The Efficient Determination of Optimal Traffic Flow Layouts with Respect to both Transportation Cost and Road Charges

Peter H. Richter, NAVIGON AG, Germany

Abstract: We present an efficient algorithm for the Traffic Flow Layout Problem TFLP that approximately minimizes both traveling expenses (time, length) and traffic intensity dependant road charges (capacity, noise, exhaustion, traffic calming, ..). The solution can be seen as a hybrid of the Quadratic Semi-Assignment Problem (QSAP) and the General Steiner Problem in Graphs (GSPG). The results below convincingly reveal that the opinion "Approximate algorithms solving either QSAP or GSPG may be sufficient enough to minimize layouts with respect to routing cost and intensity dependant road charges" does not hold at all, as long as both cost approximately contributes to the total. The performance analysis shows that the proposed algorithm outperforms surprisingly well the corresponding algorithms approximately solving QSAP and GSPG and leads to a considerable total cost reduction attractive for traffic optimization applications.

Keywords: Traffic Flow Layout Optimization, Quadratic Semi-Assignment Problem, General Steiner Problem in Graphs

1. Introduction

Traffic flow planners have to consider that the optimization for minimum length (ore time) traffic flow layouts is not only a matter of shortest or fastest paths (cost criterion "**routing cost**" = "travel expense" C_1): The traffic intensity of all the flows themselves traversing a road constitutes a second cost criterion "**road charges**" C_2 making a pure C_1-layout the less attractive the more C_2 increases (noise, pollution, capacity, toll, traffic calming, security risk (combat areas),..). The forthcoming algorithm tackles this problem. It is essentially based on a path finding strategy that provokes flow line attraction /repulsion dependent on the road charge (parameter δ, below).

Table 1 compares the Traffic Flow Layout Problem **TFLP** with **QSAP**, **GSPG**, and **GTPG**. **QSAP** is known to be ***NP***-hard [19]. Lower bounds for it have been found in [14,8,9]. In [1] polynomial classes for QSAP are presented. A vivid comparison **QSAP** and **QAP** as well as a **QSAP** algorithm based on a maximum spanning tree determination is given in [17,18]. In this paper we describe in depth the algorithm suited for direct implementation and show its results and performance regarding the different cost influence to travel expense and road charge dependent on the flow intensity and length specific road charge cost. The algorithm is in demand because it regards the simultaneous minimization of both

Routing Cost = driving time + stop time at traffic points and

Road Charges as to noise, pollution, capacity, toll, traffic calming, security risk (combat areas),.....

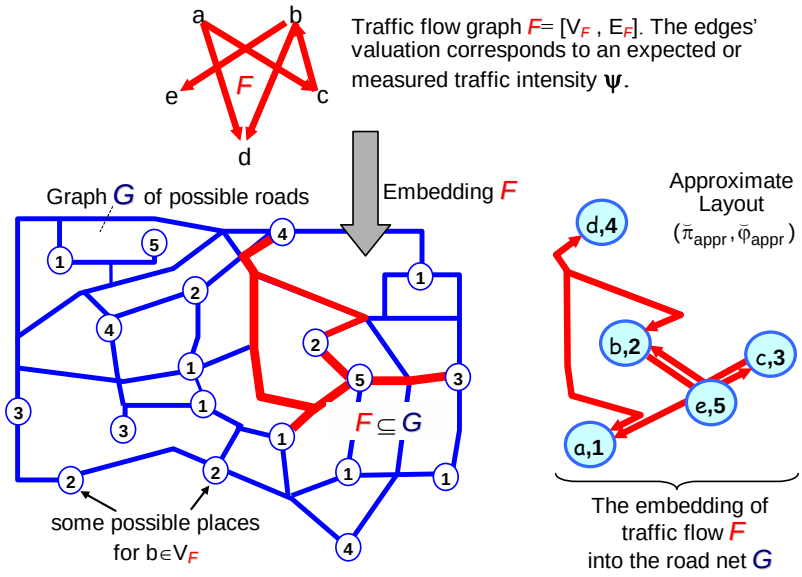

Figure 1 Depicting the task of algorithm **PR-1**

Concepts

F traffic flow graph, F= [sources/ sinks V_F, flow E_F], F is to be embedded into the road net G.

G road net graph, G= [traffic points V_G, roads between traffic points E_G] G is to embark traffic flow F.

ψ flow intensity. $\psi: E_F \cup E_G \rightarrow \boldsymbol{R}_+$ =. For flow $\lambda \in E_F$ and road $r \in E_G$ is $\psi(\{\lambda\} \cup \{r\}) = \psi(\lambda) + \psi(r)$.
$r \in E_G \Rightarrow \psi(r)$ is the flow intensity of road r before embedding F into G;
$\lambda \in E_F \Rightarrow \psi(\lambda)$ is the flow intensity of flow λ to be embedded into G.

ℓ length of the roads. $\ell: E_G \rightarrow \boldsymbol{R}_+$.

Π placement rule, $\Pi: V_F \rightarrow \mathcal{P}(V_G)$. $\Pi(s) \subseteq V_G$ is the set of possible traffic points to which source / sink s might be assigned. $s \in V_F \wedge p \in \Pi(s) \subseteq V_G \Rightarrow$ p is an admissible traffic point for placing source / sink $s \in V_F$.
$|\Pi(s)| = 1 \Rightarrow$ embedding traffic flow lines into road sequences,
$|\Pi(s)| > 1 \Rightarrow$ placing sinks /sources to traffic points and embedding traffic flow lines into road sequences.

π placement, $\pi: V_F \to V_G$, $\pi \subseteq \Pi$. We look for placement π that enables a subsequent routing φ_π to be minimal.

φ_π routing, $\varphi_\pi: E_F \to \mathfrak{P}(E_G)$ is a total function called of flow graph F in road net G considering placement π. $\varphi_\pi((s,t)) \subseteq E_G$ describes a sequence of roads from $\pi(s)$ to $\pi(t)$ embedding flow (s,t).

(π, φ_π) is called traffic layout. φ_π embeds the flow lines E_F into G such that each $\lambda=(s,t) \in E_F$ is mapped into a path $\varphi_\pi(\lambda) \subseteq E_G$ starting at $\pi(s)$ and ending in $\pi(t)$.

κ covering, $\kappa = \varphi_\pi^{-1}: E_G \to \mathfrak{P}(E_F)$ [1] is a partial function denoting which flow lines $\kappa(r) \subseteq E_F$ are conducted trough road $r \in E_G$. $\kappa = \varphi_\pi^{-1}$ is introduced to facilitate the description of algorithm PR-1.

$\oplus$ merging operation. For two coverings κ' and κ'' $\oplus$ is defined to $\kappa' \oplus \kappa'' = \{\kappa'(r) \cup k''(r))\}: \forall r \in \boldsymbol{dom}(\kappa') \cup \boldsymbol{dom}(\kappa'')\}$. [2] $\oplus$ facilitates the PR-1 description.

δ length specific intensity penalty function, $\delta: E_G \times \mathfrak{P}(E_F) \to \boldsymbol{R}_+$. With flow lines $\Lambda_1 \subseteq E_F$ and $\Lambda_2 \subseteq E_G$, $\delta(r, \Lambda_1 \cup \Lambda_2) = \psi(\Lambda_1) + \psi(\Lambda_2) + \psi(r)$ gives the length specific road charge coming from flow intensity $\psi(\Lambda_1) + \psi(\Lambda_2)$ and existing flow $\psi(r)$. δ may additionally consider further influences that indirectly depend from the traffic intensity due to embedding F into G.. For some $r \in E_G$ and two coverings κ' and κ'' it holds: $\kappa'(r) \subset \kappa''(r) \Rightarrow \delta(r, \psi(\kappa'(r))) < \delta(r, \psi(\kappa''(r)))$. δ is to set by traffic planners in the interest of traffic congestion prevention, traffic calming, traffic preference, limiting pollutant / noise level, etc.

$\hat{c}$ admissible traffic intensity, $\hat{c}: E_G \to \boldsymbol{R}_+$ gives the roads' maximum intensity (directed; road users per time unit).

$\bar{v}$ roads' civil speed $\bar{v}: E_G \to \boldsymbol{R}_+$ gives the.

μ roads' dwell time (stop time) $\mu: E_G \to \boldsymbol{R}_+$ denotes the. $\mu((x, y))$ is the dwell time at traffic point y if a road user wants to pass the road (x, y) from x to $y \in V_G$.

2. Traffic Flow Layout Problem TFLP

A traffic layout (π,φ_π) is called admissible if holds:

(i) $\pi \subseteq \Pi$ uniquely assigns each entity $s \in V_F$ a place $\pi(s) \in \Pi(s) \subseteq V_G$.

(ii) $\lambda=(s, t) \in E_F \Rightarrow \varphi_\pi(\lambda)$ describes a path from $\pi(s)$ to $\pi(t) \in V_G$ embedding (s, t) into G.

(iii) $\forall r \in \boldsymbol{rng}(\varphi_\pi): [(\sum_{\forall \lambda \in \varphi_\pi^{-1}(r)} \psi(\lambda)) \le \hat{c}(r)]$ [2] .

(2.1)

Problem Definition TFLP:

We look for an admissible layout (π,φ_π) that minimizes the objective function $\vec{C}(\pi,\varphi_\pi)$, here with criterion time:

$$\vec{C}(\pi,\varphi_\pi) = \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \left(\frac{\ell(r)}{\bar{v}(r)} \left(\psi(r) + \sum_{\lambda \in \varphi_\pi^{-1}(r)} \psi(\lambda)\right) \cdot + \mu(r) + \frac{\ell(r)}{\bar{v}(r)} \cdot \delta(r, \varphi_\pi^{-1}(r)) \right). \quad (2.2)$$

$\vec{C}_1$: Routing Cost: $\vec{C}_{11}$ road use time + $\vec{C}_{12}$ stop time $\mu(r)$

$\vec{C}_2$: Road Charge

Any layout (π,φ_π) that minimizes $\vec{C}(\pi,\varphi_\pi)$ above is called "optimal traffic flow layout". For applications preferring the length-criterion the expression $\vec{C}_{12} = \mu(r)$ is removed and $\bar{v}(r)$ is 1.

TFLP is ***NP***-hard.

Proof: We regard only $\vec{C}_1$ and introduce metric $\boldsymbol{d}: V_G^2 \to \boldsymbol{R}_+$. $\boldsymbol{d}(\pi(s),\pi(t))$ denotes a shortest distance between source $s \in V_F$ on $\pi(s)$ and sink $t \in V_F$ on $\pi(t)$ for $\pi \subseteq \Pi$. Simplifying $\vec{C}_1$ by $\vec{C}_1'$ we need only to look for an optimal placement π^* sufficing the *Quadratic Semi-Assignment Problem* **QSAP** [8,9,14,17].:

$$\textbf{QSAP:} \quad \vec{C}_1'(\pi^*) = \min_{\substack{\forall \pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{\vec{C}_1'(\pi) = \sum_{\lambda=(s,t) \in E_F} \psi(\lambda) \cdot \boldsymbol{d}(\pi(s),\pi(t))\}. \quad (2.3)$$

TFLP $\in$ ***NP*** because a right solution so far guessed is verifiable in polynomial time. Since **QSAP** is ***NP***-hard] it follows **QSAP** $\propto$ **TFLP**. [3] ■

---

[1] $\varphi_\pi^{-1} \subseteq E_G \times E_F$ denotes the reverse mapping of $\varphi_\pi \subseteq E_F \times E_G$.

[2] The "domain" $\boldsymbol{dom}(f)$ of a mapping $f: A \to B$ is the set $\{a \in A: \exists (a,b) \in f\}$. The "range" $\boldsymbol{rng}(f)$ of a mapping $f: A \to B$ is the set $\{b \in B: \exists (a,b) \in f\}$.

[3] If there is a polynomial time transformation from problem P_1 to P_2 we write $P_1 \propto P_2$, read P_1 transforms to P_2 .

Expression $\vec{C}_2 = \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \frac{\ell(r)}{\bar{v}(r)} \cdot \delta(r,\psi(\varphi^{-1}(r)))$ can be used for the same proof: If we relax δ to $\delta: E_G \times \mathfrak{P}(E_F) \to \{1\}$, and take Π as single-valued function ($|\Pi(s)|= 1$ for all $s \in V_F$), and regard the criterion 'length' we get the *Steiner Problem in Graphs* **SPG**:

$$\textbf{SPG}: \min_{\substack{\forall \varphi_\pi: E_F \to \mathfrak{P}(E_G) \\ \lambda=(s,t)\in E_F \Rightarrow \varphi_\pi(\lambda) \\ \text{is a path from } \Pi(s) \text{ to } \Pi(t) \text{ in } G}} \{\vec{C}_2(\varphi) = \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \ell(r)\} \text{ corresponds to } \min_{T \subseteq G} \{\sum_{r \in E_T} \ell(r)\}. \quad (2.4)$$

Any layout (π, φ_π) minimizing (2.4) corresponds to an optimal subgraph $T= [V_T, E_T] \subseteq G$ spanning $\Pi(V_F) \subseteq V_G$ with minimum connection cost. $\boldsymbol{rng}(\varphi_\pi)$ forms T that must be a tree, otherwise the result cannot be a minimum one. This is the Steiner Problem in Graphs **SPG**, that is ***NP***-hard [12].

Figure 1 gives an example for the **TFLP**:

Combat area G= [crossings V_G, channels of supply E_G], provision graph F= [provisioning facilities V_F, supply links E_F as to ψ]. How is F to assign to G such that the total intensity weighted connection length (ℓ) and connection risk (δ) is minimized?

3. Algorithm PR-1Equation Section (Next)Equation Section (Next)Let us denote with $T= T_1 = [V_1, E_1] \subseteq F$ the *Maximum Spanning Tree* $\overline{MST}$ in F as to ψ. $T_i = [V_i, E_i] \subseteq F$ (i=2, .. k) is derived from T_{i-1} cutting the *scan-eligible leaves* $\vec{L}_{i-1} \subseteq V_{i-1}$ of the previous tree T_{i-1}.

Let $L_i= \{b \in V_i: \boldsymbol{deg}(b) =1\}$ all the leaves of T_i with degree 1 and $A_i=\{a \in V_i \setminus L_i: |E_i(a) \setminus L_i| \le 1\}$ [4] those vertices that have at most one neighbor not being a leaf. Now, the set $\vec{L}_i = E_i(A_i) \cap L_i$ results to the *scan-eligible leaves* $\vec{L}_1$, Figure 2 . If we cut the leaves $\vec{L}_i$ from the current tree T_i the new tree T_{i+1} remains connected. For some $s \in A_i$, $\dot{P}(s) \subseteq \vec{L}_i$ denotes the leaves assigned to source/sink s within the current tree T_i .

$\tilde{\mathfrak{T}}(s) \subseteq T$ denotes the *successor tree* with $V_{\tilde{\mathfrak{T}}(s)} = \{s\} \cup \dot{P}(s) \cup \dot{P}(\dot{P}(s)) \cup \dot{P}(\dot{P}(\dot{P}(s)))\ldots$ and $E_{\tilde{\mathfrak{T}}(s)} = E_T \cap V_{\tilde{\mathfrak{T}}(s)}$ [2].

As a special writing, $\kappa_{s,p}$ represents a *covering* resulting from embedding $\tilde{\mathfrak{T}}(s)$ into **G**, where s is located on $p \in \Pi(s)$. The *length specific road charge* $\delta: E_G \times \mathfrak{P}(E_F) \to \boldsymbol{R}_+$ considers the cost specific influence $\delta(r, \Lambda)$ if flows $\Lambda \subseteq E_F$ are conducted via road $r \in E_G$.

[4] E_i is a binary relation in V_i^2, $E_i \subseteq V_i^2$. Therefore, $E_i(a) = \{x \in V_i: (a,x) \in E_i\}$ called *image of E_i at $a \in V_i$* . Similarly: $E_i(A_i) = \{x \in V_i: a \in A_i \wedge (a,x) \in E_i\}$ called *image of E_i at $A_i \subseteq V_i$* .

3. Description Algorithm PR-1

S1 Build an $\overline{MST}$ $T \subseteq F_{undir}$ using ψ. Traffic graphs are directed graphs. The $\overline{MST}$ is to build within an undirected flow graph F_{undir} as to ψ_{undir}: $V_{F_{undir}} = V_F$,

$E_{F_{undir}} = \{(a,b) \in E_F: (a,b) = (b,a)\}$, and ψ_{undir} $((a,b)) = \psi((a,b)) + \psi((b,a))$.

S2 Set $T_{i=1} = T$ as to generation i= 1. Set empty all $\kappa_{a,p}$. Go to **S4**. cutting the leaves $\vec{L}_{i-1}$ of T_{i-1}.

S3 Increment generation i. $T_i = [V_i, E_i]$ results from T_{i-1}

S4 Has T_i only one vertex? Yes: go to **S17**. Has T_i two vertices? Yes: go to **S6.**

S5 Determine leaves $L_i \subseteq V_{Ti}$ and vertices A_i.

Determine the set of scan-eligible leaves $\vec{L}_i$. Empty the covering $\kappa_{b,q}$ for all $(b,q) \in \vec{L}_i \times \Pi(b)$; Go to **S7.**

S6 Here, $|V_{Ti}|$ is 2. Take an arbitrary $b \in V_{Ti}$. Set $L_i = \{b\}$ and $A_i = V_{Ti} \setminus \{b\}$.

S7 Cycle 1: Take for $\forall a \in A_i$ their leaves $\dot{P}(a)$.

S8 Cycle 2: For $\forall p: \in \Pi(a)$: Set Q* empty. Q* is to contain the best placements $\{(b,q) \in \dot{P}(a) \times \Pi(\dot{P}(a))\}$ as to the embeddings $\kappa_{b,q}$ of the successor trees $\{\tilde{\mathfrak{T}}(b): b \in \dot{P}(a)\}$ including the embedding of the lines $(a,b) \in \{a\} \times \dot{P}(a)$ using their cumulative covering κ'.

S9 Cycle 3: For $\forall b \in \dot{P}(a)$ in decreasing $\psi_{undir}(a,b)$ to embed the strongest flow lines first: Set $ç^* = \infty$. c^* stores the cost resulting from embedding flow (a,b) on a path from p to $q \in \Pi(b)$ including the cost of previously assigned coverings $\kappa_{b,q}$.

S10 Cycle 4: For $\forall$ places $q \in \Pi(b)$ for sink b: Reset a temporary covering $\kappa'' = \kappa' \oplus \kappa_{b,q}$. $\kappa_{b,q}$ is the best covering determined when b has been selected as vertex a in **S7.**

S11 Algorithm **EMBED** calculates a minimal cost path $\mathfrak{P} \subseteq \mathbf{G}$ from p to q embedding flow (a,b). **EMBED** takes advantage from the previous best $\kappa_{b,q}$ cumulatively stored in κ^* (**S14**) and passed by κ' (**S15**).

S12 Enlarge κ'' by the new covering $E_{\mathfrak{P}} \times \{(a, b)\}$ caused by path $\mathfrak{P}$.

S13 Get the cost of κ'' corresponding to the objective function. $\delta(r, \kappa''(r))$ regards the intensity of the flows $\kappa''(r)$ and the original flow through road r.

S14 Update c*, κ^*, and Q to keep the best covering κ'' and placement (b, q).

S15 End Cycle 4. The best place q for b (to be connected from p for a) has been determined. The updated κ^* describes the least cost increase embedding of all the latest lines $\{(a, b): b \in \dot{P}(a)\}$ whose placements have been stored in Q. Pass κ^* to κ' to enable increased δ-cost influencing possibilities for the next line embedding.. Add Q to Q*. Continue with Cycle 3.

S1	$T \subseteq F_{undir}$: [T is $\overline{MST}$ as to ψ_{undir}];
S2	$i=1$; $T_{i=1} := T$; $\forall(a,p) \in V_F \times \Pi(V_F)$:[$\kappa_{a,p} = 0$]; go **S4**;
S3	$i = i+1$; Build T_i with $V_i = V_{Ti} = V_{Ti-1} \setminus \vec{L}_{i-1}$, $E_i = E_{Ti} = E_{Ti-1}) \cap (V_{Ti-1} \setminus \vec{L}_{i-1})^2$;
S4	if $\|V_i\| = 1$ goto **S17**; if $\|V_i\| = 2$; goto **S6**;
S5	$L_i = \{s \in V_i: \boldsymbol{deg}(s) = 1\}$; $A_i = \{s \in V_i \setminus L_i: \|E_i(s) \setminus L_i\| \le 1\}$; $\vec{L}_i = E_i(A_i) \cap L_i$; $\forall b \in \vec{L}_i$:[$\forall q \in \Pi(b)$:[$\kappa_{b,q} = 0$]]; goto **S7**;
S6	$b: \in V_i$; $L_i = \{b\}$; $A_i = V_i \setminus L_i$;
S7	$\forall a \in A_i$:[$\dot{P}(a) = \{b \in \vec{L}_i: (a,b) \in E_i\}$; //source a]$_1$
S8	$\forall p: \in \Pi(a)$:[$Q^* =$ empty; $\kappa' = \kappa_{a,p}$; //place p for a]$_2$
S9	$\forall b: \in \dot{P}(a)$:[$\psi_{undir}(a,b) = \max_{b' \in \dot{P}(a)} \{\psi_{undir}(a,b')\}$:[//sink b]$_3$
S10	$ç^* = \infty$; $\forall q: \in \Pi(b)$: [$\kappa'' = \kappa' \oplus \kappa_{b,q}$; //place q for b]$_4$
S11	path $\mathfrak{P}$= **EMBED**(κ'', (a, p), (b, q)) $\subseteq$ G;
S12	$\kappa'' = \kappa'' \oplus (E_{\mathfrak{P}} \times \{(a, b)\})$;
S13	$ç = \sum_{r \in \boldsymbol{dom}(\kappa'')} \ell(r)\,(\psi(r) + \sum_{\lambda \in \kappa''(r)} \psi(\lambda) + \delta(r, \kappa''(r)))$;
S14	$ç < ç^*$:[$ç^* = ç$; $\kappa^* = \kappa''$; $Q = \{(b.q)\}$]]$_4$
S15	$\kappa' = \kappa^*$; $Q^* = Q^* \cup Q$;]$_3$
S16	$\xi(a,p) = ç^*$; $\kappa_{a,p} = \kappa^*$; $\forall(b,q) \in Q^*$:[$\dot{\pi}((a,p),b) = q$];]]$_{2\ 1}$ goto **S3**
S17	$a: \in V_{Ti}$; $\xi(a,p) = \min_{p' \in \Pi(a)} \{\xi(a,p')\}$;
S18	$\breve{\pi}(a) = p$; $S = A = \{a\}$; $\breve{\kappa} = 0$; $E^* = E_F$; **while** $A \neq 0$: [$\forall a \in A$:[$B = E(a) \setminus S$; $\forall b \in B$:[$\breve{\pi}(b) = \dot{\pi}((a, \breve{\pi}(a)), b)$]; $S = S \cup B$];$A = B$];
S19	**while** $E^* \neq 0$: [$(a,b): \in E^*$:[$\psi((a,b)) = \max_{(a', b') \in E^*} \{\psi((a', b'))\}$; path $\mathfrak{P}$= **EMBED** ($\breve{\kappa}$,(a,{ $\breve{\pi}(a)$}),(b,{ $\breve{\pi}(b)$})); $\breve{\kappa} = \breve{\kappa} \oplus (E_{\mathfrak{P}} \times \{(a,b)\})$; $E^* = E^* \setminus \{(a,b)\}$];
S20	Result: Layout ($\breve{\pi}_{appr} = \breve{\pi}$, $\breve{\varphi}_{appr} = \breve{\kappa}^{-1}$);

Chart 1 Algorithm **PR-1**

S16 End Cycle 3. All the lines {a}× $\dot{P}$ (a) have been embedded by a least road-cost increase embedding κ^* for successor tree $\tilde{\mathfrak{T}}$ (a). Store the best embedding $\kappa_{a,p}= \kappa^*$ with cost $\xi(a,p)= \varsigma^*$. Put the best placements from Q* into $\dot{\pi} : V_F \times V_G \times V_F \rightarrow V_G$ using $\dot{\pi}((a,p),b) =q$ ("If a is on p then line (a, b) has an embedding from p to $q=\dot{\pi}((a,p),b)$"). Go to **S3** to build T_{i+1}.

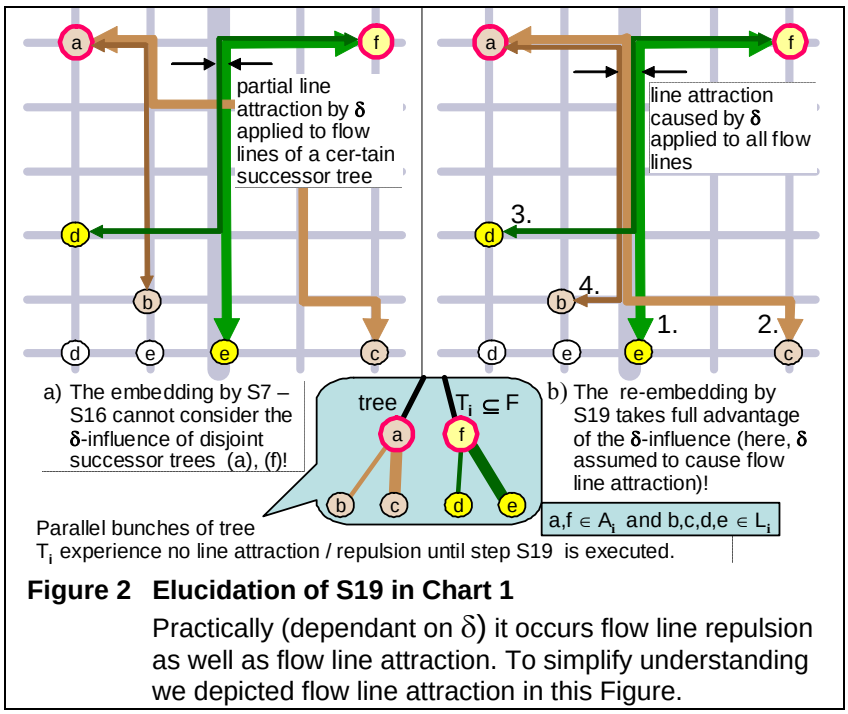

Figure 2 Elucidation of S19 in Chart 1
Practically (dependant on δ) it occurs flow line repulsion as well as flow line attraction. To simplify understanding we depicted flow line attraction in this Figure.

S17 End Cycle 1 and 2: Now, tree T_i consists of one vertex ($|V_{T_i}|=1$). Therefore, node a is unique. The best location p for this last node a can be found using ξ. Cost $\xi(a, p)$ has been assigned to an approximate structure maintained in $\kappa_{a,p}$. Thus, the current $\xi: V_F \times V_G \rightarrow \boldsymbol{R}_+$ applied to (a, p)= min $\xi(a, p')\}_{p' \in \Pi(a)}$ provides that tuple (a, b) whose top-down tracking (**S18** …) reveals an minimum traffic flow layout $\kappa_{a,p}$ of successor tree $\tilde{\mathfrak{T}}$ (a).

S18 Tuple (a, p) serves as start for a backtrack procedure. First, determine the final placement $\breve{\pi} : V_F \rightarrow V_G$ using $\dot{\pi}$. $\dot{\pi}((a, \breve{\pi}(a)), b)$ reads "If vertex a is on $\breve{\pi}$ (a), the incident vertex b of a (so far b not treated (i.e. included in S)) has its best place on $\dot{\pi}((a, \breve{\pi}(a)), b)$". The 'while' causes spreading a wave through graph F taking all the processed leaves B into the set A and places again the leaves of A as long as no further leaves can be found. $\breve{\kappa} : V_G \rightarrow \mathfrak{P}(V_F)$ is the final covering and initialized for the next step.

S19 The current layout ($\breve{\pi}$, $\kappa_{a,p}$) embeds only $\overline{\text{MST}}$ T, but not the lines $E_F \setminus E_T$. $\breve{\pi}$ is the final placement. Figure 2 elucidates why the idea to complete only $\kappa_{a,p}$ with the embeddings of the remaining flows $E_F \setminus E_T$ should be abandoned in favor of re-embedding all flows E_F (with decreasing ψ (strongest first)) enabling the full δ-influence (line attraction or repulsion), see **Figure 2**.

S20 Result= final approximate layout ($\breve{\pi}_{appr}$, $\breve{\varphi}_{appr}$).

Chart 1 reveals, **PR-1** runs with $\boldsymbol{O}(n_F^2 p^2 (m_G \cdot \boldsymbol{log}\ n_G))$ [5] to get an approximate layout ($\breve{\pi}_{appr}$, $\breve{\varphi}_{appr}$).

PR-1 uses an $\boldsymbol{O}(m_G \log n_G)$-Shortest-Path-algorithm **EMBED** that can be derived from [16]. For Shortest Path algorithms we refer to [3,4,5,10,11]. **EMBED** embeds a traffic flow $\lambda=(a,b) \in E_F$ via a path $\mathfrak{P} \subseteq G$ connecting $p \in \Pi(a)$ with $q \in \Pi(b)$. It observes an existing covering κ'' (see step **S11**) such that the current potential ***pot***(y) of a scan-eligible neighbor $y \in V_G$ of vertex $x \in V_G$, $r=(x,y) \in E_G$, can be improved by value $\vec{c} = \boldsymbol{pot}(p) + \ell(r)(\psi(\lambda) + \delta(r, \kappa''(r))) \leq \boldsymbol{pot}(y)$? Yes: $\boldsymbol{pot}(y) = \vec{c}$ with setting node x as predecessor of node y and updating κ'', see **S12**.

4. Performance Analysis

Table 1 shows the objective of the **TFLP** compared to related problems.

4.1 Partial solutions by algorithm PR-1 related to TFLP

(a) **QSAP** Table 1, (4.1), emanates from **TFLP** if $\delta(r, \psi(\varphi_\pi^{-1}(r))) = 0$. In this case **PR-1** determines an approximate lower cost bound for the routing cost $\vec{C}_1$ being an essential information when estimating possible improvements as to placing (changing Π) and routing (planning new roads for **G**) for the total cost solution $\vec{C}(\breve{\pi}_{appr}, \breve{\pi}_{appr})$. Regarding **QSAP**'s lower cost bound we refer to [14]. **EMBED**, Chart 1 , has to observe

---

[5] $|V_F| = n_F$; $|\Pi(V_F)| = p$, placement points $\Pi(V_F) \subseteq V_G$; $|E_F| = m_F$, $|V_G| = n_G$; $|E_G| = m_G$;

cost criterion $\ell(r)$ when it executes its *Shortest Path* extension step (tentatively embedding a flow line λ on road r corresponding to step **S11**, vertex potential improvement control).

(b) **GTPG** Table 1**,** (4.3), emanates from **TFLP** if $\sum_{\lambda \in \varphi^{-1}(r)} \psi(\lambda)$ is not observed. **EMBED** has to observe cost criterion $\delta(r, \psi(\lambda)+\overline{\psi}(r))$.

(c) **SPG** (2.4) emanates from **TFLP** if ψ and δ are neglected and Π is single valued. Then, **PR-1** approximately solves the Steiner Problem in Graphs **SPG** observing $C_{appr} = 2 \cdot C_{opt}(1 - 1/n_f)$, [21,15]. **PR-1** achieves this by using

$\delta(r, \psi(\varphi_\pi^{-1}(r))) = \begin{bmatrix} 1 \text{ if } \varphi_\pi^{-1}(r)=0 \\ 0 \text{ if } \varphi_\pi^{-1}(r)\neq 0 \end{bmatrix}$. **EMBED** takes every non-empty road as short-circuit when it tries to embed further flow lines. Thus, it observes the paradigm "Connect to the current tree the next reachable vertex that should be in the Steiner tree but is not in the current tree!", what guarantees the cost bound above [15]. For algorithms as to Steiner trees on graphs see [2,6,7,13,15].

(d) **TFLP** Table 1**,** (4.4): **EMBED** has to observe cost criterion $\ell(r) + \delta(r, \psi(\lambda)+\overline{\psi}(r))$, (step **S11**) embedding the current flow λ. considering the existing flow intensity $\overline{\psi}(r)$.

QSAP Quadratic Semi-Assignment Problem $\Rightarrow$ optimizes π: $\min_{\substack{\pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{ \sum_{\lambda=(s,t) \in E_F} \psi(\lambda) \cdot \boldsymbol{d}(\pi(s), \pi(t)) \}$ *corresponds to:* $\min_{\substack{\pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{ \min_{\varphi_\pi: E_F \to \mathcal{P}(E_G)} \{ \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \ell(r) \sum_{\lambda \in \varphi^{-1}(r)} \psi(\lambda) \} \}$ **(4.1)**	no line attraction / repulsion	Routing Cost Optimization
GSPG General Steiner Problem in Graphs $\Rightarrow$ optimizes π and T: $\min_{\substack{\pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{ \min_{\substack{\text{subgraph } T \subseteq G \\ \pi(V_F) \subseteq V_T}} \{ \sum_{r \in E_T} \ell(r) \} \}$ **(4.2)**		Steiner Cost Optimization
GTPG General Tracing Problem in Graphs .....$\Rightarrow$ optimizes (π, φ_π): $\min_{\substack{\pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{ \min_{\varphi_\pi: E_F \to \mathcal{P}(E_G)} \{ \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \ell(r) \cdot \delta(r, \varphi_\pi^{-1}(r)) \} \}$ **(4.3)**	line attraction / repulsion induced by δ	
GCPG General Connection Problem in Graphs $\cong$ **TFLP** Transport Flow Layout Problem (relaxed equation (2.2)): $\Rightarrow$ optimizes (π, φ_π) $\min_{\substack{\pi: V_F \to V_G \\ \pi \subseteq \Pi}} \{ \min_{\varphi_\pi: E_F \to \mathcal{P}(E_G)} \{ \sum_{r \in \boldsymbol{rng}(\varphi_\pi)} \ell(r) \cdot \big(\psi(r) + \sum_{\lambda \in \varphi^{-1}(r)} \psi(\lambda) + \delta(r, \varphi_\pi^{-1}(r))\big) \} \}$ **(4.4)**		Transport Cost and Road Charge Optimization

Table 1 Comparing Optimization Problems **TFLP** with **QSAP**, **GSPG**, and **GTPG**

4.2 Test

The program and random graph generator has been developed with MS Visual Studio 6.0 on a 1,6 GHz PC with the following settings:

- Traffic flow F: directed, $| V_F |= 9$, average vertex degree= 4, $|E_F|= 24$,
  flow intensity ψ: $0{,}01 \le \psi(\lambda) \le 1{,}0$ uniformly distributed.
- Road net G: directed and undirected, $| V_G | \le 50.000$, $|E_G| \le 199104$, average vertex degree= 4,
  road length ℓ: $0{,}1 \le \ell(r) \le 10$ uniformly distributed,
  road charge factor τ: $E_G \rightarrow \boldsymbol{R}_+$, randomly selected as follows:
  percentage of E_G: 10% 20% 20% 20% 20% 10%, edges for each part uniformly distributed
  road load factor $\tau(r)$: 0,2 0,3 0,5 1,0 1,5 3,0.
- The length specific road charge $\delta(r, \varphi_\pi^{-1}(r))$ has been simplified directly accessing the corresponding flow intensity using a road load factor $\tau(r)$ for road r and $\psi(r)= 0$ resulting to $\delta(r,\psi(\varphi_\pi^{-1}(r))) = \tau(r) \cdot \psi(\varphi_\pi^{-1}(r))^2$, $r \in E_G$. The higher $\tau(r)$ the more intensive increases the road charges, that were here simplified to a quadratic dependency on the flow.

Table 2 gives the results, each as average of three measurements. All cases (a), ..(c) of 0 have been tested with the same program system on the same machine using algorithm **PR-1**. The nominal time has been determined using system function `GetSystemTime.` Due to the small time differences between (b) and (c) we only give the time once behind (c). The results each graph and competing strategy heavily depend on the number of possible spots $|\Pi(a)|$ that come into consideration placing source / sink $a \in V_F$. Clearly, the more placements were randomly generated within the road net **G** the more possibilities are existent to enable a contraction of the connection structure in G for embedding F.

4.3 Examples for Practical Applications

Traffic analysis and simulation

Starting from flow analyzing systems that determine main traffic flows F in conurbations G, PR-1 enables the simulation of re-embeddings of F into G for different scenarios (seasons, day, time of day) with different road charge parameter preference as to noise, pollution, capacity, toll, traffic calming, security risks, .. coming to better layouts ($\breve{\pi}_{appr}$, $\breve{\varphi}_{appr}$).

Evacuation Management for disaster areas, preparation of contingency plans

How can anticipated mass evacuation traffic flows F conducted out of a coming hurricane disaster area G such that all possible roads come into consideration to enable the mass escape (very necessary in the case of the hurricanes Katrina and Rita). Here, the one-criterion approach as to routing expense fails totally. Observing road charges keeps the outgoing traffic as fluent as possible.

Temporarily re-routing of traffic flows through a conurbations

A known traffic flow $\lambda= (s,t)$ with high intensity $\psi(\lambda)$ be temporarily re-routed through a conurbation G with respect to the existing traffic intensity on the roads in G..
Traffic planners decide that λ enters G at crossings x or y and leaves G at traffic the points u, v, or w.
$F= [V_F=\{s,t\}, E_F=\{(s,t)\}]$, $\Pi=\{(s, \{x,y\}), (t, \{u,v,w\})\}$, $\{x,y,z\} \subseteq V_G$, i.e. $\Pi: V_F \rightarrow \mathfrak{P}(V_G)$.

Algorithm PR-1 might calculate $\pi= \{(s, y), (t, v))\}$ assigning the source s to y and sink t to v. φ_π embeds flow λ into G minimizing travel expense and road charges.
In the case that λ must pass first crossing $c_1 \in V_G$ and then crossing $c_2 \in V_G$ algorithm PR-1 runs thrice with Π as follows :
1.: $\Pi=\{(s, \{x, y\}), (t, \{c_1\})\}$, 2.: $\Pi=\{(s, \{c_1\}), (t, \{c_2\})\}$, 3.: $\Pi=\{(s, \{c_2\}), (t, \{u,v,w\})\}$.
Or λ is to divide into $\lambda_1= (s_1, t_1)$, $\lambda_2= (s_2, t_2)$, $\lambda_3= (s_3, t_3)$,
$\Pi= \{((s_1, \{x, y\}), (t_1, \{c_1\}), (s_2, \{c_1\}), (t_2, \{c_2\}), (s_3, \{c_2\}), (t_3, \{u,v,w\})\}$.

Single traffic user guided by future navigation systems

A future road user should guided trough a conurbation not only in his own interest but in the interest of all users, i.e. in the interest of least traffic jams, moderate environmental stress, administrative preferences, .. (expressed in δ). I.e. the consideration of road charges will play an important rule for future navigation systems that consider not only the pure road map but take into account online information about road charges.

F directed, G directed																
|V(G)|=	|E(G)|=	∖ a∈V(F): |∖ (a)|=	**(a) QSAP with PR-1**					**(b) GTPG with PR-1**					**(c) TFLP with PR-1**				
			Routing cost	Steiner Cost	Total	%	time in sec	Routing cost	**Steiner Cost**	Total	%	Routing cost	Steiner Cost	**Total**	%	time in sec
100	360	1	**220**	385	605	145	0,015	361	**148**	509	122	253	165	**418**	100	0,016
		2	**147**	197	344	131	0,016	215	**85**	300	114	164	99	**263**	100	0,032
		3	**158**	120	278	109	0,031	226	**78**	304	119	163	92	**255**	100	0,062
748	2880	1	**685**	717	1402	125	0,031	972	**316**	1288	115	750	373	**1123**	100	0,094
		2	**445**	399	844	116	0,078	662	**209**	871	120	479	247	**726**	100	0,187
		3	**414**	303	717	110	0,156	585	**163**	748	115	457	196	**653**	100	0,375
10000	39600	1	**2008**	2140	4148	132	0,593	2798	**797**	3595	114	2198	955	**3153**	100	1,157
		2	**1366**	1347	2713	128	1,391	1920	**557**	2477	117	1454	665	**2119**	100	2,641
		3	**1385**	1194	2579	120	2,922	1964	**532**	2496	116	1514	643	**2157**	100	5,906
20000	79434	1	**2885**	2724	5609	125	1,187	4072	**1174**	5246	117	3144	1359	**4503**	100	2,328
		2	**1646**	1928	3574	141	3,094	2325	**632**	2957	117	1779	754	**2533**	100	6,078
		3	**2044**	1276	3320	116	6,218	2814	**681**	3495	122	2025	832	**2857**	100	12,157
35000	139250	1	**3186**	3218	6404	126	2,047	4639	**1336**	5975	118	3468	1598	**5066**	100	3,797
		2	**2489**	2167	4656	120	5,734	3672	**960**	4632	119	2671	1215	**3886**	100	10,907
		3	**2134**	1441	3575	110	10,578	3050	**803**	3853	118	2279	976	**3255**	100	20,187
50000	199104	1	**4080**	3850	7930	122	3,203	5924	**1682**	7606	117	4430	2070	**6500**	100	6,016
		2	**2434**	2131	4565	118	15,828	3743	**1015**	4758	123	2700	1172	**3872**	100	29,547
		3	**2361**	1839	4200	116	16,422	3599	**913**	4512	124	2540	1092	**3632**	100	31,359
			best		aver.:	**123**			**best**	aver.:	**118**			**best**	**100**	
					max:	**145**				max:	**124**					
F directed, G undirected																
50000	99552	1	**4121**	7995	12116	179	2,719	6066	**1787**	7853	116	4607	2177	**6784**	100	5,687
		2	**2378**	3520	5898	153	6,406	3800	**1045**	4845	126	2668	1181	**3849**	100	13,078
		3	**2653**	3380	6033	161	13,391	3487	**961**	4448	119	2630	1115	**3745**	100	27,406
			best		aver.:	**164**			**best**	aver.:	**120**			**best**	**100**	
					max:	**179**				max:	**126**					

Table 2 **Results of PR-1 subject to the objectives (a), (b), (c) corresponding to 4.2 on large sparse random graphs**

4.4 Conclusion

Table 2 shows, that the opinion "Approximate algorithms solving **QSAP** or **GTPG** may be sufficient enough to minimize layouts with respect to travel expense and road charges!" does not hold at all, as long as both cost comparably contribute to the total. Test series (c) **TFLP** with PR-1' outperforms surprisingly well the cases (a) **QSAP** with PR-1 and (b) **GTPG** with PR-1 and leads to a considerable total cost reduction very attractive for practical applications.

The proposed procedure meets the needs of today's traffic planners that have to observe more than ever before not only Routing Cost (Travel Expense) but also Road Charges in the interest of a future oriented traffic policy in conurbations.

References

1. Bokhari, S.H., "A shortest tree algorithm for optimal assignments across space and time in a distributed processor system". IEEE Transactions on Software Engineering, 1981. SE-7: pp. 583-589.
2. Chen N. P.: New algorithms for Steiner tree on graphs. In. Proceedings of the IEEE International Symposium on Circuits and Systems, pages 1217-1219, (1983).
3. Deo N., Pang C.: Shortest Path Algorithms: Taxonomy and Annotations, NETWORKS, Vol. 14 , (1984), 275-323
4. Dial R.: Algorithm 360 Shortest Paths Forest with topological Ordering, Communications of the ACM, 12, (1969) 632-633
5. Dijkstra E.W.: A Note on two Problems in Connection with Graphs, Numerische Mathematik 1, (1959) 269-271
6. Dreyfus S.F., Wagner R.A.: The Steiner Problem in Graphs, NETWORKS, 1, (1972) 195-207
7. Duin C. W.: Steiner's Problem in Graphs, PhD thesis, Amsterdam University; (1994).
8. Gallo, G. and B. Simeone, "Optimal grouping of researchers into departments". Ricerca Operativa, 1993. 57: pp. 45-69.
9. Gallo, G., E.M. Tomasin, and A.M. Sorato, "Lower bounds for the quadratic semi-assignment problem. 1986, Rutgers University, New Brunswick, NJ 08903
10. Gilsinn I., Witzgall, C.: A Performance Comparison of Labeling Algorithms for Calculating Shortest Paths Trees, NBS Technical Note 772, US. Department of Commerce, (1973)
11. Glover F., Glover R., Klingman D.: Computational Study of an Improved Shortest Path Algorithm, NETWORKS 14, (1984) 25-36
12. Karp, K. M.: Reducibility Among Combinatorial Problems, in R. E. Miller, I. W. Thatcher (Eds.) Complexity of Computer Computations, Plenum Press, New York, 1972, 85-103
13. Karpinski M. and Zelikovsky A. Z.: New approximation algorithms for the Steiner tree problems. Technical report, International Computer Science Institute, Berkeley, (1995).
14. Malucelli F., Pretolani D.:Lower Bounds for the Quadratic Semi-Assignment Problem, Department of Informatica, University of Pisa, Corso Italia 40, 56125 Pisa, Italy (1995)
15. Richter P.: Present Approximate Algorithms for Steiner's Problem in Graphs - Classification and Two New Fast Approaches, Systems Science (PL ISSN 0137-1223); Volume 17 - Number 2 (1991) 2-28.
16. Richter P.: Efficient Shortest Path Algorithms for Road Traffic Optimization, 27th International Symposium on Advanced Transportation Applications (ISATA): Dedicated Conference on Advanced Transport Telematics, ISBN 0947719652, Aachen, 31.10.-4.11.1994, 79-86
17. Richter P.: Semi Quadratic Assignment Problem – Pragmatic Change of the QAP's Domain Enables Attractive ε-Approximation, Advances in Systems, Signals, Control and Computers, Vol. I, ed. V.B. Bajic, published by IAAMSAD and the South African Branch of the Academy of Nonlinear Sciences, ISBN 0-620-23137-8, Sep (1998) 69-75
18. Richter, P.: Wiring Layout Design Reducing Cable and Trace Cost, ISATA 2000, Dublin, Ireland, 25. - 27.9. 2000. Simultaneous Engineering and Rapid Product Development, ISBN 1-902856-13-9, page 61-68 (Award for the best paper in the Track on Simultaneous Engineering)
19. Sahni, S. and T. Gonzalez, "P-complete Approximation Problems". ACM Journal, 1976 (23): pp. 555-565.
20. Shier, Witzgall: Properties of Labeling Methods for Determining Shortest Path Trees, J. Res. Natl. Bur. Stand., (1986) 323-333
21. Takahashi H., Matsuyama A.: An Approximate Solution for the Steiner Problem, Math. Japonica 24, No. 6, (1980), 573-577