Wiring Layout Design Reducing Cable and Trace Cost

Peter H. Richter,
SATCON GmbH Teltow, Germany

00SE002

Abstract

We give an efficient constructive deterministic approximation algorithm that determines minimum installation layouts with respect to cable cost and trace cost (the latter for supporting and safeguarding the cables) in order to embed intensity weighted flow nets (wiring diagrams, power supply systems, …) into industrial, technical, or urban environments. We show, that the algorithm observes a convenient upper cost bound and prove its polynomial time behavior. The resulting layout structures may be viewed as hybrids derived from General Steiner Trees as well from solutions for the Quadratic-Semi-Assignment Problem. The success of the algorithm is based on the search for partial structures that enable common line conduction on common traces with only moderate increased cable cost (compared with shortest path conduction) but with more reduced trace cost. Practical results are given.

Introduction

Today's discrete optimization with respect to the embedding of flow nets primarily concerns the minimization of routing ***cost*** (pure flow line cost ⇒ algorithms for the Quadratic Semi-Assignment Problem QSAP) or trace ***cost*** (cost for bearing and safeguarding the flow lines ⇒ approximate algorithms for the Generalized Steiner Problem in Graphs GSPG). There are no convincing strategies that meet the layout optimization of both (***routing cost*** + ***trace cost***). In practice, one cost part is usually taken as a certain percentage of the other. This cannot be longer accepted if routing cost as well as trace cost contribute with nearly the same cost parts to the total. We are confronted with a structure that is neither a solution for the QSAP nor for the GSPG, Fig. 3. This is the case for the overwhelming majority of industrial applications where flow nets concern paths with mutual interdependence. An *attraction* tendency occurs if the cost conducting some flow lines via some common traces is smaller than that for their separated flow line conduction. *Repulsion* occurs when the cost for conducting some flow lines together are higher than single line conduction.

We start our consideration from a finite set of *entities* M (machines, consumers, distributors, transformers in a workshop or in a ship / entities of a car's electrical equipment / production plants in a factory, …) to be connected observing a *flow intensity* ψ: $M^2 \rightarrow \boldsymbol{R}_+$ seen as a relationship diagram in a wider sense. Obviously, ψ induces a *flow graph* $\mathfrak{F} = [M, \boldsymbol{dom}(\psi)] = [\text{vertices } V(\mathfrak{F}), \text{edges } E(\mathfrak{F})]$ we assume to be a connected graph (***dom***(*f*), ***rng***(*f*) ≅ *domain* and *range* of a mapping or relation *f*). The structured environments (like cars, planes, ships, workshops, …) should be given as graphs $\mathbf{G} = [V(\mathbf{G}), E(\mathbf{G})] = [$vertices including possible locations for the entities, traces possibly embedding flow lines$]$. With the traces' *length* $\ell_{\mathbf{G}}$: $E(\mathbf{G}) \rightarrow \boldsymbol{R}_+$ there is a graph metric (*shortest path distance*) $\boldsymbol{d}_{\mathbf{G}}$: $V(\mathbf{G})^2 \rightarrow \boldsymbol{R}_+$. The entities' possible locations within the *structured environment* **G** are predestined by a *placement rule* $\Pi \subseteq M \times V(\mathbf{G})$ assigning each $m \in M$ to the set of possible locations $\Pi(m) \subseteq V(\mathbf{G})$. In contrast to Π, a *placement function* π: $M \rightarrow V(\mathbf{G})$, we look for, assigns each $m \in M$ uniquely to a location $\pi(m) \in \Pi(m)$. A *routing* $\varphi_\pi \subseteq E(\mathfrak{F}) \times E(\mathbf{G})$ assigns each line $(a,b) \in E(\mathfrak{F})$ to a simple path via the edges $\varphi_\pi((a,b)) \subseteq E(\mathbf{G})$ between $\pi(a)$, $\pi(b) \in V(\mathbf{G})$. We use the concept *covering* $\kappa = \varphi_\pi^{-1} \subseteq E(\mathbf{G}) \times E(\mathfrak{F})$ to describe that trace $t \in E(\mathbf{G})$ is covered with cables $\kappa(t) \subseteq E(\mathfrak{F})$. $\mathfrak{T} = \mathfrak{T}_1 = [V_1, E_1] \subseteq \mathfrak{F}$ is a *Maximum Spanning Tree* $\overline{\text{MST}}$ as to ψ. $\mathfrak{T}_i = [V_i, E_i] \subseteq \mathfrak{F}$ (i=2, .. k) is derived by $\mathfrak{T}_{i-1}$ cutting the *scan-eligible leaves* B_{i-1} (below) $\subseteq V_{i-1}$ of the previous -tree $\mathfrak{T}_{i-1}$. Let $L_i = \{b \in V_i: \underbrace{\boldsymbol{deg}(b)}_{\text{degree of } b} = 1\}$ the leaves of $\mathfrak{T}_i$.

$A_i = \{a \in V_i \setminus L_i: |E_i(a) \setminus L_i| \leq 1\}$ are the parents of the *scan-eligible leaves* $B_i = \underbrace{E_i(A_i)}_{\text{neighbors of } A_i} \cap L_i$.

For some $a \in A_i$, $\dot{P}(a) \subseteq B_i$ denotes the leaves assigned to entity as to the current tree $\mathfrak{T}_i$.

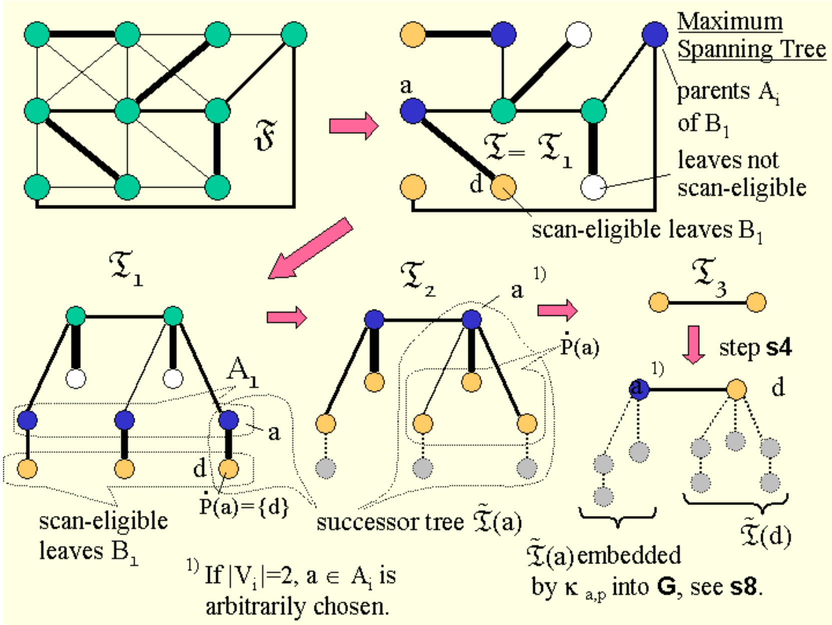


Figure 1 Generation Trees and Successor Trees

$\tilde{\mathfrak{T}}(a) \subseteq \mathfrak{T}$ denotes the *successor tree* with $V(\tilde{\mathfrak{T}}(a)) = \{a\} \cup \dot{P}(a) \cup \dot{P}(\dot{P}(a)) \cup \dot{P}(\dot{P}(\dot{P}(a)))\ldots$ and $E(\tilde{\mathfrak{T}}(a)) = E(\mathfrak{T}) \cap V(\tilde{\mathfrak{T}}(a))^2$. As a special writing, $\kappa_{a,p}$ represents a covering resulting from embedding $\tilde{\mathfrak{T}}(a)$ into **G**, where a is located on $p \in \Pi(a)$. A *length specific trace cost function* δ: $E(\mathbf{G}) \times \mathfrak{P}(E(\mathfrak{F})) \rightarrow \boldsymbol{R}_+$ considers the cost specific influence $\delta(t, \Lambda)$ so far the cables $\Lambda \subseteq E(\mathfrak{F})$ are laid out onto trace $t \in E(\mathbf{G})$, Fig. 2. Derived from general applications we use the reasonable implication $L \subseteq E(\mathfrak{F})$: $\psi(\alpha) > \psi(\beta) \Rightarrow \delta(t, L \cup \{\alpha\}) > \delta(t, L \cup \{\beta\})$, i.e.: High flow intensity induces high specific trace cost Fig. 2 elucidates how the specific trace cost function δ influences line attraction or line repulsion.

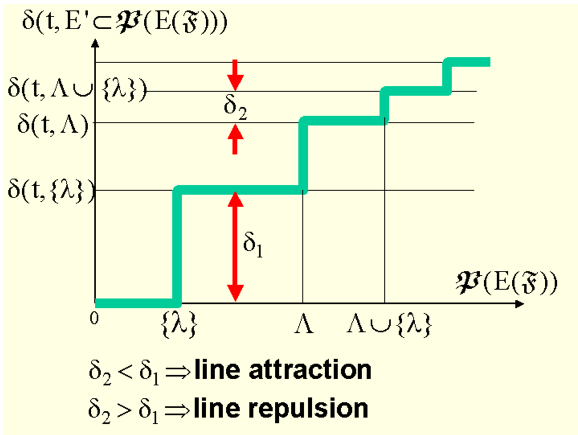


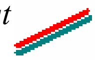
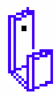
Figure 2 Why δ exerts line attraction or line repulsion

Problem 1: We look for a layout (π, φ_π), $\kappa = \varphi_\pi^{-1}$, that minimizes the sum

$$\sum_{t \in \boldsymbol{dom}(\kappa)} \ell_{\mathbf{G}}(t) \Big(\sum_{r \in \kappa(t)} \psi(r) + \delta(t, \kappa(t)) \Big) =$$

trace t
line r

$$\sum_{r \in E(\mathfrak{F})} \Big(\psi(r) \cdot \sum_{t \in \varphi_\pi(r)} \ell_{\mathbf{G}}(t) \Big) + \sum_{t \in \boldsymbol{dom}(\kappa)} \ell_G(t)\, \delta(t, \kappa(t))\}.$$

Routing cost  *Trace cost* 

Problem 1a $\Rightarrow$ *Quadratic Semi-Assignment Problem* **QSAP, *NP***-hard [1,14,18]

Problem 1b $\Rightarrow$ *General Tracing Problem in Graphs* **GTPG** derived from *General Steiner Prblem in Grap* **GSTP *NP***-hard [7, 2,8,12, 13,17, 20, 21]

An optimal layout (π, φ_π) assigns entities M to locations $\pi(M) \subseteq V(\mathbf{G})$ and embeds between them the flow net $\mathfrak{F}$ onto the traces $\varphi_\pi(E(\mathfrak{F})) \subseteq E(\mathbf{G})$ of environment **G** with minimum total cost.

Due to the problems' ***NP***-hardness, normal problem sizes of industrial applications compel to look for heuristics solving Problem 1. Below, Fig. 3 and Table 1 give the essentials how **Problem 1**, **QSAP**, **GTPG**, and **GSPG** differ from each other.

QSAP Quadratic Semi-Assignment Problem $\min_{\substack{\pi: M \to V(\mathbf{G}) \\ \pi \subseteq \Pi}} \sum_{(a,b) \in E(\mathfrak{F})} \psi(a,b)\, d_{\mathbf{G}}(\pi(a),\pi(b))$	**No conside-ration of line attraction / repulsion**	**Routing Cost Optimization** considers cost for the pure flow lines $E(\mathfrak{F})$
GSPG General Steiner Problem in Graphs $\min_{\substack{\pi: M \to V(\mathbf{G}) \\ \pi \subseteq \Pi}} \min_{\substack{\text{tree } \mathbf{T} \subseteq \mathbf{G} \\ \pi(M) \subseteq V(\mathbf{T})}} \sum_{t \in E(\mathbf{T})} \ell_{\mathbf{G}}(t) \} \}$		**Tracing Cost Optimization:** considers the cost for the traces $\subseteq E(\mathbf{G})$ bearing the cables $E(\mathfrak{F})$. The **GSPG** is contained as special case within the **GTPG**.
GTPG General Tracing Problem in Graphs $\min_{\substack{\pi: M \to V(\mathbf{G}) \\ \pi \subseteq \Pi}} \min_{\substack{\kappa \subseteq \\ E(\mathbf{G}) \times E(\mathfrak{F})}} \sum_{t \in \mathit{dom}(\kappa)} \ell_{\mathbf{G}}(t)\delta(t,\kappa(t)) \} \}$	**Observing line attraction / repulsion**	

Table 1 Comparing Optimization Problems QSAP, GSPG, and GTPG

2 Approximate Solution Procedure TRACE (please proceed comparing with Chart 1!)

The strategy:

Determine a maximum spanning tree $\overline{\text{MST}}$ $\mathfrak{T} \subseteq \mathfrak{F}$ as to ψ. Take the leaves $B_{i=1}$ of $\mathfrak{T}=\mathfrak{T}_{i=1}$ and the leaves' parents $A_{i=1}$. Determine for $\forall(a,p) \in A_i \times \Pi(A_i)$ and $\forall(d,s) \in (\dot{P}(a) \cap B_i) \times \Pi(B_i)$ the embeddings $\kappa_{a,p}$ of each bunch $\{a\} \times \dot{P}(a)$ using line attraction / repulsion determined by algorithm EMBED. Cut the leaves B_i to get the next generation tree $\mathfrak{T}_{i+1}$ and proceed like above till $\mathfrak{T}_k$ is a singleton. Embed $\mathfrak{T}$ into **G** taking the best placement of that singleton and go top down via the stored successor trees whose best placements and best successor trees' embeddings (coverings) have been stored before. Embed the remaining flow lines $E(\mathfrak{F})\backslash E(\mathfrak{T})$.

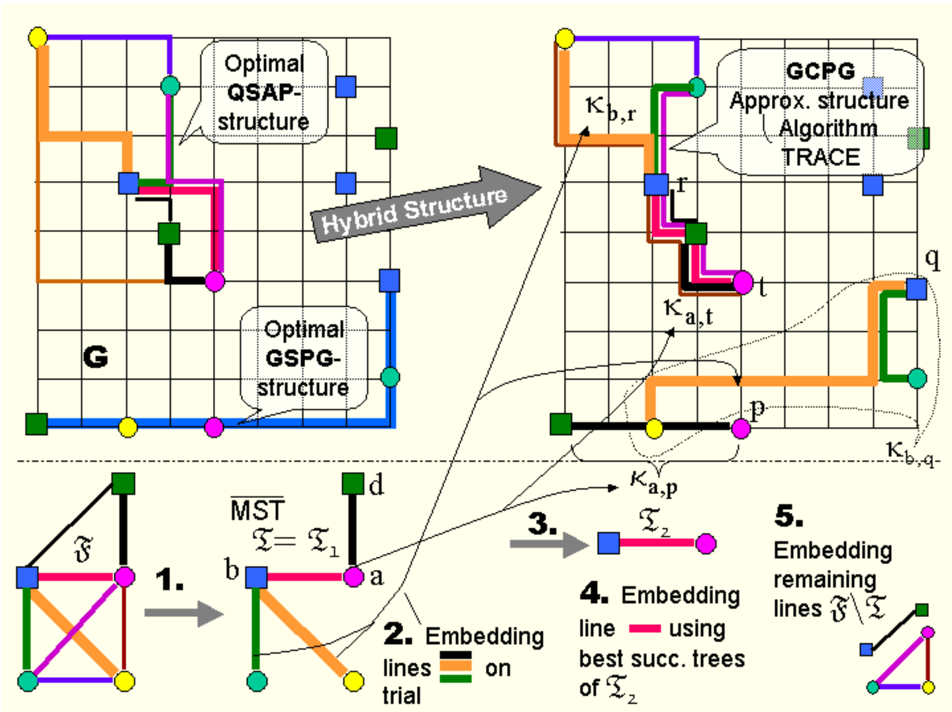


Figure 3 Problems and Solutions with respect to Problem 1

S1 Build a maximum spanning tree $\overline{\text{MST}}$ $\mathfrak{T} \subseteq \mathfrak{F}$ as to intensity ψ. Take flow tree $\mathfrak{T}_{i=1} = \mathfrak{T}$ belonging to generation i= 1. Set empty all $\kappa_{a,p}$. **Goto S4**.

S2 Increment generation i. Build the new tree $\mathfrak{T}_i = [V_i, E_i]$ from $\mathfrak{T}_{i-1}$ by cutting the leaves B_{i-1} of $\mathfrak{T}_{i-1}$.

S3 If the current tree $\mathfrak{T}_i$ consists of only one vertex then **goto S17**. If it consists of only two vertices **goto S6**.

S4 Determine the leaves $L_i \subseteq V(\mathfrak{T}_i)$ of $\mathfrak{T}_i$ and the neighbors $A_i := \{a \in V_i \setminus L_i : |E_i(a) \setminus L_i| \le 1\}$ that have at most one neighbor not being a leaf (i.e. $\mathfrak{T}_{i+1}$ remains connected when cutting the leaves B_i for the next generation). Determine the set of scan-eligible leaves to $B_i = E_i(A_i) \cap L_i$. Empty $\kappa_{b,q}$ for all $(b,q) \in B_i \times \Pi(b)$; **Goto S7.**

S5 Here, $|V(\mathfrak{T}_i)|$ is 2. Take an arbitrary $b \in V(\mathfrak{T}_i)$ and set $L_i := \{b\}$ and $A_i := V(\mathfrak{T}_i) \setminus \{b\}$.

S6 Cycle 1: Take successively $\forall a \in A_i$ and their leaves $\dot{P}(a) := \{b \in B_i : aE_ib\} \cong$ "All neighbors of a contained in B_i ".

S7 Cycle 2: For $\forall\ p :\in \Pi(a)$: Set Q* empty. Q* is to contain the best determined placements $\{(d,s) \in \dot{P}(a) \times \Pi(\dot{P}(a))\}$ as to the embeddings $\kappa_{d,s}$ of the successor trees $\{\tilde{\mathfrak{T}}(d) : d \in \dot{P}(a)\}$ including the embedding of the lines $(a,d) \in \{a\} \times \dot{P}(a)$ using their cumulative covering κ'. Notice: Only for the final treatment and the special case $|V_i| = 2$ (see **S4**) $\kappa' := \kappa_{a,p}$ is not empty because for the two vertices of A_i a covering of their successor trees already exists. Therefore, $\kappa' := \kappa_{a,p}$.

S8 Cycle 3: For all leaves $d \in \dot{P}(a)$: Initialize $ç^* = \infty$. c* stores the trace cost resulting from embedding line (a,d) on a path from p to $s \in \Pi(d)$ including the cost of previously assigned coverings $\kappa_{d,s}$.

S9 Cycle 4: For all possible locations $s \in \Pi(d)$ of $d \in \dot{P}(a)$: Build a temporary covering κ" $\kappa'' := \kappa' \oplus \kappa_{d,s} =$

$$\bigcup_{t \in \mathbf{dom}(\kappa') \cup \mathbf{dom}(\kappa_{d,s})} \{(t, \kappa'(t) \cup \kappa_{d,s}(t))\}$$

Notice, if we want to embed the current line (a,d) then $\kappa_{d,s}$ previously determined in **S16** will serve as δ-cost reduction possibility for algorithm EMBED.

S1 $\mathfrak{T} \subseteq \mathfrak{F} : [\mathfrak{T}$ is $\overline{\text{MST}}$ as to $\psi : M^2 \to \mathbb{R}_+]$;

S2 $i := 1;\ \mathfrak{T}_{i=1} := \mathfrak{T};\ \forall (a,p) \in M \times \Pi(M):$
$[\kappa_{a,p} :=$ empty]; **goto S4;**

S3 $i := i+1$; Build $\mathfrak{T}_i : V_i := V(\mathfrak{T}_i) := V(\mathfrak{T}_{i-1}) \setminus B_{i-1}$,
$E_i = E(\mathfrak{T}_i) := E(\mathfrak{T}_{i-1}) \cap (V(\mathfrak{T}_{i-1}) \setminus B_{i-1})^2$;

S4 **if** $|V_i| = 1$ **goto S17;** **if** $|V_i| = 2$ **goto S6;**

S5 $L_i := \{b \in V_i : \mathbf{\mathit{deg}}(b) = 1\}$;
$A_i := \{a \in V_i \setminus L_i : |E_i(a) \setminus L_i| \le 1\};\ B_i = E_i(A_i) \cap L_i$;
$\forall b \in B_i : [\forall q \in \Pi(b) : [\ \kappa_{b,q} := 0]]$; **goto S7;**

S6 $b :\in V_i;\ L_i := \{b\};\ A_i := V_i \setminus L_i$;

S7 $\forall a \in A_i : [_1\ \dot{P}(a) := \{b \in B_i : aE_ib\}$;

S8 $\forall p :\in \Pi(a) : [_2\ Q^* :=$ empty; $\kappa' := \kappa_{a,p}$;

S9 $\forall d :\in \dot{P}(a) : [_3\ ç^* := \infty$;

S10 $\forall s :\in \Pi(d) : [_4\ \kappa'' := \kappa' \oplus \kappa_{d,s}$;

S11 path $\mathbf{P} :=$ EMBED$(\kappa'', (a,p), (d,s)) \subseteq \mathbf{G}$;

S12 $\kappa'' := \kappa'' \oplus (E(\mathbf{P}) \times \{(a,d)\})$;

S13 $ç := \sum_{t \in \mathbf{dom}(\kappa'')} \ell_{\mathbf{G}}(t) \left(\sum_{r \in \kappa(t)} \psi(r) + \delta(t, \kappa(t)) \right)$

S14 $ç < ç^* : [\ ç^* := ç;\ \kappa^* := \kappa'';\ Q := \{(d,s)\}]\ ]_4$

S15 $\kappa' := \kappa^*;\ Q^* := Q^* \cup Q;\ ]_3$

S16 $\xi(a,p) := ç^*;\ \kappa_{a,p} := \kappa^*$;
$\forall (x,y) \in Q^* : [\ \dot{\pi}((a,p),x) := y];\ ]_2\ ]_1$ **goto S3**

S17 $a :\in V(\mathfrak{T}_i);\ \xi(a,p) := \min_{p' \in \Pi(a)} \{\xi(a,p')\};\ \breve{\kappa} := \kappa_{a,p}$;

S18 $\breve{\pi}(a) := p;\ S := A := \{a\}$;
while $A \ne 0$: **[** $\forall a \in A : [B := E(a) \setminus S;\ \forall b \in B :$
$[\breve{\pi}(b) := \dot{\pi}((a, \breve{\pi}(a)), b)];\ S := S \cup B];\ A := B$**]**;

S19 $\forall (a,b) :\in E(\mathfrak{F}) \setminus E(\mathfrak{T})$:**[**
$\mathbf{P} :=$ EMBED$(\breve{\kappa}, (a, \{\breve{\pi}(a)\}), (b, \{\breve{\pi}(b)\}))$;
$\breve{\kappa} := \breve{\kappa} \oplus (E(\mathbf{P}) \times \{(a,b)\})$ **]**;

S20 Result: Layout $(\breve{\pi}_{appr} := \breve{\pi},\ \breve{\varphi}_{appr} := \breve{\kappa}^{-1})$;

Chart 1 Flow chart of Algorithm TRACE

S10 Call algorithm EMBED to determine a cheapest cost path $\mathbf{P} \subseteq \mathbf{G}$ from p to s for embedding line (a,d). EMBED takes advantage from the previous best $\tilde{\mathfrak{T}}(d)$- embeddings

$\{\kappa_{d,y}: d\in \dot{P}(a), y\in\Pi(d)\}$ cumulatively stored in κ^* (**s14**) and passed by κ' (**s15**).

S11 - S13 Enlarge on trial κ'' by the new covering $E(\mathbf{P})\times\{(a, d)\}$ caused by **P** and determine the cost ç of κ''.

S14 Update c*, κ^* to keep the best covering κ'' found and store placement (d,s) into Q.

S15 End Cycle 4. The updated κ^* describes the least cost increase embedding of all the latest lines $\{(a,d_j): d_j \in \dot{P}(a)\}$ whose placements have been stored in $Q=\{(d_1, s_1), (d_2, s_2), \ldots\}$. Pass κ^* to κ' in order to enable increased δ-cost reduction possibilities for the next line embedding search. Update Q*. Continue with Cycle 3.

S16 End of Cycle 3. All the lines $\{a\}\times\dot{P}(a)$ have been embedded by a least trace-cost increase procedure. That means we have found an approximate embedding for succesor tree $\tilde{\mathfrak{T}}(a)$ now available with κ^* (cost ç*, placement Q*). Set $\xi(a,p):=$ ç*; $\breve{\kappa}:=\kappa_{a,p}:=\kappa^*$; $\forall(x,y)\in Q^*$: $[\dot{\pi}((a,p),x):=y]$. Read $\dot{\pi}((a,p),x)$ as follows: If a is on p then line (a,x) has been assigned a path **P** from p to $y=\dot{\pi}((a,p),x)$ where $\breve{\kappa}^{-1}(a,x)$ are the edges of path $\mathbf{P}\subseteq\mathbf{G}$.

S17 $|V(\mathfrak{T}_i)|=1$. The best location p for entity a can be found via $\xi(a,p):=\min\{\xi(a,p')\}_{p'\in\Pi(a)}$.

S18 (a, p) provides the start for a backtrack procedure. Cost ξ(a,p) belongs to an approximate structure maintained in $\kappa_{a,p}$. Determine the remaining placements via
while $A\neq 0$: **[** $\forall a\in A$:[B:= E(a)\S; $\forall b\in B$: $[\breve{\pi}(b):=\dot{\pi}((a,\breve{\pi}(a)),b)]$; S:=S ∪ B]; A:=B**]**;.

S19 All a∈M were assigned a fixed location $\breve{\pi}(a)$. Embed all lines $(a,b)\in E(\mathfrak{F})\setminus E(\mathfrak{T})$ not yet been treated at all with the successive updating of $\breve{\kappa}$ in order to enable further δ-cost reduction for them.

S20 Layout ($\breve{\pi}_{appr}:=\breve{\pi}, \breve{\varphi}_{appr}:=\breve{\kappa}^{-1}$).

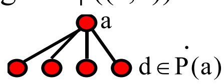
Lemma 1:

- TRACE runs with $\boldsymbol{O}(\mathbf{p}^2\ \mathbf{k}^2\cdot\mathbf{m}\log\mathbf{n})$.
- If Π is single valued and $\delta(t,\breve{\kappa}(t))=\begin{bmatrix}1 \text{ if } \kappa(t)\neq 0\\ 0 \text{ if } \kappa(t)=0\end{bmatrix}$ then TRACE observes cost bound

$C_{appr}\leq 2C_{opt}(1-\frac{1}{k})$, k=|M|, C_{opt} = cost of an optimal layout. Thus, it approximately solves also Steiner's Problem in Graphs with the known effort $\boldsymbol{O}(k\cdot m\cdot\log n)$.

Proof (n=|V(**G**)|, m=|E(**G**)|, k= |M|, p=|Π(M)|):

S3 - S16 is executed dependent on the number of generation i=1, 2, ... ≤ k ⇒ $\boldsymbol{O}(\mathbf{k})$. **S7**: Every time cycle $[_1$ is called it is passed another A_i disjoint to the previous ones. I.e. $|A_1|+|A_2|+\ldots\Rightarrow$ $\boldsymbol{O}$(k). **S8**: Cycle $[_2$ $\boldsymbol{O}(\Pi(A_1)+\Pi(A_2)+\Pi(A_i)+..)\Rightarrow$ $\boldsymbol{O}(k\cdot p)$. **S9**: Cycle $[_3$: The treated $d\in B_i$ are disjoint and belong to one uniquely defined branching $a\in A_i$, $(a,d)\in E(\mathfrak{T}_i)$. $k(p\,|\dot{P}(a)|+\rho\,|\dot{P}(b)|+\rho\,|\dot{P}(c)|+\ldots)\Rightarrow\boldsymbol{O}(k^2\cdot\rho)$. **S10**: Cycle $[_4$ including **S11 - S14:** Calling $\boldsymbol{O}$(m log n)- EMBED needs an effort $\boldsymbol{O}(\mathbf{p}(\mathbf{m}\log\mathbf{n}))$, total: ⇒ $\boldsymbol{O}(\rho^2\cdot k^2\cdot(\mathbf{m}\log\mathbf{n}))$.
The remaining steps need less time. (first part)■

If a non-empty $t\in E(\mathbf{G})$ is used embedding (a,d), $\delta(t,\breve{\kappa}(t))=\begin{bmatrix}1 \text{ if } \kappa(t)\neq 0\\ 0 \text{ if } \kappa(t)=0\end{bmatrix}$, then $\delta(t,\breve{\kappa}(t)\cup\{(a,d)\})-\delta(t,\breve{\kappa}(t))=0$. No cost arises when (a,d) is embedded into a non-empty trace. **SPG** considers the quality Y=Π(M) be connected ⇒ All edges $(a,d)\in E(\mathfrak{F})$ can be assigned $\psi((a,d)):=1$. I.e., any spanning tree $\mathfrak{T}\subseteq\mathfrak{F}$ is a maximum spanning tree $\mathfrak{T}$. If $\mathfrak{T}$ is a *bunch* 

it holds

λ=(a,d)$\in$E($\mathfrak{T}$): t$\in$***dom***($\breve{\kappa}$): $\ell_{\mathbf{G}}$(t)(δ(t, $\breve{\kappa}$(t)$\cup$\{λ\}) $-\delta$(t, $\breve{\kappa}$(t))= 0, t$\notin$***dom***($\breve{\kappa}$): $\ell_{\mathbf{G}}$(t) δ(\{λ\}) =$\ell_{\mathbf{G}}$(t).

Let **H** $\subseteq$**G** the underlying graph embedding tree $\mathfrak{T}_i \subseteq \mathfrak{F}$. To embed line (a,d) using a path $\mathbf{P}_{p,s}$ from p$\in$ V(**H**) to s=Π(d) we distinguish: **a)** s$\in$V(**H**): TRACE uses current covering, i.e. the edges E(**H**). All vertices V(**H**) can successively be reached without any expense. **b)** s$\notin$V(**H**): $\mathbf{P}_{p,s} = \mathbf{P}_{p,q} \cup \mathbf{P}_{q,s}$ where $\mathbf{P}_{q,s}$ = corresponds to a cheapest connection a pair (q,s)$\in$V(**H**) $\times$ V(**G**)\V(**H**). .The embedding of a new line (a,d) is executed corresponding to the strategy **H**:= **H** $\cup$ $\mathbf{P_G}$(V(**H**), Y\V(**H**)) $\cong$ *Enlarge the current tree **H** by a shortest path in **G** from V(**H**) to a vertex out from Y\V(**H**)!*. This is identical with the algorithmic extension step of a Steiner heuristic that tries to find a structure **H** $\subseteq$ **G** observing Y$\subseteq$V(**H**) such that $\sum_{t\in E(H)} \ell_{\mathbf{G}}(t)$ is smallest guaranteeing $C_{appr} \leq 2 \cdot C_{opt}(1-\frac{1}{k})$. Term $(1-\frac{1}{k})$ arises because the most expensive path $P_{x,y}$, x$\in\Pi$(M), y $\in\Pi$(M)\\{x\}, is excluded from the Steiner heuristic's selection. TRACE, however, can easily be modified such the very first two paths P_1:= EMBED(empty, (a, $\breve{\pi}$(a)), (d_1, $\breve{\pi}$(d_1)), P_2:= EMBED(empty, (a, $\breve{\pi}$(a)), (d_2, $\breve{\pi}$(d_2)) determined in step **S11** are compared before they are assigned to the initial covering κ''. Select that path that has smaller trace cost and repeat the EMBED-call for the other. Now, the longest of the possible paths between all the different Y=$\breve{\pi}$(M) $\subseteq$ V(**G**) cannot be embedded at first what is the presupposition for the cost bound iuluding term $(1 - \frac{1}{k})$, [20, 13]. We regard the time requirement: Step **<u>S8</u>** selects only one fixed location for the only entity a. I.e. step **<u>S7</u>** and **<u>S8</u>** provide overhead $\Rightarrow$ ***O*(1).** $[_3$, step **<u>S9</u>,** selects leaf d that has one and only one location s:=$\breve{\pi}$(d) what means that the interior effort of cycle **<u>S10</u>** is ***O***(1). EMBED is called (k−1)-times requiring ***O***(m log n).
Totally $\Rightarrow$ ***O*(k · m log n)** (second part)■

3. Algorithm EMBED

TRACE demands an algorithm that determines a path **P**:= EMBED(κ'',(a,p), (d,s)) from p to s for laying out line (a,d) with smallest trace cost considering an existing covering κ''.

<u>Lemma 2</u>: EMBED can be derived from an ***O*(m log n)** Shortest Path **P** (label setting) algorithm that embeds line λ=(a,d) $\in$E($\mathfrak{F}$) connecting p$\in\Pi$(a) and s$\in\Pi$(d) observing an existing covering κ'' such that $\sum_{e\in E(\mathbf{P})} \ell_{\mathbf{G}}(e)\,\big((\delta(e,\kappa''(e)\cup \{\lambda\}) - \delta(e, \kappa''(e))) + \psi(\lambda)\big)$ is smallest.

<u>Proof</u>: We introduce a new edge cost τ applied to the traces E(**G**) assumed to be covered with the line λ=(a,d)$\in$E($\mathfrak{F}$) observing an existing covering κ'' of lines already embedded :

$\tau(t) := \ell_{\mathbf{G}}(t)\,\big((\delta(t,\kappa''(t) \cup\{\lambda\}) - \delta(t,\kappa''(t)) + \psi(\lambda)\big)$; κ''(t) is empty $\Rightarrow$ $\tau(t) = \ell_{\mathbf{G}}(t)\cdot(\delta(t,\{\lambda\} + \psi(\lambda))$.
τ(t) expresses the cost increase necessary to traverse trace t$\in$E(**G**) while embedding λ. Using any Shortest Path Tree (SPT) algorithm starting in p, τ remains unchanged during its execution. A path **P** from p to s is "shortest" with respect to the <u>additional</u> cost λ consumes when conducted via t. Regarding SPT algorithms we refer to [3-5,9,10,15,16,19] $\Rightarrow$ ***O***(m · log n). ■

4. Layout Optimization of a Power Supply System

The studies go back to an R&D tracing project for a power supply system [6,11] to be laid out into a strongly structured workshop hall **G** (*100 m times 200 m, 446 main power consumers required up to 150 kW, 2 power sources on given locations, 406 consumers to connect with the sources via 1 of 9 distribution boxes on eligible locations, 137 different possibilities for line specification had been derived; project-specific information for the entire cable network which (e.g. the assignment of customers to distributors) had to be observed*).

Table 1 and 2 are given only to show that the cost parts of a real industrial project are about uniquely distributed and that a <u>pre-version</u> of Algorithm TRACE [6,11] succeeded to determine a layout with a 13%-total cost reduction compared to an 1:1 project realized without the pre-version's optimization, [6,11] when embedding 11 power supply clusters $\mathfrak{F}= \{\mathfrak{F}_1, .. , \mathfrak{F}_{11}\}$ into hall **G**.

<u>Trace cost</u> = material (costs for cable line support (cable chambers, plank beds)) + wages (installation cost)

<u>Routing cost</u> = material (pure cable costs) + wages (manual laying cost).

	Expenses for	in DM
routing	pure cable cost laying the cables: material laying the cables: wages	139.699 5.566 67.329
tracing	installing traces: material installing traces: wages	125.243 50.879
Total		388.716
	total routing = 55% total cost	212.594
	total tracing = 45% total cost	176.122

Table 2 The project's cost parts

5. Conclusion

The new approximation algorithm TRACE tackles the problem " Wiring Layout Design Reducing Cable and Trace Cost" surprisingly well. At a first glance, the algorithm proceeds in a strategy used to find approximate solutions for the QSAP [14, 18] via the successive cutting of leaves of a starting flow tree ($\overline{\text{MST}}$ $\mathfrak{T}$). But only the introduction and application of "coverings" (κ) proposed by algorithm TRACE enables the lines' embedding such that the next generation trees can take advantage from the best embeddings of the corresponding successor trees previously determined so far. This strategy conducts lines on common traces so far small cable cost increase leads to higher total cost decrease (cable cost and trace cost). The algorithm can be applied in a variety of other fields like transportation. We only mention the problem of the efficient redirection of certain traffic streams in urban environments where the repulsion influence of δ compels algorithm TRACE to find paths surrounding overcrowded areas as short (or fast) as possible.

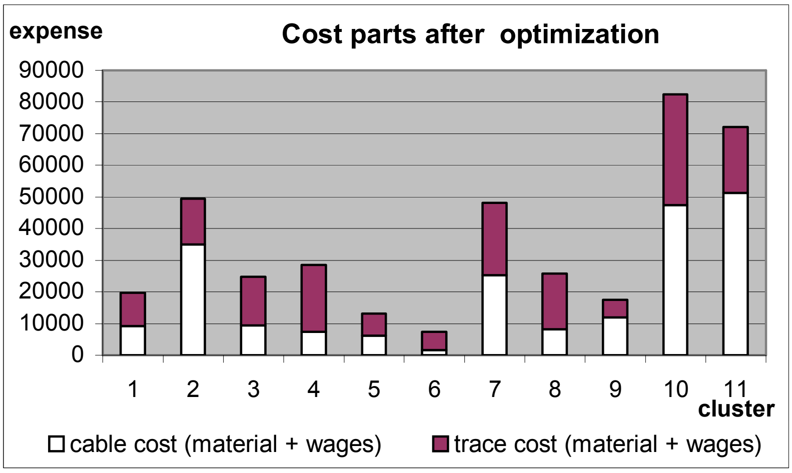


Table 3 Cost distribution each cluster 1 to 11

References

[1] Carraresi P., Malucelli F.: A NEW Lower Bound for the Quadratic Assignment Problem, *Operations Research,* Vol. 40, Supp. No. 1 Jan-Feb (1992), 522-527

[2] Chen N. P.: New algorithms for Steiner tree on graphs. In. Proceedings of the IEEE International Symposium on Circuits and Systems, pa-ges 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] Döring S., Iwainsky A., Richter P.: Some Application of NETOPT, in: Iwainsky A.: Optimization of Connection Structures in Graphs, Akademie der Wissenschaften der DDR, ZKI-Berlin, (1985)

[7] Dreyfus S.F., Wagner R.A.: The Steiner Problem in Graphs, NETWORKS, 1, (1972) 195-207

[8] Duin C. W.: Steiner's Problem in Graphs, PhD thesis, Amsterdam University; (1994).

[9] Gilsinn I., Witzgall, C.: A Performance Comparison of Labeling Algorithms for Calculating Shortest Paths Trees, NBS Technical Note 772, US. Department of Commerce, (1973)

[10] Glover F., Glover R., Klingman D.: Computational Study of an Improved Shortest Path Algorithm, NETWORKS 14, (1984) 25-36

[11] Iwainsky A., Döring, S., Richter P., Schiemangk C.: Optimierung der räumlichen Anordnung von Automatisierungsanlagen, Messen Steuern Regeln 12, Verlag Technik Berlin, (1986) 535-538

[12] Karpinski M. and Zelikovsky A. Z.: New approximation algorithms for the Steiner tree problems. Technical report, International Computer Science Institute, Berkeley, (1995).

[13] Kou L., Markowsky G., Berman L.: A Fast Algorithm for Steiner Trees, Acta Informatica 15, (1981), 41-45

[14] Malucelli F., Pretolani D.:Quadratic Semi-Assignment Problems on Structured Graphs, Ricerca Operativa 69, (1994) 57-78

[15] Moore E.F.: The Shortest Path trough a Maze, Proceedings Intern. Symposium on the Theory of Switching, Part 2, April 2-5, (1957), The Annals of the Computation Laboratory of Harvard University 30, (1959)

[16] Pape U.: Algorithm 562 Shortest Path Length [H], ACM Transactions on Mathematical Software, Vol. 6, No. 3., SEP (1980) 450-455

[17] 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.

[18] Richter P.: Semi Quadratic Assignment Problem – Pragmatic Change of the QAP's Domain Enables Attracive ε-Approximation, Advances in Systems, Signals, Control and Computers, Vol I, ed. V.B. Bajic, pblished by IAAMSAD and the South African Branch of the Academy of Nonlinear Sciences, ISBN 0-620-23137-8, Sep (1998) 69-75

[19] Shier, Witzgall: Properties of Labeling Methods for Determining Shortest Path Trees, J. Res. Natl. Bur. Stand., (1986) 323-333

[20] Takahashi H., Matsuyama A.: An Approximate Solution for the Steiner Problem, Math. Japonica 24, No. 6, (1980), 573-577

[21] Zelikovsky A.Z.: A faster approximation algorithm for the Steiner tree problem in graphs . Information Processing Letters, 46 (1993) 79-83

Peter, H. Richter: *Wiring Layout Design Reducing Cable and Trace Cost*, ISATA 2000, Dublin 25.-27.9.2000, in *Unigraphics Solutions – integrated solutions for complex processes in automotive product development,* Norwich House, Knoll Road, Camberlay, Surrey, GU15 3SY, U.K.
Tel.: +44 (0) 1276 705 171, Fax: +44(0) 1276 705 191. www.ugsolutions.com