SELECTION OF RELAXATION PROBLEMS FOR A CLASS OF ASYMMETRIC TRAVELING SALESMAN PROBLEM INSTANCES

Seiji Kataoka Susumu Morito National Defense Academy Waseda University

(Received March 2, 1990; Revised March 25, 1991)

Abstract It is commonly believed that the use of the assignment problem works well when one selects a relaxation problem within the framework of a branch and bound algorithm to solve an asymmetric traveling salesman problem (TSP) optimally. In this paper, we present asymmetric TSP instances found in real-life setting, and show that the above common belief is not necessarily appropriate. Based on the real-life example, a family of asymmetric TSP instances called SLOPE is considered, which are generated by *deforming* arc lengths of standard two-dimensional TSPs on a plane in a specific manner.

For this type of instances, we show that the assignment relaxation yields poor performance, and propose a minimum 1-arborescence relaxation similar to the minimum 1-tree relaxation that has been successfully applied to the symmetric TSPs. In order to make the algorithm more efficient, the proper selection of a root node and the determination of Lagrange multipliers to increase the lower bounds are explored.

Computational experiments with instance SLOPE and also with the real-life instance show that the proposed algorithm gives better computational performance than the algorithm with the assignment relaxation.

1 Introduction

Most papers in the field of mathematical programming refer to applications of the models discussed. Presentation and discussion of computational experiments are often regarded as necessary ingredients of research on algorithm development, and in fact many papers give results of computational experiments. It is rather rare, however, to find papers which present computational results based on real-life instances. Most studies somehow "generate" instances artifically to evaluate algorithms.

Nothing is wrong to use such randomly generated instances, and this would be necessary for the systematic evaluation of proposed algorithms. However, algorithm evaluation could sometimes be based on experiments with "improper" instances. It would be important to perform algorithm evaluation which reflects characteristics of instances which arise in real-life situations.

For those who solve problems in real-life using mathematical programming, there is no question that the proper choice of models and algorithms is crucial. There exist some guidelines which should be followed to come up with a proper choice of models and algorithms, but more information would be desired for their proper choice. This paper considers the popular traveling salesman problem (TSP) for which a huge number of algorithms have been proposed and evaluated, and shows that the commonly believed guideline for selecting a relaxation problem within a branch and bound scheme is not necessarily appropriate. Specifically, the standard guideline for the selection of a relaxation problem recommends to use "the assignment problem if the instance is asymmetric, whereas the minimum 1-tree if symmetric". See, e.g., [6], [7], [9], [11, p.370, p.378, p.392], [12], [13].

This paper presents a practical problem scenario which could be modeled as an asym-

metric TSP, and shows that the assignment relaxation does not produce good performance for the particular type of instances, presenting a case where a commonly-believed guideline leads us to an erroneous decision on algorithm design.

It is well-known for combinatorial optimization problems that computational time may vary substantially even for the instances of the identical size, and also that one may encounter exceptionally difficult instances by chance. However, if a specific class of instances is found to be difficult for some reasons, then it is desired to find a proper algorithm to deal with the class of instances. It is shown that a particular type of asymmetric TSP instances could better be solved by the minimum 1-arborescence relaxation which is proposed in this paper.

This paper is organized as follows. Section 2 describes the real-life problem scenario which motivated our study, and then defines a general class of asymmetric TSP instances called SLOPE. Computational results are given which show that the assignment relaxation produces poor performance for the real-life instance. Similar results are presented for a subclass of instances within SLOPE. The next section proposes the use of the 1-arborescence relaxation to solve SLOPE instances efficiently. To obtain stronger lower bounds, 1)the choice of a root node for the 1-arborescence relaxation, and 2)the proper determination of Lagrange multipliers to increase the lower bounds, are explored. Section 4 demonstrates computational results which indecate the effectiveness of the proposed algorithm to solve the difficult class of SLOPE instances and the real-life instance.

In Appendix A, the real-life instance from which the proposed family of asymmetric TSPs originate is given. Appendix B shows a different type of "deformed" 2-dimensional TSPs, for which computational results similar to instances SLOPE can be observed. They imply the existence of a rather large class of asymmetric TSPs in the real world which are some "modifications" of the Euclidean TSPs. Finally, Appendix C gives the description of a heuristic algorithm to find an initial tour.

2 Real-Life Problem and Modeling—Instance SLOPE

2.1 Real-Life problem

Assume that at a factory, there exist several distinct products to be produced. Production of these products will be performed on a machine, and there are two parameters, for example, temperature and product size, that basically determine the production process. The temperature and size specifications are assumed to be mutually independent. As for temperature, it is desirable to go from high to low as much as possible. There is cost incurred when temperature is changed in either directions, but the cost is substantially higher when it is raised. So there is a penalty, so to speak, to raise the temperature. Sizes, on the other hand, may go from large to small, or vice versa, yet, frequent size changes are certainly undesirable, say, due to required setup losses. Thus just sorting products in the decreasing order of temperature specifications would not generally give a good production sequence. In essence, we seek a production sequence so that changes in both temperature and sizes are as "smooth" as possible, with a particular attention given to temperature raises.

One way of formulating this problem is to plot products on a two-dimensional plane with axes corresponding to temperature and size, somehow assign "cost" of moving from product i to j, and finally find a path which visits all points with the minimum cost. Then, this problem can be formulated as the Traveling Salesman Problem (TSP).

With regard to the "cost" of moving between two products (points), it would be reasonable to assume that the basis for the cost assignment is the (Euclidean) distance between the points. For temperature, however, extra penalty is accessed when it goes from low to high. Because of this "directionality", the resultant "cost" matrix becomes asymmetric, and thus we have an instance of an asymmetric TSP:

235

Minimize $\{C(T) \mid T \in \mathcal{T}\}$

Here T is a set of tours that visit all the points exactly once and C(T) is the sum of arc cost on a tour T.

This problem originates from a rael-life problem which arose at a steel plant. Note that this type of problm is expected to exist widely for many sequencing problems in productions and other settings. Numerical data of one real-life instance with 35 products are shown in Appendix A.

To solve the particular asymmetric instance, we first applied a branch and bound algorithm based on the assignment relaxation, and the results of computational experiment are given in Table 4-2. Fundamentally, our program is based on a FORTRAN translation of a BASIC program given in Kobayashi [8, pp.107-128]. To make it more efficient and to get stronger lower bounds, at each subproblem, subroutine AP given in Carpaneto, Martello and Toth [5] and the Balas and Christofides bounding procedures [1] are coded and incorporated into the algorithm.

Throughout this paper, time shown corresponds to CPU seconds on APOLLO DOMAIN Series 4000. The 35-product instance required 3030831 branches and the 971223 CPU seconds. The poor performance of the algorithm motivated us to perform this study.

2.2 Modeling and instance SLOPE

We now define a family of Asymmetric TSP (ATSP) instances, referred to as instances SLOPE, which take two-dimensional Symmetric TSPs (STSPs) on a plane and *deform* their distances of a particular direction $p(\geq 1)$ times as described below. To get an intuitive feel, think of a situation where one moves around a ski slope, from which the above name originated. (Refer to Figures 2-1,2-2.)

Points are originally given on a two-dimensional plane (x, y). The x and y coordinates



Figure 2-1: A geometric image of instance SLOPE

of point i, $(x_i, y_i), i \in V$, where V is node set, are assumed to be known and are sorted in nondecreasing order of x axe. Assume also that these points are randomly distributed on a square. "Distances" between two points are evaluated as follows:

Copyright © *by ORSJ. Unauthorized reproduction of this article is prohibited.*



Figure 2-2: Calculation of arc length

(2.1)
$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$
 if $x_i \ge x_j$

(2.2)
$$c_{ij} = \sqrt{p^2 (x_i - x_j)^2 + (y_i - y_j)^2} \qquad if \quad x_i < x_j$$

Note that we "penalize" a move from smaller x_i to larger x_j when p > 1. In reality, a starting node was fixed to the node at the top of the slope and it was not required to come back to the starting node. To conform with this practical requirement, we fix the starting node and also add a dummy node D.

Thus, if the original instance consists of N(N = |V|) products, then the instance to solve becomes an N + 1 node problem. For example, the real-life instance with 35 products becomes 36-node instance. Define the lengths of arcs that are incident to D, as in (2.3) and (2.4).

(2.3)
$$c_{Di} = \infty$$
 for all i except $c_{D1} = 0$

$$(2.4) c_{iD} = 0 for all i$$

Note that node 1 corresponds to the node located at the top of the slope, since nodes are numbered in nondecreasing order of the values of x axe.

2.3 Applications of ATSP algorithms to instance SLOPE

The performance of a standard branch and bound algorithm with the assignment relaxation and with the Balas and Christofides (BC) procedures is examined for instances SLOPE. We parametrically change the number of nodes N (Throughout the paper, N stands for the number of nodes excluding the dummy node.) and slope parameter p, and study their effects on the total number of branches(NB), the ratio of the optimal value of the assignment relaxation of the original problem to its optimal value (AP/OPT), and the ratio of the value of lower bound improved by the BC procedures to the optimal value (BC/OPT), and CPU time. Values of p examined range from 1.0 to 2.0 with an increment of 0.2, together with 3.0, 5.0 and 10.0.

IV	<i>p</i>	NB	AP/OPT	BC/OPT	CPU
	1.0	35.85	0.7883	0.9493	0.314
	1.2	34.45	0.7914	0.9357	0.297
	1.4	31.35	0.8002	0.9348	0.275
	1.6	30.80	0.8095	0.9352	0.268
10	1.8	28.20	0.8164	0.9321	0.249
	2.0	27.65	0.8210	0.9251	0.248
	3.0	15.45	0.8584	0.9348	0.161
	5.0	5.10	0.9017	0.9545	0.089
	10.0	4.30	0.9237	0.9499	0.080
	RND	7.85	0.9600	0.9711	0.087
	1.0	97.35	0.8242	0.9543	1.109
	1.2	97.30	0.8314	0.9478	1.097
	1.4	83.85	0.8370	0.9413	0.949
	1.6	69.00	0.8414	0.9380	0.776
12	1.8	97.45	0.8468	0.9291	1.054
	2.0	92.00	0.8598	0.9252	0.986
	3.0	37.80	0.8685	0.9174	0.465
	5.0	11.25	0.8988	0.9408	0.194
	10.0	4.30	0.9296	0.9598	0.118
	RND	17.25	0.9471	0.9619	0.215
	1.0	841.80	0.7987	0.9363	11.808
	1.2	848.45	0.8039	0.9308	12.091
	1.4	1023.95	0.8095	0.9237	14.400
	1.6	890.00	0.8148	0.9203	12.033
14	1.8	457.65	0.8170	0.9160	6.447
	2.0	369.95	0.8211	0.9130	5.208
	3.0	134.05	0.8500	0.9185	2.046
	5.0	32.50	0.8784	0.9210	0.593
	10.0	10.30	0.9122	0.9410	0.239
	RND	27.15	0.9487	0.9646	0.427
	1.0	1063.15	0.8080	0.9333	17.029
	1.2	1161.65	0.8132	0.9252	19.466
	1.4	852.60	0.8192	0.9208	13.731
	1.6	773.05	0.8233	0.9187	12.576
15	1.8	757.95	0.8258	0.9127	12.378
	2.0	691.25	0.8297	0.9077	11.280
	3.0	515.65	0.8421	0.8916	8.146
	5.0	99.10	0.8648	0.9073	1.828
	10.0	19.00	0.8921	0.9180	0.445
	RND	61.55	0.9612	0.9702	0.932
	1.0	4251.75	0.7967	0.9366	80.871
	1.2	4317.10	0.7993	0.9169	79,104
	14	30104.75	0.8035	0.9113	520.350
	16	35272 50	0.8067	0.9042	607 400
16	1.0	25150 55	0.8137	0.3042	444 000
10	20	18983.65	0.8135	0.8979	333 859
	3.0	723.00	0.0100	0.0010	13 980
	5.0	148.15	0.0220	0.0000	3 07/
	10.0	16.80	0.8747	0.0013	0 502
	RND	88.40	0.0141	0.0110	1 478
	L IVIND	00.40	0.9090	0.3143	1.4(0

Table 2-1: Experiments for instances SLOPE with the assignment relaxation

The preformance of the algorithm on random ATSPs of the corresponding size is also shown for comparisons. Those rows denoted as RND in Table 2-1 give results for N + 1 node random ATSPs in consideration of the added dummy node for SLOPE instances.

Values of x and y are generated as uniform real random numbers between 1 and 100. Distances between nodes are calculated by (2.1), (2.2) and for the dummy node as defined by (2.3), (2.4) whose fractional parts are truncated to make them integers. With regard to RND, the arc lengths are generated as uniform random numbers between 1 and 100. All figures reflect average results of 20 trials.

Table 2-1 summarizes the results from which the following observations can be made: 1) There is a general tendency that as p increases, NB decreases when p gets greater than a certain value. This is natural because instances become less symmetric as p increases, and because the algorithm tends to work well for totally random and totally asymmetric ATSPs, where each c_{ij} is independent. Note that the maximum NB occurs when p is between 1 and 2. We emphasize the fact that NB is not uniformly decreasing as p increases. This is clearly shown when N = 16.

2) The ratio AP/OPT increases almost uniformly as p increases.

3) The ratio BC/OPT deteriorates almost uniformly as p increases for the range of p less than or equal to 2, contributing to the increase of NB as well as CPU.

4) As N exceeds 15, NB increases rapidly.

5) It may be possible to view that instances RND roughly correspond to instances SLOPE with parameter p ranging between 3 and 10, from the standpoint of NB and CPU. Note, however, that with respect to the strength of lower bounds such as AP/OPT and BC/OPT, instances RND always yields stronger (i.e., higher) lower bounds than SLOPE, irrespective of values of p.

The above observations would imply that the guideline which recommends to "use the assignment relaxation for ATSPs" does not necessarily apply to all instances within the wide class of ATSPs. More specifically, one can recognize within a class of SLOPE instances a non-trivial subclass for which the superiority of the assignment relaxation is doubtful.

3 A Proposed Algorithm for Instance SLOPE

3.1 Minimum 1-arborescence relaxation

The computational experience shown in Section 2 indicates that the real-life instance and SLOPE instances presented in this paper have properties similar to those of STSPs. It then is natural to consider the introduction of the minimum 1-arborescence relaxation, i.e., the directed version of the minimum 1-tree relaxation which is known to work well for STSPs [6]. We thus develop in this section a branch and bound algorithm based on the 1-arborescence relaxation for the type of ATSPs which preserve some flavor of symmetry.

There exist few studies on the applications of the 1-arborescence relaxation to solve ATSPs. Smith [12] is the only one known to the authors to study such algorithms. He used random instances for his computational experiments and concluded that the assignment relaxation works better for ATSPs than the 1-arborescence relaxation. It is not clear, however, the assignment relaxation uniformly outperforms the 1-arborescence relaxation across the wide variety of asymmetric instances.

In Section 3.2, we discuss a method for selecting a root node of the 1-arborescence, and then discuss in Section 3.3 the determination of Lagrange multipliers.

For completeness, we give definitions of some basic terminologies. An arborescence rooted at node q is a spanning subgraph of a directed graph which contains no subtours, exactly one edge directed into every node except q. A value of an arborescence is a sum of its arc lengths, and a numinum arborescence is one whose value is minimum. A minimum 1-arborescence rooted at node q is a minimum arborescence rooted at node q, plus a minimumlength incoming arc to node q.

3.2 Choice of root node for the 1-arborescence relaxation

Generally, any node can be chosen as a root to give a lower bound when the minimum 1-arborescence relaxation is used. It then is natural to select a root node so that the resultant lower bound is higher. We can show that the strongest lower bound can be obtained by selecting node 1, the node at the top of the slope, as the root node.

Theorem 1

The value of a minimum 1-arborescence becomes the largest when node 1 is chosen as the root node under the conditions (2.3) and (2.4).

Proof

First we define the following.

arb(i) = the value of a minimum arborescence rooted at node i

one(i) = the value of a minimum 1-arborescence rooted at node i

Note that the dummy node D is included in each case.

For arbitrary node $q \ (\neq 1, D), c_{qD} = c_{D1} = 0$ from the definitions of (2.3) and (2.4), and thus

(3.1)
$$one(1) = arb(1) + c_{D1}$$
.

In one(1), we call the arc which goes into node q as arc a, and define the length of the arc a as c(a). Removing arc a from one(1), produces an arborescence rooted at node q (Figure 3-1), which implies

$$(3.2) arb(q) \le one(1) - c(a).$$

If we call the minimum-length incoming arc to node q as arc b, and define the length of the arc b as c(b), the definition of a minimum 1-arborescence implies

$$(3.3) one(q) = arb(q) + c(b).$$

From (3.2) and (3.3),

$$(3.4) \qquad one(q) \le one(1) - c(a) + c(b).$$

Since $c(b) \leq c(a)$, we obtain

$$(3.5) one(q) \le one(1).$$

Theorem 1 is valid even when some arcs that are not incident to dummy node D are fixed in a branch and bound procedure, and thus it is best to select node 1 as the root of a 1-arborescence throughout the process of a branch and bound procedure.

3.3 Improvement of lower bounds by the Lagrange relaxation

We now consider, based on Smith [13], how the Lagrange relaxation can be applied to instances SLOPE, where improvement of lower bounds is tried by putting the degree constraints of the 1-arborescence relaxation into the objective function.

To put those relaxed degree constraints into the objective function, we consider an arbitrary multiplier vector $\{\pi_i \mid i \in V\}$, and transform arc lengths c_{ij} into $c'_{ij} = c_{ij} + \pi_i$. The lower bounds L is obtained by subtracting $\sum_{i \in V} \pi_i$ from the value of a 1-arborescence based on



Figure 3-1: Arborescences rooted at node 1 and at node q

the transformed c'_{ij} . This lower bound forms a piecewise linear convex function with respect to $\{\pi_i \mid i \in V\}$, based on which we obtain a new and improved multiplier vector $\{\pi'_i \mid i \in V\}$ by the following formula:

(3.6)
$$\pi'_i = \pi_i + t(d_i - 1)$$
 $i \in V.$

Here $d_i (i \in V)$ denotes the out-degree of the 1-arborescence of node *i*. The step size *t* can be obtained by:

(3.7)
$$t = \lambda (U-L) / \sum_{i \in V} (d_i - 1)^2 \qquad 0 < \lambda \le 2,$$

where U stands for an upper bound for the value of the optimal tour.

Smith [13] states that the algorithm for the 1-arborescence cannot be used when some arc lengths are negative, and thus forces those Lagrange multipliers which become negative to be 0 as in (3.8):

(3.8)
$$\pi'_i = \max\{0, \pi_i + t(d_i - 1)\} \qquad i \in V.$$

Those nodes whose degree is zero should have negative Lagrange multipliers, but because of (3.8) the actual multipliers assigned by Smith's method are zeros. This leads to the solution of the 1-arborescence not increasing the out-degree of those nodes whose associated multiplier should have been negative. The restriction of nonnegativity of arc lengths in the 1-arborescence algorithm is noted also by Suzuki [14]. However, the following simple result holds:

Theorem 2

Given a graph, add a given constant M to all arc lengths. Then a minimum 1-arborescence for the original graph gives the same minimum 1-arborescence for the transformed one, and vice versa.

Proof

Suppose that G is the given graph and G_M a graph where a constant M is added to length of each arc on G. Let T be an arbitrary 1-arborescence, and z(T) the value of T on G. Let T^* be a minimum 1-arborescence. If N denotes the number of nodes in G, any 1-arborescence T consists of N arcs, and thus the value of T^* on graph G_M would be $z(T^*) + MN$. If there exists an optimal minimum 1-arborescence $T_M^* (\neq T^*)$ on G_M , then

(3.9)
$$z(T_M^*) + MN \le z(T^*) + MN,$$

which implies

(3.10)
$$z(T_M^*) \le z(T^*).$$

Unless the inequality in (3.10) holds, however, this contradicts that T^* is a minimum 1-arborescence on G. Therefore, T^* is also the same minimum 1-arborescence on G_M .

The above theorem shows that even if some arcs have negative lengths, one can add some (possibly big) constant M to all arc lengths to make them nonnegative, and apply a 1-arborescence algorithm. The value of the 1-arborescence is the optimal value obtained for the transformed graph minus MN.

To see the effect of adding constant M instead of forcing negative multipliers to 0 as done by Smith, the following experiments are performed:

1) λ in (3.7) is fixed to 1.25. This is based on the knowledge obtained from the experiments, where various values of λ are tested ranging from 0.25 to 2.0 with a step of 0.25.

2) An initial value of π is set to 0.

3) The termination criteria is as follows:

a. When lower bound exceeds or is equal to an upper bound.

b. Maximum of 100 repetitions if the above condition is not satisfied, where iterations correspond to the number of times π is updated.

With regard to the reason why the number of repetitions is limited to 100, we performed experiments on the real-life instance to see the effect of changing the limit, which showed that the upper limit of 100 repetitions outperforms the other cases with limits ranging from 40 to 160. Refer to computational results given in Table 4-2.

As preliminary computational experiments, lower bounds obtained by the following four cases are compared:

- (1) a proposed method based on Theorem 2
- (2) the Smith's method
- (3) the BC procedures
- (4) the assignment relaxation

Evaluations are made based on the ratio between the lower bounds obtained and the optimal value. The instance used is the real-life one in Appendix A. For this instance, the optimal value is known, which is used as an upper bound U in (3.7). The results are shown in Figure 3-2.

The results show that the Smith's method could not improve the lower bounds to the level of the BC lower bound even after 100 iterations, whereas the proposed method dominates the BC lower bound with approximately 20 iterations, and moreover, yields lower bounds which is around 99 % of the optimal value after roughly 80 iterations.

4 Computational Experiments

Computational experiments are performed to see the effectiveness of the proposed algorithm. First we describe the results for instances SLOPE, and then for the real-life instance.

4.1 Computational experiments with instances SLOPE

Experiments are performed under the following conditions:

1) A root node is defined to be node 1 as discussed in 3.2.

2) Lower bounds are improved by the Lagrange relaxation as discussed in 3.3. In the algorithm, an initial upper bound is calculated by the heuristic algorithm explained in Appendix C.



Figure 3-2: Increase of lower bounds

3) The branching rule is the same as that of Smith [13].

4) Problem sizes of N = 15,20,25 are tested.

Results are shown in Table 4-1. The performance measures shown include the number of branches(NB), the ratio of the value of the minimum 1-arborescence to the optimal value (ARB/OPT), the ratio of the final lower bound after the Lagrange relaxation to the optimal value (LAG/OPT), and CPU time. Moreover, Table 4-1 shows the total number of 1arborescence problems solved (TN1A) including the number of repetitions needed to increase the lower bounds with the Lagrange relaxation, and the average number of 1-arborescence problems per subproblem (AN1A), where AN1A = TN1A/NB. For comparisons, figures in parentheses provide corresponding results obtained from the algorithm given in Section 2. Dashes (-) indicates the fact that 20 instances could not be solved within the time limit of 3 days.

Following observations are in order from Table 4-1.

1) With respect to NB (i.e., the number of nodes or subproblems in a branch and bound tree), the proposed algorithm uniformly dominates the assignment relaxation with the only exception of N = 20, p = 10.

The 1-arborescence relaxation generally yields stronger lower bounds than the assignment relaxation. Specifically, in the range of $p \leq 2$ of instances SLOPE with N = 15, the lower bounds generated by the 1-arborescence relaxation (ARB/OPT) are stronger than those by the assignment relaxation (AP/OPT). For N = 20 and 25, the similar trend is expected, even though these ratios are unknown because too much computation time was required for these cases. Recall that instances SLOPE with smaller p(> 1) tend to be difficult. More importantly, the final lower bounds as measured by LAG/OPT are uniformly stronger than those generated by the assignment relaxation after the BC procedures. Obviously, this contributes to the superiority of the 1-arborescence relaxation to the assignment relaxation.

On the other hand, one can easily observe that the ratios AP/OPT as well as BC/OPT are very high for instances RND, indicating that instances RND are substantially different from instances SLOPE with even large values of p. This would imply that instances RND fall

N	p	NB		TN1A	ΔΝΙΔ	ARD/OPT	(AP/OPT)	LAGIORT	(BC(OBT)		anti
	1.0	18.00	(1063.15)	493.90	27.44	0.9023	(0.8080)	0 0707	(0.0222)	10 010	(17.020)
	1.2	25.05	(1161.65)	709.85	28.38	0.8898	(0.8132)	0.0702	(0.8333)	10.012	(17.029)
1	1.4	28.15	(852.60)	857.00	30 44	0.8753	(0.8102)	0.9792	(0.9202)	20.202	(19.450)
1	1.6	31.40	(773.05)	957.30	30 49	0.0700	(0.0192)	0.9801	(0.9208)	26.891	(13.731)
15	1.8	37.70	(757.95)	1193 20	31.65	0.8530	(0.8255)	0.9802	(0.9187)	28.313	(12.576)
	2.0	43.55	(691.25)	1355 60	31 13	0.8437	(0.0200)	0.9712	(0.9127)	33.875	(12.378)
	3.0	34.55	(515.65)	1007.00	21 70	0.0427	(0.8297)	0.9685	(0.9077)	36.920	(11.230)
	5.0	22.00	(99.10)	719 50	31.70	0.8002	(0.8421)	0.9709	(0.8916)	30.186	(8.146)
	10.0	13.80	(19.00)	430.10	32.70	0.7031	(0.8048)	0.9850	(0.9073)	17.709	(1.828)
	RND	50.05	(61.55)	439.10	00.62	0.7022	(0.8921)	0.9885	(0.9180)	9.908	(0.445)
	10	50.05	(01.00)	1478.00	29.53	0.7057	(0.9612)	0.8941	(0.9702)	37.270	(0.932)
1	1.0	67.05		2314.20	37.30	0.9050	(-)	0.9716	(-)	141.293	(—)
1	1.2	96.06	()	2507.00	37.28	0.8947	(-)	0.9721	(-)	148.087	(—)
	1.9	60.93	()	3/54.55	43.18	0.8834	(-)	0.9716	(—)	187.806	()
20	1.0	141.95	()	5275.85	37.17	0.8715	()	0.9684	()	267.317	(—)
20	1.0	176.85	()	6658.60	37.65	0.8611	(-)	0.9650	(—)	335.660	()
	2.0	192.35	()	7392.40	38.43	0.8503	(-)	0.9664	(—)	363.356	()
	3.0	103.95	(514550.40)	4184.50	40.25	0.8023	(0.8289)	0.9648	(0.8974)	188.456	(15082.950)
	5.0	43.70	(10234.25)	1779.65	40.72	0.7391	(0.8564)	0.9704	(0.9062)	70.740	(328.350)
	10.0	47.95	(24.30)	1998.50	41.68	0.6616	(0.9027)	0.9820	(0.9319)	67.499	(1.181)
	RND	100.50	(141.85)	3166.60	31.51	0.6873	(0.9449)	0.8652	(0.9514)	133.990	(3.080)
	1.0	224.85	()	8826.15	39.25	0.9016	(-)	0.9739	(-)	835.458	(-)
1	1.2	316.80	()	12289.35	38.79	0.8895	(—)	0.9696	(-)	1140.250	(—)
	1.4	843.65	()	32699.35	38.76	0.8759	(-)	0.9636	(-)	2912.100	(-)
	1.6	1031.45	()	40903.60	39.66	0.8606	()	0.9621	(—)	3599.900	(—)
25	1.8	1438.20	()	58622.25	40.76	0.8477	(-)	0.9546	(-)	5030.700	$\dot{(}-\dot{)}$
	2.0	1890.30	()	78272.80	41.41	0.8328	(-)	0.9486	(-)	6527.300	()
1	3.0	1002.75	()	46167.95	46.04	0.7796	(-)	0.9460	<u>i</u> —i	3268,900	$\tilde{c} = \tilde{s}$
1	5.0	719.85	()	35648.80	49.52	0.7087	(-)	0.9542	$\dot{(-)}$	2162.000	(-)
	10.0	168.75	(191.55)	7670.75	45.46	0.6263	(0.8883)	0.9782	(0.9130)	342.518	(12.193)
L	RND	168.35	(3390.75)	5933.45	35.24	0.7147	(0.9762)	0.8557	(0.9804)	412.359	(11.248)

Table 4-1: Experiments for instances SLOPE with the 1-arborescence relaxation

into a very special, and peculiar, class of ATSPs.

Note also that values of lower bounds improved by the Lagrange relaxation are relatively independent of the value of the slope parameter p, and give good(i.e., large) lower bounds. 2) These results together with Table 2-1 indicate that, irrespective of the algorithm used,

difficult SLOPE instances which require much computational time are those whose parameter values are around p = 2. It is interesting to note that as p is increased from 1, instances SLOPE first become more difficult to solve, and as p exceeds 2, they become easier. For those difficult SLOPE instances, the increase of computational time for the proposed algorithm based on the 1-arborescence relaxation is not as substantial as that for the assignment counterpart.

3) As will be expected from the computational experiments given in Section 2, the computational time requirement of the assignment relaxation exploded even for p = 3 or 5 when N = 25. When N is 20, the assignment relaxation took excessive time for $p \leq 3$. As N increases, the assignment relaxation becomes inefficient even for large values of p.

4) When the assignment relaxation is used for instances SLOPE, NB increases rapidly as N exceeds 15 as shown in Table 2-1. The proposed algorithm, on the other hand, indicates much preferable characteristics in terms of the growth of NB as N increases.

Computational time requirement per subproblem (i.e., a node in a branch and bound tree) is not so heavy when the assignment relaxation is used even with the BC procedures. However, because of the weakness of the resultant lower bounds for instances SLOPE, NB tends to grow rapidly, resulting in an excessive overall computational time.

The 1-arborescence relaxation together with the Lagrange relaxation, on the other hand, requires more time per subproblem, because many 1-arborescence problems must be solved

repeatedly as the Lagrange multipliers get modified. The subgradient optimization, however, yields comparatively better lower bounds, which helps reduce NB.

4.2 Computational experiments with the real-life instance

Two branch and bound algorithms, one based on the assignment relaxation with the BC procedures, and the other based on the 1-arborescence relaxation discussed in the present paper, are compared, and the results are shown in Table 4-2.

In Table 4-2, RX denotes the type of relaxation problem to be used, and with regard to

		-			
RX	NB	TN1A	RX/OPT	LB/OPT	CPU
assignment	3030831		0.7231	0.8786	971223.000
1-arb(160)	475	28130	0.7675	0.9493	3149.611
1-arb(140)	475	25330	0.7675	0.9493	2845.613
1-arb(120)	475	22530	0.7675	0.9493	2512.327
$1 \text{-} \operatorname{arb}(100)$	478	19946	0.7675	0.9493	2242.828
$1 \text{-} \operatorname{arb}(80)$	547	21309	0.7675	0.9493	2653.280
$1 \text{-} \operatorname{arb}(70)$	654	25287	0.7675	0.9493	3438.836
$1 - \operatorname{arb}(60)$	887	34624	0.7675	0.9403	5163.555
$1 - \operatorname{arb}(50)$	3798	155181	0.7675	0.9306	25251.000
1 - arb(40)	18705	635209	0.7675	0.9174	93882.000

Table 4-2: Experiment for the real-life instance

the 1-arborescence relaxation, the value k in "1-arb(k)" is the maximum number of repetitions allowed for a subproblem in the process of the Lagrange relaxation. Two ratios, RX/OPT and LB/OPT are nothing but AP/OPT and BC/OPT, respectively, for the assignment relaxation (see Table 2-1), and ARB/OPT and LAG/OPT, respectively, for the 1-arborescence relaxation (see Table 4-1).

The results suggests the following observations:

1) The proposed algorithm solves the real-life ATSP instance more than 400 times faster than the algorithm that uses the assignment relaxation. Also, the number of branches required for the assignment relaxation algorithm is more than 6000 times of that for the proposed algorithm. Though the 1-arborescence relaxation takes more time for solving a subproblem, the better improvement of lower bounds contributes more to yield an efficient algorithm.

2) Even within the 1-arborescence relaxation, one can observe the dramatic dependence of the strength of lower bounds on the algorithmic efficiency. A small difference of 0.01 in LB/OPT could change CPU time by one digit. This indicates that the proper selection of the iteration limit k is important for the 1-arborescence relaxation. Setting k too small would likely to hurt the computational efficiency, as seen from Table 4-2.

5 Conclusions and Future Researches

This paper is motivated by a real-life instance of the asymmetric TSP, for which an algorithm based on the assignment relaxation fails to provide good performance, despite a widely believed guideline which recommends to "select the assignment relaxation for ATSPs, whereas the 1-tree relaxation for STSPs".

It is a matter of course that solution algorithms should take full advantages of structure and/or characteristics of the problem of interest. This is especially true for computationally

244

tough (say, \mathcal{NP} -hard) combinatorial optimization problems such as the TSP studied in this paper.

It is natural that problem instances reflect characteristics of a real problem scenario. Since information concerning a TSP instance is condensed in its distance matrix, characteristics and structure of the problem would be hidden in the distances. As in the case presented in this paper, even rough ideas known about how distances are "defined" would help identify any potential problem structures. Knowledges of problem structures can be, and should be exploited to determine suitable solution strategy such as the selection of relaxation problems used within the branch-and-bound framework.

For TSP instances of a particular application, the distances (c_{ij}) reflect relative "undesirableness" of going from one node to another. Each node normally corresponds to a particular "entity" in a real world and entities have their own "properties". Distances between two nodes are often defined as some functions of their properties, reflecting the underlying problem structure. This implies that so-called asymmetric TSP instances often used to evaluate ATSP algorithms are very peculiar problem instances in the sense that individual entries in the distance matrix are totally independent of each other, which may be difficult, if not impossible, to find in real world.

The above argument leads us to believe, without proof nor definite evidence, that many of real-life instances of TSPs are some deformations of Euclidean TSPs, without regard to whether they are symmetric or not. In fact, many such problems are expected, as in our example, to retain flavour of Euclidean TSPs. For much wider range of real-life ATSPs than we anticipate, it is estimated that the 1-tree or 1-arborescence relaxation provides much stronger bounds than the assignment relaxation.

Whether a TSP is symmetric or asymmetric is certainly one characterization of the problem. Standard references (say, [11]) suggest to base the selection of relaxation problem on this symmetric/asymmetric dichotomy. This paper warns that this dichotomy should be taken with caution. Moreover, the discussions given here imply that one should pay more attention, in selecting relaxation problems, to whether or not a problem maintains strong flavour of Euclidean TSPs.

We have not yet come up with definite measures obtained from given distance data of an ATSP nor with a formula that tells us which relaxation problem to choose, even if we restrict our attention to the assignment relaxation and the 1-arboresence relaxation. We guess that independence/dependence of c_{ij} 's play a key factor to come up with some measures. More concretely, the independence/dependence of c_{ij} 's can be measured by, say, 1) the ratio of distances c_{ij} and c_{ji} between two nodes, and 2) relative relationship of nodes. The former gives a local (i.e., pairwise) measure of the independence/dependence, whereas the latter a global measure.

With regard to the local measure, when the variability of these ratios c_{ij}/c_{ji} is higher, distances c_{ij} and c_{ji} can be regarded as more independent. Our results show that for ATSPs with high variability of the ratios, the assignment relaxation dominates the 1-arborescence relaxation.

On the other hand, global measures try to extract some "geometric" structure based on the global and relative relationship of nodes. Recognition of certain forms of "directionality" might lead to a good potential measure. One possibility for recognizing a directionality would take an original (complete) graph, and transform it by removing a longer arc (i.e., an arc with $\max\{c_{ij}, c_{ji}\}$) between each pair of nodes. The sum of in-degrees and out-degrees of each node of the resultant graph would be constant N, where N is the number of nodes. If the out-degrees of the transformed graph are distributed uniformly, i.e., $N, N - 1, \ldots, 1$, this would imply the existence of a certain directionality, as this is the case for SLOPE. For instances RND, however, the out-degrees of the tarnsformed graph may not be distributed uniformly between 1 and N.

In the mean time, one quick and dirty, yet easily implementable, test before selecting the type of a relaxation problem would be comparisons of the relative strength of lower bounds, without following blindly unproven criteria such as "use the assignment relaxation for ATSPs, whereas the 1-tree relaxation for STSPs".

In summary, the inherent problem structure is a key factor for selecting a proper solution strategy. More research will be needed to associate problem structures with solution strategies, which allows us to design a mechanism for an "automatic" identification of a proper solution strategy.

Acknowledgements

The authors would like to express their appreciations to the referees and the editor for their comments which greatly improved the readability of this paper. A part of the work of the second author was done while he was visiting School of Industrial Engineering, Purdue University. He would like to express his appreciations to Professors Bruce Schmeiser, Ron Rardin, Don Wagner and Ferdinand Leimkuhler.

References

- Balas, E. and Christofides, N., "A Restricted Lagrangean Approach to the Traveling Salesman Problem", Mathematical Programming, Vol.21(1981), pp.47-69.
- [2] Carpaneto, G. and Toth, P., "Some New Branching and Bounding Criteria for the Asymmetric Travelling Salesman Problem", *Management Science*, Vol.26(1980), pp.736-7435.
- [3] Chu,Y. and Lin,T., "On the Shortest Arborescence of a Directed Graph", Scientia Sinica, Vol.4(1965), pp.1396-1400.
- [4] Fulkerson, D., R., "Packing Rooted Directed Cuts in a Weighted Directed Graph", Mathematical Programming, Vol.6(1974), pp.1-13.
- [5] P., L., Hammer, "Algorithms and Codes for The Assignment Problem", Annals of Operations Research, Vol.13(1988), pp.193-223.
- [6] Held, M. and Karp, R., M., "The Traveling Salesman Problem and Minimum Spanning Trees : Part 2", Mathematical Programming, Vol.1 (1971), pp. 6-25.
- [7] Ibaraki, T., Combinatorial Optimization (in Japanese), San-Gyou Tosho (1983).
- [8] Kobayashi, T., Network Programming Problem (in Japanese), Baifukan (1981).
- [9] Konno, H. and Suzuki, H., Integer Programming Problem and Combinatorial Optimization (in Japanese), Nikka-Giren Shuppan (1982).
- [10] Kubo, H. and Okino, N., "An Algorithm for Large Scale Traveling Salesman Problem" (in Japanese), Journal of the Operations Research Society of Japan, Vol.19 (1976), pp.272-285.
- [11] Lawler, E., L., Lenstra, J., K., Rinnooy Kan, A., H., G. and Shmoys, D., B., The Traveling Salesman Problem, Wiley (1984).
- [12] Smith, T., H., C., Srinivasan, V., and Thompson, G., L., "Computational Performance of Three Subtour Elimination Algorithms for Solving Asymmetric Travelling Salesman Problems", Annals of Discrete Mathematics, Vol.1(1977), pp.495-506.
- [13] Smith, T., H., C., "A LIFO Implicit Enumeration Algorithm for the Asymmetric Travelling Salesman Problem Using a One-Arborescence Relaxation", Mathematical Programming Study, Vol.12(1980), pp.108-114.
- [14] Suzuki, H., "A Shortest Path Problem with Visiting Order Constraint", Proceedings of the 7th Mathematical Programming Symposium of Japan, (1986), pp.127-142.

A A Real-Life ATSP Problem Instance

Figure A-1 shows the numerical data of a real-life instance.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 2	99999 1180	0	9999	9999 14	9999 78	9999 252	9999 296	9999 330	9999	9999 288	9999	9999	9999	9999	9999	9999	9999	9999
3	1180	9999	9999	6	24	198	242	276	281	286	306	326	316	326	328	330	332	343
:	1176	9999	10	9999 27	40	214	258	292	275	280	300	342	332	320	344	346	348	337
6	828	9999	367	375	348	9999	44	78	144	137	184	123	118	217	130	132	134	194
7	740	9999	455	463	436	88	9999	34	100	93	140	79	74	173	86	88	90	150
8	672	9999	523	531	504	156	68 176	9999	66	59 18	106	45	40	139	52	54	56	116
10	620	9999	569	559	606	258	170	102	25	9999	20	74	64	40	76	78	80	57
11	606	9999	603	593	679	331	243	175	45	33	9999	126	116	20	128	130	132	70
13	592	9999	603	611	584	234	149	80	99	61	87	17	9999	99	12	14	16	76
14	578	9999	640	630	736	388	300	232	83	71	37	174	152	9999	143	145	147	65
16	568	9999	627	635	612	260	172	104	123	85	111	41	24	102	9999	2 9999	1 2	64
17	560	9999	635	643	616	268	180	112	131	93	119	49	32	110	8	4	9999	60
18	506	9999	683	673	719	371	283	215	140	114	113	157	135	91	111	107	103	9999
20	494	99999	725	715	821	473	385	317	167	155	122	259	237	84	213	209	205	42
21	480	9999	715	723	695	348	260	192	211	173	199	129	112	190	88	84	80	59
22 23	460	99999	726	716	756	408	320	252	188	162	161	194	172	139	148	144 221	140	48 74
24	450	9999	745	753	726	378	290	222	241	203	229	159	142	220	188	114	110	89
25	420	9999	784	792	762	414	326	258	280	242	268	180	178	259	154	150	146	128
27	398	3333	826	816	932	584	496	428	268	256	223	370	348	185	324	320	316	143
28	390	9999	1083	1049	1447	1099	1011	943	453	476	363	916	863	287	839	835	831	364
29	384	9999	1162	1206	1599	1251	1163	1095	532	554	442 520	1068	1015	365	991	987 1137	983 1133	443 521
31	372	9999	1318	1284	1900	1552	1464	1396	688	711	599	1369	1316	522	1292	1288	1284	599
32	250	9999	936	926	966	618	530	462	398	372	371	404	382	349	358	354	350	258
33	76	9999	11119	1127	1100	752	664	596	615	577	603	185	468	594	492	488	484	463
35	32	9999	1163	1171	1144	796	708	640	659	621	647	577	560	638	536	532	528	507
36	0	9999	1195	1203	1176	828	740	672	691	653	679	609	592	670	568	564	560	539
	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
	19 9999	20 9999	21 9999	22 9999	23 9999	24 9999	25 9999	26 9999	27 9999	28	29	30 9999	31 9999	32 9999	33	34	35	36
1 2 3	19 9999 344 356	20 9999 357 369	21 9999 426 372	22 9999 372 364	23 9999 371 383	24 9999 111 387	25 9999 470 416	26 9999 479 425	27 9999 408 420	28 9999 530 590	29 9999 582 642	30 9999 633 693	31 9999 684 744	32 9999 477 469	33 9999 604 550	34 9999 628 574	35 9999 650 596	36 9999 666 612
1 2 3 4	19 9999 344 356 350	20 9999 357 369 363	21 9999 426 372 388	22 9999 372 364 358	23 9999 371 383 377	24 9999 441 387 403	25 9999 470 416 432	26 9999 479 425 441	27 9999 408 420 414	28 9999 530 590 568	29 9999 582 642 620	30 9999 633 693 671	31 9999 684 744 722	32 9999 477 469 463	33 9999 604 550 566	34 9999 628 574 590	35 9999 650 596 612	36 9999 666 612 628
1 2 3 4 5 6	19 9999 344 356 350 399 225	20 9999 357 369 363 434 260	21 9999 426 372 388 348 174	22 9999 372 364 358 385 211	23 9999 371 383 377 435 261	24 9999 141 387 403 363 189	25 9999 470 416 432 387 213	26 9999 479 425 441 396 222	27 9999 408 420 414 492 318	28 9999 530 590 568 834 660	29 9999 582 642 620 934 760	30 9999 633 693 671 1033 859	31 9999 684 744 722 1133 959	32 9999 477 469 463 490 316	33 9999 604 550 566 526 352	34 9999 628 574 590 550 376	35 9999 650 596 612 572 398	36 9999 666 612 628 588 414
1 2 3 4 5 6 7	19 9999 344 356 350 399 225 181	20 9999 357 369 363 434 260 216	21 9999 426 372 388 348 174 130	22 9999 372 364 358 385 211 167	23 9999 371 383 377 435 261 217	24 9999 141 387 403 363 189 145	25 9999 470 416 432 387 213 169	26 9999 479 425 441 396 222 178	27 9999 408 420 414 492 318 274	28 9999 530 590 568 834 660 616	29 9999 582 642 620 934 760 716	30 9999 633 693 671 1033 859 815	31 9999 684 744 722 1133 959 915	32 99999 477 469 463 490 316 272	33 9999 604 550 566 526 352 308	34 9999 628 574 590 550 376 332	35 9999 650 596 612 572 398 354	36 9999 666 612 628 588 414 370
1 2 3 4 5 6 7 8	19 9999 344 356 350 399 225 181 147 74	20 9999 357 369 363 434 260 216 182	21 9999 426 372 388 348 174 130 96	22 9999 372 364 358 385 211 167 133	23 9999 371 383 377 435 261 217 183	24 9999 441 387 403 363 189 145 111	25 9999 470 416 432 387 213 169 135 200	26 9999 479 425 441 396 222 178 144 209	27 9999 408 420 414 492 318 274 240 138	28 9999 530 568 834 660 616 582 260	29 9999 582 642 620 934 760 716 682 212	30 9999 633 693 671 1033 859 815 481 262	31 9999 684 744 722 1133 959 915 881 414	32 9999 477 469 463 490 316 272 238	33 9999 604 550 526 352 308 274 234	34 9999 628 574 590 550 376 332 298	35 9999 650 596 612 398 354 320 380	36 9999 666 612 628 588 414 370 336
1 2 3 4 5 6 7 8 9	19 9999 344 356 350 399 225 181 147 74 69	20 9999 357 369 363 434 260 216 182 87 87 82	21 9999 426 372 388 348 174 130 96 156 120	22 9999 372 364 358 385 211 167 133 102 83	23 9999 371 383 377 435 261 217 183 101 97	24 9999 141 387 403 363 189 145 111 171 135	25 9999 470 416 432 387 213 169 135 200 164	26 9999 479 425 441 396 222 178 144 209 173	27 9999 408 420 414 492 318 274 240 138 134	28 9999 530 568 834 660 616 582 260 279	29 9999 582 642 620 934 760 716 682 312 330	30 9999 633 673 671 1033 859 815 481 363 381	31 9999 684 744 722 1133 959 915 881 414 433	32 9999 477 469 463 490 316 272 238 207 188	33 9999 604 550 566 352 308 274 334 298	34 9999 628 574 590 550 376 332 298 358 328 328	35 9999 650 596 612 398 354 320 380 344	36 9999 666 612 628 588 414 370 336 396 360
1 2 3 4 5 6 7 8 9 10 11	19 9999 344 356 350 399 225 181 147 74 69 60	20 9999 357 369 363 434 260 216 182 87 82 62	21 9999 426 372 388 348 174 130 96 156 120 172	22 9999 372 364 358 385 211 167 133 102 83 96	23 9999 371 383 377 435 261 217 183 101 97 77	24 9999 441 387 403 363 189 145 111 171 135 187	25 9999 470 416 432 387 213 169 135 200 164 216	26 9999 479 425 441 396 222 178 144 209 173 226	27 9999 408 420 414 492 318 274 240 138 134 114	28 9999 530 568 834 660 616 582 260 279 206	29 9999 582 642 620 934 760 716 682 312 330 258	30 9999 633 671 1033 859 815 481 363 381 309	31 9999 684 744 722 1133 959 915 881 414 433 360	32 99999 477 469 463 490 316 272 239 207 188 201	33 9999 604 550 566 526 352 308 274 334 298 350	34 9999 628 574 590 376 332 298 328 328 322 374	35 9999 650 596 612 572 398 354 354 320 380 344 396	36 9999 666 612 628 588 414 370 336 396 360 412
1 2 3 4 5 6 7 8 9 10 11 12 13	19 9999 344 356 350 399 225 181 147 74 69 60 120 120	20 9999 357 369 363 434 260 216 182 87 87 82 62 155 142	21 9999 426 372 388 348 174 130 96 156 120 172 66 56	22 9999 372 364 358 211 167 133 102 83 96 106 93	23 9999 371 383 377 435 261 217 183 101 97 77 156	24 9999 441 387 403 363 189 145 111 171 135 187 81 71	25 9999 470 416 432 387 213 169 135 200 164 216 90 95	26 9999 479 425 441 396 222 178 144 209 173 226 99 104	27 9999 408 420 414 492 318 274 240 138 134 114 213 200	28 9999 530 568 834 660 616 582 260 279 206 576 542	29 9999 582 642 620 934 760 716 682 312 330 258 676 642	30 9999 633 693 671 1033 859 815 481 363 381 309 775 741	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841	32 99999 477 469 463 316 272 238 207 188 201 188 201 198	33 9999 604 550 526 352 308 274 334 298 350 244 234	34 9999 628 574 590 376 332 298 328 322 374 268 258	35 9999 650 596 612 398 354 320 380 344 396 344 396 290 280	36 9999 666 612 628 414 370 336 396 360 412 306 296
1 2 3 4 5 6 7 8 9 10 11 12 13 14	19 9999 344 356 350 225 181 147 74 69 60 120 107 56	20 9999 357 369 363 434 260 216 182 87 82 62 155 142 42	21 9999 426 372 388 348 174 130 96 156 120 172 66 56 187	22 9999 372 364 358 385 211 167 133 102 83 96 106 93 93 91	23 9999 371 383 377 435 261 217 183 101 97 77 156 143 72	24 9999 441 387 403 363 189 145 111 171 135 187 81 71 202	25 9999 470 416 432 387 213 169 135 200 164 216 95 231	26 9999 479 425 441 396 222 178 144 209 173 226 99 104 241	27 9999 408 420 414 492 318 274 240 138 134 114 213 200 93	28 9999 530 590 568 834 660 616 582 260 279 206 576 542 160	29 9999 582 642 620 934 760 716 6822 330 258 676 642 211	30 9999 633 693 671 1033 859 815 481 363 381 309 775 741 262	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 314	32 9999 477 469 463 490 316 272 238 207 188 201 211 198 196	33 9999 604 550 526 352 308 274 334 298 350 244 234 234 365	34 9999 628 574 590 350 376 332 298 358 322 374 268 374 258 389	35 9999 650 596 612 572 398 354 320 380 380 344 396 290 280 411	36 9999 666 612 628 588 414 370 336 360 412 306 296 296 427
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	19 9999 344 356 350 399 225 181 147 74 60 120 107 56 95	20 9999 357 369 363 434 260 216 182 87 82 62 155 142 155 142 120	21 9999 426 372 388 348 174 130 96 156 120 172 66 56 56 187 44	22 9999 372 364 385 211 167 133 102 83 96 106 93 91 81 79	23 9999 371 383 377 435 261 217 183 101 97 77 156 143 72 131 120	24 9999 441 387 403 363 189 145 111 171 135 187 81 71 202 59 57	25 9999 470 416 432 387 213 169 135 200 164 216 90 95 231 83 83	26 9999 479 425 441 396 222 178 144 209 173 226 99 104 241 92 90	27 9999 408 420 414 492 318 274 240 138 134 114 213 200 93 188	28 9999 530 568 834 660 616 582 260 279 206 576 576 576 542 260 576 576	29 9999 582 642 934 760 716 682 312 330 258 676 642 211 630 628	30 9999 633 693 671 1033 859 815 481 363 381 309 775 741 262 729 727	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 314 829 827	32 9999 477 469 463 490 316 272 238 207 188 201 211 198 196 186	33 9999 604 556 526 352 308 274 334 298 350 244 234 350 244 234 365 222	34 9999 628 574 550 376 322 298 3388 3388 3388 3388 3388 3388 3	35 9999 650 596 612 572 398 354 320 384 396 290 280 280 280 411 268 296	36 9999 666 612 628 588 414 370 336 360 412 306 296 296 296 427 284
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	19 9999 344 356 350 399 225 181 147 74 60 120 107 56 95 93 91	20 9999 357 369 363 434 260 216 182 87 82 62 155 142 42 130 128	21 9999 426 372 388 348 174 130 96 156 120 172 66 56 187 44 44 42	22 9999 372 364 358 385 211 167 133 102 83 96 106 93 91 81 79 77	23 9999 371 383 377 435 261 217 183 101 97 77 156 143 72 131 129 127	24 9999 441 387 403 363 149 145 111 171 135 187 81 71 202 59 57 57	25 9999 416 432 387 213 169 135 200 164 216 90 95 231 231 83 81 79	26 9999 425 441 396 222 178 144 209 173 226 99 104 241 99 104 292 90 88	27 9999 408 420 414 492 318 274 240 138 134 114 213 200 93 188 188 186	28 9999 530 590 568 834 660 616 582 260 279 206 576 542 160 530 528 528	29 9999 582 642 620 934 766 682 312 330 258 676 642 258 676 641 630 628 626	30 9999 633 671 1033 859 815 481 363 381 363 381 309 775 741 262 729 727 725	31 9999 684 744 722 1133 959 915 8814 414 433 360 875 841 314 429 827 825	32 9999 477 469 463 490 316 272 238 207 188 201 211 198 196 186 184	33 9999 604 556 526 352 308 274 334 234 234 350 244 234 365 222 220 218	34 9999 628 574 590 376 322 298 358 322 374 268 258 258 258 246 244 244	35 9999 650 596 612 572 398 354 320 384 320 344 396 290 280 280 280 411 268 264	36 9999 666 612 628 588 414 370 336 360 412 306 296 296 427 284 222 282 282 282
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	19 9999 344 356 350 399 225 181 147 74 60 120 107 56 95 93 91 13	20 9999 357 369 216 182 87 82 62 155 142 42 128 128 128 128 26	21 9999 426 372 388 348 174 130 96 156 120 172 66 56 187 44 42 40 62	22 9999 364 358 385 211 167 133 102 83 96 106 93 91 102 103 81 79 77 77 77 726	23 9999 371 383 377 435 261 217 183 101 97 77 755 143 72 131 129 127 40	24 9999 441 387 403 363 189 145 111 171 135 187 81 71 202 59 57 55 77	25 9999 416 432 387 213 169 135 200 164 216 90 95 231 83 83 81 79 9106	26 9999 425 441 396 222 178 144 209 173 226 99 104 241 92 90 88 81 15	27 9999 408 420 414 492 318 274 240 138 134 114 213 200 93 188 186 184 77	28 9999 530 568 834 660 616 582 260 279 206 576 542 160 530 530 528 526	29 9999 582 642 620 934 760 716 682 330 258 676 642 211 630 628 628 628 628	30 9999 633 673 671 1033 859 815 481 363 381 309 775 741 262 729 727 725 326	31 9999 684 744 722 1133 959 915 8814 414 433 360 875 841 314 329 825 827 825 377	32 9999 469 463 490 316 272 238 207 188 201 211 198 196 186 186 184 182 131	33 9999 604 550 556 352 308 274 334 298 350 244 234 364 222 220 220 220 220 220 220 220 220 22	34 9999 628 574 590 336 332 298 332 258 322 374 268 258 389 246 244 244 244 244	35 9999 650 596 612 399 354 320 380 344 396 290 280 411 268 266 266 286 286	36 9999 666 612 628 388 414 370 336 396 360 412 306 296 427 284 282 284 282 284 302 302
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	19 9999 344 356 350 225 181 147 74 60 120 107 56 93 91 103 9999 20	20 9999 367 369 363 434 4260 216 182 87 82 62 155 142 42 130 128 126 26 13 3	21 9999 372 388 348 174 130 96 156 120 172 666 187 44 42 40 62 999 146	22 9999 372 364 358 211 167 133 102 83 96 106 93 95 93 91 81 79 77 76 55	23 9999 371 383 377 461 217 77 77 77 77 77 156 143 72 131 129 127 40 27 31	24 9999 411 387 403 363 189 145 111 171 137 81 71 202 59 55 57 77 71 14	23 9999 410 416 432 213 165 205 201 216 206 231 83 81 79 106 142	26 9999 425 441 396 222 178 241 209 104 241 90 80 80 81 52 200	27 9999 408 420 414 424 318 274 138 138 134 114 213 200 93 188 186 184 77 64	28 9999 5300 568 834 660 616 582 260 279 206 542 160 528 542 160 528 528 528 528 174	29 9999 582 642 620 9716 682 312 312 312 312 258 676 642 211 630 628 628 628 628 625 225 226	30 9999 633 693 671 1033 859 815 481 363 361 309 775 729 729 727 725 3266 277 219	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 314 827 825 377 328	32 9999 469 469 463 316 273 238 207 218 196 196 196 184 182 131 140	33 9999 604 550 556 352 308 274 334 234 350 244 234 234 234 234 234 244 234 222 220 218 240 217 324	34 9999 628 574 580 376 332 298 322 374 268 332 2374 268 339 246 244 244 244 244 301 348	35 9999 650 596 612 572 398 354 320 380 380 384 396 290 280 411 268 266 264 266 323 370	36 9999 6612 628 538 414 370 336 3360 3360 3412 296 412 296 427 282 280 302 282 280 339 338
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 7 18 19 20 21	19 9999 344 356 350 399 225 181 147 74 60 120 120 120 120 120 56 93 91 13 9999 20 78	20 9999 357 369 363 434 260 216 182 87 82 62 155 142 128 128 128 126 13 9999 107	21 9999 372 388 348 174 130 966 156 120 172 666 187 44 42 40 62 99 999	22 9999 374 358 385 211 167 133 102 83 96 106 93 991 81 799 77 76 25 50 037	23 9999 371 383 377 435 261 217 183 101 97 77 155 143 72 131 129 127 40 027 31	24 9999 411 387 403 363 363 189 145 111 171 133 187 81 71 202 59 57 77 114 161	23 9999 410 416 432 213 165 200 164 216 90 95 231 83 81 79 106 142 190 39	26 9999 473 425 441 296 99 173 226 99 104 241 90 88 115 152 200 48	27 9999 408 420 414 424 318 274 240 138 134 114 213 200 93 188 186 184 77 764 51	28 9999 500 590 568 834 660 616 582 260 279 206 576 542 160 528 528 528 528 528 174 117 486	29 9999 582 642 620 934 760 716 682 312 330 258 676 642 211 630 628 642 211 632 628 642 211 830 628 642 618 858 658	30 9999 633 693 671 1033 859 815 481 363 381 309 775 741 262 729 727 725 326 277 725 326 327 9685	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 314 829 827 377 328 271 785	32 99999 469 469 463 316 272 238 207 188 201 211 211 211 211 211 211 211 211 211	33 9999 604 550 556 526 3352 335 274 334 234 234 350 244 234 234 222 220 218 244 230 241 244 232 220 218 244 118	34 9999 628 574 590 376 322 299 358 328 328 338 338 268 268 268 244 244 244 244 244 244 264 264 244 301 348 2001	35 9999 650 596 612 572 398 354 320 380 344 320 344 326 2200 411 266 266 266 266 266 323 370 224	36 9999 6612 628 538 414 376 3360 412 306 412 296 427 282 282 282 282 339 338 6 240
1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 16 17 8 19 20 20 22	19 99999 344 356 350 390 225 181 147 74 69 60 120 107 56 60 120 107 13 95 995 991 13 9999 200 78 8999	20 99999 363 434 260 216 182 87 82 62 155 142 42 130 128 126 26 213 9999 107 56	21 9999 426 372 388 174 130 956 120 172 66 56 120 172 66 56 187 44 42 40 62 99 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	22 9999 364 388 211 133 102 83 96 93 91 11 81 77 26 35 50 37 79999	23 9999 371 383 261 217 183 101 101 109 129 127 40 27 127 40 27 31 87 99	24 99999 441 387 403 189 145 111 171 135 187 81 71 202 59 55 77 55 55 77 114 161 15 1 54 56	25 9999 470 416 422 387 213 135 200 164 216 201 64 216 95 231 231 231 24 216 95 231 231 231 231 24 20 5 231 231 23 20 164 216 42 22 23 23 20 164 20 20 20 20 20 20 20 20 20 20 20 20 20	26 9999 425 441 396 222 178 144 209 173 226 99 104 241 241 292 90 00 88 115 152 200 88 83	27 9999 408 420 414 492 318 274 240 138 138 134 114 213 200 93 188 188 188 188 188 184 77 64 51 144 56	28 9999 530 568 834 660 616 582 260 279 206 576 542 160 528 526 223 117 117 486 210	29 9999 582 620 934 760 716 682 312 330 258 676 642 231 238 676 642 275 226 168 586 586 586	30 99999 633 693 671 1033 859 815 481 363 381 363 775 741 262 729 727 725 326 277 219 685 313	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 329 825 377 328 271 785 325 3271 785	32 9999 477 469 469 316 272 238 201 211 198 198 201 211 198 198 198 198 198 198 198 198 198 1	33 9999 604 550 526 332 368 274 3350 244 234 222 222 224 224 224 224 224 224	34 9999 628 574 550 375 322 298 322 376 322 374 268 248 248 248 242 264 244 242 264 301 348 2232	35 9999 6500 596 612 572 398 384 320 384 320 384 290 280 481 268 266 2264 286 323 370 224 224	36 9999 666 612 628 838 414 370 336 3360 412 306 296 427 284 280 302 338 302 3386 2210 240 240 2210 202 2339 386 240 240 220 220 220 220 220 220 220 220
1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 16 17 8 19 20 22 23 4	19 9999 344 356 350 399 225 181 147 74 69 60 120 107 56 93 91 13 9999 20 78 49 952 20	20 99999 357 369 363 434 226 216 55 182 82 62 125 142 42 130 128 26 26 128 9999 107 56 48	21 9999 426 372 388 174 130 96 120 172 66 56 120 172 66 56 187 44 42 40 62 999 146 9999 146 9999 146 30 30 730	22 9999 372 364 358 211 133 102 83 96 106 106 93 91 81 79 77 26 35 50 37 9999 31 37	23 99999 371 383 377 435 261 217 183 101 97 77 156 143 131 129 127 129 127 40 27 31 87 19 9999 75	24 99999 441 387 403 363 145 111 135 187 81 71 202 59 57 55 77 71 144 161 15 45 10 9999	25 9999 4700 416 432 387 213 135 2000 164 216 95 231 83 81 79 95 231 106 166 120 190 39 974 139 24	26 9999 479 425 441 396 222 178 144 209 173 226 90 104 241 92 90 88 8115 152 200 48 83 149 33	27 9999 408 420 414 492 318 274 240 138 134 114 213 200 93 3188 186 186 188 186 188 186 51 144 51	28 9999 590 568 834 660 616 582 260 279 206 542 160 530 528 526 223 174 117 486 210 129	29 9999 582 642 620 934 716 682 330 258 676 642 2330 258 676 642 211 630 628 628 628 628 628 5262 181 571	30 9999 633 673 1033 859 815 481 363 381 363 775 741 262 729 725 326 277 219 685 313 232	31 9999 684 744 722 1133 959 915 881 414 433 360 875 841 314 314 327 827 827 827 827 328 271 785 364 271 785 364 273	32 9999 477 469 463 490 316 272 238 201 211 198 196 188 198 198 186 184 182 131 140 155 142 105 124	33 9999 604 550 526 352 368 274 335 224 234 350 244 234 222 220 218 240 240 218 240 214 240 217 374 178 208	24 9999 628 574 5500 376 372 299 374 268 258 389 246 244 244 244 244 244 244 244 264 301 348 202 297 187	35 9999 6500 596 612 572 398 320 380 344 396 290 280 411 268 266 264 286 266 264 286 370 224 233 370 224 319 209	36 9999 6666 612 628 414 370 336 3360 412 306 412 306 4296 427 284 282 280 302 339 386 240 240 335 225
1 3 4 5 6 7 8 9 10 1 12 3 4 5 6 7 8 9 10 1 12 3 4 5 6 7 8 9 10 1 12 3 4 5 6 7 8 9 10 1 12 3 4 5 6 7 8 9 10 1 12 3 14 5 6 7 8 9 10 1 12 3 14 5 6 7 8 9 10 1 12 3 14 5 6 7 8 9 10 11 12 3 14 5 6 7 8 9 10 11 12 3 14 5 6 7 8 9 10 11 12 3 14 5 6 7 8 9 10 11 12 3 14 5 6 7 8 9 10 11 12 3 14 5 16 7 8 9 10 11 12 3 14 5 16 7 8 9 10 11 2 3 14 5 16 7 8 9 10 11 12 3 14 5 16 7 8 9 10 11 12 3 14 5 16 7 8 9 10 11 12 3 14 15 16 1 11 11 12 3 11 11 11 11 11 11 11 11 11 11 11 11 1	19 9999 344 356 350 399 225 181 147 74 60 120 107 56 95 93 91 13 9999 20 78 49 52 107	20 9999 363 434 260 182 87 82 155 142 128 126 128 126 128 9999 9099 107 56 48 137	21 9999 426 378 388 174 130 966 120 172 172 172 166 566 566 566 566 587 44 42 999 9146 9999 146 9999 137 306	22 99599 372 364 358 385 211 167 133 102 83 91 91 91 91 91 91 91 77 77 26 35 50 37 999 91 31 37 76	23 99999 371 383 261 161 97 77 77 156 163 163 163 163 72 131 129 127 40 27 31 129 9999 75	24 9999 441 387 403 363 189 145 111 171 135 135 81 71 71 71 71 71 71 71 71 71 71 71 1202 59 95 77 77 114 161 15 55 55 110 9999 336	25 9999 470 416 432 387 213 169 135 200 164 216 216 216 216 216 216 216 216 216 216	26 9999 479 425 441 209 178 144 209 178 144 299 104 241 90 88 115 152 200 200 200 200 200 200 200 200 200 2	27 9999 408 420 414 492 318 274 240 138 134 134 213 200 93 188 186 184 77 64 51 144 51 144 51 237 200 200 200 203 203 203 203 203 203 203	28 9999 530 568 834 660 279 279 279 279 279 279 279 279 279 279	29 9999 582 642 620 934 760 716 6812 330 258 676 642 211 630 628 676 642 211 630 628 626 275 226 168 586 225 2181 571 576	30 9999 633 693 671 1033 859 815 481 363 381 363 381 363 727 725 727 725 326 277 227 729 727 725 326 277 23 262 277 23 326 267 5 313 322 675 565 313 675 767 767 767 767 767 767 767 767 767	31 9999 684 744 722 1133 959 915 881 414 433 875 841 433 827 324 271 782 377 782 377 782 377 785 364 283 7705	32 99999 477 469 463 316 272 238 207 188 207 211 211 211 211 211 211 211 211 211 21	33 9999 604 550 556 352 308 274 238 334 234 238 350 244 244 234 242 222 220 218 240 277 377 377 377 3168 163	34 9999 628 574 550 376 332 288 322 374 268 258 258 268 244 242 264 244 244 264 264 264 264 264	35 9999 650 556 612 572 398 354 320 240 241 266 323 373 373 324 244 286 323 373 373 373 373 224 254 284 286 323 373 373 373 209 209 200	36 9993 666 612 628 588 396 396 396 296 296 296 296 302 282 282 282 339 386 240 270 335 228 240 270 270 270 282 282 282 282 282 282 282 282 282 28
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 9 20 21 22 23 24 25 26 6 77	19 9999 344 356 399 225 181 147 74 60 120 107 56 93 91 107 59 93 91 107 59 920 78 49 9999 20 78 49 20 78 49 20 78 49 20 78 52 13 91 52 50 20 51 14 50 107 107 50 107 50 107 50 50 50 50 50 50 50 50 50 50 50 50 50	20 9999 357 369 363 434 216 216 216 215 5 5 142 42 130 128 26 26 215 30 128 230 9999 107 56 36 35 126 137 176 176 195	21 9999 426 378 348 348 144 130 96 156 127 66 56 187 44 40 62 99 99 99 99 99 99 99 90 137 30 66 67 30 66 67 27 27 27 27 27 20 20 20 20 20 20 20 20 20 20 20 20 20	22 99999 372 364 358 385 211 167 133 102 96 96 93 96 93 91 6 77 77 26 35 80 37 776 595 95	23 9999 371 383 377 435 261 217 185 143 77 77 186 143 127 40 27 127 31 87 19 9999 9999 99999 99999 99999 99999 114 133	24 9999 441 387 403 363 189 145 145 171 171 171 171 187 81 171 202 59 55 777 114 161 15 45 110 9999 9999 935 55	25 9999 470 416 432 387 213 169 135 200 169 93 231 83 81 79 95 231 183 83 81 162 142 190 39 74 139 74 139 199 99 99 99 99 99 99 99 99 99 99 99 9	26 9999 425 441 396 202 178 144 209 104 241 90 90 90 88 88 152 2008 48 83 152 2008 48 33 9999	27 9998 408 420 414 492 374 240 138 134 114 114 240 138 138 136 188 186 188 186 51 184 51 144 51 144 51 129 123 113	28 9999 568 834 6582 260 279 206 542 160 520 520 520 520 520 520 522 174 117 466 210 471 466 476	29 9999 882 642 642 934 716 682 312 312 312 312 312 312 682 642 211 630 628 642 275 226 168 586 262 181 188 586 586 586	30 9999 633 693 671 1033 859 815 481 361 363 361 262 272 725 326 277 219 635 313 227 219 663 313 236 675 315 675 315 675 315 675 315 675 315 675 315 675 315 675 315 675 775 775 775 775 775 775 77	21 9999 684 744 722 1133 959 915 881 414 414 414 429 825 841 314 825 827 825 377 328 271 785 364 271 785 364	32 9999 477 469 463 490 316 272 201 211 211 198 198 198 198 198 198 198 198 198 1	33 9595 604 556 526 352 308 274 334 234 234 222 220 218 240 217 324 178 273 3153 154 154	24 9999 528 374 590 376 332 233 232 232 232 245 245 242 242 242 242 242 242 242 24	35 9999 650 612 577 398 354 320 380 380 280 411 268 268 268 268 268 268 268 268 228 264 228 370 224 319 209 2254 319 209 200 224	36 9993 666 612 628 588 414 370 336 396 396 396 296 427 284 282 282 309 302 302 330 224 282 282 282 282 282 282 282 282 282
1 2 3 4 5 6 7 8 9 10 11 12 23 24 25 26 27 28	19 9999 344 356 325 181 147 74 60 120 107 56 93 91 107 56 93 91 13 9999 20 78 8 49 52 108 49 52 108 147 166 121 288	20 9999 357 369 363 434 216 182 62 155 142 42 130 128 26 132 9999 107 56 487 137 56 487 137 56 20 20 20 20 20	21 9999 426 372 388 348 174 170 66 120 172 66 156 127 66 56 187 40 62 999 146 999 146 8999 60 137 30 65 575	22 99999 372 358 385 211 167 133 96 93 96 93 95 93 91 81 79 77 26 35 50 37 79999 31 37 76 59999 31 37	23 9999 371 383 377 435 261 217 135 161 101 177 77 77 135 6 143 72 131 129 129 129 129 129 129 129 129 131 133 131 57 75 114 131 69 209	24 9999 141 387 483 189 145 111 111 171 137 81 71 71 202 55 77 77 114 161 161 161 161 161 161 9999 366 555 206 721	25 9999 470 416 432 387 213 169 135 200 164 216 90 95 231 83 81 79 106 142 190 39 9 74 139 74 139 24 9999 190 736	26 9999 479 425 441 396 222 178 144 249 173 226 99 104 241 241 241 241 241 241 241 25 30 88 83 152 200 483 153 33 99 9999 170	27 9999 408 420 4492 374 274 274 210 138 134 114 213 138 186 186 186 186 51 14 51 14 51 14 51 14 51 299 123 123 123 123 1999 880	28 9999 530 568 834 616 582 260 2799 205 542 160 528 528 528 528 528 174 117 486 210 129 471 486 476 529	29 9999 582 642 642 934 716 682 312 330 258 676 258 642 211 642 275 226 262 168 586 262 262 181 571 576 576 104	30 9999 633 691 1033 859 815 411 361 361 361 361 262 725 725 326 277 219 685 3226 277 219 685 675 313 232 2670 685 675 5155	21 9999 684 744 743 743 881 4133 881 414 433 860 875 841 314 827 827 827 827 827 827 827 827 827 827	22 9999 463 469 316 272 207 188 207 198 196 196 196 184 182 131 198 185 144 182 131 131 185 144 185 127 121 127 127 121 137 127 127 127 127 127 127 127 127 127 12	33 9990 5604 550 5322 308 274 334 234 234 234 234 240 213 240 213 240 213 240 213 163 163 163 163 154 144 291	24 9999 528 550 376 332 293 293 232 232 232 232 244 244 244 244 201 348 201 2232 232 232 217 187 178 168 315	35 9999 630 596 612 577 398 354 329 290 280 411 266 286 286 286 286 286 286 286 283 370 225 4 319 209 200 337 840	36 9993 666 612 628 389 414 370 3360 360 360 296 412 296 427 284 298 299 386 240 240 240 240 240 225 225 225 225 225 225 225 225 225 22
1 2 3 4 5 6 7 8 9 0 11 12 13 14 15 16 17 18 19 20 21 22 3 24 25 26 27 8 26 27 8 26 27 8 26 27 8 26 27 8 26 27 8 9 9 20 10 11 12 23 24 25 26 27 27 27 28 20 20 20 20 20 20 20 20 20 20 20 20 20	$\begin{array}{c} 19\\ 9999344\\ 3560\\ 39995\\ 1817\\ 74\\ 649\\ 600\\ 120\\ 107\\ 566\\ 953\\ 911\\ 13\\ 9999\\ 20\\ 78\\ 953\\ 911\\ 13\\ 9999\\ 20\\ 108\\ 147\\ 166\\ 121\\ 288\\ 67\\ \end{array}$	20 9999 357 369 434 2162 2162 87 82 62 155 142 42 130 126 130 126 13 9999 9099 107 56 48 137 176 195 100 201	21 9999 426 372 388 348 130 156 156 156 166 566 567 187 44 44 40 999 146 5999 137 300 665 855 2366 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 855 2365 2365 2372 855 2372 855 2372 855 245 245 255 255 255 255 255 255 255 2	22 9999 372 364 358 345 211 167 133 906 906 906 906 906 903 91 81 79 999 31 37 999 31 37 76 9599 31 37 959 931 37 26 25 20 20 20 20 20 20 20 20 20 20 20 20 20	23 9999 3711 3833 2611 217 183 101 97 77 75 156 143 722 131 131 127 43 127 31 87 9999 99999 75 5 114 133 69 209	24 9999 441 387 403 363 189 145 111 171 135 187 81 113 71 202 29 57 77 75 55 57 77 71 114 15 15 10 9999 36 35 206 771 32	25 9999 470 415 422 387 213 169 135 200 164 216 95 2311 81 79 95 231 165 142 190 95 95 95 95 95 95 95 95 95 95 95 95 95	26 9999 479 421 396 222 178 144 209 144 209 104 217 90 80 80 80 152 200 200 200 80 80 152 99 99 99 99 99 99 170 164 83 164 83 164 83 164 83 164 83 164 83 164 83 164 152 164 152 164 164 164 164 165 164 165 164 165 165 165 165 165 165 165 165	27 9999 408 414 492 318 274 240 138 134 114 200 903 188 186 186 186 186 186 186 186	28 3999 5300 568 834 6606 582 2609 2066 542 2609 2076 542 1600 5308 5285 5265 542 1600 5308 5283 5265 542 1600 5308 542 542 1600 5308 542 542 542 542 542 542 542 542	29 9999 582 642 934 760 682 312 312 312 312 312 312 312 312 312 31	30 9999 633 671 1033 859 815 481 363 381 363 375 741 2729 727 725 373 313 3222 670 685 375 515 522 675 125 252 252 252 252 252 252 25	31 9959 644 742 1133 959 915 381 414 414 414 414 429 825 364 271 271 275 271 785 328 271 775 206 388 2775 206 388 25	32 9999 477 463 490 316 272 238 201 211 198 198 198 198 198 198 198 198 198 1	33 9999 604 5566 5162 3182 274 334 234 334 234 234 234 222 222 218 240 214 240 214 277 324 154 154 154 154 154	34 9993 628 5740 350 332 298 332 374 268 238 338 338 332 374 268 248 248 248 248 248 244 242 244 244 24	35 9999 650 512 328 320 380 380 344 320 280 411 268 2264 2264 2264 2264 2264 2264 224 370 370 370 370 319 319 200 200 337 30 347 369 200 200 200 200 200 200 200 200 200 20	369 9999 666 612 628 336 336 336 336 336 2306 2496 2496 2492 2492 3302 332 339 336 249 249 249 249 230 332 332 335 225 225 235 335 335 335 335
1 2 3 4 5 6 6 7 8 9 10 11 12 13 14 15 16 6 7 8 9 10 11 22 23 24 25 26 26 27 28 29 30	19 9999 344 356 325 181 147 74 60 120 107 56 60 120 107 59 5999 20 78 91 13 39999 20 78 49 52 108 147 166 1288 166 1288 166 145 166	20 9999 357 369 363 434 282 87 82 62 155 142 42 130 128 26 26 142 130 128 126 26 137 126 26 137 126 26 137 126 26 26 358 437	21 9999 426 372 388 348 174 130 96 120 172 66 56 120 172 66 56 187 44 42 40 60 9999 146 60 60 60 137 30 60 85 236 751 303 3034	22 9999 372 364 358 385 311 167 133 102 83 93 91 106 106 106 106 106 106 106 106 106 10	23 9999 371 383 435 261 217 183 101 101 197 77 125 127 127 127 129 127 129 127 129 129 129 129 129 129 129 129 129 129	24 9999 441 387 403 363 189 145 111 171 135 187 81 202 57 77 75 55 55 55 55 57 77 71 114 15 10 9999 366 55 5206 6 721 873 1023 1174	25 9999 470 412 213 135 213 135 216 216 216 216 216 216 216 216 216 216	26 9999 479 421 222 144 144 222 229 173 299 999 999 170 83 48 83 155 152 200 88 83 199 9999 170 716 81016	27 9999 408 414 14 2492 27 138 134 134 134 134 134 134 134 134 134 134	28 9999 590 590 590 590 590 590 268 542 542 542 542 542 542 160 528 528 528 528 528 528 528 528	29 9999 842 642 934 766 716 682 312 330 258 676 676 676 678 678 678 678 67	30 9999 633 673 671 1033 815 815 841 363 363 363 363 363 363 363 36	31 9999 744 722 959 915 881 4133 375 871 841 4133 375 875 875 875 875 877 877 864 2711 775 2711 775 775 263 864 273 864 2712 775 2999	32 9999 477 469 463 490 316 272 201 211 211 211 111 140 185 165 165 124 140 185 124 140 185 124 140 185 124 140 185 124 140 185 124 140 185 124 140 185 124 140 185 124 140 185 186 186 186 186 186 186 186 186 186 186	33 9999 604 556 526 332 309 274 334 238 350 244 233 222 222 218 208 277 324 178 208 277 3163 154 144 277 794 937 794 937	34 9999 574 574 350 350 352 298 358 358 358 358 358 358 358 358 358 35	35 9999 650 596 612 577 373 354 320 330 344 396 290 220 220 220 220 220 220 220 220 220	36 9999 6666 612 628 888 414 370 3366 2966 2966 2966 296 296 296 296 297 202 302 302 297 202 302 297 202 302 202 202 202 202 202 202 202 202
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 8 19 20 21 22 23 24 25 26 27 28 29 20 31 32	19 39999 225 225 181 147 74 69 69 1200 107 56 93 913 9999 20 78 913 9999 20 78 49 52 80 147 147 147 147 147 147 147 147	20 9999 357 369 363 434 216 216 216 216 216 25 142 42 130 26 139 999 107 36 437 176 195 100 2358 437 266	21 9999 426 388 348 174 1306 120 176 6 56 56 56 56 56 56 56 56 56 56 56 56	22 9999 364 388 388 398 391 103 102 83 991 81 81 799 35 80 37 9999 31 37 76 9393 31 37 76 955 100 333 412 412 412 412	23 9999 371 383 217 425 261 217 183 101 97 77 77 77 156 143 129 127 31 129 127 31 129 9999 75 114 133 69 209 288 326 445 217	24 9999 441 387 403 363 363 189 141 135 187 81 202 59 57 77 114 161 15 45 55 577 77 114 161 15 45 102 36 55 206 721 873 1023 873 1023 873 1024 873 1025 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 873 1026 875 1026 875 1026 875 1026 875 1026 875 1026 875 1026 875 1026 875 1026 875 1026 875 1026 1026 1026 1026 1026 1026 1026 1026	25 9999 470 410 412 213 168 213 200 164 215 231 83 81 164 142 190 95 231 190 74 139 24 190 399 74 139 24 190 190 74 139 24 190 190 24 190 24 190 24 190 25 20 190 20 20 20 20 20 20 20 20 20 20 20 20 20	26 9999 425 396 222 178 209 209 104 241 241 241 90 88 155 152 200 48 83 149 33 9999 170 786 83 149 33 99999 170 706 88 83 149 33 205 88 165 80 106 106 106 106 106 106 106 106 106 10	27 9999 408 420 1414 492 274 240 138 138 134 134 134 134 134 134 134 134 134 134	28 9999 590 590 590 616 528 576 528 526 528 528 528 528 528 528 528 528	29 9999 642 620 924 924 924 924 924 924 925 312 258 626 2211 628 626 2211 628 626 225 226 168 586 526 262 181 586 576 104 12 586 576 104 338	30 9999 693 693 693 859 815 361 361 361 361 361 361 361 361 361 361	31 9999 744 744 722 1133 959 915 841 414 433 360 875 841 414 829 827 328 271 328 271 785 327 328 271 785 327 328 271 785 326 271 785 206 38 25 25 29 9992	32 9999 477 463 463 272 273 273 273 207 188 198 198 198 198 198 198 198 198 198	33 9999 604 5566 5266 322 308 274 238 224 228 244 224 244 224 246 2220 220 220 244 246 244 246 273 163 154 154 154 154 154 154 154 154 154 154	34 9993 628 574 350 350 332 298 352 258 352 268 362 244 226 268 244 226 268 244 226 268 244 222 268 244 202 233 297 187 178 168 315 815 815 815 815 815 815 815 815 815 8	33 99999 5965 5966 398 3200 3804 398 398 398 398 398 2900 2900 2900 2900 2900 2900 2900 29	36 9999 662 612 336 336 336 320 206 296 296 296 242 282 282 282 282 282 282 282 282 282
1 2 3 4 5 5 6 6 7 8 9 9 10 11 12 13 14 15 16 16 17 18 19 20 21 22 3 24 22 24 22 30 31 12 23 24 25 30 31 12 23 24 25 26 27 28 29 20 20 21 21 22 23 24 23 24 23 24 24 25 26 26 27 28 29 20 20 20 20 20 20 20 20 20 20 20 20 20	19 9999 344 356 3999 20 1817 74 69 60 120 107 56 95 95 913 20 20 20 20 20 20 20 20 20 20 20 20 20	20 9999 357 363 434 216 216 216 216 35 35 142 130 126 130 126 130 9999 99999 9007 358 107 5 5 6 176 195 280 2016 176 195 280 2016 2016 2016 2016 2016 2016 2016 201	21 9999 126 372 388 348 130 156 156 156 156 156 156 157 44 44 40 999 146 56 56 56 56 56 56 56 56 56 56 56 56 56	22 9999 364 358 385 211 167 133 102 83 96 106 93 91 81 79 9999 97 77 266 35 500 37 9999 931 31 31 35 100 331 31 100 3412 458 858 851 851 851 851 851 851 851 851 8	23 9999 371 383 261 217 183 101 97 77 75 6 143 72 131 129 127 31 129 9999 731 14 87 19 9999 208 83 69 208 83 69 208 83 69 208 83 69 208 83 69 208 83 69 208 83 69 208 83 69 208 83 10 10 10 10 10 10 10 10 10 10 10 10 10	24 9999 4411 387 463 363 145 145 145 145 145 145 145 145 145 55 57 55 77 77 161 45 100 9999 366 721 8731 1023 1023 1023 1023	25 9999 416 432 323 135 213 135 213 135 213 135 231 833 81 79 164 216 95 95 95 95 95 95 95 95 95 95 95 95 95	26 9999 425 425 222 222 124 144 241 241 299 104 241 241 290 88 81 152 2008 48 83 149 33 9999 170 7168 868 869 10189 10189 10189	27 9999 400 414 444 240 134 134 134 134 134 134 186 186 186 186 186 186 186 186	28 9999 590 590 590 590 279 260 279 279 279 279 279 279 279 279	29 9999 842 642 934 760 716 682 312 330 258 676 275 275 275 266 275 266 275 266 168 586 266 275 266 168 586 266 275 266 168 586 266 275 266 275 266 168 586 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 266 275 275 267 275 275 275 267 275 275 275 275 275 267 275 275 275 275 275 275 275 27	30 9999 633 673 673 1033 815 481 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 363 381 365 365 365 365 365 365 365 365	31 9999 744 722 1133 959 959 959 881 414 433 375 875 875 827 827 827 827 827 827 827 827 827 827	32 9999 463 463 316 272 238 238 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 188 196 199 9999 103 477	33 9999 604 550 526 322 368 274 234 234 244 240 247 324 240 247 324 178 208 277 324 163 154 163 153 153 153 153 153 155 155 15	34 9999 578 574 570 350 350 350 322 298 322 268 258 389 268 268 244 201 348 201 244 201 202 202 203 203 203 203 203 203	35 9999 650 596 612 572 398 320 344 396 2900 2400 240 2260 2244 3370 2244 3197 2019 2009 1377 344 246 319 201 201 2009 2009 2010 3197 344 200 1307 343 943 943 1266 1266 943 943 1264 46	36 9999 6612 628 3700 3366 412 3066 427 284 240 302 240 302 3353 228 240 240 270 335 228 240 240 270 335 228 240 240 270 335 228 240 240 270 335 228 240 240 270 270 335 228 240 240 270 258 240 240 240 258 240 240 258 240 240 258 240 240 258 240 240 258 240 240 258 240 240 258 240 258 240 258 240 258 258 258 258 258 258 258 258 258 258
1 2 3 4 5 6 7 8 9 10 11 12 3 14 15 16 6 7 8 9 9 10 11 12 3 14 15 16 20 21 22 23 24 25 26 27 7 28 30 31 32 33 4 35 33 4 35 33 4 35 33 4 35 35 34 35 34 35 34 35 35 35 35 35 35 35 35 35 35 35 35 35	$\begin{array}{c} 19\\ 9999\\ 344\\ 356\\ 350\\ 359\\ 225\\ 181\\ 147\\ 74\\ 60\\ 120\\ 107\\ 107\\ 107\\ 107\\ 107\\ 107\\ 107\\ 10$	20 9999 363 434 260 216 82 82 155 142 42 155 142 128 26 26 128 128 128 128 128 128 128 128 128 128	21 9999 426 372 388 348 174 130 956 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 66 120 172 120 172 174 120 174 120 174 120 174 120 174 120 174 120 174 120 174 120 174 120 175 175 175 175 175 175 175 175 175 175	22 9999 364 364 388 388 385 395 106 91 81 81 81 81 81 81 81 81 81 81 81 81 81	23 9999 371 377 435 435 435 435 435 435 435 437 437 437 437 437 437 437 437	24 9999 411 387 463 363 143 143 143 143 143 143 143 143 143 14	25 9999 470 432 213 135 213 135 213 216 216 216 216 216 216 231 33 33 3 3 3 3 3 3 3 199 999 24 9999 24 199 190 736 888 81 189 924 303 214 335 235 3352	26 9999 425 222 222 222 209 09 909 104 241 92 90 88 83 155 2200 48 83 155 200 48 83 155 200 83 83 199 9999 170 716 83 33 33 33 333 337	27 9999 400 414 414 213 240 134 134 134 134 134 134 134 136 188 188 186 186 186 177 64 51 144 51 143 143 145 166 177 129 999 137 137 137 147 147 147 147 147 147 147 14	28 9999 590 590 590 268 344 660 279 260 279 279 260 542 160 530 528 526 223 117 486 476 476 476 476 476 476 476 47	29 9999 842 642 934 760 716 682 312 330 258 676 676 676 628 628 628 628 628 628 628 62	30 9999 633 673 673 1033 815 815 815 815 839 741 262 729 727 727 727 727 727 727 72	31 9999 684 744 722 1133 915 915 915 915 941 414 433 341 314 329 327 327 327 327 327 327 327 327	32 9999 463 463 203 203 203 203 203 203 203 203 203 20	33 9999 604 550 326 3274 338 350 2744 234 234 234 234 240 277 374 318 240 277 374 178 208 273 163 154 154 163 154 163 154 163 154 937 1078 1200 103 9999 99 99 91	34 9999 574 574 370 350 350 352 298 358 358 358 358 358 358 358 366 244 244 244 244 244 244 204 301 348 202 207 137 187 178 168 315 818 818 961 1102 1244 41 102 2999 9999	35 9999 650 596 612 577 374 320 3344 320 230 240 240 240 240 240 240 240 240 240 24	36 9999 612 612 612 336 336 336 336 336 336 336 336 336 33

Figure A-1: Numerical data of a real-life instance

B Different Versions of "Deformed" Euclidean TSP

If we consider road traffic, we often encounter a situation where going into the center of a city takes more time than going out, thus resulting in asymmetric cost matrix. Since vehicles can be assumed to move on a two-dimensional plane, distances would be same and the resultant "cost" tends to preserve some Euclidean properties, or symmetry.

Consider a "center" on a plane. There is an extra penalty associated with approaching the center, whereas regular Euclidean distance is accessed when going away from the center. This, as opposed to instances SLOPE, may be called instances CONE, as it requires extra

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

efforts or costs to go toward the top of a mountain or a cone. Details of the definition of "distance" between two points i and j on a cone will not be explained here.

The above discussion implies the existence of many TSP problems which are not sym-

		NB		RX/OPT		LB/OI	γŢ	CPU		
N	p	assignment	1-arb	assignment	1-arb	assignment	1-arb	assignment	1-arb	
	1.0	1086.05	15.40	0.7823	0.8744	0.9167	0.9988	17.3349	16.3490	
	1.2	693.95	21.35	0.7858	0.8759	0.9128	0.9956	11.6626	22.4133	
15	1.4	497.85	19.55	0.8627	0.8411	0.9132	0.9968	8.7810	18.3576	
	1.6	364.10	23.20	0.7887	0.8230	0.9126	0.9924	5.9230	20.1919	
	1.8	263.00	29.25	0.7906	0.8057	0.9137	0.9931	4.8492	26.3282	
	2.0	232.65	30.90	0.7908	0.7898	0.9104	0.9910	3.5679	26.8591	
	1.0	(—)	56.65	(-)	0.8775	(-)	0.9935	(-)	128.7699	
	1.2	(-)	63.40	(-)	0.8584	(-)	0.9919	(-)	139.5196	
20	1.4	(—)	50.80	(-)	0.8414	(—)	0.9913	(—)	109.2427	
	1.6	()	79.70	()	0.8223	(—)	0.9853	(—)	150.4980	
	1.8	(—)	99.65	()	0.8048	(—)	0.9829	()	189.2531	
	2.0	(—)	100.70	(—)	0.7894	(—)	0.9866	(—)	190.8522	

Table B-1: Computational experiments with instances CONE

metric, yet closer to symmetric TSPs than those whose arc lengths are totally random. Our computational experiments for other types of "deformed" Euclidean TSPs show that our finding here would apply equally well to other such cases.

C Heuristic Algorithms for Instance SLOPE

Since instances SLOPE are based on the two-dimensional Euclidean TSPs, it appears natural to consider heuristic algorithms originally designed for the Euclidean TSPs. Heuristic algorithms for the Euclidean TSPs can be classified into two categories, namely, 1)tour construction procedures, in which a subtour is gradually expanded to come up with an initial tour, and 2)tour improvement procedures, where a tour is replaced with an another improved tour [11]. In many cases these two procedures are used in combination, as will be done here, too.

Tour construction procedures start with an initial subtour, and consist of two basic steps, a)node selection step which determines a node to be included in the current subtour, and b)node insertion step which decides where in the existing subtour the selected node will be inserted[11]. Repeated applications of a node selection and a node insertion steps would eventually produce a tour.

To come up with a heuristic algorithm for our instances, the following two approaches are considered as candidates for the node selection step:

(1)Farthest Insertion (FI) method, which selects a node farthest away from the current subtour, and

(2)Nearest Insertion (NI) method, which selects a node closest to the current subtour.

FI is known to work well for the Euclidean TSPs. See, e.g., Chapter 7 of [11]. NI, on the other hand, tries to include nodes easier to include in the existing subtour based on the geometric image.(Fig.2-1)

In order to adopt these methods to our asymmetric instances, we introduce the following

concepts of "distance":

(i) Distance between a subtour and a selected node is defined as the minimum of distances between those nodes in the subtour and the selected node.

(ii) Distance d_{ij} between two points *i* and *j* which is used in the selection step is calculated by one of the following three methods:

(C.1)
$$d_{ij} = \min(c_{ij}, c_{ji})$$
 - choose the shorter distance

(C.2)
$$d_{ij} = (c_{ij} + c_{ji})/2$$
 — use the average of the two distances

(C.3)
$$d_{ij} = \max(c_{ji}, c_{ji})$$
 - choose the longer distance

Criteria used in the insertion step is to choose the location such that the cost increase is kept minimum.

After an initial tour is obtained based on the framework just outlined, 3-opt will be performed to improve the tour, and the resultant tour will be used as an initial tour.

Table C-1 compares three different methods to come up with "distances" of tours obtained by the heuristic algorithms against the optimal value for the averages of 20 instances with N = 15 and p = 2.0.

Table C-1 shows that a combination of NI with 3-opt where distances are calculated

d_{ij}	FI	FI+3-opt	NI	NI+3-opt
(C.1)	1.174	1.124	1.079	1.063
(C.2)	1.078	1.057	1.051	1.042
(C.3)	1.073	1.056	1.112	1.061

Table C-1: The accuracy of heuristics

by (C.2) gives the best heuristic solutions, and thus the initial tours used for our algorithm is generated based on the above combination.

Seiji KATAOKA : Department of Computer Science, the National Defense Academy, 1-10-20, Hashirimizu, Yokosuka, Kanagawa 239 Japan

Susumu MORITO : Department of Industrial Engineering and Management, Waseda University, 3-4-1, Okubo, Shinjuku, Tokyo 169 Japan