Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks

Di Yuan and Dag Haugland

Supplementary Material

N.B.: When citing this work, cite the original article.

©2012 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

http://dx.doi.org/10.1109/TMC.2011.231
Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-85844
Appendix: Computing the best 1-edge exchange

**Algorithm 1** 1-edge exchange($T$)

**Require:** A spanning tree $T$

**Ensure:** Returns leaving and entering edges $(k^*, l^*)$ and $(m^*, n^*)$ and the new power $P^*$ in the best 1-edge exchange. If no improving move exists, dummy-edges and the current power are returned.

$p(i, s) ← \max_j \{ p_{ij} : (i, j) ∈ T^* \} \forall i, s ∈ V$

$(k^*, l^*, m^*, n^*, P^*) ← (0, 0, 0, 0, \sum_{s ∈ V} \sum_{i ∈ V} p(i, s))$ // No improving move is found

// Try all possible edge removals:

for all $(k, l) ∈ T$

$p'(i, s) ← p(i, s) \forall i, s ∈ V$

for all $s ∈ T_k$

$p'(k, s) ← \max_j \{ p_{kj} : (k, j) ∈ T^*_k \}$

for all $s ∈ T_l$

$p'(l, s) ← \max_j \{ p_{lj} : (l, j) ∈ T^*_l \}$

// (i) Power needed for internal forwarding of messages from internal sources:

$P_{EI}^I ← \sum_{i ∈ T_k} \sum_{s ∈ T_k} p'(i, s) + \sum_{i ∈ T_l} \sum_{s ∈ T_l} p'(i, s)$

// (ii) Power needed for internal forwarding of messages from external sources:

for all $m ∈ T_k$

$P_{m}^{EI} ← \sum_{i ∈ T_k} |V(T_i)| p'(i, m)$

for all $n ∈ T_l$

$P_{n}^{EI} ← \sum_{i ∈ T_l} |V(T_i)| p'(i, n)$

// (iii) Power increment needed for external forwarding of messages from internal sources:

for $U ← T_k, T_l$

for all $m ∈ U$

$N_m ← 0, \bar{p}_{m1} ← 0, \bar{p}_{m2} ← 0$ // Correct if $|V(U)| = 1$

if $|V(U)| > 1$ then

Find $m_1 ∈ \arg \max_{m} \{ p_{mi} : (m, i) ∈ U \}$ // Most power-demanding old neighbor

$\bar{p}_{m1} ← p_{m,m_1}$ // The corresponding power

$\bar{p}_{m2} ← \max_{i} \{ p_{mi} : (m, i) ∈ U, i ≠ m_1 \}$ // Second most, if any

$N_m ← |V(U) \cap V(T_{m_1})|$ // Counting sources requiring power $\bar{p}_{m2}$

for all $m ∈ T_k$

for all $n ∈ T_l$

$P_{EI}^T ← P_{EI}^T(T_k, p_{mm}, \bar{p}_{m1}, \bar{p}_{m2}, N_m) + P_{EI}^T(T_l, p_{mn}, \bar{p}_{n1}, \bar{p}_{n2}, N_n)$

// Check quality of the move $(k, l, m, n)$:

if $P_{EI}^I + P_{m}^{EI} + P_{n}^{EI} + P_{EI}^T < P^*$ then

$(k^*, l^*, m^*, n^*, P^*) ← (k, l, m, n, P_{EI}^I + P_{m}^{EI} + P_{n}^{EI} + P_{EI}^T)$

return $(k^*, l^*, m^*, n^*, P^*)$
Algorithm 2 \pie(U, p_{mn}, \bar{p}_{m1}, \bar{p}_{m2}, N_m)

Require: A tree $U$, power $p_{mn}$ of the new edge, the two largest power demands $\bar{p}_{m1}$ and $\bar{p}_{m2}$ of edges incident to node $m$, number $N_m$ of sources demanding power $\bar{p}_{m2}$ at $m$.

Ensure: Returns power increment necessary for external forwarding of internal messages at node $m$.

$p_{IE} \leftarrow 0$ // No power increment needed for $(m, n)$ so far

if $p_{mn} > \bar{p}_{m2}$ then
  $P_{IE} \leftarrow N_m (p_{mn} - \bar{p}_{m2})$ // $N_m$ sources ask for increment from $\bar{p}_{m2}$ to $p_{mn}$
if $p_{mn} > \bar{p}_{m1}$ then
  // All but $N_m$ sources in $U$ ask for increment from $\bar{p}_{m1}$ to $p_{mn}$
  $P_{IE} \leftarrow P_{IE} + (|V(U)| - N_m) (p_{mn} - \bar{p}_{m1})$

return $P_{IE}$