ELSEVIER

Discrete Applied Mathematics 70 (1996) 37–55

DISCRETE APPLIED MATHEMATICS

Polygon scheduling

Johann Hurink

Universität Osnabrück, Fachbereich Mathematik/Informatik, 49069 Osnabrück, Germany

Received 26 August 1994; revised 20 June 1995

Abstract

Consider a set of circles of the same length and r irregular polygons with vertices on a circle of this length. Each of the polygons has to be arranged on a given subset of all circles and the positions of the polygon on the different circles are depending on each other. How should the polygons be arranged relative to each other to minimize some criterion function depending on the distances between adjacent vertices on all circles?

A decomposition of the set of all arrangements of the polygons into local regions in which the optimization problem is convex is given. An exact description of the local regions and a sharp bound on the number of local regions are derived. For the criterion functions minimizing the maximum weighted distance, maximizing the minimum weighted distance, and minimizing the sum of weighted distances the local optimization problems can be reduced to polynomially solvable network flow problems.

Keywords: Cyclic scheduling

1. Introduction

In connection with railway and road transportation problems Cerny [5] mentioned the following scheduling problem. A set of train-routes is passing through some railway station or is using a common track. Trains of the same route arrive regularly, i.e. the time intervals between two successive trains of the same route are constant. How should the routes be scheduled such that the safety interval between successive trains in the station or on the track becomes as large as possible?

A mathematical description of this problem may be given as follows. We represent the scheduling period by a circle. The arrivals of trains of one route correspond to the vertices of a regular polygon. The vertices of this polygon are lying on the circle. So each route is represented by a regular polygon with vertices on the circle. How should these polygons be moved relative to each other, such that the minimal distance between neighboring vertices on the circle is maximized?

Guldan [8] gave an algorithmic solution method for this problem. The main idea is to decompose the set of all possible schedules of the polygons into an exponential

SSDI 0166-218X(95)00100-X

number of local regions. Each local region can be characterized by an acyclic graph and the corresponding local optimization problem can be solved by longest path calculations.

Another solution method for the problem is given by Vince [14]. His method depends on a reformulation of the problem as a graph theoretic problem. The running times of both mentioned algorithms are exponential.

In connection with the above-mentioned railway problem, one can also think about some other optimization criterion. If all the trains use a common track, then it is in the interest of the passenger that the average waiting time or the maximum waiting time becomes as small as possible. These two optimization criteria and the optimization criteria to maximize the safety interval are special cases of one criterion function $f(u_1, \ldots, u_m)$ depending on the time intervals u_k between the arrivals of consecutive trains. This function has the form

$$f(u_1, \ldots, u_m) = \sum_{k=1}^{m} u^p, \tag{1.1}$$

where p is an arbitrary but fixed value with $-\infty \leqslant p \leqslant 0$ or $1 \leqslant p \leqslant \infty$. In the case $p = \infty$ and $p = -\infty$ the problem of minimizing the function (1.1) reduces to problems of the form

$$\text{minimize} \max_{1 \leqslant k \leqslant m} u_k \tag{1.2}$$

and

$$\text{maximize} \min_{1 \leqslant k \leqslant m} u_k, \tag{1.3}$$

which describe the objectives to minimize the maximum waiting time and to maximize the safety interval.

Burkard [4] has shown that for the case of two regular polygons and for an arbitrary value of p the function (1.1) attains its minimum if one of the distances between vertices of the two polygons is equal to

$$A \cdot [2\,\mathrm{lcm}(m_1, m_2)]^{-1},$$

where A denotes the length of the circle, m_1 and m_2 the number of vertices of the two polygons and $\mathrm{lcm}(m_1, m_2)$ the least common multiple of the numbers m_1 and m_2.

Another analytic result for problems of this type is given by Hurink [9]. Here the optimal value of the criterion function (1.3) for three regular polygons is given explicit in dependence of the number of vertices of the polygons. For problems with four or more polygons there are only analytic results for a few very special type of problems (see [4, 2]).

The papers mentioned so far are all dealing with scheduling regular polygons. But in practice it appears that the arrivals of trains are not regular. It may happen that there are fixed periodic patterns for the arrivals of the trains of one route. In the mathematical description this leads to a replacement of the regular polygons by irregular polygons. The length of the scheduling period (i.e. the length of the circle) will now be the least common multiplier of the length of all period patterns for the different routes.

This type of polygon scheduling problems was first discussed by Brucker and Meyer [3]. For the case of two irregular polygons and the criterion functions (1.2) and (1.3) they have developed efficient algorithms. The running time of these algorithms depends polynomially on the number of vertices of the two polygons.

The problem of scheduling r irregular polygons with the criterion function (1.1) is considered in Brucker et al. [1]. A solution of the problem, based on a decomposition of the set of all schedules into local regions in which the optimization problem is convex, is given. Furthermore, it is shown that the problem is NP-hard for every value of p.

In this work we will consider the problem of scheduling r irregular polygons with a generalized version of the function (1.1) as optimization criterion. Each distance on the circle will be weighted by a nonnegative number. In connection with railway problems a weight for a distance for instance may be the number of passengers for whom this distance is of importance.

For the problem of scheduling trains this polygon scheduling problem is only able to optimize the train schedule on one track or in one station. However in practice the schedules in different stations or on different tracks are not independent and so one has to coordinate these schedules. This leads to a generalization of the polygon scheduling problem where polygons have to be scheduled on several circles simultaneously. More precisely, we have a set of circles and a set of polygons. The circle length is again determined as the least common multiplier of the length of all period patterns for the different routes. Each of the polygons has to be scheduled on a given subset of the circles and between the position of one polygon on two different circles there are fixed differences. The criterion function again will be the weighted version of the function (1.1), where $u_1, \dots, u_m$ now denote the distances between neighboring vertices on all circles.

A problem of this type is mentioned by Cerny and Guldan [6]. In connection with transportation problems they define several polygon scheduling problems. One of these problems is related to a special version of the problem formulated above, in which there are only regular polygons. However, Cerny and Guldan do not give a solution method for their problem.

We will start with a mathematical formulation of the considered problems in Section 2. In Section 3 we will reformulate these continuous optimization problems as a discrete optimization problem by decomposing the set of all possible schedules into a finite number of local regions. This decomposition is based on the decomposition given in Brucker et al. [1]. We will refine this approach to get a one-to-one characterization of the local regions and a compact description of the number of local regions.

In Section 4 we will consider the local optimization problems in more detail. For the optimization criteria of maximizing the minimum weighted distance, minimizing the maximum weighted distance and minimizing the sum of weighted distances the local optimization problems are reduced to network flow problems which are polynomially solvable. Topics for further research and some concluding remarks are mentioned in Section 5.

2. Formulation of the problem

Cerny [5] used a railway scheduling problem as motivation for the polygon scheduling problem. A set of train-routes passes through some railway station. For each route the distances between successive trains are known. The problem is to find a schedule of the trains which minimizes some objective depending on the distances between successive trains. If we represent the scheduling period by a circle, the arrivals of trains corresponds to vertices on the circle. Since the distances between successive trains of one route are fixed, we may consider the vertices belonging to the trains of one route as a polygon with vertices on the circle. This leads to the following problem.

General problem. *Given are r polygons denoted by $P_1, \ldots, P_r$ with vertices on a circle of length A. How should the polygons be moved relative to each other, such that an objective function depending on the distances between adjacent vertices is optimized?*

In practice there are many different stations which are connected by tracks. The trains of a route pass through some of these stations in a given order. The arrivals of a train at two consecutive stations on the route differs by the time the train uses to travel between the stations. Therefore, the position of the polygons at the different stations corresponding to this route are depending on each other. However, with the general problem we just can optimize the schedule for each of the stations independently. This will lead to contradictions between the calculated arrival and departure times for trains of a route at the different stations.

Therefore, we will generalize the general problem in such a way, that we schedule polygons on different circles simultaneously. The circles will correspond to the different stations. Between the polygons on the different circles there will be connections which are expressed by paths in a graph. A path will express the order in which the trains of a route pass through the stations and the travel times will be expressed by the length of the arcs in the graph. To simplify the notation, we will assume that the travel times between two stations are the same for all trains. If we want to model the more realistic situation that the travel times are depending on the routes, we have to associate with each arc a vector of arc length. However, the derived results can be obtained for this case in the same way as for the case of constant travel times.

Path constrained problem. *The following data are given:*

(i) *s circles of length A denoted by $C_1, \ldots, C_s$. On each circle there exists a fixed point* 0.

(ii) *A complete graph $G_p = (V_p, E_p)$ with node set $V_p = \{1, \ldots, s\}$ and arc length l_{ij} for all arcs $(i,j) \in E_p$. The nodes of the graph G_p correspond to the circles.*

(iii) *r polygons denoted by* $P_1, \ldots, P_r$.

(iv) *For each polygon* P_i *a path* $p_i = (i_1, \ldots, i_{n_i})$ *in the graph* G_p; $i = 1, \ldots, r$.

A schedule for a path constrained problem will be called feasible *if polygon* P_i *is scheduled on the circles* $C_{i_1}, \ldots, C_{i_{n_i}}$ *and if the position of polygon* P_i *on the circle* $C_{i_{j+1}}$ *differs from the position of* P_i *on* C_{i_j} *by a clockwise movement of* $l_{i_j i_{j+1}}$ *units* $(j = 1, \ldots, n_i - 1;\ i = 1, \ldots, r)$. *The last condition implies that if on the circle* C_{i_j} *the distance between* 0 *and a vertex* v *of polygon* P_i *is* t_i *then on the cricle* $C_{i_{j+1}}$ *the corresponding distance is* $(t_i + l_{i_j i_{j+1}}) \bmod A$.

The aim is to find a feasible schedule of the polygons, such that an objective function depending on the distances between adjacent vertices on the circles $C_1, \ldots, C_s$ *is optimized.*

If we want to get an exact mathematical description of these problems we must find a unique mathematical description of a schedule of the polygons and we must specify the objective function. Let 0 be a fixed point on the circles called *origin*. Then a schedule can be described by a vector $\boldsymbol{t} \in \mathbb{R}^r$ of distances t_i (taken clockwise) between 0 and a fixed vertex of polygon P_i for the general problem and between 0 and a fixed vertex of polygon P_i on circle C_{i_1} for the path constrained problem $(i = 1, \ldots, r)$. We may assume that

$$t \in [\mathbb{R}/A]^r.$$

In order to define the criterion function we denote for a given schedule t by $U(t)$ the multiset of all distances between all pairs of neighboring vertices on all circles. If vertices coincide, the distance between them is zero. Furthermore, a weight $a(u) \in \mathbb{R}_0^+$ is associated with each distance $u \in U(t)$. The value of the criterion function for a schedule t is

$$f(t) := \sum_{\substack{u \in U(t) \\ a(u) \neq 0}} (a(u)u)^p, \tag{2.1}$$

where p is a fixed value with $-\infty \leqslant p < 0$ or $1 \leqslant p \leqslant \infty$. We set $f(t) = \infty$, if $-\infty < p < 0$ and there exist some zero distance u in $U(t)$.

The two problems now can be formulated as follows:

$$\min_{t \in [\mathbb{R}/A]^r} f(t). \tag{2.2}$$

For $p = -\infty$ this problem reduces to

$$\max_{t \in [\mathbb{R}/A]^r} \min_{\substack{u \in U(t) \\ a(u) \neq 0}} a(u)u$$

and for $p = \infty$ to

$$\min_{t \in [\mathbb{R}/A]^r} \max_{u \in U(t)} a(u)u.$$

In general (2.2) is a nonlinear nonconvex optimization problem which has many local optima. Therefore no standard solution methods can be used to solve this problem. Furthermore, this problem is NP-hard for every value of p even for the general problem with all weights equal (see [1]).

3. The combinatorial structure of the polygon scheduling problem

Brucker et al. [1] presented a decomposition approach for the unweighted version of the general problem. They decompose the set $[\mathbb{R}/A]^r$ into a finite number of *local regions* such that for each of these local regions $\mathscr{L}$ the problem

$$\min_{t \in \mathscr{L}} f(t) \tag{3.1}$$

is a convex optimization problem with linear constraints. The idea behind this decomposition is, that the hard problem (2.2) is replaced by a finite number of easier solvable optimization problems. A solution of (2.2) may be found by solving each of these problems (3.1) and comparing the objective function values. However, Brucker et al. do not give a one-to-one characterization of the local regions.

Since the structure of the solution space does not depend on the weights for the distances, the ideas of Brucker et al. [1] can also be used for the weighted case (2.2). We will first refine their approach such that we get an exact description of the local regions for the general problem. Afterwards, we will shortly mention how this approach may be applied to the path constrained problem.

3.1. The decomposition of the general problem

The decomposition idea is very simple. We define a local region as a set of schedules with the same sequence of vertices on the circle. However, if we want to use this decomposition idea for a solution method, we first have to give a description of the set of all local regions, which can be used to enumerate this set.

Let t_0 be a schedule such that no two vertices coincide. According to Guldan [8] schedules with this property will be called *free*. For free schedules the cyclic sequence of vertices is uniquely defined. Another schedule t (which may be nonfree) has the same cyclic sequence of vertices as t_0, if by a proper choice of the ordering of coinciding vertices the vertices of t can be arranged in the same sequence as the vertices of t_0. Therefore, we can associate with t_0 the local region

$$\mathscr{L}(t_0) := \{t \in [\mathbb{R}/A]^r \mid t \text{ has the same cyclic sequence of vertices as } t_0\}.$$

A nonfree schedule t can be mapped into a free schedule t' by the following procedure. Consider a pair of coinciding vertices of t. Move one of the polygons which belong to this pair of coinciding vertices clockwise by an amount $\delta > 0$. If δ is chosen sufficiently small the order of the vertices will not change. This procedure is repeated until all coincidences are eliminated. The result is a free schedule t'.

Obviously, the ordering of the coinciding vertices of t can be chosen in such a way that t has the same cyclic sequence of vertices as t'. Therefore, the system $\mathscr{L} = \{\mathscr{L}(t) \mid t$ is a free schedule$\}$ is a decomposition of $[\mathbb{R}/A]^r$. Unfortunately, the number of free schedules is not finite and it is not possible to enumerate all local regions by enumerating all free schedules.

In order to enumerate the local regions from $\mathscr{L}$, we will identify them by labeled intrees T, with node set $\{1, 2, \dots, r\}$ and root r. The nodes of the trees T correspond to polygons and an arc (i, j) of a tree T will be labeled with a pair $[v, w]$, where v is a vertex of polygon P_i and w a vertex of polygon P_j.

We define that for each arc (i, j) with label $[v, w]$ in the tree T the vertices v and w coincide and that v is arranged before w in the cyclic order of vertices on the circle. This leads in a unique way to a nonfree schedule $t(T)$ with a unique order for the coinciding vertices which appear as arc labels. But it may happen that also vertices that do not appear as arc labels coincide, i.e. the order of the vertices in the schedule $t(T)$ is not uniquely defined. Therefore, the schedule $t(T)$ does not define a unique local region.

In order to get a uniquely defined local region, we have to define a unique order for all coinciding vertices. This can be done by extending the labeled intree T to an arc labeled graph $S(T)$ that contains for each pair of coinciding vertices a corresponding arc:

For a node i of T we denote by *depth*(i) the number of arcs on the unique path from i to the root r in the tree T. $S(T)$ is now constructed in the following way. We start with $S(T) := T$. Next we consider iteratively all pairs (i, j) with coinciding vertices v of P_i and w of P_j. If *depth*$(i) >$ *depth*(j) (*depth*$(i) <$ *depth*(j), respectively) we add an arc (i, j) with label $[v, w]$ (an arc (j, i) with label $[w, v]$, respectively) to $S(T)$. If *depth*$(i) =$ *depth*(j) we choose the lexicographic order, i.e. we add an arc (i, j) with label $[v, w]$ if $i < j$.

The procedure given above yields in a unique way an acyclic arc labeled graph, such that for all pairwise coinciding vertices there exists an arc between the nodes of the corresponding polygons. If we now define, that for coinciding vertices v of polygon P_i and w of polygon P_j, the vertex v is arranged before the vertex w, iff (i, j) is an arc in the graph $S(T)$, we get a unique sequence for the vertices of the schedule $t(T)$.

Therefore, we are able to define a local region $\mathscr{L}(T)$ for an arc labeled intree T by

$$\mathscr{L}(\boldsymbol{T}) := \{t \in [\mathbb{R}/A]^r \mid t \text{ has the same cyclic sequence of vertices as } t(T)\}.$$

This local region $\mathscr{L}(T)$ can be described by linear inequalities in the following way. Let $d_{ij}(T)$ be the minimal distance (taken clockwise) between vertices of P_i and the next vertices of P_j in the schedule $t(T)$. Then $\mathscr{L}(T)$ is defined by

$$\mathscr{L}(\boldsymbol{T}) := \{t(T) + x \mid d_{ij}(T) + x_j - x_i \geqslant 0 \text{ for all } i, j = 1, \dots, r,\ i \neq j\}.$$

From the above discussion we know that each labeled intree defines a local region. But if we want to use the labeled intrees for enumeration of all local regions, we have to show that for each free schedule t_0 one can find a corresponding intree $T(t_0)$, such that the schedule $t(T(t_0))$ has the same cyclic sequence of vertices as t_0.

Theorem 3.1. *For each free schedule t_0 there exists an arc labeled intree $T(t_0)$ with root r, such that the schedule t_0 belongs to $\mathscr{L}(T(t_0))$.*

Proof. Starting from the schedule t_0 we get iteratively an intree $T(t_0)$ as follows: In the first step we move polygon P_r in counterclockwise direction until for the first time vertices of P_r hit vertices of some other polygon, say P_{i_1}. For the pair (i_1, r) we choose coinciding vertices v of P_{i_1} and w of P_r and define an arc (i_1, r) with label $[v, w]$.

In the $(k+1)$th step we move the polygons $P_r, P_{i_1}, \ldots, P_{i_k}$ simultaneously in counterclockwise direction, until for the first time vertices of $P_r, P_{i_1}, \ldots, P_{i_k}$ hit some vertices of one of the remaining polygons, say $P_{i_{k+1}}$ (the distance by which the polygons are moved may be 0). Let $P_{j_1}, \ldots, P_{j_l}$ be the subset of the polygons $P_r, P_{i_1}, \ldots, P_{i_k}$, which have coinciding vertices with the polygon $P_{i_{k+1}}$. Furthermore, let j be the index in the set $j_1, \ldots, j_l$ with maximal depth in the tree, which has been constructed in the first k steps. Now we add an arc (i_{k+1}, j) with a label $[v, w]$, where v and w are coinciding vertices of the two polygons $P_{i_{k+1}}$ and P_j, to the existing arcs. The choice of the index j guarantees that for all coinciding vertices of the polygon $P_{i_{k+1}}$ and the polygons $P_{j_1}, \ldots, P_{j_l}$ the arcs $(i_{k+1}, j_1), \ldots, (i_{k+1}, j_l)$ are added to the graph $S(T)$.

After $r-1$ steps we have added $r-1$ arcs and therefore we get an intree. It is obvious, that $t_0 \in \mathscr{L}(T)$. □

The above construction shows that for each local region $\mathscr{L}(t_0)$ there exists an arc labeled intree T such that $\mathscr{L}(T) = \mathscr{L}(t_0)$. Therefore, the system $\{\mathscr{L}(T) \mid T$ is an arc labeled intree with root $r\}$ is a decomposition of $[\mathbb{R}/A]^r$ and it is possible to enumerate all local regions via constructing all possible arc labeled intrees with r vertices and a fixed root.

Furthermore, the number of arc labeled intrees is an upper bound on the number of local regions. This number is given by the following Theorem (see [1]).

Theorem 3.2. *The number T_r of different arc labeled intrees with r vertices and a fixed root is given by*

$$T_r = \left(\prod_{i=1}^{r} m_i\right)\left(\sum_{i=1}^{r} m_i\right)^{r-2}, \tag{3.2}$$

where m_i is the number of vertices of Polygon P_i.

However, this bound on the number of local regions is not sharp, because different arc labeled intrees may lead to the same graph $S(T)$, thus representing the same local region, as Fig. 1 shows. In both cases (1) and (2) the trees T_1 and T_2 lead to the same

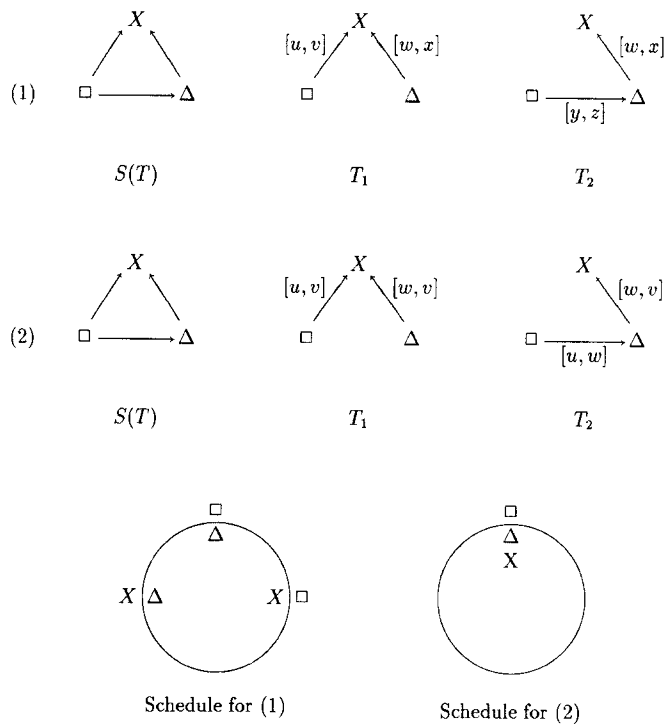

Fig. 1. Different trees define the same graph $S(T)$.

graph $S(T)$ and therefore define the same local region. In Case (1) a perturbation of the data will overcome this problem, because in the perturbed problem an additional coincidence of vertices, which is not expressed by the tree, may not occur. In Case (2) both trees T_1 and T_2 describe that the vertices u, v and w coincide at one point of the circle. Therefore, a perturbation of the data will not prevent the additional coincidence of vertices.

The crucial point in Case (2) is that more than two vertices coincide at one point on the circle. In such situations the graph $S(T)$ has to define a linear order for the coinciding vertices. The tree T_2 in Case (2) itself defines the linear order for the vertices u, v and w, whereas the tree T_1 does not define an order for the vertices u and v. But it is easy to see that for each choice of the order of u and v there exists a tree which defines the same linear order for the three vertices without using an arc of the graph $S(T)$.

This observation can be formulated more generally by:

Theorem 3.3. *If for a given tree T vertices of the polygons $P_1, \dots, P_k$ coincide in the schedule $t(T)$ at a point x on the circle we have:*

(1) *$S(T)$ contains a unique path*

$$(i_1, \dots, i_k); \tag{3.3}$$

where $i_1, \dots, i_k$ is a permutation of the indices $\{1, \dots, k\}$.

(2) *There exists a tree T' that contains no arc (i_j, i_l) with $j, l \in \{1, \ldots, k\}$ and $l \neq j+1$, such that*

$$S(T') = S(T).$$

Proof. (1) For all pairs (i,j); $i,j \in \{1, \ldots, k\}$, $i \neq j$ there are vertices of polygon P_i and polygon P_j which coincide at the point x. Therefore, $S(T)$ contains either the arc (i,j) or (j,i). Since the graph $S(T)$ is acyclic there exists a path of the mentioned type (3.3) in $S(T)$.

(2) Let *depth*(i) denote the maximum number of arcs on paths from node i to node r in $S(T)$; $i = 1, \ldots, r$. These values exist and are unique, since the graph $S(T)$ is acyclic and contains the tree T. For the values *depth*(i) we have the following inequalities:

$$depth(i) > depth(j) \quad \text{for all arcs } (i,j) \text{ in } S(T). \tag{3.4}$$

For each node $i \in \{1, \ldots, r-1\}$ there exists a node i' such that (i, i') is an arc in $S(T)$ and $depth(i) = depth(i') + 1$. Denote by T' the graph containing the arcs (i, i') with corresponding coinciding vertices of the schedule $t(T)$ as arc labels; $i = 1, \ldots, r-1$. Obviously, T' is an intree and $t(T') = t(T)$. The restrictions (3.4) imply that $S(T') = S(T)$.

It remains to show that T' does not contain arcs (i_j, i_l) with $j, l \in \{1, \ldots, k\}$ with $l \neq j+1$. Since $|depth(i_j) - depth(i_l)| > 1$ for all $j, l \in \{1, \ldots, k\}$ with $|j - l| > 1$ this condition is fulfilled. □

As a conclusion of Theorem 3.3 we can restrict the search for an optimal tree to trees T which have the following property.

Property free. For all arcs (i,k) with label $[u, w_1]$ and (j,k) with label $[v, w_2]$ in T the vertices w_1 and w_2 are different; i.e. $w_1 \neq w_2$.

Trees with this property will be called *free trees*. For the calculation of an upper bound on the number of different local regions it is of interest to calculate the number of different free trees.

Theorem 3.4. *The number FT_r of different free arc labeled intrees with r vertices and a fixed root r is given by*

$$FT_r = \left(\prod_{i=1}^{r} m_i\right) \prod_{i=1}^{r-2} \left(-i + \sum_{j=1}^{r} m_j\right). \tag{3.5}$$

Proof. First notice that a given tree leads in a unique way to an intree if we specify the root. Therefore, T_r is equal to the number of different arc labeled trees with r vertices. In order to calculate this number, we will look at the degrees of the nodes of a tree.

If we denote by d_i the degree of node i, each tree T has a unique degree sequence $(d_1, \dots, d_r)$. It is obvious, that

$$\sum_{i=1}^{r} d_i = 2(r-1) \text{ and } 1 \leqslant d_i \leqslant r-1 \quad \text{for } i = 1, \dots, r. \tag{3.6}$$

For a given sequence $(d_1, \dots, d_r)$ which fulfills (3.6) there exists a corresponding tree and the number $T(r, d_1, \dots, d_r)$ of different trees with r vertices and degree sequence $(d_1, \dots, d_r)$ is given by the multinomial coefficient:

$$T(r, d_1, \dots, d_r) = \binom{r-2}{d_1 - 1, \dots, d_r - 1}. \tag{3.7}$$

(see e.g. [12, Theorem 3.1]).

Furthermore, note that there are

$$f_k(d) := \begin{cases} m_k \cdot (m_k - 1) \cdots (m_k - (d-1)) & \text{if } \; k = r, \\ m_k \cdot m_k \cdot (m_k - 1) \cdots (m_k - (d-2)) & \text{otherwise} \end{cases} \tag{3.8}$$

different ways to choose the labels involving a vertex of polygon P_k, if d is the degree of node k. This follows from the fact that the labels of all incoming arcs of a node must be different to get a free intree. (Note that there exists no free intree with the degree sequence $(d_1, \dots, d_r)$ if $d_r > m_r$ or $d_k > m_k - 1$ for some $k \in \{1, \dots, r-1\}$.)

From (3.8) it follows that there are

$$\prod_{i=1}^{r} f_i(d_i) \tag{3.9}$$

different ways to label the arcs of a tree with degree sequence $(d_1, \dots, d_r)$ in order to get a free intree.

With (3.7) and (3.9) we have

$$\begin{aligned}
FT_r &= \sum_{\substack{(d_1, \dots d_r) \in N^r \\ \sum_{i=1}^{r} d_i = r-2}} \left(T(r, d_1 + 1, \dots, d_r + 1) \prod_{i=1}^{r} f_i(d_i + 1) \right) \\
&= \left(\prod_{i=1}^{r} m_i \right) \sum_{\substack{(d_1, \dots d_r) \in N^r \\ \sum_{i=1}^{r} d_i = r-2}} \left(\binom{r-2}{d_1, \dots, d_r} \prod_{i=1}^{r} \frac{f_i(d_i + 1)}{m_i} \right) \\
&= \left(\prod_{i=1}^{r} m_i \right) \sum_{\substack{(d_1, \dots d_r) \in N^r \\ \sum_{i=1}^{r} d_i = r-2}} \left(\binom{r-2}{d_1, \dots, d_r} \left(D^{d_r}(x^{m_r - 1})(1) \cdot \prod_{i=1}^{r-1} D^{d_i}(x^{m_i}(1) \right) \right) \\
&= \left(\prod_{i=1}^{r} m_i \right) D^{r-2}(x^{-1 + \sum_{i=1}^{r} m_i})(1) \\
&= \left(\prod_{i=1}^{r} m_i \right) \prod_{i=1}^{r-2} \left(-i + \sum_{j=1}^{r} m_j \right).
\end{aligned}$$

The fourth equality follows from the generalized Leibnitz formula

$$D^n(u_1 \dots u_s) = \sum_{\substack{(k_1, \dots k_s) \in N^s \\ \sum_{i=1}^{s} k_i = n}} \left(\binom{n}{k_1, \dots, k_s} \prod_{i=1}^{s} D^{k_i}(u_i) \right),$$

where u_i denotes a differential function, $i = 1, \dots, s$ and $D^k(u)$ denotes the kth derivative of a function u (see e.g. [13]). □

The number FT_r from Theorem 3.4 gives another upper bound on the number of different local regions. It will be shown that this bound is sharp if the polygon scheduling problem has the following property.

Property ND (*Nondegenerated*). For all possible free labeled intrees T we have: If vertices v of P_i and w of P_j coincide in the schedule $t(T)$, there exists a path between i and j in T, such that all vertices, which belong to labels of this path, coincide with v and w on the circle.

This property ND is not restrictive, because it is possible to perturb each instance of a polygon scheduling problem in such a way that it satisfies the property ND. This perturbation can be done by arbitrary small changes in the data. Therefore, it has no influence on the optimal schedule, because the objective function depends continuously on the perturbation. We have the following lemma.

Lemma 3.1. *For a problem satisfying property ND two different free labeled intrees T_1 and T_2 lead to different graphs $S(T_1)$ and $S(T_2)$.*

Proof. Assume a multigraph $S(T)$ contains two arcs (i, j_1) and (i, j_2) with $j_1 \neq j_2$. We will show that only one of these two arcs may belong to a free tree T' with $S(T') = S(T)$.

Due to property ND the free tree T' must contain a path from i to j_1 and a path from i to j_2. All the vertices which belong to labels of these paths coincide on the circle. Therefore, $S(T)$ contains either a path from j_1 to j_2 or a path from j_2 to j_1 such that all the vertices which belong to labels of the paths coincide on the circle. Due to property ND the free tree T' must also contain such a path. In the first case (i, j_2) and in the second case (i, j_1) cannot be an arc in the free tree T' with $S(T') = S(T)$.

The above discussion shows that for each node i there is a unique successor in any free intree T' with $S(T') = S(T)$. This completes the proof of the lemma. □

Using Lemma 3.1 we can prove the following lemma.

Lemma 3.2. *Let $T^1 \neq T^2$ be free labeled intrees for a problem satisfying property ND. Then $\mathcal{L}(T^1) \neq \mathcal{L}(T^2)$.*

Proof. Let us assume that the two different free labeled intrees T^1 and T^2 describe the same local region; i.e. $\mathscr{L}(T^1) = \mathscr{L}(T^2)$. Furthermore, we may assume that for the schedules $t^1 = t(T^1)$ and $t^2 = t(T^2)$ we have

$$t_r^1 = t_r^2. \tag{3.10}$$

From the assumption $\mathscr{L}(T^1) = \mathscr{L}(T^2)$ it follows that $t^1 \in \mathscr{L}(T^2)$ and $t^2 \in \mathscr{L}(T^1)$. Therefore there must exist vectors x^1 and x^2 with

$$t^1 = t^2 + x^2, \tag{3.11}$$

$$d_{ij}(T^2) + x_j^2 - x_i^2 \geqslant 0, \quad i,j = 1, \ldots, r;\ i \neq j, \tag{3.12}$$

$$t^2 = t^1 + x^1, \tag{3.13}$$

$$d_{ij}(T^1) + x_j^1 - x_i^1 \geqslant 0, \quad i,j = 1, \ldots, r;\ i \neq j. \tag{3.14}$$

From (3.11) and (3.13) it follows that

$$x^2 = -x^1. \tag{3.15}$$

On the other hand, we have for all arcs (i,j) in the tree T^k:

$$d_{ij}(T^k) = 0, \quad k = 1,2.$$

Therefore, we get with (3.12) and (3.14):

$$x_i^k \leqslant x_r^k, \quad i = 1, \ldots, r-1;\ k = 1,2. \tag{3.16}$$

Furthermore, it follows from (3.10) that $x_r^1 = x_r^2$. Together with (3.15) this yields $x_r^1 = x_r^2 = 0$. Now (3.16) implies that

$$x^1 = x^2 = 0.$$

Therefore,

$$t^1 = t^2 \quad \text{and} \quad S(T^1) = S(T^2).$$

This is a contradiction to Lemma 3.1. □

Since by Lemma 3.2 each free labeled intree defines a different local region for problems satisfying property ND, the bound of Theorem 3.4 is sharp.

Theorem 3.5. *For a polygon scheduling problem satisfying property ND the number of different local regions is given by the number*

$$\left(\prod_{i=1}^{r} m_i\right)\prod_{i=1}^{r-2}\left(-i + \sum_{j=1}^{r} m_j\right)$$

of different free arc labeled intrees.

3.2. The decomposition of the path constrained problem

If we want to apply the decomposition idea to the path constrained problem, we first have to adapt the definition of a local region. Here a local region will be a set of schedules which have the same cyclic sequence of vertices on all circles. To identify such a local region, it is again possible to use labeled rooted intrees with node set $\{1, \dots, r\}$ and root r. But for the path constrained problem we have to extend the labels of the arcs. It is not sufficient to label an arc (i,j) merely with a tuple $[v, w]$ of vertices, because it is not clear on which circle these vertices have to coincide. Therefore we will label the arcs with triples $[v, w, k]$, where k denotes the index of a circle and v and w again denote vertices of the polygons which correspond to this arc. An arc (i,j) with label $[v, w, k]$ describes that the vertex v of P_i coincide with the vertex w of P_j on the circle C_k and that we arrange v before w in the cyclic order of vertices on the circle C_k.

It is easy to see that not all labelings for the arcs of a tree lead to a possible schedule. If an arc (i,j) is labeled with a triple $[v, w, k]$ and k is a circle not contained in both paths p_i and p_j, then one of the polygons P_i or P_j will not be scheduled on the circle C_k and therefore it is not possible that the vertices v of P_i and w of P_j coincide on the circle C_k.

Due to this observation we treat a triple $[v, w, k]$ only as a label for an arc (i,j) if v is a vertex of P_i, w is a vertex of P_j and $k \in \{p_i \cap p_j\}$. Here $\{p_i \cap p_j\}$ denotes the set of nodes which appear in both paths p_i and p_j. An arc labeled intree with such labels will be called a *proper* labeled intree.

For a proper labeled intree T we again can define as for the general problem a corresponding schedule $t(T)$, a graph $S(T)$ which defines the order of coinciding vertices in $t(T)$, and therefore a local region $\mathscr{L}(T)$ (see Section 3.1). Also it is obvious that the result from Theorem 3.1 holds for the path constrained problem, i.e. the system $\{\mathscr{L}(T) \mid T \text{ is a proper labeled intree with root } r\}$ is a decomposition of $[\mathbb{R}/A]^r$ and it is possible to enumerate all local regions via constructing all possible proper labeled intrees with r vertices and a fixed root. If we generalize the definitions of the properties Free and ND to proper labeled trees, we can show that the number of different local regions for the path constrained problem is given by the number of free proper labeled intrees. However, for this number we will not get a compact description as in Theorem 3.5, since this number depends on the structure of the graph G_p.

As a conclusion of Theorems 3.4 and 3.5 it is possible to formulate the considered problem (2.2) in the following way:

$$\min_{T \in M_{FT}} f(T), \tag{3.17}$$

where M_{FT} denotes the set of all possible free (proper) arc labeled intrees with r vertices and a fixed root and where $f(T)$ is defined by

$$f(T) = \min_{t \in \mathscr{L}(T)} f(t), \tag{3.18}$$

Therefore it is possible to solve the global optimization problem (2.2) by solving the corresponding local optimization problem (3.18) for each free labeled intree T and compare all solutions to find the best. Methods for enumerating trees can be found for instance in Christofides [7].

4. Local optimization

In the last section we have reduced the general problem (2.2) to a finite number of local optimization problems (3.18). It remains to show how the optimal value $f(T)$ of this local optimization problem can be calculated for a given tree T. For the unweighted case of the general problem Brucker et al. [1] considered these problems for $p = -\infty$ and $p = \infty$. They show that the local optimization problems can be solved by network optimization techniques. In this section we will study these problems for the path constrained problem in connection with some special class of weights $a(u)$ for the distances u. For $p = -\infty$ and $p = \infty$ this will lead to a slight generalization of the methods presented by Brucker et al. [1]. For $p = 1$ (this case does not make sense for the unweighted criterion function) we will show that the local optimization problem can also be solved by a network flow problem.

First we will describe the problem (3.18) for the general criterion function (2.1). Therefore let $U_{ij}(T)$ be the multiset of all distances on all circles (taken clockwise) between vertices v of P_i and w of P_j which are neighbors in the cyclic sequence of the schedule $t(T)$ $(i, j = 1, \dots, r)$. The value $d_{ij}(T)$ defines the minimum distance in $U_{ij}(T)$. If $U_{ij}(T)$ is empty we define $d_{ij}(T) = \infty$. The local optimization problem $\boldsymbol{P(T)}$ for a tree T now has the following form:

$$\text{minimize} \quad \sum_{i,j=1}^{r} \sum_{\substack{u \in U_{ij}(T) \\ a(u) \neq 0}} (a(u)(u + x_j - x_i))^p \tag{4.1}$$

$$\text{subject to} \quad d_{ij}(T) + x_j - x_i \geqslant 0, \quad i,j = 1, \dots, r; \quad i \neq j.$$

For all $u \in U_{ij}(T)$ we have $u \geqslant d_{ij}(T)$ and thus $u + x_j - x_i \geqslant 0$. Therefore for each p with $-\infty \leqslant p \leqslant 0$ or $1 \leqslant p \leqslant \infty$ the problem $P(T)$ is a well-defined convex optimization problem with linear constraints which can be solved – in principle – by well-known methods.

For further considerations we will make use of more specific weights $a(u)$ for the distances u on the circle. The weights of the distances are classified via the polygons which lead to the distances and via the circles on which they appear. This leads to the definition of sets $U_{ij}^k(T)$, which contain the distances (taken clockwise) between all neighboring vertices of the polygons P_i and P_j on the circle C_k for the schedule $t(T)$ $(i, j = 1, \dots, r;\ k \in \{p_i \cap p_j\})$. The connection between the sets $U_{ij}(T)$ and $U_{ij}^k(T)$ is given by

$$\bigcup_{k \in \{p_i \cap p_j\}} U_{ij}^k(T) = U_{ij}(T), \quad i,j = 1, \dots, r.$$

If we introduce weights $a(u) = a_{ij}^k$ for all distances $u \in U_{ij}^k(T)$, we get the following local optimization problem:

$$\text{minimize} \quad \sum_{i,j=1}^{r} \sum_{\substack{k \in \{p_i \cap p_j\} \\ a_{ij}^k \neq 0}} \sum_{u \in U_{ij}^k(T)} (a_{ij}^k(u + x_j - x_i))^p \tag{4.2}$$

$$\text{subject to} \quad d_{ij}(T) + x_j - x_i \geqslant 0, \quad i,j = 1, \dots, r; \quad i \neq j.$$

In the following Sections 4.1 and 4.2 we will study the problem (4.2) for the values $p = -\infty$, $p = \infty$ and $p = 1$.

4.1. *The max min ($p = -\infty$) and the min max ($p = \infty$) problem*

For $p = -\infty$ and $p = \infty$ it is possible to generalize the approach of Brucker et al. [1] for the unweighted version of the general problem to solve problem (4.2). In this approach the local optimization problem is formulated as linear program and it is shown that the corresponding dual problem can be reduced to calculate a cycle of minimum average length for the case $p = -\infty$, and a cycle of maximum average length for the case $p = \infty$, respectively, in a complete loop free network with r vertices and arc length $d_{ij}(T)$; $i,j = 1, \dots, r$; $i \neq j$.

In the following we will describe in short the necessary modifications of the approach of Brucker et al. that enable us to solve problem (4.2) for $p = -\infty$ and $p = \infty$. The used techniques and proofs are similar to that of Brucker et al. and can be found in Hurink [10]. First we will look at the problem of maximizing the minimum weighted distance, i.e. we treat the case $p = -\infty$. Here (4.2) can be formulated by the linear program

$$\text{maximize} \quad z$$

$$\text{subject to} \quad \bar{a}_{ij}^k z - x_j + x_i \leqslant d_{ij}^k(T), \quad i,j = 1, \dots, r; \ i \neq j; \ k \in \{p_i \cap p_j\},$$

where $\bar{a}_{ij}^k$ is defined by

$$\bar{a}_{ij}^k := \begin{cases} 1/a_{ij}^k & \text{if } a_{ij}^k > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The corresponding dual problem can be reduced to finding a cycle of minimum cost to time ratio in a network $N(T) = (V, E, d(T))$ with $V = \{1, \dots, r\}$, $E = \{(i,j)^k \mid i,j = 1, \dots, r; \ i \neq j; \ k \in \{p_i \cap p_j\}\}$ and $d(T): E \to \mathbb{R}_0^+$ defined by $(i,j)^k \mapsto d_{ij}^k(T)$ (note that this network $N(T)$ contains parallel arcs). The *cost to time ratio* of a cycle $C: (i_1, i_2), (i_2, i_3), \dots, (i_{s-1}, i_s), (i_s, i_1)$ is defined by

$$\left(\sum_{(i,j) \in C} a'_{ij} \right)^{-1} \sum_{(i,j) \in C} d_{ij}(T)$$

and a cycle with minimum cost to time ratio can be found in polynomial time (see [11]).

The change of the network due to the approach of Brucker et al. is a consequence of the generalization to the path constrained problem, whereas the introduction of the weights in the criterion function leads to cycles with minimum cost to time ratio instead of cycles with minimum average length.

For the problem of minimizing the maximum distance ($p = \infty$) we again get a linear problem. Here (4.2) reduces to

$$\begin{aligned} &\text{minimize} \quad w \\ &\text{subject to} \quad \bar{a}_{ij}^k w - x_j + x_i \geqslant h_{ij}^k(T), \quad i,j = 1, \ldots, r; \; k \in \{p_i \cap p_j\}, \\ &\qquad\qquad - x_j + x_i \geqslant - d_{ij}(T), \quad i,j = 1, \ldots, r; \; i \neq j, \end{aligned}$$

where $h_{ij}^k(T)$ is defined as the maximum distance in the set $U_{ij}^k(T)$; $i,j = 1, \ldots, r$; $k \in \{p_i \cap p_j\}$ and $\bar{a}_{ij}^k$ is defined by

$$\bar{a}_{ij}^k := \begin{cases} 1/a_{ij}^k & \text{if } a_{ij}^k > 0, \\ \infty & \text{otherwise.} \end{cases}$$

The dual of this problem reduces to finding a cycle of maximum cost to time ratio in a network $N(T) = (V, E, h(T))$ with $V = \{1, \ldots, r\}$, $E = \{(i,j)^k \mid i,j = 1, \ldots, r; k \in \{p_i \cap p_j\}\}$ and $h(T): E \to \mathbb{R}_0^+$ defined by $(i,j)^k \mapsto h_{ij}^k(T)$.

4.2. *The sum of weighted distances problem* ($p = 1$)

If in (4.2) we take $p = 1$ we get the problem of minimizing the sum of weighted distances. For this optimization criteria the special case, that all weights are equal, is not of interest, since in this case the value of the objective function is equal for all schedules in $[\mathbb{R}/A]^r$. Therefore, we assume that not all weights are equal. The local problem $P(T)$ then can be formulated by

$$\begin{aligned} &\text{minimize} \quad \sum_{i,j=1}^{r} a_{ij}^k m_{ij}^k(T)(x_j - x_i) + \sum_{i,j=1}^{r} \sum_{k \in \{p_i \cap p_j\}} \left(a_{ij}^k \sum_{u \in U_{ij}^k(T)} u \right) \\ &\text{subject to} \quad d_{ij}(T) + x_j - x_i \geqslant 0, \quad i,j = 1, \ldots, r; \; i \neq j, \end{aligned} \tag{4.3}$$

where $m_{ij}^k(T)$ denotes the cardinality of the set $U_{ij}^k(T)$.

Since $\sum_{i,j=1}^{r} \sum_{k \in \{p_i \cap p_j\}} (a_{ij}^k \sum_{u \in U_{ij}^k(T)} u)$ is a constant for a given tree T, an equivalent formulation of (4.3) is given by

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{r} c_i x_i \\ &\text{subject to} \quad - x_j + x_i \leqslant d_{ij}(T), \quad i,j = 1, \ldots, r; \; i \neq j, \end{aligned} \tag{4.4}$$

where c_i is defined by

$$c_i := \sum_{j=1}^{r} \sum_{k \in \{p_i \cap p_j\}} (a^k_{ji} m^k_{ji}(T) - a^k_{ij} m^k_{ij}(T)), \quad i = 1, \dots, r.$$

The dual of (4.4) is

$$\begin{aligned} &\text{minimize} \quad \sum_{\substack{i,j=1 \\ i \neq j}}^{r} d_{ij}(T) y_{ij} \\ &\text{subject to} \quad \sum_{j=1}^{r} y_{ij} - \sum_{j=1}^{r} y_{ji} = -c_i, \quad i = 1, \dots, r, \\ &y_{ji} \geqslant 0, \quad i, j = 1, \dots, r; \ i \neq j. \end{aligned} \tag{4.5}$$

Problem (4.5) is a transshipment problem in a complete graph with r vertices. Because

$$\sum_{i=1}^{r} c_i = \sum_{i=1}^{r} \sum_{j=1}^{r} (a_{ji} m_{ji}(T) - a_{ij} m_{ij}(T)) = 0,$$

it has a solution and can be solved by well-known methods in polynomial time. Therefore, we can find a tree T, such that the corresponding local region $\mathscr{L}(T)$ contains an optimal solution of the global problem (2.2) by solving the dual problems (4.5) for all free proper labeled intrees.

5. Conclusion

We have considered the problem of scheduling irregular polygons with vertices on a circle under different objectives. The optimization criteria were depending on the distances between adjacent vertices on the circle. Furthermore, we have considered a more general problem where polygons on different circles have to be scheduled simultaneously. Between the polygons on the different circles there are connections which can be expressed by paths in a graph. This generalization of the polygon scheduling problem has an application in the area of railway scheduling.

These polygon scheduling problems are continuous optimization problems, which are NP-hard in the strong sense. We have reformulated them as discrete optimization problems $\min_{s \in \mathscr{S}} c(s)$ where each solution $s \in \mathscr{S}$ represents a continuous set of schedules. The objective function value $c(s)$ of a solution $s \in \mathscr{S}$ is the optimal value of a convex optimization problem with linear constrains.

For the objectives maximizing the minimum weighted distance, minimizing the maximum weighted distance, and minimizing the sum of weighted distances the problems of calculating the value $c(s)$ reduce to linear programming problems. The corresponding dual problems are special network flow problems which can be solved by well-known methods.

Since the cardinality of the set $\mathscr{S}$ is exponential, it is only practicable for small problems to use these results for an enumerative solution method. Based on the discrete formulation of the polygon scheduling problem Hurink [10] developed two neighborhood structures. They can be used to adapt local search heuristics to the polygon scheduling problem. Computational tests show that local search methods in connection with one of the two neighborhood structures lead to good results.

For further research in the area of the polygon scheduling problem, the discrete optimization problem formulation could be useful. Since each solution $s \in \mathscr{S}$ can be represented by a sequence of vertices on the circle, the polygon scheduling problem can be considered as a sequencing problem with complex subproblems (calculating $c(s)$). Many other practical problems (shop problem, batching and lot-sizing problems, etc.) can also be formulated as sequencing problems with complex subproblems. Therefore it is an interesting question if it is possible to transfer known solution methods for these problems to the polygon scheduling problem.

References

[1] P. Brucker, R. E. Burkard and J. Hurink, Cyclic schedules for r irregularly occurring events, J. Comput. Appl. Math. 30 (1990) 173–189.

[2] P. Brucker and J. Hurink, A railway scheduling problem, Zeitschrift Oper. Res. 30 (1986) 223–227.

[3] P. Brucker and W. Meyer, Scheduling two irregular polygons, Discrete Appl. Math. 20 (1988) 91–100.

[4] R. E. Burkard, Optimal schedules for periodically recurring events, Discrete Appl. Math. 15 (1986) 167–180.

[5] J. Cerny, Problems for systems of regular polygons on a circumference and their application in transport (Czech), Mate. obzory (1972) 51–59.

[6] J. Cerny and F. Guldan, Location of polygon vertices on circles and its application in transport studies, Aplikace Matematiky 32 (1987) 81–95.

[7] N. Christofides, Graph Theory, An Algorithmic Approach (Academic Press, London, New York, San Franscisco, 1975).

[8] F. Guldan, Maximization of distances of regular polygons on a circle, Aplikace Matematiky 25 (1980) 182–195.

[9] J. Hurink, Scheduling von periodischen Vorgängen, Diplomarbeit, Fachbereich Mathematik/Informatik, Universität Osnabrück, 1986.

[10] J. Hurink, Polygon Scheduling, Ph.D. Thesis, Fachbereich Mathematik/Informatik, Universität Osnabrück, 1992.

[11] N. Meggido, Combinatorial optimization with rational objective functions, Math. Oper. Res. 3 (1979) 313–323.

[12] J. W. Moon, Counting labelled trees, Canadian Mathematical Monographs, No. 1, Canadian Mathematical Congress, Montreal, 1970.

[13] A. Ostrowski, Vorlesungen über Differential- und Integralrechnung II (Birkhäuser, Basel), 1951.

[14] A. Vince, Scheduling periodic events, Discrete Appl. Math. 25 (1988) 299–310.