Journal of Computational and Applied Mathematics 30 (1990) 173–189
North-Holland

Cyclic schedules for r irregularly occurring events

Peter BRUCKER *
Fachbereich Mathematik, Universität Osnabrück, Albrechtstrasse 28, D-4500 Osnabrück, FRG

Rainer E. BURKARD **
Institut für Mathematik, Technische Universität Graz, Kopernikusgasse 24, A-8010 Graz, Austria

Johann HURINK *
Fachbereich Mathematik, Universität Osnabrück, Albrechtstrasse 28, D-4500 Osnabrück, FRG

Received 8 March 1989
Revised 22 November 1989

Abstract: Consider r irregular polygons with vertices on some circle. How should the polygons be arranged to minimize some criterion function depending on the distances between adjacent vertices? A solution of this problem is given. It is based on a decomposition of the set of all schedules into local regions in which the optimization problem is convex. For the criterion functions minimize the maximum distance and maximize the minimum distance the local optimization problems are related to network flow problems which can be solved efficiently. If the sum of squared distances is to be minimized a locally optimal solution can be found by solving a system of linear equations. For fixed r the global problem is polynomially solvable for all the above-mentioned objective functions. In the general case, however, the global problem is NP-hard.

Keywords: Cyclic scheduling.

1. Introduction

In connection with railway scheduling problems Guldan [6] considered the problem to place r regular polygons on a circle such that the minimum distance between two neighbouring vertices becomes as large as possible. This corresponds to the situation that trains which arrive in constant intervals at some railway station are scheduled such that the safety interval between two trains is as large as possible.

In this paper we drop the assumption of regularly arriving trains. We consider trains arriving according to some fixed periodic pattern and optimize different objective functions: maximizing

* The work of these authors was partially supported by the Deutsche Forschungsgemeinschaft, Project "Optimierung diskreter Systeme und Computergraphik".
** The work of this author was partially supported by the Austrian Science Foundation (Fonds zur Förderung der wissenschaftlichen Forschung), Project S32/01.

the safety intervals, minimizing the longest as well as the average waiting time. All these objective functions are special cases of one criterion function $f(u_1, \dots, u_m)$ depending on the time intervals u_i, between two arrivals of the form

$$f(u_1, \dots, u_m) = \sum_{k=1}^{m} u_k^p, \tag{1}$$

where p is an arbitrary but fixed value with $-\infty \leqslant p \leqslant 0$ or $1 \leqslant p \leqslant \infty$.

A geometric model for this situation can be set up as follows: we illustrate the periodically repeating time interval (dayly or weekly schedule for trains) by a circle. The arrivals of trains correspond to vertices of irregular polygons, where every polygon corresponds to a railway line. Let us assume that polygon P_i has m_i vertices, i.e., the corresponding line carries m_i trains in the time period. If we fix a point—called origin—on the circle and if one vertex of every polygon is specified we can uniquely describe a schedule by the vector $t = (t_1, \dots, t_r)$ where t_i is the distance between origin and specified vertex of polygon P_i. We have to find a value for the vector t (i.e., we have to move the polygons relative to each other) such that $f(u_1, \dots, u_m)$ becomes optimal.

For solving the corresponding regular problem with $p = -\infty$, Guldan decomposes the set of all parameters t into local regions and characterizes every local region by an acyclic graph. Moreover, he gives a method for solving the local problems by longest paths computations.

In [2] it was shown for the case of two regular polygons ($r = 2$) that for arbitrary p the function (1) attains its minimum for $t = (0, t^*)$ where

$$t^* = [2 \operatorname{lcm}(m_1, m_2)]^{-1}.$$

In the cases $p = \infty$ and $p = -\infty$ we get problems of the form

$$\textit{minimizing} \max_{1 \leqslant k \leqslant m} u_k \tag{2}$$

and

$$\textit{maximizing} \min_{1 \leqslant k \leqslant m} u_k. \tag{3}$$

Problem (2) minimizes the longest waiting time, whereas (3) maximizes the safety interval. In [1] efficient algorithms have been developed for (2) and (3) in the case of two irregular polygons.

In this paper the case of r irregular polygons with general objective function (1) will be discussed. In Section 2 the parameter space which describes all schedules is decomposed into local regions in which the optimization problem is convex. A general optimization procedure based on this decomposition will also be described.

In Sections 3 and 4 we consider in more detail the minmax-problem (2), the maxmin-problem (3) and the cases in which p is a nonnegative even integer. Some special problems of practical interest which can be solved efficiently are discussed in Section 5. In Section 6 we show that for fixed r the global problem for $p = -\infty$, 2 and $+\infty$ are polynomially solvable whereas the general problem is NP-hard. Topics for further research are mentioned in Section 7.

2. Decomposition of the problem

Assume that we have r polygons denoted by $P_1, \dots, P_r$ which are to be scheduled on a circle of length A. Let 0 be a fixed point on the circle called origin. Then a schedule is defined by the vector $t \in \mathbb{R}^r$ of distances t_i (taken clockwise) between 0 and a fixed vertex of polygon P_i $(i = 1, \dots, r)$. We may assume that

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

For a schedule t let $U(t)$ be the multiset of all distances between all pairs of neighbouring vertices on the circle. If vertices coincide, the distance between the vertices is zero. Then the general problem may be formulated as follows.

$$\min_{t \in [\mathbb{R}/A]^r} f(t) \quad \text{with } f(t) = \sum_{u \in U(t)} u^p, \tag{4}$$

where p is a fixed value with $-\infty \leqslant p < 0$ or $1 < p \leqslant \infty$ (we omitted the cases $p = 0$ and $p = 1$ which are trivial). We set $f(t) = \infty$ if $-\infty < p < 0$ and there exists some zero distance in $U(t)$.

In general (4) is a nonlinear nonconvex optimization problem which has many local optima. To solve (4) we will decompose $[\mathbb{R}/A]^r$ into a finite number of sets called local regions such that for each of these sets $\mathscr{L}$ the problem

$$\min_{t \in \mathscr{L}} \sum_{u \in U(t)} u^p \tag{5}$$

is a convex optimization problem with linear constraints. We may find the solution of (4) by solving each of these problems (5) and comparing the objective function values. In this section we will discuss how to find a decomposition of $[\mathbb{R}/A]^r$ into the sets $\mathscr{L}$.

A local region is a set of schedules yielding the same sequence of vertices on the circle. Let t_0 be a schedule such that no two vertices coincide. According to Guldan [6] schedules with this property will be called free. In this case the cyclic sequence of all m vertices is uniquely defined, say

$$(v_1, v_2, v_3, \dots, v_m). \tag{6}$$

A schedule t (which may be nonfree) has the same sequence of vertices as t_0 if the vertices can again be arranged as in (6) by a proper choice for the sequence of coinciding vertices. For example, let t be a schedule, where v_2 and v_3 coincide, but all other vertices are as in sequence (6). Then t leads to the same sequence as t_0, but also to the same sequence as another free schedule, namely $(v_1, v_3, v_2, \dots, v_m)$. Thus nonfree schedules belong to different local regions.

The remarks above enable us to define the local region of t_0 by

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

By the preceding remarks, $\mathscr{L}(t_0)$ is a closed convex set in $[\mathbb{R}/A]^r$. The interiors of two different local regions are disjoint. Thus the system $\{\mathscr{L}(t) \mid t \text{ is a free schedule}\}$ yields a decomposition of $[\mathbb{R}/A]^r$.

The local region $\mathscr{L}(t)$ associated with a free schedule t can be characterized by linear inequalities in the following way: let d_{ij} be the minimal distance (taken clockwise) between vertices of P_i and the next vertices of P_j in the schedule t. Then $\mathscr{L}(t)$ is the set of all schedules $t + x \in [\mathbb{R}/A]^r$ satisfying

$$d_{ij} + x_j - x_i \geqslant 0 \quad \text{for all } i,\ j = 1, \dots, r,\ i \neq j. \tag{7}$$

Now we shall turn to the question how we can generate all local regions. It will turn out that arc-labelled intrees will play a crucial role.

Let T be an arc-labelled intree with node set $\{1, 2, \ldots, r\}$ and root r. The nodes of T correspond to polygons, an arc (i, j) with label $[v, w]$ says that vertex v of polygon P_i coincides with vertex w of polygon P_j and that we arrange v *before* w in the cyclic order of vertices on the circle.

Starting from a free schedule t_0 we get an intree $T(t_0)$ as follows: we move polygon P_r counterclockwise until for the first time vertices of P_r hit vertices of other polygons, say $P_{i_1}, \ldots, P_{i_k}$. For every pair (i_κ, r), $\kappa = 1, 2, \ldots, k$, we choose coinciding vertices v of P_{i_κ} and w of P_r and define an arc (i_κ, r) with label $[v, w]$. Then we move the polygons $P_r, P_{i_1}, \ldots, P_{i_k}$ simultaneously counterclockwise, until for the first time again vertices of $P_r, P_{i_1}, \ldots, P_{i_k}$ hit some vertices of the remaining polygons. For an isolated node i, $i \notin \{r, i_1, \ldots, i_k\}$ we add an arc (i, j) with label $[v, w]$ if the vertex w of polygon P_j, $j \in \{r, i_1, \ldots, i_k\}$ meets the vertex v of polygon P_i. We continue in this way, until we get $r-1$ arcs and therefore an intree. For a node i of T we denote by depth(i) the number of arcs on the unique path from i to r.

It might happen, that not only the vertices which appear as arc labels in the intree coincide, but that there are further coincidences. In order to get a uniquely defined sequence of vertices on the circle we have to state which of the coinciding vertices comes before the other. This can be done by enlarging the labelled intree T to an arc-labelled multigraph $S(T)$.

$S(T)$ is iteratively constructed in the following way. At first, let $S(T)$ be T. Now we choose the lexicographic smallest pair (i, j) for which there are coinciding vertices v of P_i and w and P_j such that neither $[v, w]$ nor $[w, v]$ appears as an arc label in $S(T)$. We add to $S(T)$ an arc (i, j) with label $[v, w]$, if either depth$(i) >$ depth(j) or if depth$(i) =$ depth(j) and there is no path in $S(T)$ from j to i.

This procedure yields in a unique way an acyclic arc-labelled multigraph with as many arcs as there are pairwise coinciding vertices on the circle. Thus a unique sequence for these vertices is defined which describes the local region of the free schedule t_0.

Notice that different arc-labelled intrees may lead to the same multigraph S, thus representing the same local region. See, e.g., Fig. 1.

In both cases (1) and (2) the trees T_1 and T_2 lead to the 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) a perturbation of the data will not prevent the additional coincidence, because the tree itself defines the coincidence of the three vertices u, v and w.

The preceding remarks enable us to generate all local regions via generating arc-labelled trees. For methods generating trees, see, e.g., [3, pp.125–135].

An upper bound on the number of local regions is given by the number of labelled trees which is equal to

$$\left(\prod_{i=1}^{r} m_i\right)\left(\sum_{i=1}^{r} m_i\right)^{r-2}, \tag{8}$$

where m_i is the number of vertices of polygon P_i. This follows from [8, Theorem 6.1].

This bound on the number of local regions is not sharp, because there may exist different trees which lead to the same graph $S(T)$, as the examples in Fig. 1 show. In particular for all problems with $r \geqslant 3$ the number of local regions is smaller than the upper bound (8).

In connection with a labelled tree T we consider a local optimization problem $P(T)$. Let

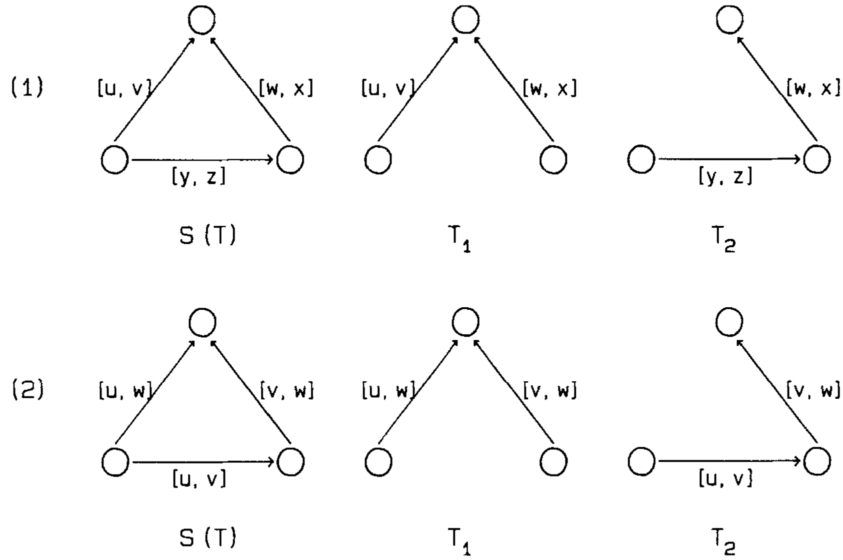

Fig. 1.

$U_{ij}(T)$ denote the multiset of all distances (taken clockwise) between vertices v of P_i and w of P_j which are neighbours in the cyclic sequence given by the schedule described by the multigraph $S(T)$. Define $d_{ij}(T) = \min U_{ij}(T)$ and notice, that for all arcs (i, j) in the graph $S(T)$ the corresponding distances $d_{ij}(T)$ are zero. Now the local optimization problem $P(T)$ has the following form:

$$\min \sum_{i,j=1}^{r} \sum_{u \in U_{ij}(T)} (u + x_j - x_i)^p, \tag{9}$$

$$\text{subject to} \quad d_{ij}(T) + x_j - x_i \geqslant 0, \quad i,\ j = 1, \dots, r,\ i \neq j. \tag{10}$$

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

To solve the global optimization problem (4) we may solve for each labelled tree T the local problem $P(T)$ and compare all solutions to find the best one. We shall see later how this procedure can be speeded up in certain special cases.

3. The maxmin u_k- and minmax u_k-problem

For $p = -\infty$ and $p = \infty$ problem (4) specializes to

$$\max_{t \in [\mathbb{R}/A]^r} f(t) \quad \text{with } f(t) = \min_{u \in U(t)} u \tag{11}$$

and

$$\min_{t \in [\mathbb{R}/A]^r} f(t) \quad \text{with } f(t) = \max_{u \in U(t)} u. \tag{12}$$

To solve these problems we have to solve problems $P(T)$ which turn out to be specially structured linear programs. We will study these linear programs in Sections 3.1 and 3.2.

3.1. The maxmin u_k-problem

If we are dealing with the maximum u_k-problem we only have to consider the minimal distances $d_{ij}(T)$ of the sets $U_{ij}(T)$.

Therefore the problem $P(T)$ which corresponds with (11) is

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

Note that $\min\{d_{ii}(T)+x_i-x_i \mid i=1,\dots,r\}$ is a constant and that all $d_{ij}(T)$ are nonnegative. Since $x_i=0$ for all i is a feasible solution the maximum of $\min\{d_{ij}(T)+x_j-x_i \mid i,\ j=1,\dots,r\}$ will be nonnegative in an optimal solution.

Therefore the problem above may be replaced by

$$\max \min\{d_{ij}(T)+x_j-x_i \mid i,\ j=1,\dots,r;\ i\neq j\}$$

or the linear program

$$\begin{aligned}&\max z,\\ &\text{subject to} \quad z-x_j+x_i \leqslant d_{ij}(T), \quad i,\ j=1,\dots,r,\ i\neq j.\end{aligned} \tag{13}$$

The dual of (13) is

$$\begin{aligned}&\min \sum_{\substack{i,j=1\\ i\neq j}}^{r} d_{ij}(T)\, y_{ij},\\ &\text{subject to} \quad \sum_{\substack{i,j=1\\ i\neq j}}^{r} y_{ij}=1,\\ &\qquad \sum_{i=1}^{r} y_{ji}-\sum_{i=1}^{r} y_{ij}=0, \quad j=1,\dots,r,\\ &\qquad y_{ij}\geqslant 0, \quad i,\ j=1,\dots,r,\ i\neq j.\end{aligned} \tag{14}$$

Next we will derive some properties of (13) and (14).

Let $N(T)=(V,E,d(T))$ be the network with $V=\{1,\dots,r\}$, $E=\{(i,\ j) \mid i,\ j=1,\dots,r;\ i\neq j\}$ and $d(T): E\to \mathbb{R}^+$ defined by $(i,\ j)\mapsto d_{ij}(T)$.

Let C: $(i_1, i_2), (i_2, i_3),\dots,(i_{s-1}, i_s), (i_s, i_1)$ be a cycle in $N(T)$. By $|C|$ we denote the number of arcs in C. We also write $(i,\ j)\in C$ if arc $(i,\ j)$ belongs to C. The *average length* of C is defined by

$$\frac{1}{|C|}\sum_{(i,j)\in C} d_{ij}(T).$$

Finally, let $f(t)$ be the optimum distance on the circle line for schedule t.

Theorem 1. *Let* C: $(i_1, i_2), \ldots, (i_s, i_1)$ *be a cycle of minimum average length in* $N(T)$. *Then* y *defined by*

$$y_{ij} = \begin{cases} \frac{1}{|C|} & \text{if } (i, j) \in C, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

is an optimal solution of (14).

Proof. Problem (14) is a circulation problem. Let y be an optimal circulation. Then y may be split into a sum of circulations in cycles $C_1, \ldots, C_e$, i.e., we have

$$y_{ij} = \sum_{(i,j) \in C_\nu} k_\nu \quad \text{for all } (i, j) \in E,$$

where k_ν $(\nu = 1, \ldots, e)$ are positive real numbers with $\sum_{\nu=1}^{e} k_\nu |C_\nu| = 1$. Assume without loss of generality that

$$\frac{1}{|C_1|} \sum_{(i,j) \in C_1} d_{ij}(T) \leqslant \frac{1}{|C_\nu|} \sum_{(i,j) \in C_\nu} d_{ij}(T) \quad \text{for } \nu = 1, \ldots, e.$$

Then

$$\begin{aligned} \sum_{(i,j) \in C_1} d_{ij}(T)\left(\frac{1}{|C_1|}\right) &= \left(\sum_{\nu=1}^{e} k_\nu |C_\nu|\right)\left(\sum_{(i,j) \in C_1} d_{ij}(T)\left(\frac{1}{|C_1|}\right)\right) \\ &\leqslant \sum_{\nu=1}^{e} k_\nu |C_\nu| \frac{1}{|C_\nu|} \sum_{(i,j) \in C_\nu} d_{ij}(T) \\ &= \sum_{\nu=1}^{e} \sum_{(i,j) \in C_\nu} d_{ij}(T) k_\nu = \sum_{\substack{i,j=1 \\ i \neq j}}^{r} d_{ij}(T) y_{ij}. \end{aligned}$$

Thus, if we take $C = C_1$, then (15) is an optimal solution of (14). □

According to Guldan [6] a sequence $P_{i_1}, P_{i_2}, \ldots, P_{i_s}$ of polygons is called a *polygon cycle* with respect to t if there exist $P_{i_\nu}P_{i_{\nu+1}}$-distances ($\nu = 1, \ldots, s-1$) and a $P_{i_s}P_{i_1}$-distance all equal to $f(t)$. Using this concept of a polygon cycle the optimal solution of the primal problem (13) has the following property.

Corollary 2. *Let* (z, x) *be an optimal solution of* (13) *and let* C: $(i_1, i_2), \ldots, (i_s, i_1)$ *be a cycle of minimum average length in* $N(T)$. *Then* $P_{i_1}, P_{i_2}, \ldots, P_{i_s}$ *is a polygon cycle for* $t + x$.

Proof. Due to complementary slackness we have

$$z = d_{ij}(T) + x_j - x_i \quad \text{for all } (i, j) \in C,$$

which implies that $P_{i_1}, P_{i_2}, \ldots, P_{i_s}$ is a polygon cycle for $t + x$. □

The problem of finding a cycle with minimum average length can be solved in $O(r^3)$ steps (see [7]).

Assume that we have found a tree T such that the corresponding local region contains an optimal solution of the global problem (11). Furthermore let C: $(i_1, i_2), \ldots, (i_s, i_1)$ be a cycle with minimum average length in $N(T)$. To find an optimal solution x of (11) we may proceed as follows.

We first define the x_i-values for all $i \in V_c = \{i_1, \ldots, i_s\}$ by

$$x_{i_1} = 0, \qquad x_{i_{\nu+1}} = z - d_{i_\nu i_{\nu+1}}(T) + x_{i_\nu}, \quad \nu = 1, \ldots, s-1, \tag{16}$$

where z is the optimal objective value of (14). Thus, independently how the other components of x are defined, (z, x) and the optimal solution y defined by (15) satisfy the complementary slackness conditions. Therefore we have to define the remaining components of x such a way that x is feasible for (13).

First notice that (16) implies

$$x_{i_1} = z - d_{i_s i_1}(T) + x_{i_s}, \tag{17}$$

for otherwise addition of all equations (16) and (17) yields

$$z \neq \frac{1}{|C|} \sum_{(i,j) \in C} d_{ij}(T),$$

which is a contradiction.

Furthermore

$$x_j \geqslant z - d_{ij}(T) + x_i \quad \text{for all } i,\ j \in V_c.$$

This follows from the fact that if

$$x_j < z - d_{ij}(T) + x_i$$

for some $(i, j) \notin C$, i, $j \in V_c$, then there exists a cycle C_1 with

$$\frac{1}{|C_1|} \sum_{(i,j) \in C_1} d_{ij}(T) < z,$$

(see Fig. 2) which contradicts the optimality of z.

Thus, we have

$$z - x_j + x_i \leqslant d_{ij}(T)$$

or

$$d'_{ij}(T) + x_i \leqslant x_j \quad \text{where } d'_{ij}(T) = z - d_{ij}(T) \tag{18}$$

for all i, $j \in V_c$.

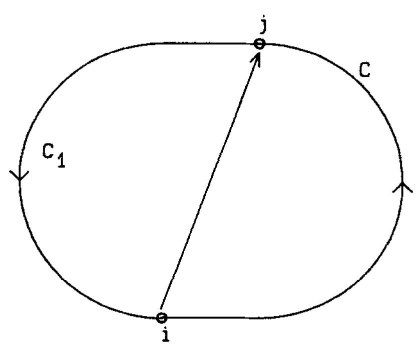

Fig. 2.

From (18) it follows that the other x_i-values can be found by solving a longest path problem. For this purpose consider a network $N'(T)$ derived from $N(T)$ by eliminating all arcs of the complete graph with node set V_c and by adding one new node w and arcs (w, i) with length x_i for all $i \in V_c$. Furthermore, all the other arc lengths $d_{ij}(T)$ are replaced by $d'_{ij}(T) = z - d_{ij}(T)$. We are interested in the longest paths from w to all other nodes in $N'(T)$. These longest paths exist because in $N'(T)$ there exists no cycle of positive length. For all $i \in V$ let x_i be the length of the longest path from w to i in $N'(T)$.

3.2. The minmax u_k-problem

The problem $P(T)$ which corresponds with (12) is

$$\begin{aligned}&\min \max\{h_{ij}(T) + x_j - x_i \mid i, j = 1, \ldots, r\},\\ &\text{subject to} \quad d_{ij}(T) + x_j - x_i \geqslant 0, \quad i, j = 1, \ldots, r, \ i \neq j.\end{aligned} \tag{19}$$

In (19) the $d_{ij}(T)$-values are defined as before. Furthermore the $h_{ij}(T)$ are the maximal P_iP_j-distances for all $i, j = 1, \ldots, r$. Notice, that in this situation we also consider maximal P_iP_i-distances.

A formulation equivalent to (19) is

$$\begin{aligned}&\min w,\\ &\text{subject to} \quad w - x_j + x_i \geqslant h_{ij}(T), \quad i, j = 1, \ldots, r,\\ &\qquad\qquad\quad -x_j + x_i \geqslant -d_{ji}(T), \quad i, j = 1, \ldots, r, \ i \neq j.\end{aligned} \tag{20}$$

The dual of (20) is

$$\begin{aligned}&\max \sum_{i,j=1}^{r} h_{ij}(T) y_{ij} - \sum_{\substack{i,j=1\\ i \neq j}}^{r} d_{ji}(T) z_{ij},\\ &\text{subject to} \quad \sum_{i,j=1}^{r} y_{ij} = 1,\\ &\qquad \sum_{i=1}^{r} y_{ji} + \sum_{\substack{i=1\\ i \neq j}}^{r} z_{ji} - \sum_{i=1}^{r} y_{ij} - \sum_{\substack{i=1\\ i \neq j}}^{r} z_{ij} = 0, \quad j = 1, \ldots, r,\\ &\qquad y_{ij} \geqslant 0, \quad i, j = 1, \ldots, r,\\ &\qquad z_{ij} \geqslant 0, \quad i, j = 1, \ldots, r, \ i \neq j.\end{aligned} \tag{21}$$

Again, the dual linear program (21) is a circulation problem in a different type of network $N^*(T)$. $N^*(T)$ has the form $N^*(T) = (V, E, c)$ with node set $V = \{1, \ldots, r\}$ and arc set $E = H \cup D$ where H and D are disjoint. (V, H) is a complete digraph with loops $e_i = (i, i)$ for $i = 1, \ldots, r$. (V, D) is a loopfree complete digraph. Thus, associated with each pair (i, j), $i \neq j$, are exactly two arcs $a = (i, j) \in H$ and $\bar{a} = \overline{(i, j)} \in D$ which are parallel to each other. We define $c(a) = h_{ij}(T)$ and $c(\bar{a}) = -d_{ji}(T)$. For all loops e_i we set $c(e_i) = h_{ii}(T)$. The network we obtain by omitting all arcs in D is denoted by $N(T)$.

Problem (21) is equivalent to the problem of finding a cycle C in $N^*(T)$ containing at least one arc from H which maximizes the *modified average length*

$$\frac{1}{|H \cap C|}\left\{\sum_{(i,j)\in H\cap C} h_{ij}(T) - \sum_{\overline{(i,j)}\in D\cap C} d_{ji}(T)\right\}. \tag{22}$$

If C is such a cycle, then a circulation in C with value $1/|H \cap C|$ is an optimal circulation for (21).

For an optimal tree T a theorem similar to Theorem 1 holds.

Theorem 3. *Let T be an optimal tree and C: $(i_1, i_2),\ldots,(i_s, i_1)$ be a cycle of maximal average length in $N(T)$. Then y and z defined by*

$$y_{ij} = \begin{cases} \dfrac{1}{|C|} & \text{if } (i, j) \in C, \\ 0, & \text{otherwise}, \end{cases}$$

$$z_{ij} := 0, \tag{23}$$

is an optimal solution of (21).

We first prove the next lemma.

Lemma 4. *Let t^* be an optimal solution for* (12). *Then there exists a polygon cycle with respect to t^*.*

Proof. Assume there exists no polygon cycle with respect to t^*. Then we consider a directed graph $G = (V, E)$ with $V = \{1,\ldots, r\}$ and

$$E = \{(i, j) \mid \text{there exists a } P_iP_j\text{-distance (clockwise) equal to } f(t^*)\}.$$

Due to our assumption G must be acyclic. Starting with t^* we construct a new schedule by a clockwise movement of the polygons. This is done in the following way. We first move all polygons which correspond with sources (but not sinks) in G by a small amount ϑ. Let G_1 be the graph we get by eliminating all these sources and arcs incident with these sources. We apply the same process to G_1 but move the corresponding polygons only $\frac{1}{2}\vartheta$ units, etc. We continue until all nodes are isolated. Let k be the number of steps of this procedure which decreases all maximum distances.

Furthermore we have

$$\vartheta + \frac{\vartheta}{2} + \frac{\vartheta}{4} + \cdots + \frac{\vartheta}{2^k} < 2\vartheta.$$

Thus, distances smaller than the maximum distance are increased by at most 2ϑ. If we choose ϑ sufficiently small, $f(t^*)$ will be decreased which is a contradiction to the fact, that t^* was optimal. □

Proof of Theorem 3. Let T be an optimal tree and $t^* = t + x$ be a corresponding optimal solution for (12). From Lemma 4 we know that there exists a polygon cycle with respect to t^*. Thus, for all (i, j) which correspond to this polygon cycle we must have

$$w - x_j + x_i = h_{ij}(T).$$

By adding these equations we get

$$f(t^*) = w = \frac{1}{|C_1|} \sum_{(i,j) \in C_1} h_{ij}(T),$$

where C_1 is the cycle in $N(T)$ which corresponds with the polygon cycle. Thus, if in (23) we replace C by C_1, we get an optimal solution of (21). Because a cycle C of maximum average length in $N(T)$ leads to a feasible solution (23) of (21) this solution must be optimal. □

Corollary 5. *Let T be an optimal tree, let (w, x) be an optimal solution of* (20) *and let C: $(i_1, i_2), \ldots, (i_s, i_1)$ be a cycle of maximum average length in $N(T)$. Then $P_{i_1}, \ldots, P_{i_s}$ is a polygon cycle for $t + x$.*

Since finding a cycle of maximum average length corresponds to the free optimization problem

$$\min \max\{ h_{ij}(T) + x_j - x_i \mid i, j = 1, \ldots, r \}, \tag{24}$$

we can adopt the following strategy for finding an optimal solution t_{opt} with objective function value f_{opt} for (19).

Let T be an optimal tree. Then (24) has a solution which is also feasible for (19). This follows from Theorem 3 and from duality. Thus, we can search for a tree T for which (24) has a solution which is also feasible for (19). Among these trees one with minimum objective value is optimal.

The idea described above can be formalized in the following algorithm.

Algorithm 1

1. $f_{\text{opt}} := \infty$;
2. **FOR ALL** *labelled trees* T **DO**
3. **IF** *there exists an optimal solution* x *for* (24) *which is also feasible for* (19) **THEN**
4. **IF** $f(t + x) < f_{\text{opt}}$ **THEN**
5. **BEGIN** $f_{\text{opt}} := f(t + x)$; $t_{\text{opt}} := t + x$ **END**

To check whether there exists an optimal solution x for (24) which is also feasible for (19) it is sufficient to know the optimal objective value w^* of (24) and to check whether the system

$$x_j - x_i \leqslant m_{ij}(T) := \min\{ w^* - h_{ij}(T),\ d_{ji}(T) \}, \quad i, j = 1, \ldots, r,$$

where $d_{ii}(T) = \infty$ for all i, has a feasible solution. However, by duality feasibility of this system is equivalent with the nonexistence of negative cycles in the network $(V, A, m(T))$. Thus, an optimal tree can be found by calculating cycles with maximum average costs and shortest path calculations. After we have identified an optimal tree T a corresponding optimal solution $t + x$ may be calculated in a similar way as described in Section 3.1.

4. The $\sum u_k^p$-problem for nonnegative even integers p

In this section we will study problem (4) for nonnegative even integers p. In this case (9) is a strictly convex function on $[\mathbb{R}/A]^r$ and therefore necessary and sufficient conditions for the free optimum of f are

$$\sum_{\substack{i=1\\i\neq k}}^{r} \sum_{u\in U_{ik}(T)} (u+x_k-x_i)^{p-1} - \sum_{\substack{j=1\\j\neq k}}^{r} \sum_{u\in U_{kj}(T)} (u+x_j-x_k)^{p-1} = 0, \quad k=1,\dots,r. \tag{25}$$

Theorem 6. *For positive even integers p there exists a tree T such that an optimal solution of the global optimization problem* (4) *is given by a solution of the free optimization problem* (9) *which simultaneously satisfies* (10).

Proof. Let t^* be an optimal solution of the global problem (4). Then there exists some tree T and a vector x^* satisfying (10) such that $t^* = t + x^*$. We have to prove that x^* is a solution of the free optimization problem (9).

It is sufficient to show that (25) holds for $x = x^*$. If for some $k = 1,\dots, r$ we replace x_k^* by $x_k^* + \epsilon$, then the value of the objective function is given by

$$\begin{aligned} f(x^*+\epsilon e_k) &= f(x^*) \\ &\quad + \epsilon p\left[\sum_{\substack{i=1\\i\neq k}}^{r} \sum_{u\in U_{ik}(T)} (u+x_k^*-x_i^*)^{p-1} - \sum_{\substack{j=1\\j\neq k}}^{r} \sum_{u\in U_{kj}(T)} (u+x_j^*-x_k^*)^{p-1}\right] \\ &\quad + \mathrm{O}(\epsilon^2). \end{aligned} \tag{26}$$

Thus, (25) must hold for otherwise we could improve the global optimum by choosing a suitable ϵ-value with $|\epsilon|$ small. Notice, that if some $u + x_k^* - x_i^*$ or $u + x_j^* - x_k^*$ are zero, then we have to change the tree for negative or positive ϵ. This means that in (26) some zero values are to be shifted from one sum to the other. However, the argumentation is still valid. □

Of particular interest is the case $p = 2$. For $p = 2$ formula (25) becomes a system of linear equations

$$\sum_{\substack{i=1\\i\neq k}}^{r} \sum_{u\in U_{ik}(T)} (u+x_k-x_i) - \sum_{\substack{j=1\\j\neq k}}^{r} \sum_{u\in U_{kj}(T)} (u+x_j-x_k) = 0, \quad k=1,\dots,r. \tag{25'}$$

If we introduce

$$U_k^+ = \bigcup_{\substack{i=1\\i\neq k}}^{r} U_{ik}(T), \qquad U_k^- = \bigcup_{\substack{j=1\\j\neq k}}^{r} U_{kj}(T), \qquad c_k = \sum_{u\in U_k^-} u - \sum_{u\in U_k^+} u,$$

$$u_k^+ = |U_k^+|, \qquad u_k^- = |U_k^-|, \qquad u_k = u_k^+ + u_k^-,$$

$$u_{ik}^+ = |U_{ik}(T)|, \qquad u_{ki}^- = |U_{ki}(T)|, \qquad v_{ki} = -u_{ki}^- - u_{ik}^+,$$

the system (25′) may be written in the compact form

$$u_k x_k + \sum_{\substack{i=1\\i\neq k}}^{r} v_{ki} x_i = c_k, \quad k=1,\dots,r.$$

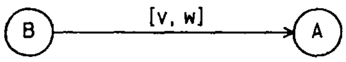

Fig. 3.

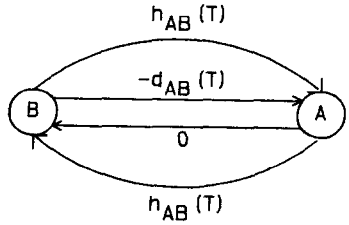

Fig. 4.

Note that one variable in this system may be chosen arbitrarily (for example $x_1 = 0$). Algorithm 1 amounts now to solving (25′)—which can be done in $O(r^3)$ steps—and to checking whether the solution of the linear equation system fulfils

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

Thus, the local problem can be solved in $O(r^3)$ steps.

5. Examples

In this section we will apply the results developed so far to problems with $r = 2$ or $r = 3$.

5.1. The minmax u_k-problem for $r = 2$

The trees T are given by Fig. 3, where A and B denote the two polygons and w and v are arbitrary vertices of A and B.

We may create all trees T by fixing polygon B and moving polygon A clockwise around the circle. Each T corresponds with a situation in which a vertex w in A coincides with some vertex v in B. For such a situation let $U_{AB}(T)$ and $U_{BA}(T)$ be the multisets of all AB-distances and BA-distances on the circle.

Define

$$d_{AB}(T) = \min U_{AB}(T), \qquad h_{AB}(T) = \max U_{AB}(T), \qquad h_{BA}(T) = \max U_{BA}(T).$$

If we ignore AA- and BB-distances (which is appropriate in some applications) we get the network of Fig. 4.

The minimal modified average cycle length for this network is

$$\text{OPT}(T) = \max\{h_{AB}(T) - d_{AB}(T),\ \tfrac{1}{2}[h_{AB}(T) + h_{BA}(T)],\ h_{BA}(T)\}$$

or

$$\text{OPT}(T) = \begin{cases} h_{BA}(T) & \text{if } h_{BA}(T) > h_{AB}(T), \\ \tfrac{1}{2}[h_{AB}(T) + h_{BA}(T)] & \text{if } h_{BA}(T) \leqslant h_{AB}(T), \\ & \quad d_{AB}(T) > \tfrac{1}{2}[h_{AB}(T) - h_{BA}(T)], \\ h_{AB}(T) - d_{AB}(T) & \text{if } h_{BA}(T) \leqslant h_{AB}(T), \\ & \quad d_{AB}(T) \leqslant \tfrac{1}{2}[h_{AB}(T) - h_{BA}(T)]. \end{cases}$$

If we update the sets $U_{AB}(T)$ and $U_{BA}(T)$ efficiently when moving polygon A relative to polygon B we obtain an algorithm with running time $O(nm \log m)$. Here m and n are the numbers of

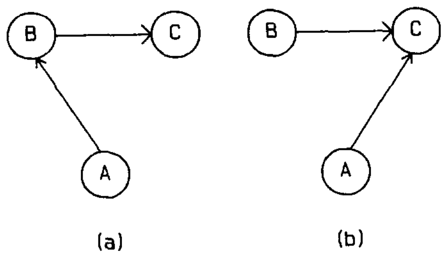

Fig. 5.

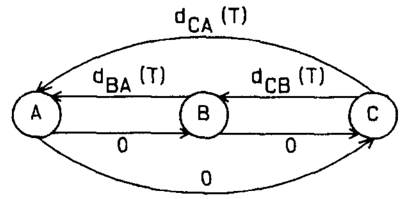

Fig. 6.

vertices of polygons A and B. More details can be found in [1] where the maxmin u_k-problem for two polygons is also discussed.

5.2. The maxmin u_k-problem for $r = 3$

We have three polygons denoted by A, B, C and fix polygon C. To create all trees we apply two passes. In the first pass we first move the B-polygon clockwise relative to the C-polygon. For each situation in which a B-vertex and a C-vertex coincide we move the A-polygon around the whole circle. In the second pass the roles of A and B are interchanged.

In the first pass we get 2 types of trees shown in Fig. 5. Tree arcs are presented without their labels.

The network shown in Fig. 6 corresponds with the tree of Fig. 5(a). The corresponding optimal objective value OPT(T) of $P(T)$ is given by

$$\text{OPT}(T) = \min\{\tfrac{1}{2}d_{BA}(T),\ \tfrac{1}{2}d_{CB}(T),\ \tfrac{1}{3}(d_{BA}(T) + d_{CB}(T)),\ \tfrac{1}{3}d_{CA}(T),\ d\},$$

where

$$d = \min\{d_{AA}(T),\ d_{BB}(T),\ d_{CC}(T)\}.$$

The network which corresponds with the tree shown in Fig. 5(b) is shown in Fig. 7. The corresponding optimal value is

$$\text{OPT}(T) = \min\{\tfrac{1}{2}d_{CA}(T),\ \tfrac{1}{2}d_{CB}(T),\ \tfrac{1}{2}(d_{BA}(T) + d_{AB}(T)),\ \tfrac{1}{3}(d_{CA}(T) + d_{AB}(T)),\ \tfrac{1}{3}(d_{CB}(T) + d_{BA}(T)),\ d\}.$$

The trees of the second pass are completely symmetric to those of the first pass. All we have to do is to interchange A and B.

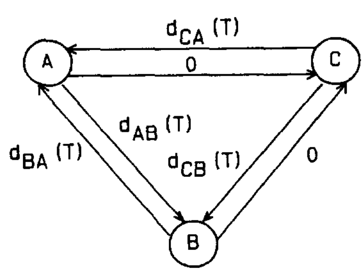

Fig. 7.

5.3. The min $\sum u_k^2$-problem for $r = 2$

For $r = 2$ the system (25′) of linear equations and the nonnegative constraints (10) reduces to

$$0 = d_{12}(T) \geqslant -x, \qquad d_{21}(T) \geqslant x, \qquad v_{12}x = c_1,$$

if we set $x_1 = 0$ and $x_2 = x$. Thus, we only have to check whether

$$0 \leqslant \frac{c_1}{v_{12}} \leqslant d_{21}(T),$$

with

$$c_1 = \sum_{u \in U_{12}(T)} u - \sum_{u \in U_{21}(T)} u, \qquad v_{12} = |U_{12}(T)| + |U_{21}(T)|,$$

holds. If in this case

$$f(t + x) = \sum_{i,j=1}^{2} \sum_{u \in U_{ij}(T)} \left(u + \frac{c_1}{v_{12}}\right)^2 < f_{\text{opt}},$$

we have to replace f_{opt} by $f(t + x)$ (see step 3–5 of Algorithm 1).

6. Complexity

The number of local regions is bounded by $(\prod_{i=1}^r m_i)(\sum_{i=1}^r m_i)^{r-2}$ (cf. [8]), i.e., we get polynomially many local regions for a fixed number r of polygons. Since for $p = -\infty$, $p = 2$ and $p = +\infty$ the local optimization problems are polynomially solvable this amounts to a *polynomial method* for solving the minmax problem, the min average problem and the maxmin problem for fixed r. However, the complexity of algorithms presented for solving the general problem grows exponentially with r. In this section we show that the general problem is NP-hard for each p ($-\infty \leqslant p < 0$ *or* $1 < p \leqslant \infty$).

In this connection it is crucial how the encoding of the input for the polygon scheduling problem is chosen. If we are dealing with regular polygons it is naturally to describe a polygon by the number of circle segments. On the other hand if a polygon is irregular, it will be described by two sequences $l_1, \ldots, l_s$ and $m_1, \ldots, m_s$ of numbers where $l_i \neq l_{i+1}$ for $i = 1, \ldots, s - 1$. The meaning of this encoding is as follows: m_i counts the number of repetitions of an interval of length l_i within the sequence of intervals on the circle created by the polygon.

To prove that (4) is NP-hard we will reduce the 3-partitioning problem to problem (4). The 3-partitioning problem which is known to be NP-complete (see [5]) can be formulated as follows.

3-partitioning problem

Let s, $s(1), \ldots, s(3n)$ be positive integers with

$$\sum_{i=1}^{3n} s(i) = ns \quad \text{and} \quad \tfrac{1}{4}s < s(i) < \tfrac{1}{2}s \quad \text{for all } i = 1, \ldots, 3n.$$

Does there exist a partition $I_1, \ldots, I_n$ of the index set $\{1, \ldots, 3n\}$ such that

$$\sum_{i \in I_j} s(i) = s \quad \text{for } j = 1, \ldots, n?$$

Theorem 7. *For each p $(-\infty \leqslant p < 0$ or $1 < p \leqslant \infty)$ the 3-partitioning problem is polynomially reducible to the $\sum u_k^p$-problem with $3n$ polygons.*

Proof. Let p be finite. Then we associate with the 3-partitioning problem a $\sum u_k^p$-problem with the following data. First let the circle length A be equal to s. Furthermore, let polygon P_i $(i = 1,\ldots,3n)$ have $s(i)$ vertices. One distance between these vertices is defined to be equal to $s - s(i) + 1$. The other $s(i) - 1$ distances are equal to 1. We will show that the 3-partitioning problem has a solution if and only if the cyclic scheduling problem has a solution with

$$\sum u_k^p \leqslant ns\left(\frac{1}{n}\right)^p. \tag{27}$$

If the 3-partitioning problem has a solution, then there exists a partition $I_1,\ldots, I_n$ such that for each $j = 1,\ldots,n$ the total number of vertices of polygons P_i with $i \in I_j$ is equal to s. Furthermore, there exists a schedule t such that for each $j = 1,\ldots, n$ these vertices are placed on the s integer positions on the circle. By (clockwise) moving all polygons P_i with $i \in I_j$ exactly $(j-1)/n$ units we get a schedule in which all ns vertices are distributed equidistantly on the circle. Thus, for this new schedule inequality (27) holds.

If on the other hand inequality (27) holds, then the ns vertices of all polygons must be equidistantly distributed because $ns(1/n)^p$ is the optimal objective value even in the case in which ns points can be moved freely on the circle. Due to the fact that all distances between vertices of the same polygon are integral we must have

$$\sum_{i \in I_j} s(i) = s \quad \text{for } j = 1,\ldots, n,$$

where I_j is the set of indices of polygons P_i with vertices on positions $(j-1)/n + k$ for $k = 0,\ldots, s-1$.

NP-hardness for the maxmin u_k-problem $(p = -\infty)$ and the minmax u_k-problem $(p = \infty)$ can be proved in a similar way. □

Although the general problem (4) is NP-hard the algorithms presented in this paper are feasible approaches to solve the problems for small r as indicated in the examples of the last section. Furthermore, we can solve the local optimization problem belonging to a given labelled tree T. This process may be a useful subroutine in connection with heuristic algorithms.

7. Conclusion

We have developed methods for scheduling irregular polygons with vertices on a circle under different objectives. In applications the polygons may be interpreted as railway lines which pass a station. In this context the following modifications and generalizations are of interest:

- the distances u_k can be weighted by the number of waiting passengers;
- only distances between trains heading for different destinations are of importance;
- a network of lines instead of lines through a single station.

The question how to solve the modified problems may be settled on the basis of the model and solution methods developed in this paper. These issues are topics of current and future research in this area.

Acknowledgement

We thank G. Rote for his valuable remarks on an earlier version of this paper, in particular for pointing us out formula (8) for the number of labelled trees.

References

[1] P. Brucker and W. Meyer, Scheduling two irregular polygons, *Discrete Appl. Math.* **20** (1988) 91–100.
[2] R.E. Burkard, Optimal schedules for periodically recurring events, *Discrete Appl. Math.* **15** (1986) 167–180.
[3] N. Christofides, *Graph Theory, an Algorithmic Approach* (Academic Press, New York, 1975).
[4] H. Edelsbrunner, *Algorithms in Combinatorial Geometry* (Springer, Berlin, 1987).
[5] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, San Francisco, CA, 1979).
[6] F. Guldan, Maximization of distances of regular polygons on a circle, *Apl. Mat.* **25** (1980) 182–195.
[7] R.M. Karp, A characterization of the minimum cycle mean in a digraph, *Discrete Math.* **23** (1978) 309–311.
[8] J.W. Moon, Counting labelled trees, Canad. Math. Monographs, No. 1, (Canadian Mathematical Congress, Montreal, 1970).