Tolerances in Lagrangian relaxation and Branch and Bound

Marcel Turkensteen

Center of Operations Research Applications in Logistics
Department of Business Studies and Business Studies
Aarhus School of Business, Aarhus University
matu@asb.dk

May, 2011

Introduction

- This presentation is about two types of frequently used methods for Combinatorial Optimization Problems, namely Lagrangian relaxation and Branch and Bound (BnB).
- In addition, we discuss the computation and use of tolerances.
- The example problems are the Asymmetric Traveling Salesman Problem (ATSP) for BnB and the Degree-Constrained Minimum Spanning Tree Problem (DCMSTP) for Lagrangian relaxation.
- Based on the papers [14], [8] and a paper in progress.

The graph-theoretical notation

In order to formally define tolerances, the ATSP and AP are considered within the framework of the following combinatorial minimization problem; see [10]. $(\mathcal{E}, C, \mathcal{D}, f_C)$ is the problem of finding

$$S^* \in \arg\min\{f_C(S) \mid S \in \mathcal{D}\},$$

where $C : \mathcal{E} \rightarrow \Re$ is the given *instance* of the problem with a *ground set* $\mathcal{E}$ satisfying $|\mathcal{E}| = m$ $(m \geq 1)$, $\mathcal{D} \subseteq 2^{\mathcal{E}}$ is the *set of feasible solutions*, and $f_C : 2^{\mathcal{E}} \rightarrow \Re$ is the *objective function* of the problem. By $\mathcal{D}^* = \arg\min\{f_C(S) \mid S \in \mathcal{D}\}$ the set of optimal solutions is denoted.

The Asymmetric Traveling Salesman Problem

For the ATSP, we use the undirected graph $G = (V, A, C)$ with set of vertices V, set of arcs A and cost matrix C.
The Asymmetric Traveling Salesman Problem (ATSP) is the problem of constructing a tour through n locations such that every location is visited exactly once.
It is asymmetric in the sense that the distance from i to j may not be equal to the distance from j to i.

The Assignment Problem relaxation

The Assignment Problem (AP) is also defined on the graph $G = (V, A, C)$.
The AP is the problem of assigning each job j to worker i such that each worker performs one job.
A feasible solution is a set of subcycles in G.
The AP can be solved in $O(n^3)$ time with the Hungarian algorithm.

The Degree-Constrained Minimum Spanning Tree

Given a directed graph $G = (V, E, C)$.
The Degree-Constrained Minimum Spanning Tree Problem (DCMSTP) is the problem of connecting a set of n nodes or vertices in a network with edges in E and edge weights given in C.
The number of edges adjacent to some or all vertices, the *vertex degrees*, are limited.
Relaxation: the Minimum Spanning Tree Problem (MSTP).

The Degree-Constrained Minimum Spanning Tree

Given a directed graph $G = (V, E, C)$.
The Minimum Spanning Tree Problem (MSTP) is the problem of connecting a set of n nodes or vertices in a network with edges in E and edge weights given in C.
The MSTP can be solved in $O(n^2)$ time with Prim's algorithm.

Tolerances

- The tolerances with respect to an optimal solution are, roughly spoken, the change in solution value if an element is included or excluded from a solution.
- Tolerances are used in sensitivity analysis to obtain the sensitivity of the optimal solution at hand to changes in parameter values.
- For now, we consider minimization problems.
- An upper tolerance value is the increase in an element's cost value before the optimal solution changes.
- Likewise, a lower tolerance value is the largest decrease in an element's cost value before the optimal solution changes.

Definition of tolerances (MSTP)

The following formal definition of upper tolerances is taken from [9] and adapted to the MSTP. For any graph $G = (V, E, C)$, we define $\mathcal{T}^*$ as the set of MSTs on the graph G.

Definition

Let C_ϵ be the cost matrix of the MSTP such that $c_\epsilon(e) = c(e)$ for $e \in E \setminus \{e\}$ and $c_\epsilon(e) = c(e) + \epsilon$. Then the upper tolerance value of e with respect to any $T^* \in \mathcal{T}^*$ is defined by and denoted as $u_{T^*}(e) = \sup\{\epsilon \in \Re : \ T^* \textit{ is an optimal solution of the MSTP on } (V, E, C_\epsilon)\}$.

Computing upper and lower tolerances

Define $\mathcal{T}_{-}^{*}(e)$ as the set of optimal solutions to the MSTP on the graph $G = (V, E \setminus \{e\}, C)$.

Computation of the upper tolerances: the upper tolerance value $u_{T^*}(e)$ of any edge $e \in T^*$ equals $u_{T^*}(e) = f_C[T_{-}^{*}(e)] - f_C[T^*]$ for each $T_{-}^{*}(e) \in \mathcal{T}_{-}^{*}(e)$.

A lower tolerance value is computed with the solution with the edge e.

The same holds for the AP.

Computing upper and lower tolerances (2)

- For the MSTP, an upper tolerance value of an edge e in T^* can be computed by removing the edge from the tree and finding and adding the minimum cost edge between the two components.
- A lower tolerance value of an edge e outside T^* can be computed by adding the edge and finding and removing the maximum weight edge in the resulting cycle (minus e).
- For the AP, an upper and lower tolerance value can be computed with an additional step of the Hungarian algorithm.
- There are results for both the AP [16] and the MSTP [5] that show that multiple tolerance values can be computed fast.

Solving the ATSP

Effective methods for solving the ATSP are:

- The Concorde Solver for the STSP.
- The Branch and Bound algorithm by [3].
- Tolerance-based BnB algorithms.

Branch and Bound methods have been effective for the ATSP, but also for many other COPs.

Branch and Bound

- Branch and Bound is a methodology for (generally) NP-hard Combinatorial Optimization Problems.
- A BnB algorithm solves an easily solvable version of the problem, a *relaxation*, first.
- If the relaxation solution is infeasible for the original problem, divide the problem up into new subproblems.
- Continue solving all subproblems until there are no more left.

Ingredients of Branch and Bound

For a minimization problem, the elements of BnB are:

- A lower bound to all solutions of a subproblems;
- An upper bound, usually the best solution found so far;
- The search strategy: the order in which the open subproblems are searched through;
- The branching rule, specifying how the current subproblem should be divided into new subproblems.

Upper bounds

An upper bound is usually the value of the best solution found so far.
Such a value is e.g. the solution value of a solved subproblem, or it can be determined with a heuristic.

Lower bounds

A lower bound indicates what the minimum value of any solution of a (sub-)problem is.
If the lower bound value is higher than or equal to the upper bound value, we can fathom the (sub-)problem.

The branching rule

If a subproblem is not solved or fathomed, it is divided into mutually disjoint subproblems.
These subproblems are added to the list of *open* subproblems.
For the AP, one often takes the shortest cycle in the AP solution and remove each of the arcs in the solution.
First, select an arc e_1 from the cycle and remove it, then remove arc e_2 and include e_1, etc.
This is the branching rule from [4].

The search strategy

The search strategy determines the order in which the open subproblems are solved.

The best first search (BFS) strategy solves the most promising open subproblem first, i.e., the subproblem with the largest lower bound value.

The depth first strategy (DFS) solves the most recently generated subproblem first.

Then the question is: how to order the new subproblems generated in a branching step?

The role of upper tolerances in BnB

Upper tolerances can both be used in a branching rule and in a lower bound.

In DFS algorithms, it is important to decide on which arcs to branch on correctly.

Most BnB strategies select the arc with the highest cost first for exclusion.

We suggest to select the lowest upper tolerance value for exclusion.

Claim: the upper tolerance value is a more likely indicator of whether an arc belongs to an optimal ATSP solution than its cost.

A lower bound for the ATSP

Christofides lower bound: use reduced costs and shrink the cycles into vertices, determine the costs of connecting the cycles. ([6])

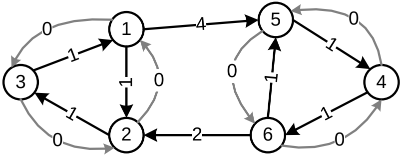


Figure: AP solution (light color)

Bottleneck upper tolerances

In the picture, at least one arc must be removed from each cycle.
Check that the upper tolerance in the left hand cycle (3,2,1) is 3 for each arc.
Then breaking the cycle costs at least 3.

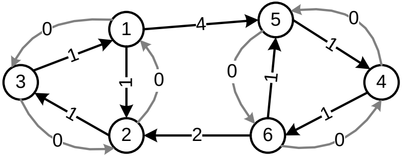


Figure: AP solution (light color)

The role of upper tolerances: Lower Bound

The upper tolerance value of each arc in the AP solution is 3.
The current AP solution has value 0.
At least one arc needs to be removed from the cycle, which makes the AP solution 3 units more expensive.
Claim: the lower bound to a solution, based on upper tolerances, is 3.

Lower tolerances

In the picture, cycles can be reversed or connected.
Check that the lower tolerances of arcs reversing the orientation of the cycles is 3.
Check that the lower tolerance values of both arcs *between* the cycles are 10.
Claim: connecting the cycles costs at least 10.

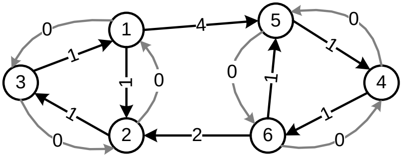


Figure: AP solution (light color)

The use of upper and lower tolerances

The use of upper tolerances only [14], both for lower bounds and branching rules.
The use of lower tolerances in addition [8], only for lower bounds.
We find that it leads to improvements in DFS BnB methods.
There are similar uses in Linear Programming (strong branching) and in heuristics (e.g. the modified Lin-Kernighan heuristic).

Lagrangian relaxation

- Lagrangian relaxation is a solution method that is often used for determining lower bounds.
- Some constraints are penalized in the objective function and the remaining problem is solved.
- Then update the penalty values until a sufficiently good / optimal value is obtained.
- The application problem is the Degree Constrained Minimum Spanning Tree Problem (DCMSTP).

Lagrangian relaxation in general

A problem:

$$\min f(x) \ s.t. \ g(x) \leq b$$

is represented as:

$$\max_{\lambda} \min_{x} f(x) + \lambda \times (g(x) - b)$$

where λ is a vector of penalty parameters.
Optimality conditions are:

$$\begin{aligned} i \quad & g(x) \leq b & (1) \\ ii \quad & \lambda_i(g_i(x) - b_i) = 0 \ \forall i & (2) \\ & & (3) \end{aligned}$$

Applications of Lagrangian relaxation

Lagrangian relaxation has been applied in:

- Held-Karp algorithm for the Symmetric Traveling Salesman Problem;
- Several location problems;
- Capacitated MSTPs.

Performing Lagrangian relaxation

- The decision variables in Lagrangian Relaxation are the Lagrangian multipliers λ.
- Try to set them in such a way that the optimality conditions hold.
- Start at some value λ^0 and adapt the multipliers in an iterative process.
- Reference: [7].

Lagrangian relaxation for Linear Programming Problems

Take the following LP problem:

$$
\begin{aligned}
\min \quad & c^T x && (4)\\
s.t. \quad & A_1 x \leq b_1 && (5)\\
& A_2 x \leq b_2 && (6)\\
& && (7)
\end{aligned}
$$

Solve it as the following Lagrangian relaxation problem:

$$
\begin{aligned}
\max_{\lambda} \min_{x} \quad & c^T x + \lambda(b_2 - A_2 x) && (8)\\
s.t. \quad & A_1 x \leq b_1 && (9)\\
& && (10)
\end{aligned}
$$

One of the exercises is to show that the optimality conditions correspond to complementary slackness.

Lagrangian relaxation: numerical example

$$min \quad 10x + 10y \tag{11}$$
$$s.t. \quad x \geq 10 \tag{12}$$
$$x, y \geq 0 \tag{13}$$

Becomes:

$$\max_{\lambda} \min_{x,y} \quad 10x + 10y + \lambda(10 - x) \tag{14}$$
$$x, y \geq 0 \tag{15}$$

Lagrangian relaxation: numerical example (2)

$$\max_{\lambda} \min_{x,y} \quad 10x + 10y + \lambda(10 - x) = \tag{16}$$

$$\max_{\lambda} \min_{x,y} \quad (10 - \lambda)x + 10y + 10\lambda \tag{17}$$

Solution for: $\lambda = 0 \rightarrow (x, y) = (0, 0)$
Value 0.

Lagrangian relaxation: numerical example (3)

$$\max_{\lambda} \min_{x,y} \quad 10x + 10y + \lambda(10 - x) = \tag{18}$$

$$\max_{\lambda} \min_{x,y} \quad (10 - \lambda)x + 10y + 10\lambda \tag{19}$$

Optimal solution for: $\lambda = 5 \rightarrow (x, y) = (0, 0)$
Value 50.

Lagrangian relaxation: numerical example (4)

$$\max_{\lambda} \min_{x,y} \quad 10x + 10y + \lambda(10 - x) = \tag{20}$$

$$\max_{\lambda} \min_{x,y} \quad (10 - \lambda)x + 10y + 10\lambda \tag{21}$$

Optimal solution for: $\lambda = 12 \rightarrow (x, y) = (\infty, 0)$
Value $-\infty$.

Lagrangian relaxation: numerical example (5)

$$\max_{\lambda} \min_{x,y} \quad 10x + 10y + \lambda(10 - x) = \tag{22}$$

$$\max_{\lambda} \min_{x,y} \quad (10 - \lambda)x + 10y + 10\lambda \tag{23}$$

Optimal solution for: $\lambda = 10 \rightarrow (x, y) = (?, 0)$ and value 100.
Note that the optimality conditions are satisfied if $(x, y) = (10, 0)$.

The Degree Constrained Minimum Spanning Tree Problem (DCMSTP)

The DCMSTP is frequently ecountered in network design problems. Solution approaches are:

- Meta-heuristics, such as VNS [12].
- Branch and Bound, Branch and Cut [2, 13].
- Lagrangian relaxation [1, 15].

Lagrangian relaxation is often used to generate lower bounds in other algorithms.

Lagrangian relaxation and the DCMSTP

The DCMSTP is formulated as:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{e\in E} c_e x_e && (24)\\
s.t \quad & && (25)\\
& x(E) = |V| - 1 && (26)\\
& x(S) \leq |S| - 1,\ S \subset V, S \neq \emptyset && (27)\\
& x(\delta(i)) \leq b_i,\ \forall i \in V && (28)\\
& x_e \geq 0 && (29)
\end{aligned}
$$

Here, $x(\delta(i))$ denotes the number of edges adjacent to vertex i in the solution.
We bring the following constraints into the objective function: $x(\delta(i)) \leq b_i,\ \forall i \in V$.

The Lagrangian relaxation problem

$$LRP(\lambda) = \max_{\lambda} \sum_{(i,j)\in E} (c_{ij} + \lambda_i + \lambda_j)x_{ij} - \sum_{i \in V} \lambda_i \times b_i; \qquad (30)$$

where x should correspond to an MST.
A solution is optimal if 1) the solution is an MST; 2) the degree-constraints are non-violated; 3) the Lagrangian multiplier value λ_v of a vertex $v \in V$ is only positive if the degree of v is equal to the maximum degree.
We modify the penalty values or *Lagrangian multipliers* λ_v in order to achieve or approach optimality.

Updating of Lagrangian multiplier values

- The Lagrangian multipliers are updated in an iterative process, so we have $\lambda_v^1, \lambda_v^2, \ldots$ for each $v \in V$.
- In existing approaches (e.g. subgradient method by [11]), penalty values are found by a converging series of steps.
- For the DCMSTP, the penalty value depends on a quantity t^k in step k of the process, converging to 0 according to the relation $t^{k+1} - 2t^k + t^{k-1} = 0$.
- Part of the step size depends on the amount of degree violation in step k.

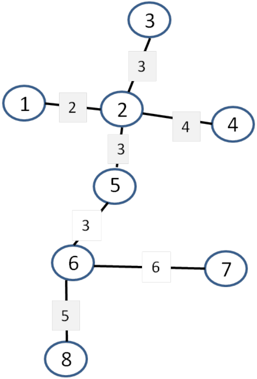
Lagrangian multipliers and tolerances

By how much can we increase each λ_v, while guaranteeing at least b_v adjacent edges in the resulting tree?
Upper tolerance values can be used to estimate Lagrangian multiplier values with the following theorem.

Theorem

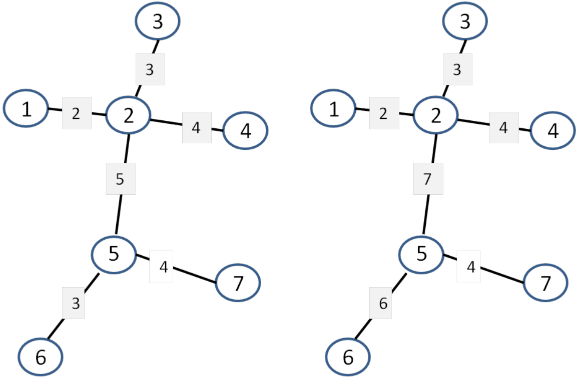
For a given MST T^* on the graph $G = (V, E, C)$, we set $\lambda_v \geq 0$ for each $v \in V$ and create the cost matrix C' with $c'(v, w) = c(v, w) + \lambda_v + \lambda_w$.
For each $(v, w) \in E$ with $u_{T^*}(v, w) > \lambda_v + \lambda_w$, it holds that $(v, w) \in T'_\lambda$, where T'_λ is an MST for the cost matrix C'.

Possible increase of λ



The example shows an MST with upper tolerance values on the edges.
We assume that $b_v = 2$ for each vertex v. Take a single 'isolated' violating vertex, e.g. vertex 2, and note that the values of λ_w of all its neighboring vertices in the tree remain 0.
An exercise in the seminar is to set λ_2.

Possible increase of λ (2)



Results of upper tolerance-based multiplier setting

We find that the use of upper tolerances for setting *initial* Lagrangian multiplier values lead to much tighter lower bounds, in particular after a small number of iterations in the subsequent Lagrangian relaxation approach.

Conclusions

- We have discussed the example problems were the ATSP and the DCMSTP.
- We have discussed upper and lower tolerance computations.
- We have discussed Branch and Bound and Lagrangian relaxation and applied them to the example problems.
- We have explained how upper tolerances can play a role within the methods.
- The seminar will contain exercises on these topics.

R. Andrade, A. Lucena, and N. Maculan.
Using Lagrangean Dual Information to Generate Degree Constrained Minimum Spanning Trees.
Discrete Applied Mathematics, 154:703–714, 2006.

L. Caccetta and S.P. Hill.
A Branch and Cut Method for the Degree-Constrained Minimum Spanning Tree Problem.
Networks, 37:74–83, 2001.

G. Carpaneto, M. Dell'Amico, and P. Toth.
Exact Solution of Large-scale Asymmetric Traveling Salesman Problems.
ACM Transactions on Mathematical Software, 21(4):394–409, 1995.

G. Carpaneto and P. Toth.
Some New Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem.
Management Science, 21:736–743, 1980.

F. Chin and D. Houck.

Algorithms for Updating Minimal Spanning Trees.
Journal of Computer and System Sciences, 16:333–344, 1978.

N. Christofides.
Graph Theory: An Algorithmic Approach. Academic Press.
1975.

M.L. Fisher.
The Langrangian Relaxation Method for Solving Integer Programming Problems.
Management Science, 27(1):1–18, 1981.

R. Germs, B. Goldengorin, and M. Turkensteen.
Lower tolerance-based branch and bound algorithms for the atsp.
Computers and Operations Research, page Corrected Proof, 2011.

B. Goldengorin, G. Jäger, and P. Molitor.
Tolerances Applied in Combinatorial Optimization.
Journal of Computer Science, 2(9):716–734, 2006.

B. Goldengorin, G. Sierksma, and M. Turkensteen.

Tolerance based algorithms for the ATSP.
Lecture Notes in Computer Science, 3353:222–234, June 2004.

M. Held, P. Wolfe, and H.P. Crowder.
Validation of the Subgradient Method.
Mathematical Programming, 6:62–88, 1974.

C. Ribeiro and M.C. Souza.
Variable Neighborhood Search for the Degree-Constrained Minimum Spanning Tree Problem.
Discrete Applied Mathematics, 118:43–54, 2002.

M. Savelsbergh and A. Volgenant.
Edge Exchanges in the Degree-Constrained Minimum Spanning Tree Problem.
Computers & Operations Research, 12 (4):341–348, 1985.

M. Turkensteen, D. Ghosh, B. Goldengorin, and G. Sierksma.
Tolerance-Based Branch and Bound Algorithms for the ATSP.
European Journal of Operational Research, 189(3):775–788, 2008.

A. Volgenant.
A Lagrangean Approach to the Degree-Constrained Minimum Spanning Tree Problem.
European Journal of Operational Research, 39:325–331, 1989.

A. Volgenant.
An Addendum on Sensitivity Analysis of the Optimal Assignment.
European Journal of Operational Research, 169:338–339, 2006.