Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

TOBIAS ACHTERBERG
THORSTEN KOCH
ANDREAS TUCHSCHERER

On the Effects of Minor Changes in Model Formulations

supported by the DGF Research Center Matheon

ZIB-Report 08-29 (August 2008)

On the Effects of Minor Changes in Model Formulations*

Tobias Achterberg† Thorsten Koch‡ Andreas Tuchscherer§

Abstract

Starting with the description of the Traveling Salesmen Problem formulation as given by van Vyve and Wolsey in the article "Approximate extended formulations", we investigate the effects of small variations onto the performance of contemporary mixed integer programming solvers. We will show that even minor changes in the formulation of the model can result in performance difference of more than a factor of 1000. As the results show it is not obvious which changes will result in performance improvements and which not.

1 Introduction

In their article [vVW06] "Approximate extended formulations" van Vyve and Wolsey describe a mixed integer programming (MIP) model for solving the Traveling Salesmen Problem (TSP). When we tried to reproduce the results we noticed an erratic behavior of the MIP solvers depending on minor variations of the model. In the following we will show that even very small changes in the formulation can have a large impact on the solvability. Trick shows in [Tri05] that with modern MIP solvers the effects of changes are hard to predict as today's solvers employ many methods to apply the usual tricks used to improve a formulation. In this article we will show that probably due to these automatic "improvements" the performance can get nearly unpredictable. In particular, omitting seemingly useful redundant information can sometimes speed up the solution process.

In the next section we will present the model as described in the original article and list possible variations. In Section 3 computational results for all combinations of the variations will be given and explained.

2 The model and its variations

The TSP instance is defined by a set of nodes $V = \{1, \ldots, n\}$, $n \in \mathbb{N}$, a set of arcs $A = \{(i, j) | i, j \in V, i \neq j\}$, and a distance (weight, cost) function c_{ij}, interpreted as the length of arc $(i, j) \in A$. The goal is to find a shortest round trip through all nodes $i \in V$.

2.1 The formulation by van Vyve and Wolsey

The formulation depends on an approximation parameter k which controls the extend of the subtour elimination constraints implied by the model. For each node $l \in V$, define a neighborhood $V_l \subseteq V$ as the set of the k nodes nearest to node l, including l itself. The model is given in terms

*Supported by the DFG Research Center Matheon *Mathematics for key technologies* in Berlin.
†ILOG Deutschland GmbH, tachterberg@ilog.de
‡Zuse Institute Berlin (ZIB), Takustr. 7, 14195 Berlin, Germany, koch@zib.de
§Zuse Instritute Berlin, tuchscherer@zib.de

of the following variables:

$$y_{ij} := \begin{cases} 1, & \text{if arc } (i,j) \in A \text{ is in the tour,} \\ 0, & \text{otherwise.} \end{cases}$$

$$u_i := \text{number of nodes visited before } i \in V \text{ (node 1 is visited first)}$$

$$w^l_{ij} := \text{flow on arc } (i,j) \in A \text{ for neighborhood } l \in V$$

Now the MIP formulation for the Traveling Salesman Problem reads as follows:

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij} \quad \text{s. t.} \tag{1}$$

$$\sum_{(j,i)\in\delta^-(i)} y_{ji} = 1 \quad \text{for all } i \in V \tag{2}$$

$$\sum_{(i,j)\in\delta^+(i)} y_{ij} = 1 \quad \text{for all } i \in V \tag{3}$$

$$y_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A \tag{4}$$

$$u_1 = 0 \tag{5}$$

$$u_i - u_j + (n-1)y_{ij} \leq n-2 \quad \text{for all } (i,j) \in A : j \neq 1 \tag{6}$$

$$u_i \geq 1 \quad \text{for all } i \in \{2,\dots,n\} \tag{7}$$

$$u_i \leq n-1 \quad \text{for all } i \in \{2,\dots,n\} \tag{8}$$

$$\sum_{(i,j)\in\delta^-(j)} w^l_{ij} - \sum_{(j,i)\in\delta^+(j)} w^l_{ji} = 0 \quad \text{for all } l \in V \text{ for all } j \in V_l : j \neq l \tag{9}$$

$$\sum_{(i,l)\in\delta^-(l)} w^l_{il} - \sum_{(l,i)\in\delta^+(l)} w^l_{li} = 1 \quad \text{for all } l \in V \tag{10}$$

$$w^l_{ij} \geq 0 \quad \text{for all } l \in V,\ (i,j) \in A(V_l) \cup \delta^-(V_l) \cup \delta^+(V_l) \tag{11}$$

$$w^l_{ij} \leq y_{ij} \quad \text{for all } l \in V,\ (i,j) \in A(V_l) \cup \delta^-(V_l) \tag{12}$$

Note that already the model (1)–(8) is a correct TSP formulation (Miller-Tucker-Zemlin formulation [PS91]). Constraints (9)–(12) are useful as they imply subtour elimination constraints as specified by

Theorem 1 (van Vyve, Wolsey)**.** *The subtour elimination constraint*

$$\sum_{(i,j)\in\delta^-(U)} y_{ij} \geq 1$$

is valid for (9)–(12) *if there exists* $l \in V$ *such that* $l \in U \subseteq V_l$*.*

Proof. Let l and U satisfying $l \in U \subseteq V_l$ be given. Equations (9)–(12) guarantee that there is enough capacity in y for one unit to flow from outside V_l to l. Thus the capacity of the cut $\delta^-(U)$ is at least 1. □

So far we have ignored one important issue: In case $k = n$, the model is infeasible since constraints (9)–(12) cannot be satisfied. In order to maintain feasibility, in the original formulation the neighborhood sets do not contain node 1, i. e., V_l is defined as the set of the k nodes nearest to node l, including l itself, but excluding node 1. However, removing node 1 from the sets V_l, for $l \in V$, degrades the solvability of the model for $k < n$ as it significantly reduces the initial lower bound. Since the whole point in using *approximate* extended formulations is to use $k < n$, we include node 1 in all computations.

2.2 Possible variations of the formulation

In the following we list some minor changes of the formulation. Regarding the model the restrictions are redundant and the relaxations only remove redundant constraints. In all cases neither the integer optimal solution is altered, nor the objective value of the linear programming relaxation is changed.

2.2.1 Restricting the formulation

1. The variables u_i have to be integer in any feasible solution.

$$u_i \in \{1, \ldots, n-1\} \quad \text{for all } i \in V \tag{13}$$

   can be added to the model.

2. There are no explicit upper bounds on the variables w^l_{ij}, even though all are implicitly bounded by inequality (12):

$$0 \leq w^l_{ij} \leq 1 \quad \text{for all } l \in V_l, (i,j) \in A(V_l) \cup \delta^-(V_l) \cup \delta^+(V_l) \tag{14}$$

3. The variables w^l_{ij} are also implicitly integer.

$$w^l_{ij} \in \mathbb{Z} \quad \text{for all } l \in V_l, (i,j) \in A(V_l) \cup \delta^-(V_l) \cup \delta^+(V_l) \tag{15}$$

   can be added to the model.

4. All w^l_{ij} in $(i,j) \in \delta^+(V_l)$ can be fixed to zero:

$$w^l_{ij} = 0 \quad \text{for all } l \in V_l, (i,j) \in \delta^+(V_l) \tag{16}$$

5. For the same reason upper bounds on w^l_{ij} in $(i,j) \in \delta^+(V_l)$ can be added in inequality (12), using instead:

$$w^l_{ij} \leq y_{ij} \quad \text{for all } l \in V,\ (i,j) \in A(V_l) \cup \delta^-(V_l) \cup \delta^+(V_l) \tag{17}$$

2.2.2 Relaxing the formulation

6. The upper bounds on the u_i variables are not needed, constraint (8) can be omitted.

7. For equations (9) and (10) equality is not required. Relaxing them to

$$\sum_{(i,j)\in\delta^-(j)} w^l_{ij} - \sum_{(j,i)\in\delta^+(j)} w^l_{ji} \geq 0 \quad \text{for all } l \in V \text{ for all } j \in V_l : j \neq l \tag{18}$$

$$\sum_{(i,l)\in\delta^-(l)} w^l_{il} - \sum_{(l,i)\in\delta^+(l)} w^l_{li} \geq 1 \quad \text{for all } l \in V \tag{19}$$

   is still valid.

3 Computational results

In this section we will report on the computational results obtained by solving all combinations of the above variations of one TSP instance with two contemporary MIP solvers.

All computing times are given as CPU seconds on a PC running Linux with a 3.6 GHz Pentium-D and 4 GB RAM. If not otherwise noted, all runs are limited to at most one hour. The instance used is *att48* from TSPLIB [Rei91]. $k = 13$ is used in all cases. The model was generated using ZIMPL[1] [Koc04] version 2.07. The source files can be found at http://www.zib.de/koch/reformulation. We used CPLEX[2] version 10.0.1 and SCIP[3] [Ach04] version 0.90e to solve the MIP instances. CPLEX and SCIP were run with default settings, with the exception that probing was disabled for SCIP (in CPLEX, only a very limited version of probing is applied by default). SCIP used CPLEX 10.0.1 as LP solver subroutine.

The formulation variants are denoted as *u-w-e-b-f*. Table 1 describes possible values and their meanings. The original formulation is denoted as *2-1-1-1-0*. Note that $u = 3$ and $w = 3$ mean the variable is an *implicit integer*, i. e., the variable will have an integral value for any solution of the LP relaxation where all normal integer variables have integral values. With ZIMPL and SCIP it is possible to notify the solver of this property. The solver has then the possibility to treat the variable alternatively as continuous or integer variable.

u		Description
1	$1 \le u_i \le \infty$	Relaxation 6 used (Constraint (8) omitted)
2	$1 \le u_i \le n-1$	as in the original formulation
3	$1 \le u_i \le n-1$	but declared as *implied integer* (SCIP only)
4	$u_i \in \{1, \ldots, n-1\}$	Restriction 1 (declared integer)
5	$u_i \in \mathbb{N}$	Restriction 1 and Relaxation 6
w		
1	$0 \le w_{ij}^l \le \infty$	as in the original formulation
2	$0 \le w_{ij}^l \le 1$	Restriction 2 (explicit upper bounds)
3	$0 \le w_{ij}^l \le 1$	Restriction 2 and declared as *implicit integer* (SCIP only)
4	$w_{ij}^l \in \{0, 1\}$	Restriction 2 and 3 (declared binary)
5	$w_{ij}^l \in \mathbb{N}$	Restriction 3 (declared integer)
e		
0		Relaxation 7 (not requiring equality for (9) and (10))
1		as in the original formulation
b		
0		Restriction 5 (bind outgoing w_{ij}^l to y_{ij})
1		as in the original formulation
f		
0		as in the original formulation
1		Restriction 4 (fix outgoing w_{ij}^l to zero)

Table 1: Possible values of formulation variants denoted as *u-w-e-b-f*.

[1] http://zimpl.zib.de
[2] http://www.ilog.com/cplex
[3] http://scip.zib.de