A Polynomial Algorithm for 2-Cyclic Robotic Scheduling

Vladimir Kats[1] and Eugene Levner[2,*]
[1]Institute for Industrial Mathematics, Beer-Sheva, Israel
kats@iimath.com
[2]Holon Institute of Technology, Holon, Israel
levner@hit.ac.il

Abstract. We solve a single-robot *m*-machine cyclic scheduling problem arising in flexible manufacturing systems served by computer-conrolled robots. The problem is to find the minimum cycle time for the so-called 2-cyclic (or "2-degree") schedules, in which exactly two parts enter and two parts leave the production line during each cycle. An earlier known polynomial time algorithm for this problem was applicable only to the Euclidean case, where the transportation times must satisfy the "triangle inequality". In this paper we study a general non-Euclidean case. Applying a geometrical approach, we construct a polynomial time algorithm of complexity $O(m^6 \log m)$.

1 Introduction

Fully automated production cells consisting of flexible machines and a material handling robot have become commonplace in contemporary manufacturing systems. Much research on scheduling problems arising in such cells, in particular in flowshop-like production cells, has been reported recently. Although there are many differences between the models, they all explicitly incorporate the interaction between the materials handling and the job processing decisions, since this interaction determines the efficiency of the cell. This paper considers a *robotic flowshop cell* which consists of m machines $M_1, \ldots, M_m$, an input station M_0, an output station M_{m+1}, and a single robot that performs all material handling operations in the cell, i.e., the transportation of parts between the machines and the stations, as well as the loading and unloading of parts onto and from the machines and stations. To simplify the presentation, all parts are assumed to be identical; however, all results presented below are valid for the production system producing two product types. The parts are initially available at the input station M_0 and must be sequentially processed on $M_1, \ldots, M_m$ in this order, until they are finally unloaded from M_m and delivered at the output station.

There is no buffer available between the machines. Therefore, once a part is being removed from a machine, without delay it must be transported by the robot to the next station according to technological order (this condition is called *the no-wait condition*). In the practices of PCB industries, violating this condition may deteriorate the product quality and cause a defect product. To move a part, the robot will first travel to the machine where the part is located, wait if necessary, unload the part, travel to the next machine specified by the technological sequence, and load the part. The robot

repeats its moves periodically, such a production process is called *cyclic*, and the corresponding sequence of robot moves is called *a cyclic schedule*. A repeatable sequence in which each processing operation and each robot move appear k times during each cycle is called a *k-cyclic, or k-part, or k-degree, schedule cycle*. During each k-part cycle, exactly k parts enter the line and k parts are unloaded at the output station; at the end of the cycle the flowshop cell returns to its original state (with each machine being loaded and unloaded exactly k times during the cycle). An optimization problem for robotic flowshops asks to specify a *sequence of robot moves*, so as to maximize a predetermined production performance measure, namely, the throughput rate of the flowshop, or, equivalently, to minimize the cycle time.

Due to the importance of the robotic scheduling problems, the vast literature has appeared in recent years being devoted to both cyclic and non-cyclic formulations of these problems. Optimal schedules have been deeply studied over the past decades – we refer the interested reader to the books by Blazewicz et al., 1996, Pinedo 2002 and Sriskandarajah *et al*. 2006, as well as to the comprehensive surveys by Sethi *et al.* 1992, Hall 1999, Crama *et al.* 2000, and Che and Chu 2005, and numerous references therein. In general, the multi-cyclic schedules may have a better throughput rate than the 1-cyclic ones, as have been reported by many researchers (Song *et al.* 1993, Lei and Wang 1994, Levner *et al*. 1996, Kats *et al*. 1999, Lei and Liu 2001, Che *et al.* 2002, Chu *et al*. 2003, to mention a few).

The literature on the k-cyclic scheduling, for $k > 1$, is not so vast, which can be explained by the increased complexity of these problems, in comparison with the 1-cyclic scheduling. To the best of our knowledge, the first works on multi-cyclic robotic scheduling have appeared in the 1960s in the former Soviet Union; Suprunenko *et al.* 1962, Aizenshtat 1963, and Blokh and Tanayev 1966 have proposed concise and elegant mathematical descriptions of the k-cyclic processes with transporting automatic devices and introduced the so-called *method of forbidden intervals* (MFI) for finding an optimal schedule; however, these authors did not establish its polynomiality. This method has been further developed and proved to be polynomial for the 1-cyclic case by Levner *et al.* 1997, where the upper bound of $O(m^3 \log m)$ has been obtained. However, this result is related to the 1-cyclic schedules only. Other early papers devoted to this class of scheduling problems have been limited to the development of heuristic and branch-an-bound methods (see, for example, Song *et al*. 1993, and Lei and Wang, 1994).

Special attention of the researchers has been devoted to the case of 2-cyclic scheduling. Based on the mentioned-above method of forbidden intervals (MFI), Levner *et al*. 1996 have proposed a geometric algorithm solving this class of problems fast in practice and have conjectured that it is polynomial time. This conjecture has been proven by Chu *et al*. 2003 for the special case of the *Euclidian metrics*, in which the transportation times satisfy the "triangle inequality". These authors have proved that, in this case, the complexity of the geometric algorithm is $O(m^8 \log m)$; for their proof, they have used an elaborate analysis of the MFI. Chu 2006 have presented a more sophisticated treatment of the MFI for the considered scheduling problem, which permitted him to improve the algorithm complexity for the Euclidian metrics from $O(m^8 \log m)$ to $O(m^5 \log m)$. The complexity for the general non-Eucledean case has remained an open question.

The aim of the present paper is a further study of the 2-cyclic robotic scheduling problem. We investigate a general *non-Euclidean metrics*, where the processing and transportation times are not required to satisfy the "triangle inequality". Here, the method of forbidden intervals, in its original form, is not applicable because the intervals cannot be simply merged together, as in the Euclidean case. We suggest a different geometric approach based on concepts of *feasible polygones* and *singular points* which is valid for the both cases, Euclidean as well as non-Euclidean. Using this approach, we enhance the geometrical scheme suggested by Levner et al. 1996 and construct an improved algorithm of complexity $O(m^6 \log m)$.

This paper is organized as follows. Section 2 gives a formal description of the problem. Section 3 present the analysis of the problem and restate it as a finite series of the linear programming problems. Section 4 introduces the singular points and estimates their total number. Based on the concepts of feasible polygons and singular points, Section 5 presents the new polynomial algorithm and estimates its complexity. Section 6 concludes the paper.

2 Problem Formulation

For any given instance of the scheduling problem introduced in the previous section, there are two associated fixed sequences, U and S:

- a fixed and *a priori* known sequence $U = \{0, 1, 2, \ldots, m, m+1\}$ which specifies that each of ~~all~~ (identical) parts is loaded at the input station M_0, processed on the machines in order $M_1, \ldots, M_m$, and then is unloaded at an output station M_{m+1};
- an *a priori* unknown sequence S of robot moves, $S = \{s(1)=0, s(2),\ldots, s(2m+2)\}$, which is to be found and specifies the ordering of $2m+2$ (material handling) operations to be performed by the robot in each cycle in the 2-cyclic schedule.

Here it is assumed that M_{m+1} denotes the unloading station;

To make a formal definition for the problem, we introduce the following parameters:

p_i The processing time of a part at machine i, $i = 1, 2,\ldots,m$; to simplify the presentation, we assume that the processing time include the durations of the loading and unloading operations performed by the robot at machine i;

d_i The transportation time of a part from machine i to $i+1$, $i= 0, 1, 2,\ldots,m$, where "machine" $m+1$ is the unloading station;

r_{ij} The traveling time of an unloaded robot from machine i to machine j, $i= 1, 2,\ldots,m+1$; $j=0. 1, 2,\ldots,m$;

Z_i The completion time of the ith operation performed at machine M_i, or, equivalently, the start time of robot's travel to machine M_i.

In the 2-cyclic schedules, exactly two parts enter and two parts leave the line during each cycle. It follows that the (identical) parts are loaded into the line at time
$\ldots - kT, -kT + T_1, \ldots, -2T, -2T+T_1, -T, -T+T_1, 0, T_1, T, T+T_1, 2T, \ldots, kT+T_1, (k+1)T, ,$
where $T_1 < T$.

We start with formulating several natural constraints of the problems. Consider the part introduced into the process at time 0. Due to the no-wait condition, this part must be completed at machine i at time $Z_i = Z_{i-1} + d_{i-1} + p_i$, $i=1,\ldots,m$, $Z_0 = 0$. Correspondingly, the part introduced into the process at time t must be unloaded from machine i at time $t + Z_i$. We assume that at each processing machine (i.e., the machines 1,2, …,m) no more then one part can be processed simultaneously. From this condition it immediately follows that the time intervals between two sequential introductions of parts, T_1 and $T\text{-}T_1$, must satisfy the following inequalities:

$$T_1 \geq max_{i=1,\ldots,m}\, p_i.$$

$$T\text{-}T_1 \geq max_{i=1,\ldots,m}\, p_i.$$

The ability of the robot to transport only one part at time, prohibits also values $d_{i-1} + p_i + d_i$ to be larger than T_1 and $T\text{-}T_1$, for any i, because in such a case the transportation of a part to machine i and from machine i will overlap in time. Thus, we have

$$T_1 \geq T_0 = max_{i=1,\ldots,m}\,(d_{i-1} + p_i + d_i) = max_{i=1,\ldots,m}\,(d_i + Z_i - Z_{i-1}); \quad (1a)$$

$$T\text{-}T_1 \geq T_0 = max_{i=1,\ldots,m}\,(d_{i-1} + p_i + d_i) = max_{i=1,\ldots,m}\,(d_i + Z_i - Z_{i-1}). \quad (1b)$$

On the other hand T_1 and $T\text{-}T_1$ are not larger than $T^0 = Z_m + d_m + r_{m+!,0}$. The schedule with $T_1 = T\text{-}T_1 = T^0$ corresponds to a *primitive cyclic robot* route $S_0 = \{0, 1, 2,\ldots,m\}$ coinciding with the technological order of machines U. Comparing T_0 and T^0, we obtain:

$$T^0 = m\,T_0 + r_{m+!,0}. \quad (1c)$$

The periodicity of the process allows us to restrict its analysis to a single cycle confined within time interval $[0, T)$. Within this interval, exactly two parts must be unloaded from each machine i. The part which was introduced at time $-kT$, where $k = floor(Z_i/T)$ will be unloaded at time $(Z_i) mod\ T = Z_i - kT$, whereas the part which was introduced at time $-hT + T_1$, where $h = floor[(Z_i + T_1)/T]$ will be unloaded at time $(Z_i + T_1) mod\ T = Z_i + T_1 - hT$. Note that station M_0 will be unloaded at time 0 and T_1.

Proposition 1. For the given time intervals T and T_1, the periodically repeated robot route is defined uniquely and can be found by ordering numbers $Y_i = (Z_i) mod\ T$ and $Y_i' = (Z_i + T_1) mod\ T$ ($i=0, 1,\ldots,m$) in increasing order

$$0 = Y_0 = Y^{(*)}_{s1} = Y^{(*)}_{s2} = \ldots = Y^{(*)}_{s(2m+2)} < T, \quad (2)$$

where $Y^{(*)}_{s1} = Y_0 = Z_0$. The sequence of indexes $S = \{s(1)=0, s(2),\ldots, s(2m+2)\}$ determines the robot route, which is the sequence of robot moves between the machines within time interval $[0, T)$.

The proof is similar to the 1-cyclic case given, for instance, in Kats and Levner 1998, and is skipped here.

The sequence S is being repeated periodically, it means that after unloading machine $M_{s(2m+2)}$ the robot moves again to station $s(1) = M_0$ and at time T introduces a new part into the process. Thus, values T and T_1 fully determine the robot schedule. Similar to the 1-cyclic case (see, for instance Kats and Levner 1998), a 2-cyclic schedule is feasible iff the following inequalities are satisfied

$$Y^{(*)}_{sk} + R_{sk,\, s(k+1)} = Y^{(*)}_{s(k+1)}, \quad (3)$$

where $R_{i,\,j} = d_i + r_{i+1,\,j}$, $k=1,2,\ldots,(2m+2)$; $r_{s(2m+2)+1,\,s(2m+3)} = r_{s(2m+2)+1,\,s1}$, $Y^{(*)}_{s(2m+3)} = Y_0+T = T$.

Definition. The sequence of robot's moves is called a *feasible route* or a *feasible schedule* if it ensures that there is sufficient time for the robot to travel between the machines. In formal terms, the robot route is feasible iff the unloading times Y_i satisfy constraints (1a), (1b) and (3).

The 2-cyclic scheduling problem under consideration can now be formulated as follows: To find a feasible 2-cyclic robot route S minimizing the cycle time T (such a route is also called the *optimal schedule*).

3 Problem Analysis

The robot route remains unchanged as far as the set of $Y^{(*)}_{sk}$ values keeps the same order defined by (2); in other words, the robot route changes if and only if the order (2) is changed for some pair of $Y^{(*)}_{sk}$ values. To study all such changes, we will examine all possible intersections between pairs of functions Y_j and Y_j.

3.1. The types of the intersection lines

The unloading times $Y_i=(Z_i) mod\ T = Z_i - kT$ are piecewise linear functions of one variable, namely, the cycle time T, whereas the times $Y_i'= (Z_i+T_1) mod\ T = Z_i+T_1-hT$ are piecewise linear functions of two variables, T and T_1. At some values of T and T_1 the different functions, say, $Y^{(*)}_i$ and $Y^{(*)}_j$ intersect which, in plain language, means that machines i and j must be unloaded simultaneously.

The intersections can be of four types, which are written out below; in what follows, without loss of generality, we will assume that $j > i$.

Type 1. Intersections of $Y_i = (Z_i) mod\ T$ and $Y_j = (Z_j) mod\ T$.
The intersections are determined by the equation

$$Z_i-k_iT= Z_j-k_jT$$

and take place at values

$$T=F_{ijk}=(Z_j-Z_i)/k, \quad (4)$$

where $k = k_j – k_i$, that is. $k = 1, 2,\ldots$.

Note that functions $Y_i = (Z_i) mod\ T$ and, respectively, $Y_j = (Z_j) mod\ T$ consist of segments of lines Z_i-k_iT ($k_i= 1, 2,\ldots$), and, respectively, Z_j-k_jT ($k_j= 1, 2,\ldots$), ranging between 0 and T: $0 = Z_i-k_iT <T$, $0 = Z_j-k_jT <T$ (see Fig.1). Therefore, at value $T=F_{ijk}$, only one pair of segments of lines Y_i and Y_j intersect, whereas the *infinite number* of straight lines Z_i-k_iT and Z_j-k_jT, where $k_j = 0, 1,\ldots,k_i + k$ ($k=1,2,\ldots$), intersect at $T=F_{ijk}$.

As we will show below, in the considered problem the number of different k in (4) is, in fact, finite and bounded from above by the number of machines m. It is due to the fact that the cycle time T, which we are interested in, cannot be arbitrarily small but is bounded from below by T_0 (see condition (1a)).

Type 2. Intersections of $Y_i^{'}= (Z_i+T_1)mod\ T$ and $Y_j^{'}= (Z_j+T_1)mod\ T$.
This type of intersections defines the same set of intersection points as the previous one:

$$Z_i+T_1-h_iT = Z_j+T_1-h_jT;\ \ T=F_{ijk}=(Z_j-Z_i)/k, \text{ where } k = 1, 2,\ldots.$$

Informally, this means that if $T=F_{ijk}$ then the unloading time on machine i and j coincides twice within each cycle.

Type 3. Intersections of $Y_i=(Z_i)mod\ T$ and $Y_j^{'}= (Z_j+T_1)mod\ T$.
For fixed T_1 the intersections take place at the points E_{ijk} similar to $T=F_{ijk}$ considered above*:*

$$T =E_{ijk}=(Z_j+ T_1-Z_i)/k,\ k = 1,2,\ldots$$

which means that those intersections are located along the lines given by equation

$$T_1= -(Z_j-Z_i)+kT,\ k =1,2,\ldots,$$

which (taking into account that $T_1<T$) is a composition of line segments.
Type 4. Intersections of $Y_j=(Z_j)mod\ T$ and $Y_i^{'}= (Z_i+T_1)mod\ T$.
Similar to Case 3, those intersections are located along the lines $T_1= (Z_j-Z_i)-kT$, k=1,2,….

Consider the plane with axes T and T_1 and suppose that values T and T_1 are changed in the plane in an arbitrary way. In such a situation, we will say that a variable point (T, T_1) is *moving along some trajectory*. The intersection analysis considered above indicates that the robot route may change only when the trajectory crosses the following lines:

$$T=(Z_j-Z_i)/k\ , \tag{5a}$$

$$T_1= -(Z_j-Z_i)+kT, \tag{5b}$$

$$T_1= (Z_j-Z_i)-kT, \tag{5c}$$

where k =0,1,2,….

3.2. The number of the intersection lines

Consider now the triangular area A in the plane with coordinates (T, T_1), bounded by the inequalities

$$T_1 = T_0,\ T_1 = T/2,\ \ T_1 = \ T - T^0.$$

Proposition 2. The search for an optimal solution can be restricted to the area A only.

Proof is evident and is skipped

The lines (5a)-(5c) divide the area A into the regions bounded by the line segments; these regions are called *polygons*.

Proposition 3. For any fixed pair of indices (i, j), the number of different lines with different k in (5a)-(5c) crossing the area A is finite and bounded from above by the number of machines, m.

Proof is based on (1a) –(1c).

3.3 A linear programming formulation of the problem

Let us take a point ($T=x$, $T_1=y$) inside of some polygon. This point uniquely determines the robot route. Because only at the polygon edges the unloading times of different machines become equal, for *all* points *(x,y) inside* of the polygon the robot route *cannot change*. Whereas the crossing of line $T_1=-(Z_j-Z_i)+kT$ or line $T_1= (Z_j-Z_i)-kT$ changes the order in robot's sequence S only for one pair of neighboring machines i and j, the crossing of the line $T=(Z_j-Z_i)/k$ changes the order of neighboring machines i and j in two different places of S.

Recall that in the scheduling problem under consideration, we have to find two types of interrelated variables: (1) the time values T and T_1, and (2) the corresponding robot route. Earlier, in Proposition 1, we have established that if T and T_1 are known then the robot route is defined in a unique way. Now suppose that the robot route is known while T and T_1 are unknown.

Proposition 4. For any *fixed* robot route S, the integers k and h in expressions $Y_i=(Z_i) mod\ T= Z_i - kT$ and $Y_i^{'}= (Z_i+T_1) mod\ T =Z_i+T_1-hT$ are uniquely determined by the route.

The proof is identical to that for the 1-cyclic case in Kats and Levner 1998, 2002.

Consider now an arbitrary polygon in A and assume that the robot route in this polygon is $S =\{s(1)=0, s(2),\ldots, s(2m+2)\}$. Then the problem of finding minimal cycle time T for a *fixed robot route S* becomes the following polynomially solvable special case of the linear programming problem defined for two variables, T and T_1:

Problem P. Minimize T

subject to

$$T_1 \geq T_0 = max_{i=1,\ldots,m} (d_{i-1} + p_i + d_i)$$

$$T - T_1 \geq T_0 = max_{i=1,\ldots,m} (d_{i-1} + p_i + d_i)$$

$$Y^{(*)}_{sk} + R_{sk,\, s(k+1)} = Y^{(*)}_{s(k+1)},$$

where $R_{i,\,j} = d_i + r_{i+1,\,j}$, $k=1,2,\ldots,(2m+2)$; $r_{s(2m+2)+1,\, s(2m+3)} = r_{s(2m+2)+1,\, s1}$, $Y^{(*)}_{s(2m+3)}= Y_0+T= T$, and all $Y_i^{(*)}$ are taken in their explicit form, $Y_i = Z_i - kT$ or $Y_i = Z_i + T_1 - kT$. Here all the parameters Z_i are known input data, and k-values for each Y_i are defined as indicated in Proposition 3.

Thus, we have arrived to the following observation: Taking into account that all points (T, T_1) in any polygon define just the same robot route, the original scheduling problem can be solved by examining all possible polygons inside the area A one after another, solving Problem **P** for each of them, and, finally, choosing, among all the obtained solutions, a schedule with the minimum T. As we will see in the next subsection, the amount of polygons within area A is at most $O(m^5)$.

3.4. The number of robot routes. The Euler formula

Let us estimate the total number of polygons in the area A, or, equivalently, the number of different feasible robot routes.

Lemma 1. The number of polygons in A is at most $O(n^5)$.

Proof. 1. First, let's estimate how many points of intersections the polygons can have. The line $T=F_{ijk}=(Z_j-Z_i)/k$ may intersect line $T_1=-(Z_g-Z_f)+hT$ or $T_1=(Z_g-Z_f)-hT$ inside area A only if $h=ceil[(Z_g-Z_f)/F_{ijk}]$ or $h=floor[(Z_g-Z_f)/F_{ijk}]$, correspondingly. It means that one line $T=F_{ijk}=(Z_j-Z_i)/k$ can cross all other lines in at most $O(m^2)$ points and then the total number of such type points, caused by all the intersections of this type, is at most $O(m^5)$.

Further, the intersection of two lines of the same type, that is, either (a) $T_1=-(Z_j-Z_i)+k'T$ and $T_1=-(Z_g-Z_f)+k''T$, or (b) $T_1=(Z_j-Z_i)-k'T$ and $T_1=(Z_g-Z_f)-k''T$, takes place at a point $T=G_{ijfgk}=[(Z_j-Z_i)-(Z_g-Z_f)]/k$, where, without loss of generality, we assume that $(Z_j-Z_i)>(Z_g-Z_f)$, k=1,2,... Note that inside the area A only one pair of lines may intersect at this point. In case (a) this pair of lines is determined by $k'=ceil[(Z_j-Z_i)/G_{ijfgk}]$ and $k''=ceil[(Z_g-Z_f)/G_{ijfgk}]$ whereas in case (b), correspondingly, by $k'=floor[(Z_j-Z_i)/G_{ijfgk}]$ and $k''=floor[(Z_g-Z_f)/G_{ijfgk}]$.

The intersection of lines of different types, that is, either (a) $T_1=-(Z_j-Z_i)+k'T$ with $T_1=(Z_g-Z_f)-k''T$ or (b) $T_1=(Z_j-Z_i)-k'T$ with $T_1=-(Z_g-Z_f)+k''T$, takes place at point $T=G'_{ijfgk}=[(Z_j-Z_i)+(Z_g-Z_f)]/k$. Inside the area A, there are only points such that $[(Z_j-Z_i)\pm(Z_g-Z_f)]/k \geq 2T_0$ and $(Z_j-Z_i)\pm(Z_g-Z_f)=2T^0$; then it immediately follows that $1=k=T^0/T_0=m$. Hence, the total number of points G_{ijfgk} and G'_{ijfgk} is $O(m^5)$. From Proposition 2 it follows that the total amount of lines (5a)-(5c) is $O(m^3)$, so the amount of intersection points of those lines with the triangular border of area A is also at most $O(m^3)$. Thus, the total number of intersection points, including those lying on the border of A, is at most $O(m^5)$.

2. Now we can estimate the total number of polygons. Denote the number of polygons in A by f, the number of intersection points in A by n, and the number of line segments connecting pairs of intersection points in A by e. Interpreting the intersection points as vertices of a planar graph, the connecting segments as edges and the polygons as faces of a planar graph (not including the outer infinitely large face), we can use the *Euler polyhedron formula* which claims:

$$f = e - n + 1.$$

For a simple, connected, planar graph with n vertices and e edges, it is well known in graph theory that, for $n \geq 3$, it holds: $e = 3n - 6$, and, therefore, $f = 2n - 5$. Thus, the total number of polygons in A is of the same order of magnitude as the number of intersection points, that is, it does not exceed $O(m^5)$. ÿ

4. SINGULAR POINTS AND THEIR PROPERTIES

Consider any polygon p_A created by the intersection of lines (5a)-(5c) in the area A. As far as the robot route $S=\{s1=0, s2,\ldots, s(2m+2)\}$ is uniquely determined for all points (T, T_1) in p_A, we can define the polygon p_A as the set of points satisfying $2m+2$ precedence relations in inequalities (2), where each inequality in the system (2) is replaced by one of the following inequalities, using the variables Z_{sk} :

$$Z_{sk} - k_{sk}T = Z_{s(k+1)} - k_{s(k+1)}T, \quad (6a)$$

$$Z_{sk} + T_1 - k_{sk}T = Z_{s(k+1)} + T_1 - k_{s(k+1)}T, \quad (6b)$$

$$Z_{sk} + T_1 - k_{sk}T = Z_{s(k+1)} - k_{s(k+1)}T, \quad (6c)$$

$$Z_{sk} - k_{sk}T = Z_{s(k+1)} + T_1 - k_{s(k+1)}T, \quad (6d)$$

For the given robot route S, the integers k_{sk} and $k_{s(k+1)}$ are uniquely determined by the route and do not depend on values T and T_1 (in the considered polygon p_A).

Along with p_A we consider an area of feasible schedules, denoted by p_B, it is also a polygon (which may be empty) which is located inside polygon p_A and determined by inequalities (3) in such a way that each inequality along the chain (3) is replaced by one of the following inequalities, using the variables Z_{sk} (which can be rewritten in a similar way as inequalities (2) are presented above in the form (6)):

$$Z_{sk} - k_{sk}T + R_{sk,\,s(k+1)} = Z_{s(k+1)} - k_{s(k+1)}T, \quad (7a)$$

$$Z_{sk} + T_1 - k_{sk}T + R_{sk,\,s(k+1)} = Z_{s(k+1)} + T_1 - k_{s(k+1)}T, \quad (7b)$$

$$Z_{sk} + T_1 - k_{sk}T + R_{sk,\,s(k+1)} = Z_{s(k+1)} - k_{s(k+1)}T, \quad (7c)$$

$$Z_{sk} - k_{sk}T + R_{sk,\,s(k+1)} = Z_{s(k+1)} + T_1 - k_{s(k+1)}T. \quad (7d)$$

For the given robot route S, the integers k_{sk} and $k_{s(k+1)}$ are uniquely determined by the route and do not depend on values T and T_1 (see ……). The inequalities (7a) and (7b) can be reduced to

$$T \geq [Z_{sk} - Z_{s(k+1)} + R_{sk,\,s(k+1)}]/(k_{sk} - k_{s(k+1)}), \text{ if } k_{s(k+1)} < k_{sk}. \quad (8)$$

or

$$T = [Z_{s(k+1)} - Z_{sk} - R_{sk,\,s(k+1)}]/(k_{s(k+1)} - k_{sk}), \text{ if } k_{s(k+1)} > k_{sk} \quad (8')$$

Let's assume that polygon p_B is not empty; then the minimal value of the cycle time T in p_B is the T-coordinate of its "most left" node (an intersection point), which must lie on the border of one of the inequalities (8), (9) or (10). These borders have the following form:

$$T = (Z_j - Z_i + R_{i,j})/k \text{ , or} \quad (9a)$$

$$T = (Z_j - Z_i + T_1 + R_{i,j})/k, \text{ or} \quad (9b)$$

$$T = (Z_j - Z_i - T_1 + R_{i,j})/k, \quad (9c)$$

where $j > i$ and $j, i \in \{0, 1, 2, \ldots, m\}$; $k = 1, 2, \ldots, m/2$. We will call the obtained lines (9a)-(9c) *the lines of possible solutions*.

Consider the points in area A lying on lines (9a)-(9c) in which the robot route changes (we call them *singular points*). Those points are defined as the intersections of lines (9a)-(9c) with lines (5a)-(5c). The total amount of all singular points, lying on the lines of possible solutions (9a)-(9c) in area A is at most $O(m^5)$. The proof is similar to the previous case.

5. ALGORITHM: DESCRIPTION AND COMPLEXITY

In this section we present a polynomial algorithm with the worst-case complexity $O(n^6 \log n)$.

The algorithm works as follows.

Step 1. Present all the intersection points and the line segments joining them for lines (5a)-(5c) in area A as, correspondingly, nodes and edges of a planar graph.
Step 2. Make the obtained planar graph Eulerian, by doubling, if needed, its edges (in order to make all the node degrees even), and build an Eulerian cycle in the obtained extended planar graph.
Step 3. Move along the Eulerian cycle and sequentially consider the polygons p_A, for instance, taking, one by one, those polygons p_A that are located on the left to each edge in the Eulerian cycle. In each polygon p_A find a robot's route determined by the system of inequalities (2).
Step 4. For the found robot route, solve the system of inequalities (3) given in form (7a) –(7d) with respect to T and T_1.
Step 5. Among all the found solutions, choose the optimal one, having the minimal T-value.

The validity of the algorithm follows from the fact that each face in the planar graph (i.e., each polygon in A) will be examined at least once.

Let's estimate its complexity. At Step 1, in order to construct the nodes-edges incidence matrix of the planar graph it is sufficient $O(n+e) = O(n^5)$ elementary operations. At Step 2, the total amount of edges, as well as the total amount of faces, in the extended Eulerian graph is increased at most twice, that is, still $O(n^5)$. The complexity of building an Eulerian cycle in the Eulerian graph is linear in the number of edges, i.e. is $O(n^5)$. At Step 3, to build a robot route in each polygon requires $O(n \log n)$ operations. At Step 4, the system of inequalities (7a)-(7d) contains only two variables, therefore, its solution can be found in $O(n \log n)$ operations [Megiddo 1983]. Hence, the total complexity of this straightforward algorithm is at most $O(n^6 \log n)$.

6. A concluding remark

In this paper we have studies the general non-Euclidean case and construct a polynomial time algorithm of complexity $O(m^6 \log m)$. We believe that this worst case upper bound is not tight and can be improved to $O(m^5 \log m)$ and even better. The algorithm has worked much faster in practice than the guaranteed worst-case upper bound. In our future research, we intend to improve the upper bound and to estimate the algorithm behavior on the average.

References

1. V.S. Aizenshtat, Multi-operator cyclic processes, Doklady of the Byelorussian Academy of Sciences, 1963, 7(4), 224-227 (Russian).
2. J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Scheduling Computer and Manufacturing Processes*, Springer Verlag, Berlin, 1996
3. A.Sh. Bloch and V.S. Tanayev, Multi-operator processes, Proceedings of the Byelorussian Academy of Sciences, (physical and mathematical sciences), 1966, (2), 5-11, (Russian).
4. Ada Che, Chengbin Chu and Feng Chu, Multicyclic hoist scheduling with constant processing times, *IEEE Transactions on Robotics and Automation*, February 2002, 18(1), 69-80.

5. Ada Che, Chengbin Chu, and Eugene Levner, A polynomial algorithm for 2-degree cyclic robotscheduling, *European Journal of Operational research*, February 2003, 145(1), 31-44.
6. Ada Che and Chengbin Chu, Multi-degree cyclic scheduling of two robots in a no-wait flowshop, *IEEE Transactions on Automation Science and Engineering*, April 2005, 2(2), 175-183.
7. Chengbin Chu, A Faster Polynomial Algorithm for 2-cyclic robotic scheduling, *Journal of Scheduling*, 2006 (in press).
8. Yves Crama, Vladimir Kats, Joris van de Klundert, and Eugene Levner, Cyclic scheduling in robotic flowshop, Annals of operations Research, 2000, 96(1-4), 97-123.
9. Milind N. Dawande, H.Neil Geismer, Suresh P.Sethi, and Chelliah Sriskandarajah, *Throughput Optimization in Robotic Cells*, Springer, 2006.
10. V. Kats and E. Levner, Cyclic scheduling on a robotic production line, *Journal of Scheduling,* 2002, 5, 23-41
11. V. Kats and E. Levner, A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling, *Operations Research Letters,* 1997, 21, pp. 171-179
12. Vladimir Kats, Eugene Levner and Leonid Meyzin, Multiple-part cyclic hoist scheduling using a sieve method, *IEEE Transactions on Robotics and Automation*, August 1999, 15(4), 704-713.
13. Lei Lei and Qing Liu, Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints, *INFOR*, May 2001, 39(2), 185-199.
14, Lei Lei and T.J. Wang, Determining optimal cyclic hoist schedules in a single-hoist electroplating line, *IEE Transactions*, March 1994, 26(2), 25-33.
15. Eugene Levner, Vladimir Kats and Chelliah Sriskandarajah, A geometric algorithm for finding two-unit cyclic schedules in no-wait robotic flowshop, *Proceedings of the International Workshop in Intelligent Scheduling of Robots and FMS, WISOR-96*, Holon, Israel, HAIT Press, 1996, 101-112.
16. Nimrod Megiddo,Towards a genuinely polynomial algorithm for linear programming, *SIAM Journal on Computing*, 1983, 12, 347-353.
17. Michael Pinedo, Scheduling. Theory, Algorithms and Systems, 2nd ed., Prentice Hall, 2002.
18, S.P. Sethi, C. Sriskandarajah, G.Sorger, J. Blazewicz, and W. Kubiak, Sequencing of parts and robot moves in a robotic cell, *International Journal of Flexible Manufacturing Systems*, 1992, 4, 331-358.
19. W.Song, Z.B. Zabinsky and L.Storch, An algorithm for scheduling a chemical process tank line, *Production Planning & Control*, June 1993, 4, 323-332.
20, D.A. Suprunenko, V.S. Aizenshtat, A.S. Metel'sky, Multi-operator transformation processes, *Doklady of the Byelorussian Academy of Sciences*,1962, 6(9), 541-544 (in Russian).