i'\}$. Since y is feasible, each arc (j', k') exists and has cost $c_{j'k'}^4$. Given arc (j', k') , we know that there exists a feasible shortest path from j to k which may pass through only the nodesets of j and k , and the cost of this path is $c_{j'k'}^4$. Simply construct x as the collection of these shortest paths defined by consecutive nodes of y . Since the cost of each arc of y is equal to the cost of the corresponding path in x , $c^4 y = c^3 x$. ■

Proof of Theorem 5: From Lemma 16, we can construct x^* which is feasible and $c^4 y^* = c^3 x^*$. Suppose x^* was not optimal and there was another tour x such that $c^3 x < c^3 x^*$. From Lemma 15 we know there would exist a (P4) tour y such that $c^4 y \leq c^3 x$. This would imply $c^4 y \leq c^3 x < c^3 x^* = c^4 y^*$ and, hence, y^* would not be optimal. By contradiction, x^* is optimal for (P3). ■

APPENDIX D

The proof of Lemma 6 follows the development of Lemma 17.

Lemma 17 : *A feasible solution to (P5) which enters cluster C_i through node i^j with $j \neq 1$, must leave the cluster from node i^{j-1} . A feasible solution to (P5) which enters cluster C_i through node i^1 , must leave the cluster from node i^r where $r = |C_i|$.*

Proof : Omitted.

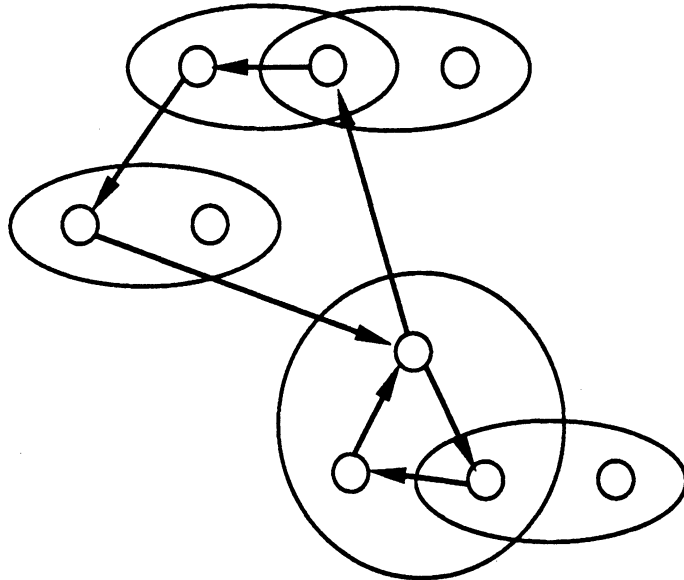
Proof of Lemma 6: Given a feasible solution to (P4), x , expressed as a sequence of nodes $\{i^j, k^l, \dots, p^q, i^j\}$, we can construct a solution y to (P5) as $y = \{i^j, i^{j+1}, \dots, i^{j-1}, k^l, k^{l+1}, \dots, k^{l-1}, \dots, p^q, p^{q+1}, \dots, p^{q-1}, i^j\}$. The solution y is feasible for (P5) by definition of a clustered TSP and by construction of \mathcal{A}^5 . Each intercluster arc of y has a cost equal to a corresponding interset arc of x . Hence, $c^5 y = c^4 x$.

A feasible solution to (P5), y , must pass through the clusters as specified in Lemma 17. This means if intercluster arc (i^j, k^l) is used in y , y must include an intercluster arc into node i^{j+1} and an intercluster arc out of node k^{l-1} . We can construct a feasible solution to (P4), x , as follows. For every intercluster arc (i^j, k^l) in y , include arc (i^{j+1}, k^l) in x . By definition, $c_{i^{j+1}k^l}^4 = c_{i^j k^l}^5$, so $c^4 x = c^5 y$. ■

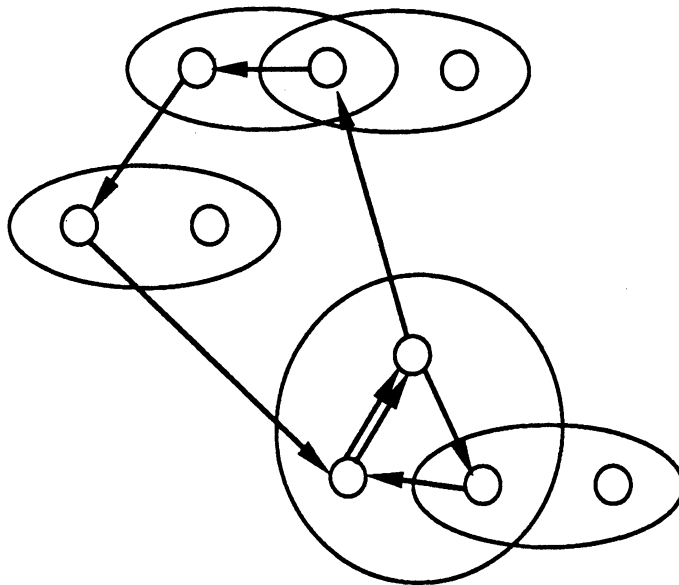
REFERENCES

- [1] J. Chisman, "The Clustered Traveling Salesman Problem," *Computers and Operations Research* 2 (1975) 115-119.
- [2] J. Current and D. Schilling, "The Covering Salesman Problem," *Transportation Science* 23(3) (1989) 208-213.
- [3] A.L. Henry-Labordere, "The Record Balancing Problem: A Dynamic Programming Solution of a Generalized Travelling Salesman Problem," *RIRO B-2* (1969) 43-49.
- [4] K. Jongens and T. Volgenant, "The Symmetric Clustered Traveling Salesman Problem," *European Journal of Operational Research* 19 (1985) 68-75.
- [5] G. Laporte and Y. Nobert, "Generalized Travelling Salesman Problem Through n Sets of Nodes: An Integer Programming Approach," *INFOR* 21(1) (1983) 60-75.
- [6] G. Laporte, H. Mercure and Y. Nobert, "Finding the Shortest Hamiltonian Circuit Through n Clusters: A Lagrangean Approach," *Congressus Numerantium* 48 (1985) 277-290.
- [7] G. Laporte, H. Mercure and Y. Nobert, "Generalized Travelling Salesman Problem Through n Sets of Nodes: The Asymmetrical Case," *Discrete Applied Mathematics* 18 (1987) 185-197.
- [8] E.L. Lawler, J.K. Lenstra, A.H. Rinnooy Kan and D.B. Shmoys, *The Traveling Salesman Problem* (Wiley, New York, 1985).
- [9] C.E. Noon, "The Generalized Traveling Salesman Problem," unpublished dissertation, Department of Industrial and Operations Engineering, The University of Michigan (Ann Arbor, 1988).
- [10] C.E. Noon and J.C. Bean, "A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem," College of Business Administration working paper No. 855, University of Tennessee (Knoxville, 1988).

- [11] C.E. Noon and L.A. Kaplan, "A TSP Transformation for the Multi-Depot MTSP," working paper, Department of Management, University of Tennessee (Knoxville, 1989).
- [12] M.W. Padberg and G. Rinaldi, "Optimization of a 532-City Symmetric Traveling Salesman Problem by Branch and Cut," *Operations Research Letters* 6 (1987) 1-8.
- [13] R.G. Parker and R.L. Rardin, "The Traveling Salesman Problem: An Update of Research," *Naval Research Logistics Quarterly* 30 (1983) 69-96.
- [14] J.P. Saksena, "Mathematical Model of Scheduling Clients Through Welfare Agencies," *CORS Journal* 8 (1970) 185-200.
- [15] S. Srivastava, S. Kumar, R.C. Garg and P. Sen, "Generalized Traveling Salesman Problem Through n Sets of Nodes," *CORS Journal* 7 (1969) 97-101.

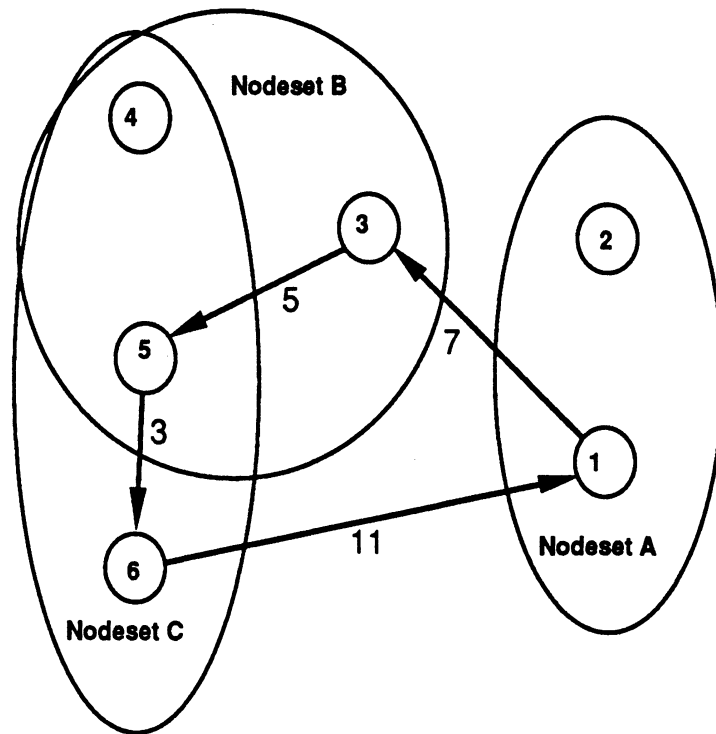


1(a) Feasible GTSP tour which passes through a node twice.



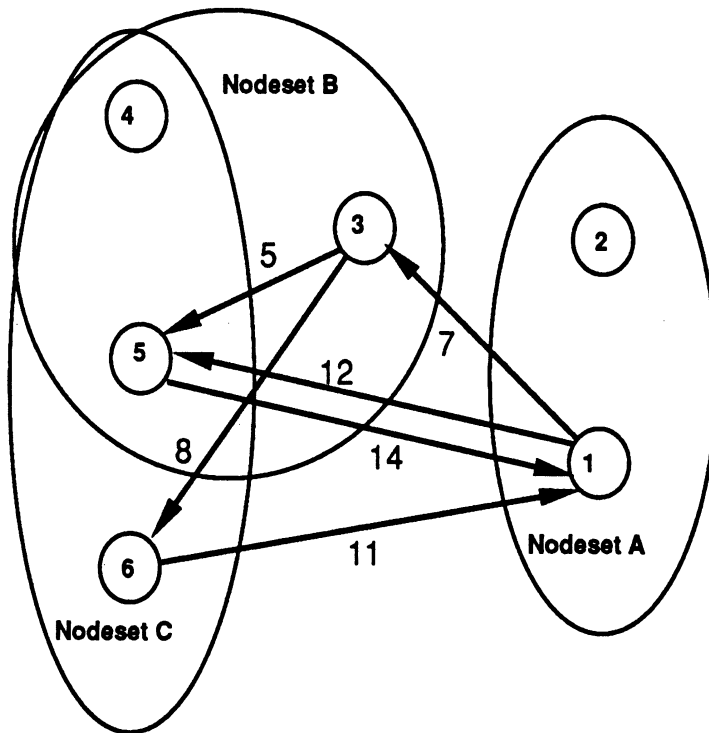
1(b) Feasible GTSP tour which passes through an arc twice.

Figure 1: Examples of feasible GTSP tours.



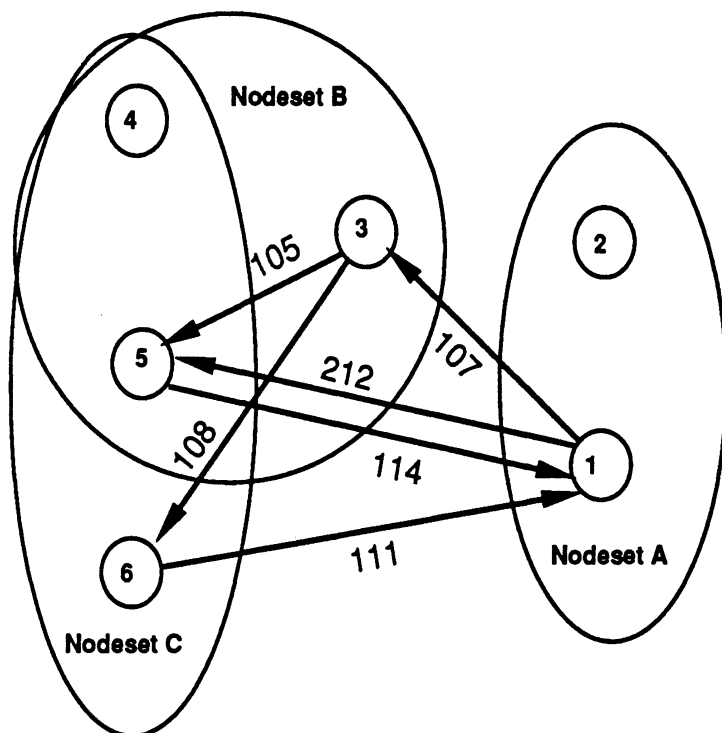
A feasible solution is given by the sequence 1-3-5-6-1 with cost of 26

Figure 2: Example problem (P_0).



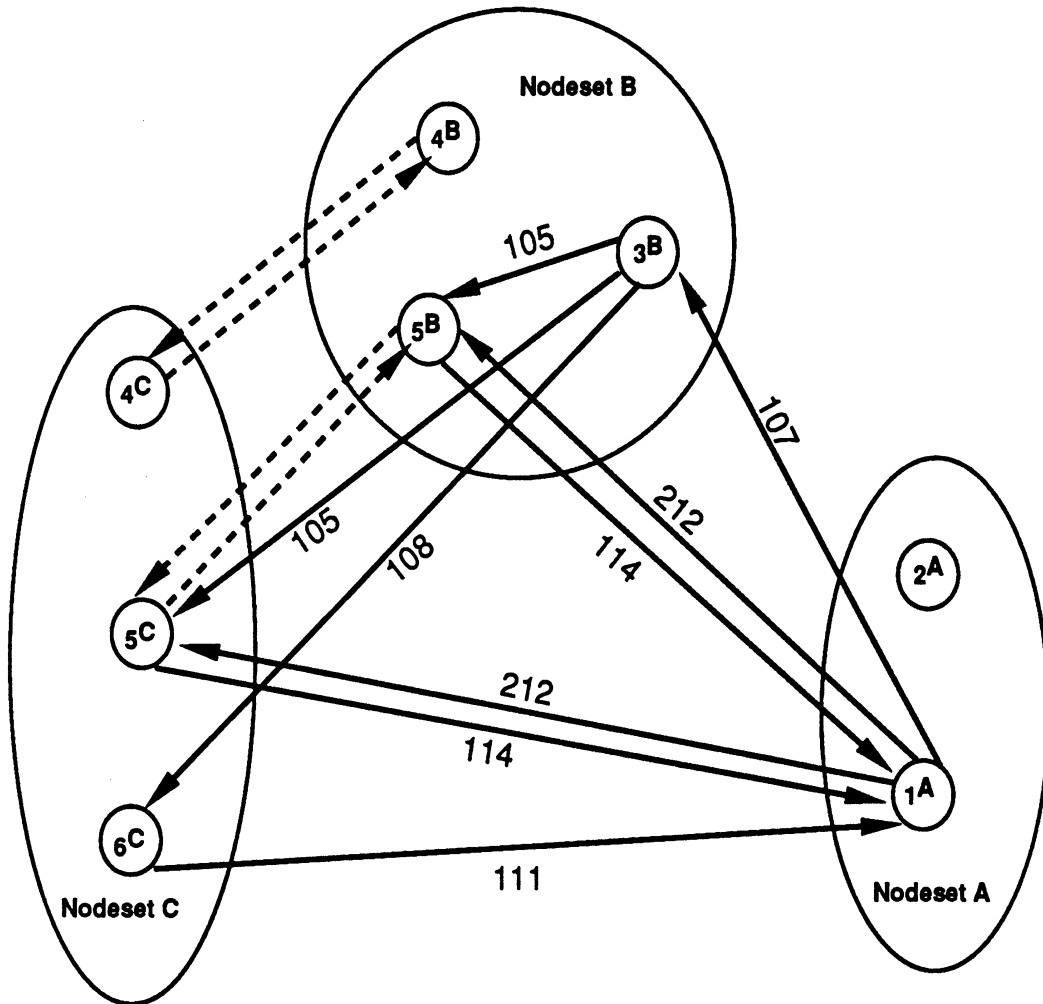
Each arc must enter at least one new nodeset.

Figure 3: Example problem ($P1$).



With $\alpha=100$, costs are adjusted according to the number of new nodesets an arc enters.

Figure 4: Example problem ($P2$).



For a node with multiple nodeset membership, we create duplicate nodes interconnected with zero cost arcs (dashed).

Figure 5: Example problem (P3).

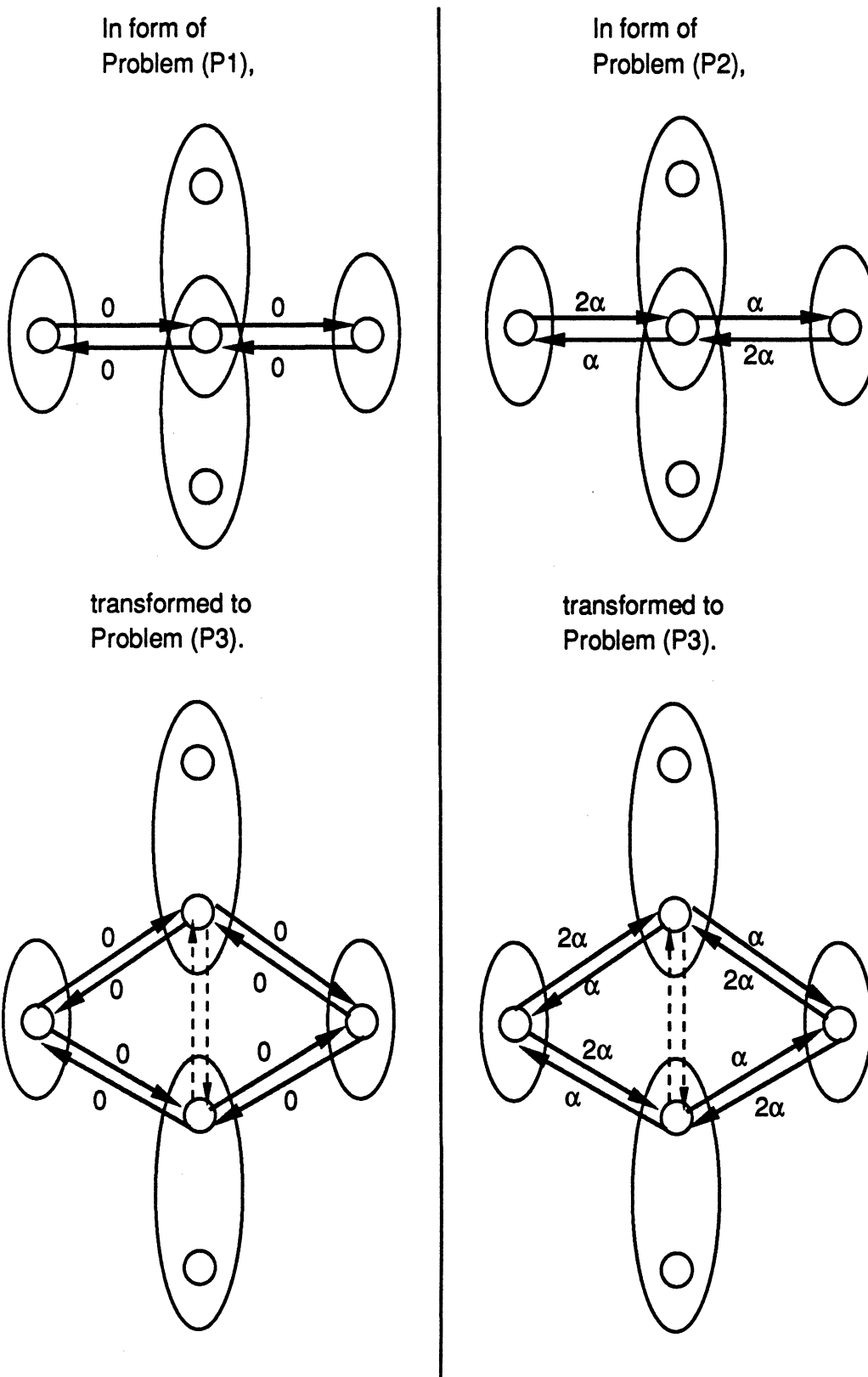
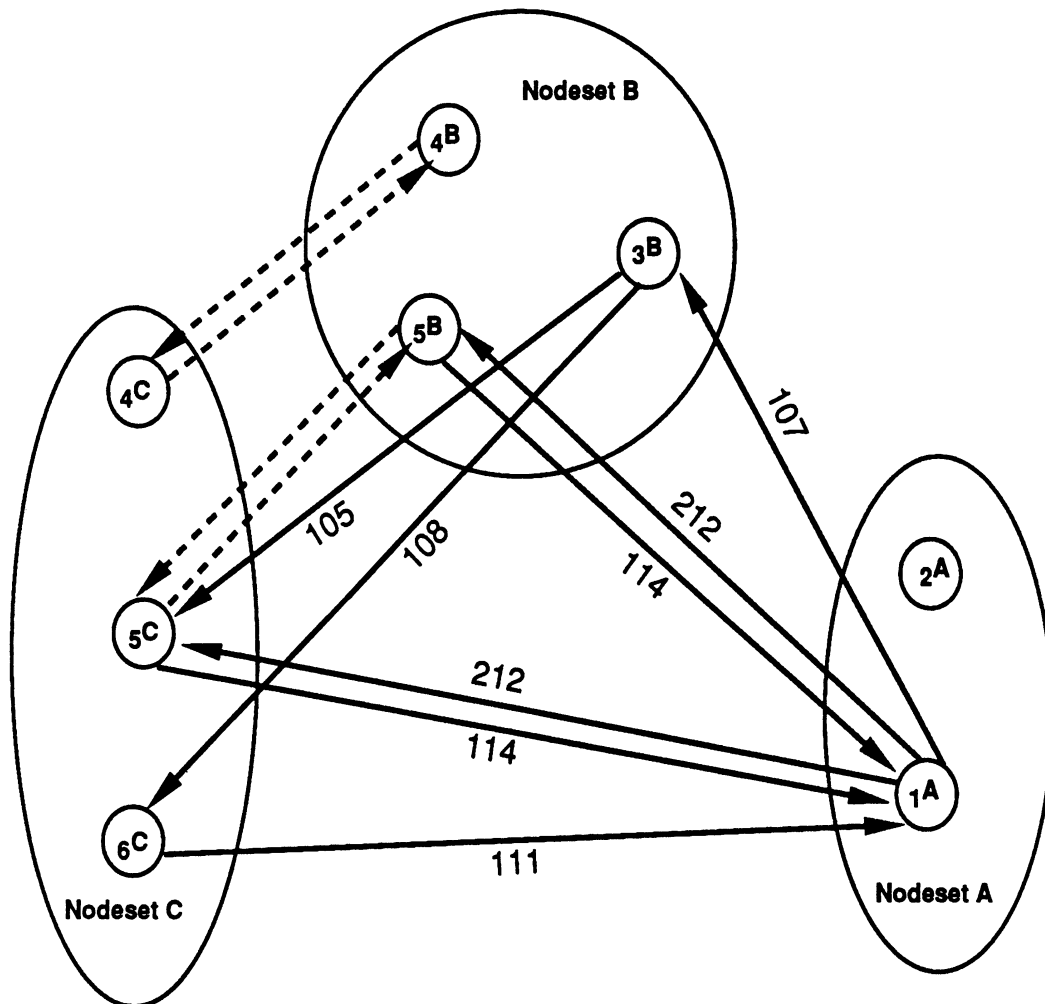
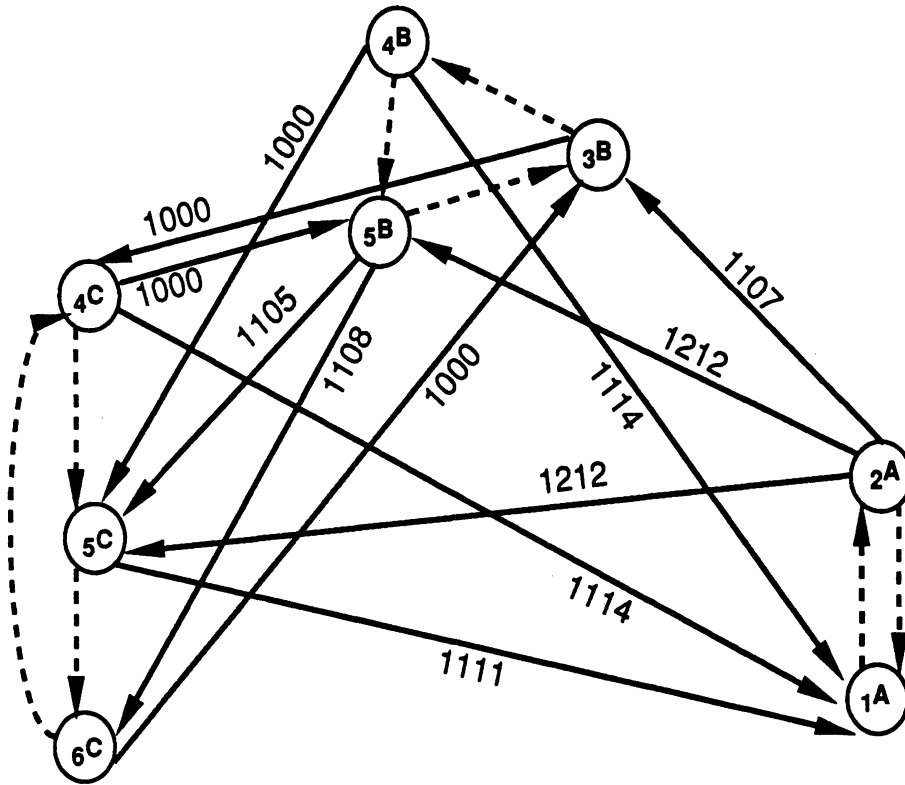


Figure 6: Example showing necessity of (P2) transformation.



Intraset arcs are eliminated by restricted shortest path computations.

Figure 7: Example problem (P4).



With $\beta=1000$, the clustered TSP can be transformed to a standard TSP. A feasible solution is given by the sequence $1^A-2^A-3^B-4^B-5^B-5^C-6^C-4^C-1^A$ with total cost = $m(\alpha+\beta) + 26 = 3326$.

Figure 9: Example problem (P6).

UNIVERSITY OF MICHIGAN



3 9015 04732 7005