Semi Quadratic Assignment Problem

Pragmatic change of the QAP's domain enables attractive ε-approximation

Peter H. Richter (peterrichter@msn.com)
Institute for Informatics, University Potsdam
Post box 601553, 14415 Potsdam, Germany

Abstract

Changing the **QAP**'s hard definition such that the facilities M are allowed to be mapped by a (single-valued, not necessarily injective) function π into the set of possible locations Y subject to a relation Π, $\pi \subseteq \Pi$, it arises the **Semi-QAP** that might be regarded as a relaxation of the **QAP**. In contrast to the **Tree-QAP** (flow graph F is a tree) the corresponding **Semi-Tree-QAP** is solvable in polynomial time. This enables us to show that there is a constructive approximation algorithm, here called **R2,** solving the **Semi-QAP** with $O(k^2 \cdot p^2)$ guaranteeing an $\varepsilon = (C_{appr} - C_{opt}) / C_{opt} = |F| + 1 - |M|$ (**Semi-QAP**'s optimal cost C_{opt}, and approximate cost C_{appr}).

1. Introduction

For about 40 years researchers have been proposing exact and heuristic methods for the Quadratic Assignment Problem **QAP**. It is one of the most considered problems regarding layout optimization and has been exhaustively studied just within the recent years by Pardalos P. M., Burkard R., Christophides N., Wolkowicz H., Rendl F. and others. As stated in [Burkard, 1984], the problem is so hard that an instance with size 36 (he take the example from [Steinberg, 1961]) has not been solved till now. For bigger sizes, heuristic methods must be used.

The **QAP** whose formulation is attributed to [Koopmans, Beckmann, 1957] reads in terms of graphs as follows:

<u>**INSTANCE**</u>: Undirected flow graph F whose vertices M= V(F) represent a set of facilities; flow $\psi: M^2 \xrightarrow{\subseteq} \boldsymbol{R}_+$ (flow, relationship intensity, assigned to the edges E(F)); undirected distance graph **G** seen as a mapping of a structured environment whose vertices Y= V(**G**), |V(F)|= |V(**G**)|, are the facilities' future locations with distances $\boldsymbol{d}_{\mathbf{G}}: E(\mathbf{G}) \to \boldsymbol{R}_+$ assigned to edges E(**G**) of **G**; placement cost $\eta: M \times Y \to \boldsymbol{R}_+$.

<u>**PROBLEM**</u>: We look for a bijection $\pi: M \to Y$, called optimal location function, with cost $\tilde{C}(\pi)$

$$\tilde{C}(\pi) := \min_{\substack{\pi': V(F) \to V(\mathbf{G}) \\ \pi' \text{ bijective}}} \{ \sum_{(a,b) \in E(F)} \psi(a,b) \cdot \boldsymbol{d}_{\mathbf{G}}(\pi'(a), \pi'(b)) + \sum_{a \in M} \eta(a, \pi'(a)) \}. \qquad [1]$$

The **QAP**'s corresponding integer programming formulation had been already formulated by [Koopmans, Beckmann, 1957]. For other **QAP** formulations like *Trace formulation* and *Kronecker Product* (Tensor product) we refer to [Hadley, 1989].

[Sahni, Gonzales, 1976] proved the corresponding decision problem's ***NP***-completeness by transformation from HAMILTONIAN CIRCUIT and showed that the determination of an ε-approximate solution of the **QAP** remains ***NP***-complete. [Lawler, 1963] showed further that TRAVELING SALESMAN is a special case of the **QAP**. Even **Tree-QAP** (flow graph F is a tree, definition below) remains ***NP***-complete since it contains also the TRAVELING SALESMAN as a special case, [Christofides, Bernavent, 1989]. We also refer to [Christofides, 1975], [Christofides., Gerrard, 1976], [Christofides, Mingozzi, Toth, 1981]. A survey of exact algorithms is given by [Pierce, Rowston, 1971] and [Burkard, 1984]. Further essential results came from [Murthy K.A., Pardalos P.M., 1990], [Bazaraa, , Sherali,1979], [Nissen, 1992]. An essential contribution about local search heuristics and recent approaches solving the **QAP** was made by [Pardalos, Murthy, Harrison,1993] and, of course, by [Pardalos, Rendl, Wolkowicz, 1994].

<u>Abbreviations used throughout this paper:</u>

- M= set of entities to be located;
- Y= set of locations;
- $\psi: M^2 \xrightarrow{\subseteq} \boldsymbol{R}_+$ is the flow intensity;
- $\eta: M \times Y \to \boldsymbol{R}_+$ is the location specific expense
- $F = [\underbrace{M}_{V(F)}, \underbrace{\boldsymbol{dom}(\psi)}_{\text{edges } E(F)}]$ flow graph;
- **G** is the graph representing the structured environment;
- **[** vertices, edges **]** = general graph notation;
- Let be **H** some graph. We use:
- V(**H**)= set of vertices of **H**;
- E(**H**)= set of edges of **H**, $E(\mathbf{H}) \subseteq V(\mathbf{H})^2$;
- $E_{\mathbf{H}}(x)$= set of adjacent vertices of $x \in V(\mathbf{H})$;
- $E_{\mathbf{H}}(X)$= adjacent vertices of $X \subseteq V(\mathbf{H})$;
- $\mathbf{G}_\pi(F) = [\pi(M), \varphi_\pi(E(F))]$ is an <u>embedding</u> of F into **G** with $\pi: M \to Y$. φ_π maps E(F) into shortest paths between $x,y \in V(\mathbf{G})$ in **G**.
- $\boldsymbol{d}_{\mathbf{G}}$ describes the distance $\boldsymbol{d}_{\mathbf{G}}(x,y)$ of a shortest path from $x \in V(\mathbf{G})$ to $y \in V(\mathbf{G})$.
- $\mathbf{P}_{\mathbf{G}}(X,Y)$ is a shortest path between X and $Y \subseteq V(\mathbf{G})$ such that $\boldsymbol{d}_{\mathbf{G}}(x,y) := \min_{(x',y') \in X \times Y} \{\boldsymbol{d}_{\mathbf{G}}(x',y')\} \subseteq \mathbf{G}$ between $x' \in X$, $y' \in V(\mathbf{G})$.

- k= |M|;
- ω= |E(F)|;
- n= |V(**G**)|;
- m= |E(**G**)|;
- p= |Y|;

2. Known Heuristic Approaches

Essential heuristics classified corresponding to the heuristic *approaches* of [Foulds, 1983] and observing the results of [Pardalos, Murthy, Harrison, 1993] are the following ones:

Construction Strategies

[Christofides, Benavent, 1989] suggested a relaxation (called algorithm REL1) for the reduced **Tree-QAP** (neglecting the part $\sum_{a \in M} \eta(a, \pi(a))$ in [1]) that first abandoned the original problem formulation's hard request for an injective location function $\pi \subseteq M \times Y$. It imposes the requirement that all immediate neighbors $b \in E_F(a)$ of entity a, with a,b∈ M= V(F), of flow tree $\mathbf{T} \subseteq F$ must be assigned to different locations $\pi(a) \neq \pi(b)$. The authors devised an algorithm (generalization of the algorithm by [Houck, 1977]) that forces solutions to determine mappings $\pi: M \to Y = V(\mathbf{G})$ of k= |M|= |Y| entities into locations that become *locally injective*, although they are not guaranteed to be *globally injective*. REL1 runs with $O(k^4)$. It can be regarded as a special case of **Semi-Tree-QAP** defined below:

[Murthy, Pardalos, 1992] started from the <u>***GRAPH PARTITIONING PROBLEM***</u>: Divide the flow graph's vertices M=

V(F) into two equal subsets A and B (|V(F)| assumed even) such that $\sum_{(a,b)\in A\times B}\psi(a,b)$ is smallest among all two subsets halving V(F) that is known to be ***NP***-complete. We refer to a local search algorithm given by [Kernighan, Lin, 1972]) that determines a local optimum for the **QAP.**

Improvement Strategies: [Burkard, Rendl, 1984], and [Wilhelm, Ward, 1987] proposed a ***SIMULATED ANNEALING*** approach where new candidate solutions of inferior quality are accepted with a certain probability in order to move out of local minima. [Burkard, Rendl, 1984] obtained suboptimal solutions to problems sizes up to 36 within 1-2% of the best known solutions. For instance, they were able to get within 1.83% of the best known solution to [Nugent et al., 1968] in 25.5 sec on a UNIVAX 1100/81.

[Taillard E., 1991] gave a ***TABOO SEARCH*** (**TS**) method based on using a taboo list used to avoid returning to the local optimum just visited by forbidding the reverse move. A special *aspiration criterion* enables a selection of some taboo moves if they are judged to be interesting. If k^2 iterations are allowed and the taboo list size is varied randomly in a range of 10% about the size of the problem, it is possible to reach solutions of excellent quality, depending more on the type of problem than on its size, for problem size $k \geq 20$. The authors think that the search of suboptimal solutions for bigger problems must be done by another procedure, using for example more elaborated concepts of **TS**.

For ***Disturbance Methods*** we refer to [Burkard, 1973].
For ***Evolutionary Methods*** we refer to [Nissen, 1992]

3. Semi-QAP versus QAP?

The **QAP** and **Tree-QAP** concern the intractable determination of k mutually exclusive locations for k entities (facilities, units,…). In contrast to the **QAP**'s and **Tree-QAP**'s hard definition that the location function π: M → Y must be injective we introduce a relation $\Pi \subseteq M \times Y$ serving as frame for the single valued but not necessarily injective location function $\pi \subseteq \Pi$. Thus, it arises a new problem we call **Semi-QAP** and **Semi-Tree-QAP**, respectively**.** The relaxation enables the design of a constructive $O(k^2\cdot p^2)$ heuristic guaranteeing an upper cost bound given by $\varepsilon = C_{appr} - C_{opt}) / C_{opt} = \omega+1-k$ (**Semi -QAP**'s optimal cost C_{opt}, approximate cost C_{appr}). Certain entities may be located together onto one common location so far Π allows that. Further, more than |M| locations are defined as available for possible locations. Note! There is a problem called ***Q**uadratic **S**emi-**A**ssignment **P**roblem* (**QSAP**) (e.g. [Pardalos, Rendl, Wolkowicz, 1994, page 9] that has no likeness at all with our **Semi-QAP** we treat here.

The **QSAP** reads as follows: *Assign via a single valued location function π: M → Y entities M to common locations Y (clusters) such that the dissimilarities μ: $M^2 \to \boldsymbol{R}_+$ between them are minimum:* $\min_{\substack{\text{function } \pi\in M\times Y\\ \pi \text{ single-valued}}} \sum_{a,b\in M}\mu(\pi(a),\pi(b))$.

The following problem formulation comprises the **QAP** and **Tree-QAP** as well as the corresponding **Semi** –Problems.

Problem Formulation | **QAP** / **Semi-QAP** | **Tree-QAP** / **Semi-Tree-QAP** | [2]

INSTANCE

- Graph **G**,
- metric d_G: $V(G)^2 \to \boldsymbol{R}_+$,
- finite set M of facilities (for **QAP** and **Tree-QAP** it holds |M|= |Y|),
- locations $Y \subseteq V(G)$,
- relation $\Pi \subseteq M \times Y$ for **Semi-QAP** and **Semi-Tree-QAP,**
- location expense η: $M \times Y \to \boldsymbol{R}_+$,
- flow ψ: $M^2 \xrightarrow{\subseteq} \boldsymbol{R}_+$ whose flow graph
  - F is not necessarily a tree => **QAP, Semi - QAP**
  - T is a tree => **Tree – QAP, Semi - Tree - QAP**

PROBLEM: Find a bijection $\pi \subset M\times Y$ (**QAP, Tree – QAP**) / total function $\pi \subseteq \Pi$ (**Semi – QAP, Semi – Tree – QAP**) such that the cost

$$\tilde{C}(\pi) = \sum_{(a,b)\in E(F)} \psi(a,b)\cdot d_G(\pi(a),\pi(b)) + \sum_{a\in M}\eta(a,\pi(a))$$

is smallest among all bijections or total functions (Semi-Problems), respectively.

QAP					
e	●	○	○	○	○
d	○	○	○	○	●
c	○	●	○	○	○
b	○	○	○	●	○
a	○	○	●	○	○
M	1	2	3	4	5

○, ● ∈ Y
π: M → Y
● ∈ π(M)

Semi QAP													
e				●					○				
d			○				●					○	
c				●	○			○				○	
b		○								●	○		
a	○					○				●			○
M	1	2	3	4	5	6	7	8	9	10	11	12	13

○, ● ∈ Π(M)
π: M → Π(M)
● ∈ π(M)

Figure 1 The different domains and a possible result (●) of **QAP** and **Semi-QAP.** The **Semi-QAP**'s location function π must not necessarily be injective.

Allowing a comparison **QAP** ⇔ **Semi-QAP** we highlight the domains' difference mainly based on the existence of an implicit location rule $\Pi_{QAP} = M \times Y$ that delivers the frame for a location function π. While the **QAP** and **Tree-QAP** use the ***Exclusive Or*** (XOR) for locating M to Y, the **Semi- QAP** and **Semi- Tree-QAP** use the ***Not-Exclusive Or*** (OR). This should be elucidated by Figure 1 below: The main difference between **QAP** and **Semi-QAP** is caused by the quality of location function π:

QAP: π is bijective ⇔ Each location enables the placing of one and only one facility. Indeed, this quality is responsible for the **QAP**'s intractability ([Lawler, 1963], [Sahni, Gonzales, 1976]).

Semi-QAP: π is unique ⇔ Each location $p\in Y$ is allowed to contain entities $M' \subseteq \Pi^{-1}(p)$, $0 \leq |M'| \leq |\Pi^{-1}(p)|$. The advantage that we are able to design a polynomial exact **Semi-Tree-**

QAP algorithm leading to an acceptable approximation **QAP** algorithm is paid with the price of some inconvenience with respect to Π: In the case that certain entities $M' \subseteq \pi^{-1}(p) \subseteq \Pi^{-1}(p)$ are not allowed to stay together on p (e.g. total size of location p is restricted) further constraints have to be introduced. We regard here the case that all entities $M' \subseteq \pi^{-1}(p) \subseteq \Pi^{-1}(p)$ are allowed to be assigned to p dependent on location function π determined.

4. Semi-Tree-QAP's tractability - Convenient Semi-QAP-Approximation

Previous researches had focused their looking more or less for efficient heuristics strongly attached to $|M| = |Y|$ and for the finding of a bijection π. Let us especially regard the **Tree-QAP** where the flow graph T= **[** M, ***dom***(ψ) **]** is a tree that is to lay out via bijection π: $M \rightarrow Y$ into **G** resulting to an optimal embedding $\mathbf{G}_\pi$ (T) $\subseteq$ **G** with respect to minimum cost $\tilde{C}(\mathbf{G}_\pi(T))$. The **Tree-QAP** remains ***NP***- hard. It contains the Traveling Salesman Problem (**TSP**) as a special case if we consider the flow graph as a path (a special type of tree) with all flows equal to 1. [Christofides, Benavent, 1989] gave in their paper a branch–and- bound algorithm to optimally solve the **Tree-QAP**, though for less than 25 machines in quite reasonable time.

Theorem 1 If there is an exact polynomial **Tree-QAP** -algorithm then there is an ε-optimal polynomial **QAP**-algorithm sufficing $\frac{\tilde{C}(\mathbf{G}_\pi(F)_{appr}) - \tilde{C}(\mathbf{G}_\pi(F))}{\tilde{C}(\mathbf{G}_\pi(F))} \leq \varepsilon = \omega + 1 - k$

Proof: Assume the **QAP**'s flow graph F with respect to ψ and let be $T \subseteq F$ a Maximum Spanning Tree (shortly $\overline{\text{MST}}$ T). Each edge $r \in E(F) \setminus E(T)$ has a smaller flow than any edge within the cycle that results from adding r to E(T), see Figure 2. Otherwise, T couldn't be denoted as $\overline{\text{MST}}$. Assume an optimal **Tree-QAP**-algorithm that embeds T into **G** through shortest links resulting to $\mathbf{G}_\pi(T)$. Let as denote by $\mathbf{G}_\pi(F)_{appr}$ an embedding that results from completing $\mathbf{G}_\pi$ (T) with the remaining shortest links F \ T. For each edge $(a, b) \in E(F)$ a shortest path $\mathbf{P}_\mathbf{G}(\{\pi(a)\}, \{\pi(b)\}) \subseteq \mathbf{G}$ cannot be longer than a shortest path $\mathbf{P}_{\mathbf{G}(T)}(\{\pi(a)\}, \{\pi(b)\})$ within subgraph $\mathbf{G}_\pi(T) \subseteq \mathbf{G}$, i.e. $\boldsymbol{d}_\mathbf{G}(\{\pi(a)\},\{\pi(b)\}) \leq \boldsymbol{d}_{\mathbf{G}(T)}(\{\pi(a)\},\{\pi(b)\})$. It follows, that $\tilde{C}(\mathbf{G}_\pi(F)_{appr}) \leq$

$$\tilde{C}(\mathbf{G}_\pi(T)) + (\omega - (k-1)) \cdot \tilde{C}(\mathbf{G}_\pi(T)) =$$

$\tilde{C}(\mathbf{G}_\pi(T)) \cdot (2+\omega- k)$. With $\tilde{C}(\mathbf{G}_\pi(T)) \leq \tilde{C}(\mathbf{G}_\pi(F))$ we get $\frac{\tilde{C}(\mathbf{G}_\pi(F)_{appr}) - \tilde{C}(\mathbf{G}_\pi(F))}{\tilde{C}(\mathbf{G}_\pi(F))} = \omega + 1 - k.$

Finally we have to show that an algorithm that carries out

- S1 Determination of an $\overline{\text{MST}}$ $T \subseteq F$
- S2 Minimum embedding of flow graph (tree) T into **G** resulting to $\mathbf{G}_\pi(T) \subseteq \mathbf{G} \Rightarrow$ **Tree-QAP**
- S3 Completing $\mathbf{G}_\pi$ (T) to $\mathbf{G}_\pi$ (F) embedding the remaining edges E(F \ T) representing shortest paths in **G** is polynomial time bounded iff the **Tree-QAP**-algorithm (Theorem 1) is too.

We regard again:

- S1 Obviously, a spanning tree $T \subseteq F$ is maximum with respect to cost function ψ: $M^2 \xrightarrow{\subseteq} \boldsymbol{R}_+$ if it is minimum with respect to $-\eta$. That means that we can use the known MST-algorithms referred of [Horowitz, Sahni, 1978], [Papadimitriou, Steiglitz, 1982], or [Richter, 1989] with a worst case time effort $\boldsymbol{O}(\omega \cdot \log k)$.
- S2 Intentionally assumed to be polynomial time bounded (Theorem 1)
- S3 Since $\boldsymbol{d}_\mathbf{G}$ is given, we don't need to refer to some $\boldsymbol{O}(|E(\mathbf{G})| \cdot \log |V(\mathbf{G})|)$ shortest path algorithm (performance analysis [Richter, 1989]). Thus, we need an effort $\boldsymbol{O}(\omega\text{-}k\cdot)$ due to the necessary additional determination of ω - (k-1) shortest distances in **G**.

■

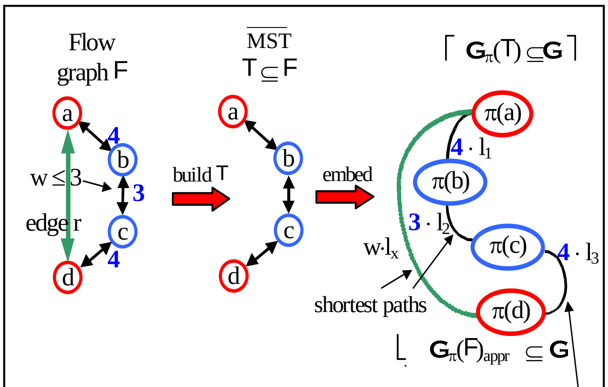

Figure 2 M={a,b,c,d}, k= |M|= 4, ω= 4. Intermediate $w \cdot l_x \leq 4 \cdot l_1 + 3 \cdot l_2 + 4 \cdot l_3$ vertices not depicted.

Experiences with the layout design of industrial applications (e.g. layout optimization with respect to electrical connection structures [Iwainsky, Döring, Richter, Schiemangk, 1986], [Döring, Iwainsky, Richter, 1985], [Schiemangk, Hofmann, Richter, 1990] showed that the **QAP**s hard definition π to be bijective (at least injective) very rarely corresponds to practical domains' definitions. Usually, it holds – at least: for the first planning stages of location/ allocation problems:

(a) The number of possible locations is greater or equal than the number of facilities that are to assign to.

(b) Some facilities are predestined not to all but to a proper subset of all locations.

(c) Some facilities are allowed to stay together on the same location.

In other words, regarding the overwhelming number of applications it is not the question to find a bijection π: $M \rightarrow Y$ but to find a single valued and not necessarily injective function. We can regard the detailed contents of (a)..(c) as *placement relation* Π that constitutes the frame for our location function π:

Function $\pi \subseteq$ relation $\Pi \subseteq M \times Y$, generally $|M| \leq |Y|$, $Y \subseteq V(\mathbf{G})$. $a \in M \Rightarrow \pi(a) \in \Pi(a)$ with $|\Pi(a)| \geq 1$.

$\Pi(a) \subseteq Y \subseteq V(\mathbf{G})$ denotes the set of possible locations capable to embark $a \in M$. $\Pi^{-1}(p) \subseteq M$ denotes the set of facilities allowed to be placed together on $p \in Y \subseteq V(\mathbf{G})$.

With this essential and pragmatic assumption we are able to give an exact polynomial **Semi-Tree-QAP**–Algorithm that efficiently determines a suboptimal solution for the **Semi-QAP.** The result is anticipated by Lemma 1:

Lemma 1 **Semi - QAP** can be solved with an effort $\boldsymbol{O}(k^2p^2)$ guaranteeing an upper cost bound

$$\frac{\tilde{C}(\mathbf{G}_\pi(F)_{appr}) - \tilde{C}(\mathbf{G}_\pi(F))}{\tilde{C}(\mathbf{G}_\pi(F))} \leq \varepsilon = \omega + 1 - k.$$

Proof Part 1: For the cost bound proof we refer to Theorem 1 (augmenting shortest path between all locations whose entities are directly joined by links E(F)\E(T)). Algorithm **R2T** below determines an $\overline{\text{MST}}$ **T** within the flow graph F with respect to flow ψ. Then, **T** is optimal embedded into **G**

by **R2T.** We give the optimal embedding proof (part 2) after explaining **R2** and **R2T.** Since the **Semi-Tree-QAP** is exactly solved by algorithm **R2T** the condition for solving **Semi-QAP** corresponding to **Lemma 1** is given. ■

5. Suboptimal Algorithm R2

In the following we give an $O(k^2 \cdot p^2)$ suboptimal **Semi-QAP** algorithm **R2** based on an optimal **Semi-Tree-QAP** algorithm **R2T** corresponding to Theorem 1. Their time effort will be proved afterwards.

Algorithm R2 approximately solving **Semi- QAP** using location function π determined by **R2T**

Step Action

(1) *Determine the Maximum Spanning Tree* $\overline{\text{MST}}$ $\mathbf{T} \subseteq \mathbf{F}$ *with respect to* ψ*:*

$$\mathbf{T} \subseteq \mathbf{F} \text{ such that } C(\mathbf{T}) := \max_{\substack{\text{tree } T' \subseteq F \\ V(T') = M}} \{ \sum_{r \in E(T')} \psi(r) \};$$

(2) *Determine the embedding* $\mathbf{G}_\pi(\mathbf{T})$ *of* $\mathbf{T}$ *into* $\boldsymbol{G}$ *via location function* π determined by **R2T.** *Observe,* π *is minimum with respect to the embedding of tree* $\boldsymbol{T}$ *into* $\boldsymbol{G}$*. The same location function used to embed flow graph* $\boldsymbol{F}$ *into* $\boldsymbol{G}$ *cannot be denoted optimal because the links* $E(\boldsymbol{F})\backslash E(\boldsymbol{T})$ *weren't considered during the optimization led by* ***R2T:***

Call R2T;

(3) Result: Suboptimal embedding of $\boldsymbol{F}$ into $\mathbf{G}$ via function $\pi: M \longrightarrow Y$, $\pi(a) \in \Pi(a)$, suboptimal cost

$$\tilde{C}(\mathbf{G}_\pi(\mathbf{F})) = \sum_{(a,b) \in E(F)} \psi(a,b) \cdot \boldsymbol{d}_{\mathbf{G}}(\pi(a),\pi(b)) + \sum_{a \in M} \eta(a,\pi(a)) .$$

Sub-Algorithm R2T
exactly solving **Semi-Tree- QAP**

Step Action

(1) *Initialization and regarding the original tree* $\mathbf{T}$ *belonging to the tree generation i=1 with* $\xi(a,p_a)$ *= least cost connecting a on* p_a *to all predecessors in direction to the original leaves:*
$\forall a \in M : [\forall p_a \in \Pi(a) : [\xi(a,p_a) := \eta(a,p_a)]];$
$\mathbf{T}. = \mathbf{T}_1$; $N = V(\mathbf{T}) := M$; $B' := 0$; $i := 1$; **goto** (3);

(2) *Make topical the new tree generation:*
$i := i+1$; $N := N \setminus B'$; $B' := 0$; $\mathbf{T}_i := [V_i, E_i] :=$
$[N, E(\mathbf{T}_{i-1}) \cap N^2]$;

(3) *If* T_i *is singleton then start the final treatment at (8)*:
If $|N| = 1$ **then goto** (8);

(4) *If* T_i *contains only two facilities(leaves) then consider a very simple tree reduction:*
if $|N| = 2$ **then goto** (6);

(5) *Determine set B consisting of all leaves of tree* T_i *and set* A_i *as those neighbors of B having less than 2 neighbors not belonging to B (i.e. the reduced tree remains always connected):*
$B := \{b \in V_i: \boldsymbol{deg}_{Ti}(b) = 1\}$;
$A_i := \{a \in V_i \setminus B: |(\{a\} \times V_i \setminus B) \cap E(T_i)| \leq 1\}$;
goto (7);

(6) *Determine B and* A_i *with respect to a two-vertex tree* $\boldsymbol{T}_i$*:*
$b :\in V_i$; $B := \{b\}$; $A_i := V_i \setminus B$;

(7) *Determine for each pair* $(a, p_a) \in A_i \times \Pi(a)$ *the nearest locations* $\{ \dot{\pi}((a, p_a), b)\}_{b \in B_a} \subseteq \Pi(B_a)$ *for the flow neighbors* $B_a := \{b \in B: a\, E_i\, b\}$ *of a and store their flow weighted distances into* $\{ \dot{c}((a, p_a), b)\}_{b \in B_a}$*, and the total flow weighted distances (a to all* B_a*) into* $\xi(a, p_a)$*:*

$\forall a \in A_i : [_1$ $B_a := \{b \in B: aE_ib\}$; $B' := B' \cup B_a$; $ind(a) := 0$;
  $\forall p_a \in \Pi(a) : [_2$
    $\forall b \in B_a : [_3$ $c(p_b^*) := \dot{c}((a, p_a), b) :=$
      $\min_{p_b \in \Pi(b)} \{\psi(a,b) \cdot \boldsymbol{d}_{\mathbf{G}}(p_a, p_b) + \underbrace{\xi(b, p_b)}_{0 \text{ when } i=1} \}$;
      $\dot{\pi}((a, p_a), b) := p_b^*;\ ]_3$
    $\xi(a,p_a) := \eta(a,p_a) + \sum_{b \in B_a} \dot{c}((a,p_a),b)\ ]_2\ ]_1$;

goto (2);

(8) *At this point,* $\boldsymbol{T}_i$ *is a singleton. Assume* $V_i = N = \{x\}$*. Determine the best location p* for x:*

Minimum cost $\xi(x, p^*) := \min_{p \in \Pi(x)} \{\xi(x,p) + \eta(x,p)\}$ when x is located on $\pi(x) := p^*$;

(9) *We regard the original flow tree* $T. := [V,E]$*. Starting from a = x and* $\pi(a) := p^*$*, backtrack* $\dot{\pi}$ *perfecting the function* $\pi: M \longrightarrow Y$ *through the following procedure:*

1. $\{(a, \pi(a))\}_{a = x}$; $S := 0$;
2. $\{(b, \underbrace{\dot{\pi}((a, \pi(a)), b)}_{\pi(b)}).: b \in E(a) \setminus S\}$; $S := S \cup \underbrace{E(a)}_{\text{neighbors of } a}$;
3. $\{(c, \underbrace{\dot{\pi}((b, \pi(b)), c)}_{\pi(c)}): c \in E(b) \setminus S\}$; $S := S \cup E(b)$;
4. **and so on** till the remotest leaves in $\mathbf{T}$ are reached.

(10) Result: Location function $\pi: M \rightarrow Y \subseteq \Pi(M)$ guaranteeing minimum cost $\tilde{C}(\mathbf{G}_\pi(\mathbf{T})) =$

$$\sum_{(a,b) \in E(T)} \psi(a,b) \cdot \boldsymbol{d}_{\mathbf{G}}(\pi(a),\pi(b)) + \sum_{a \in M} \eta(a, \pi(a)) \text{ with}$$

respect to embedding T.

Complexity Algorithm **R2T** and **R2**
We consider **R2T** exactly solving **Semi-Tree-QAP**. We regard only the most expensive steps, especially the cycle built by (2)..(7). **G** and F are assumed internally represented in list-representation and matrix representation, respectively.

Step	Action with respect to the steps in **R2T** time
(7)	The exterior cycle (2), (3),… (7) depends on the number of generations $i = 1,2,3, \ldots \lambda \leq k$ that cannot take more than $\boldsymbol{O}(k)$

- Cycle $[_1$ "$\forall a \in A_i$" regards another A_i disjoint to all previous $A_{i-1}, A_{i-2}, \ldots$ already treated. Thus, $|A_1| + |A_2| + \ldots + |A_{\lambda-1}| \Rightarrow \boldsymbol{O}(k)$. Considering the determination of B_a determined within $[_1$ totally needs: $\boldsymbol{O}(\mathbf{k}^2)$
- Cycle $[_2$ embedded within $[_1$ takes totally: $\boldsymbol{O}(\mathbf{k} \cdot \mathbf{p})$

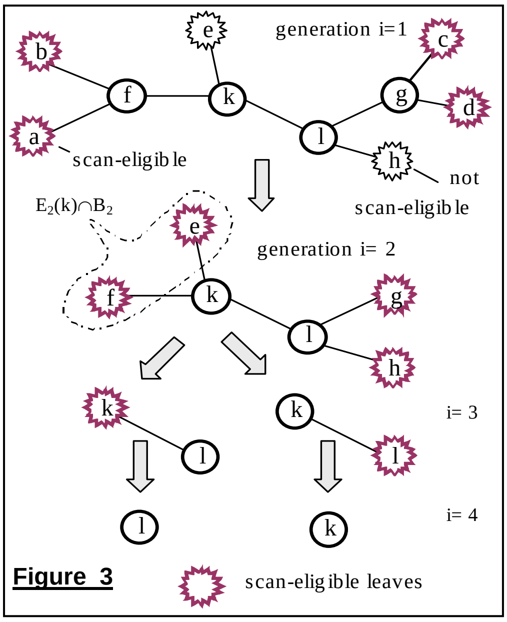

Figure 3

- Cycle $[_3$ "$\forall b \in B_a$" scans always another set of sons B_a belonging to father $a \in A_i$, totally: $O(k^2 \cdot p)$
- "$\forall p_b \in \Pi(b)$" in " $\min_{p_b \in \Pi(b)}$ " is embedded in $[_1$, $[_2$, and $[_3$ totally: $\underline{O(k^2 \cdot p^2)}$

(9) is executed in a top-down manner starting from the last generation tree $\mathbf{T}_{i=\lambda} = [\{x\},0]$ backtracking via $\{x\} \Rightarrow E(x) \Rightarrow E(E(x))\backslash\{x\} \Rightarrow E(E(E(x)))\backslash E(x) \Rightarrow$.(treating always the outer situated neighbors regarding $\mathbf{T}_1 = [V, E]$) to determine their best places stored in $\dot{\pi}$ $\Rightarrow$ totally: $O(k^2)$

Algorithm **R2** includes **R2T**, i.e. from an embedding $\mathbf{G}_\pi(\mathbf{T})$ of $\overline{\text{MST}}$ $\mathbf{T} \subseteq \mathbf{F}$ into $\mathbf{G}$, setting the remaining links $E(\mathbf{F})\backslash E(\mathbf{T})$ into $\mathbf{G}$ with an additional effort $O(k^2) \Rightarrow$ totally: $\underline{O(k^2 \cdot p^2)}$

6. Semi-Tree-QAP is in *P* (Proof Lemma 1 part 2)

First, we will prove that the embedding $\mathbf{G}_\pi(T) \subseteq \mathbf{G}$ of the $\overline{\text{MST}}$ $T = [V, E] \subseteq F$ via π into $\mathbf{G}$ is minimum as to the

$$C(\mathbf{G}_\pi(T)) := \sum_{\forall (a,b) \in E(T)} \psi(a,b)\ d_G(\pi(a),\pi(b)) + \sum_{a \in M} \eta(a,\pi(a)) :$$

1. The repeated proceeding steps (2) ..(7) of algorithm **R2T** finally enables the determination of a minimum $(a, p_a^*) \in M \times \Pi(M)$ with respect to $\xi(a,p_a^*) = C(\mathbf{G}_\pi(T))$ leading to the **Semi-Tree-QAP**'s optimal location function π determined via the following top down procedure.
2. The tracing-back via the places (a,p^*)

$$\downarrow$$

$$\{(\underbrace{b, \pi(b)} := \dot{\pi}((a, \overbrace{p^*}),b)) : b \in E(a)\}$$

$$\searrow$$

$$\{(\underbrace{c, \pi(c)} := \dot{\pi}((\overbrace{b, \pi(b)}),c)) : c \in E(E(a))\backslash\{a\}\}$$

$$\downarrow$$

$$\{(d, \pi(d) = \dot{\pi}((\overbrace{c, \pi(c)}),d)) : d \in E(E(E(a))) \backslash (\{a\} \cup E(a))\}$$

and so on.

We define a special set $B \subseteq V$ (shortly $V = V(T)$, $E = E(T)$) of leaves,, called *scan-eligible leaves*, and their fathers $A \subseteq V$ with respect to the current tree T as follows:

- $B := \{b \in V:\ \boldsymbol{deg}_T(b) = 1 \wedge$ set $E(E(b))$ contains at most 1 node not being a leaf$\}$.
- $A := E(B)$.

<u>Example</u> (see Fig. 3): $B := \{a,b,c,d\}$, $A := E(B) := \{f, g\}$. If b is a leaf, E(b) uniquely determines the father f of b and E(E(b)) provides all neighbors of father f. Thus, B contains all leaves whose fathers will evolve to leaves if cutting their sons B. In this sense, the bold marked leaves of Figure 3 belong to B. E.g., entity e will not belong to B since its father k has two neighbors not being a leaf (fathers f and l).

R2T draws advantage from this view by enabling a bottom-up procedure such that each tuple $(a, p_a) \in \{a\} \times \Pi(a)$ comes into consideration to search for the best locations out from $\Pi(b) \subseteq V(\mathbf{G})$, $b \in E(a) \cap B$ with respect to T. In the case that flow graph T has a form $T = [\{k, l\}, \{(k, l)\}]$, see Fig. 3, we arbitrarily regard either ($k \in B$ and $l \notin B$) or ($l \in B$ and $k \notin B$) as stated with statements (3) and (4) in **R2T**.

Now, we come back to our $\overline{\text{MST}}$ $T \subseteq F$. We introduce the concept *generation* in order to describe the changing structure of $T \subseteq F$. We consider the graph T_i, i=1,2,.., and special sets A_i, B_i, recursively defined by the following rules:

<u>Rules for the construction generation trees</u>

$T_{i+1} = [V_{i+1}, E_{i+1}] := [V_i \backslash B_i, E_i \parallel (V_i \backslash B_i)]$ = tree of *generation* i+1. $T := T_1$ is the original $\overline{\text{MST}}$ $\subseteq F$. i.e. i= 1 is the starting generation. Generally, T_i is the tree obtained from T_{i-1} by deleting all scan-eligible leaves B_{i-1} of the previous generation i-1 and all edges incident to them.

$B_{i+1} := \{b \in V_{i+1}:\ \boldsymbol{deg}_{T_{i+1}}(b) = 1 \wedge E_{i+1}(E_{i+1}(b))$ contains at most 1 node not being a leaf $\}$.

B_{i+1} is the set of *scan-eligible leaves* of generation i+1.

$A_{i+1} := \{a \in V_{i+1}: \exists (a,b) \in (\{a\} \times B_{i+1}) \cap E_{i+1}\} = E_{i+1}(B_{i+1})$ = set of all *scan-eligible fathers* of generation i+1= counter domain of relation E_{i+1}.

Clearly, T_i, i= 1,2,.., is a tree and there exists some $\lambda \geq 1$ such that T_λ remains left as singleton $[\{s\}, 0]$, $s \in M$. Let us call this λ the level of T..

<u>Variables introduced as to generation $1 \leq i \leq \lambda$</u>

The following definitions will already reveal the intention of the algorithmic strategy that meets our objective "Optimal Embedding Tree T into **G**" for **Semi-Tree-QAP.**

- $\dot{\pi} : A_i \times \Pi(A_i) \times B_i \rightarrow \Pi(E_i(A_i) \cap B_i)$, i.e. $\dot{\pi}((a,p_a),b) \cong$ "If a is placed on p_a then $p_b := \dot{\pi}((a,p_a), b) \in V(\mathbf{G})$ is the location for b such that there is no other one with smaller cost than $\dot{c}((a,p_a), b)$".
- $\xi(a,p_a) = \eta(a,p_a) + \sum_{b \in B_i \cap E_i(a)} \dot{c}((a,p_a),b) \in \boldsymbol{R}_+$ = minimum 'potential' necessary to link all the leaves $b \in B_i \cap E_i(a)$ within tree T_i provided entity a has been placed on $p_a \in \Pi(a)$.
- $\dot{c}((a,p_a), b) = \min_{p_b \in \Pi(B_i \cap E_i(a))} \{\psi(a,b) \cdot \boldsymbol{d_G}(p_a, p_b) + \xi(b, p_b)\}$ = connection cost assigned to the location $\dot{\pi}((a,p_a), b)$.
- $\xi(b,p_b) = \eta(b,p_b) + \sum_{c \in B_{i-1} \cap E_{i-1}(b)} \dot{c}((b,p_b),c) \in \boldsymbol{R}_+$

and so on .... Then, we can realize:

Lemma 2: Starting with generation i:= 1 and proceeding according to i:= 1, 2, ..k, . λ-1, the successive construction of T_i, A_i, B_i, and the corresponding determination of the tuples $(\dot\pi((a,p_a),b),\ \dot c((a,p_a),b)) \in V(\mathbf{G}) \times \boldsymbol{R}_+$ observing

$$\dot c((a,p_a),b) := \min_{p_b \in \Pi(B_k \cap E_k(a))} \{\psi(a,b)\cdot \boldsymbol{d}_{\mathbf{G}}(p_a,p_b) + \underbrace{\xi(b,p_b)}_{\eta(b,p_b)\text{ if } i=1}\}$$

and the potentials $\xi(a,p_a) = \eta(a,p_a) + \sum_{b\in B_k\cap E_k(a)} \dot c((a,p_a),b)$ for all $((a,p_a),b) \in A_k \times \Pi(A_k) \times (B_k \cap E_k(a))$ guarantees that $\xi(a,p_a)$ of generation k is smallest with respect to all locations in Y realizing the links directed from $A_k \Rightarrow B_k \Rightarrow B_{k-1} \Rightarrow B_{k-2} \Rightarrow \ldots \Rightarrow .. B_1 \subseteq V_1 = V$.

Proof by complete induction over the generation i, $1 \le i \le: \lambda$:
On the basis of the successive construction of the sets T_i, A_i, and B_i, we prove the assertion for

i:= 1 $\Rightarrow T_1$(original $\overline{\text{MST}}$): We regard each tuple $(a, p_a) \in A_1 \times \Pi(a)$, the neighbors $b \in E_1(a) \cap B_1$ of a, and their possible places $(b, p_b) \in (E_1(a) \cap B_1) \times \Pi(E_1(a) \cap B_1)$. The determination of the neighbors' best locations
$\{(\dot\pi((a,p_a),b), \dot c((a,p_a),b))\}_{b\in E_1(a)\cap B_1}$ for all leaves $E_1(a) \cap B_1$ on their possible locations $p_b \in \Pi(E_1(a) \cap B_1)$ is easily executed via the evaluation

$$c := \psi(a,b)\cdot \boldsymbol{d}_{\mathbf{G}}(p_a,p_b) + \underbrace{\xi(b,p_b)}_{\eta(b,p_b)\text{ if } i=1}$$ and stored into

$(\dot c((a,p_a),b) := c,\ \dot\pi((a,p_a),b) := p_b)$ iff $c < \dot c((a,p_a),b)$.

Obviously, $\xi(a,p_a) = \eta(a,p_a) + \sum_{b\in B_1\cap E_1(a)} \dot c((a,p_a),b) \in \boldsymbol{R}_+$ is then the sum of best connections so far all leaves $E_1(a) \cap B_1$ of unit a were treated to find their best places. I.e. the implicit procedure above enables the correct determination of the set $\{\xi(a,p_a): (a, p_a) \in A_1 \times \Pi(a)\}$ using best placements $\{(\dot\pi((a,p_a),b), \dot c((a,p_a),b))\}_{b\in E_1(a)\cap B_1}$. Thus, the assertion is correct for **i** = 1.

1 < i= k $\Rightarrow T_k$: Suppose the assertion is true for i= k. As we had calculated the values $\{\xi(b,p_b): (b,p_b) \in A_{k-1} \times \Pi(b)\}$ during the treatment of the last generation k-1 (b was belonging to A_{k-1}) we are able to determine the current generation's potentials

$$\dot c((a,p_a),b) = \sum_{b\in B_k\cap E_k(a)} \left(\min_{p_b\in\Pi(b)} \{\boldsymbol{d}_{\mathbf{G}}(p_a,p_b)\cdot\psi(a,b) + \xi(b,p_b)\}\right)$$

for all $((a,p_a),b) \in A_k \times \Pi(a) \times B_k \cap E_k(a)$, respectively the tuples $\{(\dot\pi((a,p_a),b), \dot c((a,p_a),b))\}_{b\in B_k\cap E_k(a)}$. Notice, that $\xi(b,p_b)$ contains the least connection costs necessary to link all the neighbors connected to b as to their best placements. Thus, $\xi(a,p_a) := \eta(a,p_a) + \sum_{b\in B_a} \dot c((a,p_a),b)$ can be determined for all

$((a,p_a),b) \in A_k \times \Pi(a)$.

i =k +1 We build the next generation tree $T_{k+1}(A_{k+1}, B_{k+1})$ considering i= k+1. As we have calculated the set of values $\{\xi(a,p_a): (a,p_a) \in A_k \times \Pi(a)\}$ during the treatment of generation k we are able to determine the current generation's potentials

$$\dot c((x,p_x),a) = \sum_{\underbrace{a\in E_{k+1}(x)\cap B_{k+1}}_{\text{leaves } B_x}} \min_{p_a\in\Pi(a)} \{\{\boldsymbol{d}_{\mathbf{G}}(p_x,p_a)\cdot\psi(x,a) + \xi(a,p_a)\}$$

for all $((x,p_x),a) \in A_{k+1} \times \Pi(x) \times \underbrace{E_{k+1}(x) \cap B_{k+1}}_{B_x}$ respectively the tuples $\{(\dot\pi((x,p_x),a), \dot c((x,p_x),a))\}_{a\in B_x}$. It follows that $\xi(x,p_x) := \eta(x,p_x) + \sum_{a\in B_x} \dot c((x,p_x),a)$ can be determined for all $((x,p_x),a) \in A_{k+1} \times \Pi(x)$ representing minimum connection expenditure. The lemma's assertion is true for i= k+1 provided it is true for i= k < λ . Therefore It follows that the assertion is true for all natural numbers 1,2,..k, .. λ-1

If k= λ, we have reached the final stage of the bottom-up like procedure implicitly given above. Only one entity a, $\{a\} = A_\lambda$, has to be considered for finding its best location $p^* \in \Pi(a)$. Corresponding to **Lemma 2** above, the set $\{\xi(a,p_a)): (a, p_a) \in \{a\} \times \Pi(a)\ \}$ for the last treated entity $a \in M = V$ represents the total expenditure depending on the places Π(a) necessary to realize the best connections from a on p_a to its sons' locations $\dot\pi((a,p_a),b) \in \Pi(E_\lambda(a) \cap B_\lambda)\}$ (and from there to the further successors directed to the original leaves B_1), we only need to build the minimum $\xi(a,p^*) := \min_{p\in\Pi(a)} \{\xi(a,p)\}$ and $\vec\pi_{appr}(a) := p^*$. Thus, we have got that placement tuple (a, p*) whose top-down sequence (see below) within the original tree T= **[**V, E**]** guarantees the finding of the best inferior locations:

1. a has to be placed on π(a)):
   $\Rightarrow \{(a, \pi(a))\}$; S:= 0;
2. The entities $b \in E(a)\backslash S$ have to be placed on $\pi(b) := \dot\pi((a,\pi(a)),b)$ that are the destinations for the lines $\{\psi(a,b): b \in E(a)\}$ via shortest paths: $\{\mathbf{P}_{\mathbf{G}}(\{\pi(a)\},\{\pi(b)\}): b \in E(a)\}$,
   $\Rightarrow \{(b, \underbrace{\dot\pi((a,\pi(a)),b)}_{\pi(b)}): b \in E(a)\backslash S\}$; S:= S ∪ E(a);
3. Destinations for the lines $\{\psi(b,c): c \in E(b)\backslash S\}$ via shortest paths $\{\mathbf{P}_{\mathbf{G}}(\{\pi(b)\},\{\pi(c)\}): c \in E(b)\backslash S\}$,
   $\Rightarrow \{(c, \underbrace{\dot\pi((b,\pi(b)),c)}_{\pi(c)}): c \in E(b)\backslash S\}$; S:= S ∪ E(b);

   ⇒ so on till the remotest leaves in T are reached.

It follows that the embedding of the $\overline{\text{MST}}$ T ⊆ F into $\mathbf{G}_\pi(T) \subseteq \mathbf{G}$ is minimum with respect to the cost $C(\mathbf{G}_\pi(T))$:= $\sum_{\forall(a,b)\in E(T)} \{\boldsymbol{d}_{\mathbf{G}}(\pi(a),\pi(b))\cdot\psi(a,b)\}$ based on $\pi: M \to \mathbf{Y}$, $\pi \subseteq \Pi$.

■

The proof above gives the certainty that **R2** really enables the minimum embedding of the maximum spanning tree T ⊆ F into **G** resulting to $\mathbf{G}_\pi(T)$ that ensures the cost bound for **Semi-QAP** corresponding to Theorem 1: The remainder links E(F)\E(T) are simply added to the already determined embedding $\mathbf{G}_\pi(T)$ resulting to $\mathbf{G}_\pi(F)$ that suffices the ε-approximation predicted.

7. Concluding Remarks

In this paper we introduced the **Semi-QAP** and **Semi-Tree-QAP** problem as an interesting alternative to the **QAP** and **Tree-QAP** because to its relevance for industrial applications. Due to the presented $\boldsymbol{O}(k^2p^2)$ optimal **Semi-Tree-QAP** algorithm **R2T** we are able to solve **Semi-QAP** guaranteeing an upper cost bound $\frac{\tilde{C}(\mathbf{G}_\pi(F)_{appr}) - \tilde{C}(\mathbf{G}_\pi(F))}{\tilde{C}(\mathbf{G}_\pi(F))} \leq \varepsilon = \omega + 1 - k$ realized by algorithm **R2**.

As we would expect, algorithm **R2** that uses a polynomial **Semi-Tree-QAP** algorithm is very attractive for

sparse graphs. e.g., in the case that flow graph F is a tree (ω= k-1) $\Rightarrow \varepsilon = 0$,

graphs with a high variation coefficient of the flow cost distribution. Clearly, if some edges E' $\subseteq$ E(F) have a flow intensity much higher than the remaining edges, set E' will be added to the edges E(**T**) of $\overline{\text{MST}}$ T and therefore minimum allocated via shortest paths within E(**G**). That means that the remaining edges E(F) \ E(T) cannot decisively influence the total cost mainly determined by the minimum embedding of the maximum flow edges E(T).

Thus, designers are well advised to draw attention whether or not the **QAP**'s hard definition might be relaxed conforming to the **Semi-QAP**'s problem formulation.

8. References

[1] Bazaraa M.S., Sherali H.D.: New approaches for solving the quadratic assignment problem, in: *Operations Research Verfahren*, Vol. **32**, (**1979**) 29--46

[2] Burkard R.E.: Die Störungsmethode zur Lösung Quadratischer Zuordnungsprobleme, Ops. Res. Verfahren, 16, (**1973**), 84-108

[3] Burkard R.E.: Quadratic Assignment Problem, *Eur. J. Opnl. Res*, 15, (**1984**) 283-289

[4] Burkard R.E., Rendl F.: A thermodynamically motivated simulation procedure for combinatorial optimization problems, *European Journal of Operations Research*, 17, (**1984**), 169-174

[5] Christofides N.: Graph Theory. An Algorithmic Approach, Academic Press London, (**1975**)

[6] Christofides N., Gerrard M.: Special cases of the Quadratic Assignment Problem. Management Science Report No. 391, Carnegie Mellon University, Pittsburgh, (**1976**)

[7] Christofides N., Benavent E.: An exact algorithm for the Quadratic Assignment Problem on a tree, *Operations Research*, Vol. 37, No.5, Sep-Oct (**1989**) 760-768

[8] Christofides N., Mingozzi A. and Toth P.: Contributions to the Quadratic Assignment Problem, *European Journal Operational Research* , Vol 4, (**1980**), 243-247

[9] Döring, S., Iwainsky A., Richter P.: Some Application of NETOPT, In: Iwainski A. (Ed.) Optimization of Connection Structures in Graphs, Central Institute for Cybernetics and Information Processes (CICIP) Berlin, (**1985**) 240-259

[10] Floyd R.W.: Algorithm 97: Shortest Path; *Comm. ACM 5*, (**1962**), 345

[11] Foulds L.R.: The heuristic problem solving approach, *J. Op. Res. Soc*., Vol. 34, No. 10, (**1983**), 927-934

[12] Hadley S.W.: Continuous optimization approaches for the Quadratic Assignment Problem, PH.D. Thesis, University Waterloo, (**1989**)

[13] Horowitz E., Sahni S.: Fundamentals of Computer Algorithms, Computer Science Press, Inc. (**1978**)

[14] Houck D., Picard J.C., Queyranne M., Vemuganti R.R.: The Traveling Salesman Problem and Shortest n-Paths, University of Maryland, College Park, (**1977**)

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

[16] Kernighan B., Lin S.: An efficient heuristic procedure for partitioning graphs, *Bell Systems Journal*, 49, (**1972**) 291-307

[17] Koopmans T., Beckmann M.: Assignment problems and the Location of Economic Activities, Econometrica 25, (**1957**) 55-76

[18] Lawler E.: Quadratic Assignment Problem, *Management Science*, Vol 9, (**1963**) 586-599"

[19] Murthy K.A., Pardalos P.M.: A polynomial-time approximation algorithm for the Quadratic Assignment Problem, The Pennsylvania State University", CS-33-90, July, (**1990**)

[20] Murthy, K.A., Pardalos, P.M. and Li, Y.: A local search algorithm for the Quadratic Assignment Problem, *Informatica*, Vol 3, #4, (**1992**) 524--538"

[21] Nissen V.: A new efficient evolutionary algorithm for the Quadratic Assignment Problem, *Operations Research Proceedings*, editor K.W. Hansmann and A. Bachem and M. Jarke and W.E. Katzenberg and A. Marusev, Springer Berlin, (**1992**) 259--267

[22] Nugent C.E., Vollman T.E., Ruml J.: An Experimental Comparison of Techniques for the Assignment of Facilities to Locations ", *Operations Research* , Vol 16, (**1968**) 150-173

[23] Pardalos P. M., Murthy K. A., Harrison T. P.: A Computational Comparison of Local Search Heuristics for Solving Quadratic Assignment Problems, *Informatic*a, Vol. 4, No.1-2, (**1993**), 172-187

[24] Pardalos P. M., Rendl F., Wolkowicz H.: The Quadratic Assignment Problem. In: Quadratic Assignment and Related Problems, DIMACS Series Vol. 16, American Mathematical Society (**1994**) 1- 42

[25] Papadimitriou C.H., Steiglitz K.: Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, (**1982**)

[26] Pierce J., Crowston W.: Tree Search Algorithms for the Quadratic Assignment Problem, *Naval Res. Logist. Quart*. 18, (**1971**), 1-36

[27] Richter P.: Neue effektive Algorithmen zur Optimierung von Verbindungsstrukturen in Graphen, Dissertation, Akademie der Wissenschaften, Zentralinst. für Kybernetik und Informationsprozesse, Berlin (**1989**) $\Rightarrow$ Library of the Humboldt University Berlin/ Germany, 90 HB 2553, # 00001100141237

[28] Sahni S., Gonzalez T.: P-complete Approximation Problems, *J.Assoc. Comput. Mach*. 23, (**1976**), 555-565

[29] Schiemangk C., Hofmann J., Richter P.: Mathematical Modeling and Optimization of FMS-Layouts. 11th IFAC World Congress: “Automatic Control in the Service of Mankind”, Vol.9, Tallin, Estonia, USSR, August 13-17 (**1990**)

[30] Taillard, E.: Robust Tabu search for the Quadratic Assignment Problem", PARCOMPUT, Vol 17, (**1991**), 443--455

[31] Wilhelm M.R., , Ward T.L.: Solving Quadratic Assignment Problems by Simulated Annealing, *IIE Transactions*, Vol 19, (**1987**), 107-119