

**The Complexity of Dynamic/Periodic
Languages and Optimization
Problems**

James B. Orlin

Sloan W.P. No. 1679-85

July 1985

1. Introduction

Deterministic dynamic/periodic optimization problems arise naturally in various quantitative disciplines including Computer Science, Economics, and Operations Research. These periodic models may be applied to long-range economic planning, workforce scheduling, vehicle routing, machine maintenance, and a host of industrial applications. In this paper we offer a unifying framework for dynamic/periodic problems in "terms of dynamic/periodic languages", and we discuss the complexity of these languages. In particular, many such languages derived from NP-complete languages can be shown to be polynomial space (PSPACE) complete. Among these are the dynamic/periodic variants of the following problems: the 3-satisfiability problem, the 3-dimensional matching problem, the number partition problem, the hamiltonian path problem, and the independent set problem. We provide a straightforward technique for showing how to prove the PSPACE-completeness of these periodic problems derived from NP-complete problems.

PSPACE may be characterized in a number of different ways. Chandra and Stockmeyer [CS] and Schaefer [Sc] showed that problems in PSPACE could be reduced to certain 2 person games. More recently, Papadimitriou [P] showed that the class PSPACE could be described in terms of certain generalizations of Markov decision chains, or as he phrased it, "games against nature". Our approach here is to describe the class PSPACE as those languages for which there are "periodic certificates".

This interpretation lends itself well to proving that many NP-complete languages have dynamic/periodic counterparts which are PSPACE-complete. Moreover, these dynamic/periodic languages arise naturally via periodic optimization problems.

In addition, Reif [R] showed that a number of other dynamic problems are PSPACE-complete. Although his class is different from the class described here, his approach is conceptually similar to ours and his proof technique is similar.

The results in this paper were presented in a preliminary form in [01].

The Outline of This Paper

In Section 2, we review some of the research on various dynamic/periodic problems. In Section 3, we illustrate how to derive periodic problems from problems in the class NP.

In Section 4, we define "dynamic/periodic languages" and prove some elementary properties of these languages. In Section 5, we prove the PSPACE-completeness of certain periodic languages derived from NP-complete languages. These results contrast with those by Graves and Orlin [GO] and by Orlin [02], [03], [04], [05] and [06] who provide polynomial time recognition algorithms for various dynamic/periodic languages derived from polynomial time languages, including the following: linear programming, network flows, coloring interval graphs, maximum matching, and 2-satisfiability.

In section 7, we discuss the real-time, complexity of dynamic/periodic optimization problems. Here, periodic optimization problems are viewed as real-time scheduling problems with known periodic data. We show that there is a polynomial real-time scheduling algorithm for the periodic optimization problems presented in Section 3 if and only if $P = PSPACE$.

In Section 6, we show that the periodic knapsack problem may be solved in pseudo-polynomial time (i.e., in polynomial time if the data are unary encoded), and thus it is meaningful to describe other PSPACE problems as "strongly PSPACE complete", a counterpart to strong NP-completeness as described by Garey and Johnson [GJ]. Finally, in Section 8, we present some variations of dynamic/periodic language problems.

2. Some Periodic Optimization Problems

Dynamic/periodic optimization problems occur in a variety of settings. Below we reference some examples of such optimization problems. The list below is intended as a sampler and is far from comprehensive.

Cyclic Staffing

Two types of periodic optimization problems arise typically in employment: shift scheduling and day-off scheduling. Both of these are surveyed by Baker [B]. Many problems of this type arise in service industries, and they may be classified according to the following parameters: types of workshifts allowed, the demand structures, and the types of additional constraints, possibly imposed by union contracts.

Vehicle Routing

Dantzig, Blattner, and Rao [D] consider the problem of determining an infinite horizon tour for a tramp steamer so as to maximize its average profit per day. The tramp steamer is required to travel among a fixed finite set of

ports and receives a profit p_{ij} each time that it travels from port i to port j . The above problem, also called the minimum cost-to-time ratio circuit problem, deals with a transportation problem in which the opportunities for profit are periodic over time. Efficient algorithms for the tramp steamer problem have been given by Dantzig et.al [D], Lawler [L], and Megiddo [M].

Another setting for periodic problems is the area of airplane scheduling (scheduling vehicles to meet a daily repeating set of demands) as surveyed by Simpson [Si].

Scheduling Tasks on Processors

Many job shop problems may be formulated in an environment in which the tasks arrive periodically over time, and the task due-dates also repeat periodically. Subclasses of these problems have been studied by many authors including Labetoulle [La], Dhall and Liu [DL], Lawler and Martel [LM], and Mok [Mo].

Continuous Problems

An entire literature exists on periodic problems in control with applications to many industrial settings, as described in detail in [Ma]. Furthermore, discrete approximations of these models are, in general, special cases of the class of periodic languages defined in Section 4.

3. Dynamic/Periodic Languages Derived from Languages in the Class NP

In this section we describe dynamic/periodic languages that may be considered extensions of languages in the class NP. All of the languages below can be proved to be PSPACE-hard using the technique illustrated in Section 5. These problems also serve to motivate the definitions and theorems of Section 4 on dynamic/periodic languages.

Notation

We let Z denote the set of integers. We let 1 denote a row vector each of whose components has value one. The notation $\| \cdot \|$ refers to the sup norm.

In general, we will let superscripts denote a time dimension. If $S = \{s_1, \dots, s_n\}$ is a finite set of elements then

$$S^{\infty} = \{s_i^p : i \in [1..n] \text{ and } p \in \mathbb{Z}\}.$$

We interpret the element s_i^p as the element s_i in period p . We will also refer to s_i^p as the p -th copy of element s_i . The successor of element s_i^p is s_i^{p+1} and is denoted $SUC(s_i^p)$. If $T \subseteq S^{\infty}$ is a finite subset, then the successor of T is the subset of S^{∞} obtained by replacing each element of T by its successor. We let the successor of T be denoted $SUC(T)$.

Rather than use the cumbersome prefix "dynamic/periodic" throughout this paper, we will typically use the abbreviation D/P.

BOUNDED D/P INTEGER PROGRAMMING

INPUT: $m \times n$ matrices A and B , m -vector d , scalar k .

QUESTION: Is there an infinite sequence $\{x^i : i \in \mathbb{Z}\}$ of non-negative integer n -vectors such that $\|x^i\| \leq k$ and $Ax^i + Bx^{i+1} = d$ for all $i \in \mathbb{Z}$?

D/P PARTITION

INPUT: n -vectors a and b .

QUESTION: Is there an infinite sequence $\{x^i : i \in \mathbb{Z}\}$ of boolean n -vectors such that $ax^{i+1} + bx^i = d$ (where $d = (a+b)/2$) for all $i \in \mathbb{Z}$?

Suppose that $U = \{u_1, \bar{u}_1, \dots, u_n, \bar{u}_n\}$ is a finite set of literals. We say that a set C of clauses defined on U^{∞} is periodic if the following property holds: $c \in C$ if and only if $SUC(c) \in C$. (Here the successor of c is obtained by replacing each literal of c by its successor in U^{∞}). We note that to specify the input for a periodic set of clauses, it suffices to list those clauses whose initial literal is in period 1.

D/P SATISFIABILITY

INPUT: A finite set U of literals and a periodic collection C of clauses defined on U^{∞} .

QUESTION: Is there a truth assignment for all variables such that C is satisfied?

One can carry the analogy to NP-complete problems further. For example, one can define D/P 3-Satisfiability to be special case of D/P Satisfiability in which each clause has exactly 3 literals.

PERIODIC GRAPH THEORETIC LANGUAGES

Let $V = \{1, \dots, n\}$ be a finite set of vertices. A set E of edges on V^{ω} is said to be periodic if it has the following property: $(i^p, j^r) \in E$ if and only if $(i^{p+1}, j^{r+1}) \in E$. (Equivalently, $e \in E$ iff $SUC(e) \in E$.) In addition, there are a finite number of edges incident to each vertex.

An example of a periodic graph is given in Figure 1. To specify the input of a periodic graph it suffices to specify those edges whose tail is incident to a vertex in period 1.

[INSERT FIGURE 1 HERE]

Given the richness of graph theoretic applications, it is not surprising that periodic graphs are useful descriptors of dynamic/periodic problems. For several applications of periodic graphs see [G0], [02] and [05].

Periodic graphs are an infinite extension of time expanded networks. These networks are well known and have been used, for example, by Ford and Fulkerson [FF] in their work as dynamic network flows.

PERIODIC HAMILTONIAN PATH

INPUT: A periodic graph $G = (V^{\omega}, E)$

QUESTION: Is there a path P in G that passes through each vertex exactly once?

PERIODIC 3-COLORABILITY

INPUT: A periodic graph $G = (V^{\omega}, E)$.

QUESTION: Is it possible to color all of the vertices of G with only three colors such that adjacent vertices have different colors?

PERIODIC INDEPENDENT SET

INPUT: A periodic graph $G = (V^{\omega}, E)$ and an integer k with $1 \leq k \leq |V|$.

QUESTION: Is there a set $S \subseteq V^{\omega}$ such that no two vertices in S are adjacent and such that $|S \cap \{i^p : i \in V\}| = k$ for all $p \in \mathbb{Z}$?

D/P 3-DIMENSIONAL MATCHING

A collection of triples $M \subseteq W^{\omega} \times X^{\omega} \times Y^{\omega}$ is called periodic if M has the

following property: $e \in M$ if and only if $SUC(e) \in M$.

INPUT: A periodic collection $M \subseteq W^{\omega} \times X^{\omega} \times Y^{\omega}$.

QUESTION: Is there a subset $M' \subseteq M$ such that each element of $W^{\omega} \cup X^{\omega} \cup Y^{\omega}$ appears in exactly one element of M' ?

THEOREM 1. The following language recognition problems are all PSPACE-hard: Bounded D/P Integer Programming, D/P Partition, D/P 3-Satisfiability, Periodic Hamiltonian Path, Periodic 3-Colorability, Periodic Independent Set, and D/P 3-Dimensional Matching. \square

We will sketch the proof of Theorem 1 in Section 5. We note that the first two problems of Theorem 1 are easily shown to be in the class PSPACE. Without additional conditions it is not clear whether any of the periodic graph problems listed above or D/P 3-SAT or D/P 3-Dimensional matching is in class PSPACE.

4. Dynamic Periodic Languages

In this section we define the class D/P, generalizing several examples of the previous section. This description parallels one of the methods of describing the class NP. As such, it is suggestive of the proof technique given in Section 5. In addition, we show that $D/P = PSPACE$. For alternate descriptions of the class PSPACE, see [GJ] or [CS] or [P].

In the following, the symbol $\$$ will serve as a delimiter. A k -input language L is a language for which every element has $k-1$ delimiters. The class kP refers to the subclass of k -input languages that may be recognized in time polynomial in the size of the first input (the string occurring prior to the first delimiter.)

For each language $L \in 2P$, let

$$N(L) = \{x: x\$y \in L \text{ for some } y\}.$$

REMARK 1. The class NP is equal to the set of languages \mathcal{L} such that $\mathcal{L} = N(L)$ for some $L \in 2P$.

The above remark is a well known alternate characterization of the class

NP. For example, Papadimitriou and Steiglitz [PS] use essentially this characterization as their definition of the class NP.

We will shortly show that a periodic variation of the generation of NP from 2P can be used to generate the class PSPACE from the class 3P.

For each language $L \in 3P$, let $D(L)$ be the dynamic/periodic language defined as follows:

$$D(L) = \{x: \text{there exists an infinite sequence } \{y^i: i \in \mathbb{Z}\} \\ \text{such that } x \# y^i \# y^{i+1} \in L \text{ for each } i \in \mathbb{Z}\}.$$

It is easy to see that the bounded D/P Integer Programming Problem can be written so that it is of the form $D(L)$. Here the x would stand for the input and the strings y^i would correspond to the solution vectors. We also note that without boundedness it is unlikely that D/P Integer Programming could be described in this form. Here the difficulty occurs because the recognition of the string $x \# y^i \# y^{i+1}$ must be in time polynomial in $|x|$. (That is how we defined the class 3P.) Thus we could not allow the sizes of the elements y^i and y^{i+1} to be exponentially large in $|x|$.

Suppose $L \in 3P$ and $x \in D(L)$. We say that a certificate for x is a sequence $\{y^i: i \in \mathbb{Z}\}$ such that $x \# y^i \# y^{i+1} \in L$ for each $i \in \mathbb{Z}$. We say that a certificate (y^i) is periodic with period p if $y^i = y^{i+p}$ for each $i \in \mathbb{Z}$.

THEOREM 2. Suppose that $L \in 3P$. Then $D(L) \in PSPACE$. Moreover, there is a polynomial function $q(\cdot)$ with the following property: for any $x \in D(L)$ there is a periodic certificate (y^i) whose period length is at most $2^{q(|x|)}$.

PROOF. Because $L \in 3P$, there is a polynomial $p(\cdot)$ for which the following is true: if $x \# y \# y' \in L$ then $|y| < p(|x|)$. Now suppose that x is any string. We will show how to determine if $x \in D(L)$ and, if so, construct a periodic certificate for x using a standard technique in dynamic programming. Let G_x be a graph whose vertex set V consists of all strings of size less than $p(|x|)$. Let the edge set E consist of all edges (u,v) such that $x \# u \# v \in L$. Then $x \in D(L)$ if and only if G_x has a directed circuit. Moreover, the length of the circuit is at most $|V|$ which is bounded by $C^{p(|x|)}$, where $C-1$ is the number of symbols in the alphabet.

If $x \in D(L)$, then we may sequentially guess the periodic certificate in polynomial space. Thus $D(L) \in \text{NPSpace}$, the class of languages accepted by non-deterministic turing machines using a polynomially space bounded tape. Also $\text{NPSpace} = \text{PSPACE}$, as proved by Savitch [Sa]. \square

Let D/P be the set of all languages such that $L = D(L)$ for some $L \in 3P$. We also refer to D/P as the set of dynamic/periodic languages.

THEOREM 3. $D/P = \text{PSPACE}$

PROOF. We have already shown that $D/P \subseteq \text{PSPACE}$.

Conversely, suppose that \mathcal{L} is accepted by a polynomially bounded turing machine M . Let $p(\cdot)$ be the polynomial that bounds the space of M .

Let C_x denote the set of all possible configurations of M with the restriction that the space is bounded by $p(|x|)$. Let $L \in 3P$ consist of all triples

$x \# \langle n_1, y \rangle \# \langle n_2, z \rangle$ satisfying the following:

- (i) n_1, n_2 are positive integers less than $|C_x|$.
- either (ii) $n_2 = n_1 + 1$ and z is the configuration that would follow y in M ,
- or else (ii') $n_2 = 1$, y is an accepting (terminal) configuration and z is the initial configuration.

If $x \in \mathcal{L}$, it is apparent that $x \in D(L)$. Moreover, by (ii) and (ii'), any periodic certificate for x with respect to $D(L)$ corresponds to the sequence of steps taken by the turing machine M . Thus $\mathcal{L} = D(L)$. \square

5. PSPACE-Completeness Proofs

The description of PSPACE in terms of dynamic/periodic language suggests a range of problems that are candidates for the class PSPACE-complete. The obvious first candidate is D/P Satisfiability. Unfortunately, the language as defined is not necessarily in the class D/P or the class PSPACE. We remedy this by adding an additional restriction.

We say that a clause c on U^∞ is narrow if any two literals u^p and v^r of

the clause are such that $|r-p| \leq 1$. (We could relax the condition so that $|r-p| \leq q(|U|)$ for some polynomial q . The relaxed condition would still be restrictive enough for our purposes.) We say that a set C of clauses is narrow if each clause is narrow. Similarly, a narrow periodic graph is a periodic graph $G = (V^{\infty}, E)$ such that each edge (i^p, j^r) is narrow, i.e., $|r-p| \leq 1$.

REMARK 2. If we restrict attention to narrow inputs, the following problems are all in the class D/P: D/P Satisfiability, D/P 3SAT, Periodic 3-Colorability, Periodic Independent Set and Periodic 3-Dimensional matching. \square

The proof of Remark 2 is straightforward. Nevertheless it is not always so straightforward to prove that graph problems on narrow periodic graphs are in PSPACE. In particular, it is an open question as to whether the hamiltonian path problem on narrow periodic graphs is in the class PSPACE. The difficulty here is due to the fact that the property of being "hamiltonian" is not locally defined, whereas each of the other graph problems listed above is a local property. For example, to determine if S is an independent set of a narrow periodic graphs, it suffices to verify that $S \cap (V^p \cup V^{p+1})$ is independent for each $p \in \mathbb{Z}$, where V^p is the set of vertices in period p .

A Scheme for Proving the PSPACE-Completeness of Dynamic/Periodic Problems

Suppose that \mathcal{L} and \mathcal{L}' are dynamic periodic variants of NP-complete problems L and L' . Furthermore, suppose that \mathcal{L} is known to be PSPACE-complete. To prove that \mathcal{L}' is PSPACE complete, first transform L into L' in polynomial time. Then transform \mathcal{L} into \mathcal{L}' in polynomial time.

This approach might be viewed (albeit, cynically) as the "greedy approach to proof generation". If 3SAT is used to prove that 3-Colorability is NP-complete, then one might optimistically hope that D/P 3SAT could be used to prove the PSPACE-completeness of periodic 3-Colorability. The remarkable thing is that this greedy approach to PSPACE-completeness proofs usually works.

Intuitively, the description of PSPACE in terms of periodic certificates is sufficiently close to the description of NP in terms of certificates. Thus one can prove Theorem 1 by essentially copying the corresponding NP-completeness proofs from Garey and Johnson [GJ] while keeping careful track of the superscripts which denote the period. We will omit all the proofs except

for D/P Partition, whose proof we include for a different reason.

Of course, such a proof technique is not easily formalized, and such a mimicking does not exist in general. Moreover, the "mapping" of NP problems into D/P problems is not clearly defined. For example, the periodic counterpart to the hamiltonian circuit problem may not be the D/P hamiltonian path problem. Maybe the correct counterpart is the problem finding a subgraph S such that when restricted to the vertex set $V^P \cup V^{P+1}$, S is a hamiltonian circuit.

The above definition is the "natural" generalization in terms of the class D/P, but it is less natural from a graph theoretic point of view. The D/P generalization of the clique problem is just as unnatural from a graph theoretic point of view, viz., select a set S such that $S \cap (V^P \cup V^{P+1})$ is a clique for all $p \in Z$.

Although the greedy approach to PSPACE-completeness proofs works usually, we illustrate below a case in which an added degree of subtlety is required. In the remainder of this section we show how to transform the D/P knapsack problem into the D/P partition problem.

KNAPSACK

INPUT: non-negative integer n -vector a , and integer d .

QUESTION: Is there a 0-1 n -vector x such that $ax = d$?

D/P KNAPSACK

INPUT: non-negative integer n -vectors a and b , and integer d .

QUESTION: Is there an infinite sequence $x = \{x^i : i \in Z\}$ of 0-1 n -vectors such that $ax^{i+1} + bx^i = d$ for all $i \in Z$?

The partition problem is the special case of the knapsack problem in which $d = \frac{1}{2} \sum a_i$.

The usual polynomial time transformation from the knapsack problem into the partition problem is straightforward. Let a and d be the input for the knapsack problem. Let a' be the $(n+1)$ -vector a, a'_{n+1} where $a'_{n+1} = |2d - \sum a_i|$, and let $d' = \frac{1}{2} \sum a'_i$. Then there is a 0-1 vector x such that $ax = d$ if and only if there is a 0-1 vector x' such that $a'x' = d'$. The validity of the "if" direction depends on the following simple

observation: if $a'x' = d'$, then $a'(1-x') = d'$. The solution to the knapsack problem is either x' or $1-x'$ according as $x'_{n+1} = 0$ or $x'_{n+1} = 1$.

The same argument does not apply to the D/P knapsack problem. Suppose one were to attempt to transform the D/P knapsack problem with inputs a, b, d into the D/P partition problem with input (a', b', d') defined as follows:

$$a' = (a, a'_{n+1}), \quad b' = (b, 0) \quad \text{and} \quad a'_{n+1} = |2d - la - lb|, \quad \text{and} \quad d' = (la' + lb')/2.$$

If there is a solution to the D/P knapsack problem then this is a solution to the D/P partition problem, but the converse is not true. A simple example is the problem " $3x^i + 4x^{i+1} = 4$ ", which admits no periodic solution. Nevertheless, the problem " $3x^i + y^i + 4x^{i+1} = 4$ " does have a periodic solution.

There are two remedies: change the problem or change the transformation. The first remedy is actually not frivolous, since it raises a relevant issue. Perhaps the appropriate formulation of knapsack and D/P knapsack are as follows.

KNAPSACK II

INPUT: non-negative integer n -vector a , and integer d .

QUESTION: Is there a 0-1 n -vector x such that either $ax = d$ or else $ax = (la-d)$?

D/P KNAPSACK II

INPUT: non-negative integer n -vectors a and b , and integer d .

QUESTION: Is there an infinite sequence $x = \{x^i : i \in \mathbb{Z}\}$ of 0-1 n -vectors such that $ax^{i+1} + bx^i = d$ or $la + lb - d$ for each $i \in \mathbb{Z}$?

Knapsack and knapsack II are the same problem. However, D/P knapsack II is different from D/P knapsack; moreover, the above transformation shows that D/P Partition is PSPACE-complete assuming that D/P knapsack II is PSPACE-complete. The point is that the failure of the mimicking of the NP-completeness proof may be due in part to the ambiguity in how one should define D/P knapsack.

To transform D/P knapsack into D/P partition is not difficult, but it is more difficult than the corresponding NP-complete transformation. Let a, b, d be the input for D/P knapsack. Let $a' = a'_{n+1}, a'_{n+2}$ and let $b' = b, b'_{n+1}, b'_{n+2}$, where

$$\begin{aligned}
b'_{n+1} &= M + 2d - la - lb \\
b'_{n+2} &= 2M \\
a'_{n+1} &= M \\
a'_{n+2} &= 0 \\
d' &= (la' + lb')/2,
\end{aligned}$$

where M is greater than $la + lb$. The reader can verify that the transformation is correct. The key difference between this transformation and the previous one is that in any feasible solution to the D/P Partition problem we must have that $x_{n+1}^p = x_{n+1}^{p+1}$ and $x_{n+2}^p = x_{n+2}^{p+1}$ for each $p \in \mathbb{Z}$.

6. A Pseudo-Polynomial Algorithm for the D/P knapsack Optimization Problem

The following rules of thumb appear to be widely applicable. If a language L is NP-complete then the analogous dynamic/periodic language is PSPACE complete. If a language L is in the class P, then the analogous dynamic/periodic language is also in the class P. Examples of dynamic/periodic problems soluable in polynomial time may be found in Graves and Orlin [G0] and in Orlin [02], [03], [04], [05] and [06]. Although the above rules of thumb are not true in general, they do appear to be true for many "natural" problems.

The above rules of thumb suggest that the D/P knapsack problem should be soluable in pseudo-polynomial time and should be PSPACE-complete. In fact, both properties are true. Below we will show how to solve the following optimization version of the D/P knapsack problem in pseudo-polynomial time.

$$\begin{aligned}
&\text{Minimize } p^{-1} \sum_{i=1}^p cx^i \\
&\text{Subject to } \begin{aligned}
&ax^i + bx^{i+1} = d \quad \text{for } i = 1, \dots, p-1 \\
&ax^p + bx^1 = d \\
&0 \leq x^i \leq 1, x^i \text{ integer} \\
&p \geq 0 \text{ integer.}
\end{aligned}
\end{aligned}$$

Here, for convenience, we have written a periodic description in which the period length p is a variable.

To solve the problem, we create a directed graph $G(a,b,c,d) = (V,E)$,

where $V = \{0, \dots, d\}$. Moreover, there is an edge (i, j) with cost k in E if there is a 0-1 n -vector x such that $ax = j$, $bx = d-i$, and $cx = k$. The graph may be constructed in $O(nd^2)$ steps using a dynamic programming recursion.

THEOREM 4. There is a feasible solution to the D/P knapsack problem if and only if $G(a, b, c, d)$ has a directed circuit. The optimum solution, if one exists, is induced by the circuit C in G which minimizes the ratio of the cost to the number of edges. (This ratio is called the minimum cycle mean).

PROOF. Let C be a directed circuit in G with edges e_1, \dots, e_p . Let x^1, \dots, x^p be the 0-1 n -vector that induced the edge. Then x^1, \dots, x^p in a periodic solution. Moreover the average cost of the circuit C is $p^{-1}(cx^1 + \dots + cx^p)$, which is the cycle mean of C . It is easy to verify that the correspondence between circuits of G and periodic solutions is 1:1. □

We note that the minimum cycle mean of G may be determined in $O(d_3)$ steps using an algorithm of Karp [K].

7. Real time scheduling

In the dynamic/periodic optimization problems described in Section 2, the schedule (certificate) is the crucial output to obtain. Moreover, even if one had an oracle to solve the language recognition problem, there is no apparent polynomial algorithm to find the certificate. Indeed, the certificate will often be exponentially long.

In most of the scheduling problems of Section 2, the schedule is to run over an infinite horizon. Generally, it is unnecessary or unimportant to obtain the infinite-horizon schedule in polynomial time. It suffices to determine the schedule for the first i periods in time polynomial in the data and in i . We might view such an algorithm as a polynomial real time algorithm.

We will show below that none of the D/P problems of Theorem 1 have polynomial real time algorithms, unless $P = PSPACE$. In fact, we will show that just to determine the certificate for period 1 is PSPACE-hard.

Let $L \in 3P$. The initialization problem for $D(L)$ is as follows.

INPUT: $y \in D(L)$

PROBLEM: Determine x_1 such that there is a periodic certificate $(x^i: i = 1, \dots, p)$.

THEOREM 5. The initialization problem is PSPACE-hard for all of the D/P problems in Theorem 1.

SKETCH OF PROOF. Modify the proof that D/P satisfiability is PSPACE-complete as follows. Starting with an arbitrary language $\mathcal{L} \in \text{PSPACE}$ and an instance x , transform x to an instance x' such that $x' \in \text{D/P Satisfiability}$, and any certificate (y^i) for x' has the following property

- (i) The first symbol of y^i is 0 for all i if $x \in \mathcal{L}$.
- (ii) The first symbol of y^i is 1 for all i if $x \notin \mathcal{L}$.

The details of the transformation are detailed but straightforward. Also, if y^i is part of a valid certificate for x' , one can determine whether $x \in \mathcal{L}$, and thus the initialization problem for D/P satisfiability is PSPACE-hard. The other proofs are analogous.

We note that in our proof that $\text{D/P} = \text{PSPACE}$, all the PSPACE languages $\mathcal{L} = D(L)$ so constructed are polynomially real time solvable.

8. Extensions

Suppose that $L \in 3P$ and $S, T \in P$, and $t \in \mathbb{Z}$. One can consider variations of the language $D(L)$ as follows.

FIXED PERIOD PROBLEM

INPUT: x, t

QUESTION: Is there a periodic certificate $(y^i: i=1, \dots, t)$ for x ?

INITIAL CONDITION PROBLEM

INPUT: x, S

QUESTION: Is there a certificate $(y^i: i \in \mathbb{Z}^+)$ for x such that $y^1 \in S$. (i.e., $x \# y^i \# y^{i+1} \in L$ for $i > 1$ and $y^1 \in S$).

INITIAL AND TERMINAL CONDITION PROBLEM

INPUT: s, S, T

QUESTION: Is there a certificate y^1, \dots, y^t such that $x \oplus y^1 \oplus y^{i+1} \in L$ for $i = 1, \dots, t-1$ and $y^1 \in S$ and $y^t \in T$.

One may also form the latter problem with a fixed period length. As long as the period length is not bounded, all of the PSPACE-completeness Theorems of this paper extend to the above generalizations.

9. Open Problems

In order to prove that the periodic graph problems are in the class PSPACE, we restricted attention to narrow periodic graphs. It is an interesting open question as to whether the non-narrow periodic graph problems are in the class PSPACE. (This seems quite unlikely.) If not, they may form an interesting class of problems that provably require an exponential amount of space. Also, the complexity of the periodic hamiltonian path problem on narrow periodic graphs is of interest. (We know that it is PSPACE-hard.)

On a broader scope, it is of interest to know if there are general properties that an NP-complete problem may have so as to guarantee that the corresponding D/P problem is PSPACE-complete. Currently, we have no meta-theorems concerning these PSPACE-completeness results.

Acknowledgements

I wish to thank Christos Papadimitriou for several helpful discussions. This work was supported in part by the National Science Foundation.

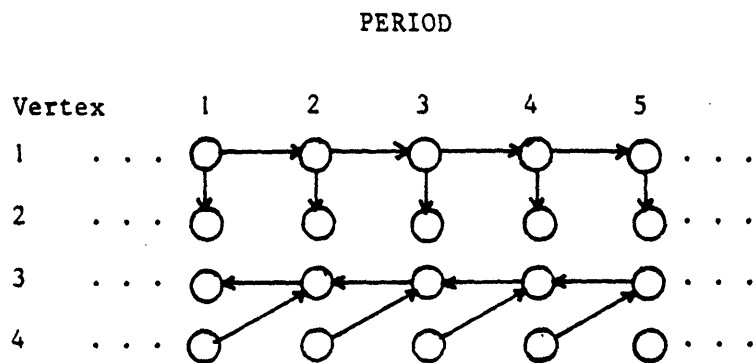


Figure 1. A subgraph of a (narrow) periodic graph.

References

- [B] Baker, K.R., "Workforce Allocation in Cyclical Scheduling Problems", Operational Research Quarterly, 27, (1976), pp.155-167.
- [C] Cook, S.A., "The Complexity of Theorem-Proving Procedures", Proc. 3rd ACM Symposium on The Theory of Computing, New York, (1971), pp.151-158.
- [CS] Chandra, A.K. and L. Stockmeyer, "Alternation", Proceedings of the 17th FOCS, (1976), pp.98-108.
- [D] Dantzig, G.B., W. Blattner, and M.R. Rao, "Finding a Cycle in a Graph with Minimum Cost to Times Ratio with Application to a Ship Routing Problem", in Theory of Graphs, P. Rosentiehl, editor, Dinot, Paris; Gordon and Breach, New York, (1967), pp.77-84.
- [DL] Dhall, S.K., and C.L. Liu, "On a Real-Time Scheduling Problem", Operations Research, 26, (1978), pp.127-140.
- [FF] Ford, L.R., and D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N.S., 1962.
- [GJ] Garey, M.R., and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, 1979.
- [GO] Graves, S.C., and J.B. Orlin, "A Minimum Concave Cost Dynamic Network Flow Problem with an Application to Lot-Sizing", Networks 15, (1985), 59-71.
- [K] Karp, R.M., "Characterization of the Minimum Cycle Mean in a Digraph", Discrete Mathematics 23, (1978), pp.309-311.
- [L] Lawler, E.L., "Optimal Cycles in Doubly Weighted Linear Graphs", in Theory of Graphs, P. Rosentiehl, editor, Dinod, Paris; Gordon and Breach, New York, (1967), pp.209-214.
- [La] Labetoulle, J. "Some Theorems on Real-Time Scheduling", Computer Architecture and Networks, E. Gelenbe and R. Mahl editors, North Holland Publishing Company, (1974), pp. 285-298.
- [LL] Liu, C.L., and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment", JACM, 20, (1973), pp.46-61.
- [LM] Lawler, E.L. and C.U. Martel, "Scheduling Periodically Occurring Tasks on Multiple Processors, submitted for publication.
- [M] Megiddo, N., "Combinatorial Optimization with Rational Objective Functions", in Proceedings of the 10th ACM Symposium on Computation, San Diego, California, (1978), pp.1-12.

- [Ma] Periodic Optimization Vol. 2, A. Marzollo editor, CISM Courses and Lectures 135, Springer Verlag, 1972.
- [Mo] Mok, A., "Complexity of Static Schedule Design for Hard Real-time Environments", Working Paper, Laboratory for Computer Science, MIT, 1980.
- [O1] Orlin, J.B., "The Complexity of Dynamic Languages and Dynamic Optimization Problems", Transactions of the 13th Annual ACM Symposium on the Theory of Computing, May 1981, pp.218-227.
- [O2] Orlin, J.B., "Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: an Application of Periodic Posets", Operations Research 30, (1982), pp.768-776.
- [O3] Orlin, J.B., "Maximum Throughput Dynamic Network Flows", Mathematical Programming 27, (1983), pp.214-223.
- [O4] Orlin, J.B., "Dynamic Matchings and Quasi-dynamic Fractions Matchings I and II", Networks 13, (1983), pp. 551-580.
- [O5] Orlin, J.B., "Minimum Convex Cost Dynamic Network Flows", Mathematics of Operations Research 9, (1984), pp.190-207.
- [O6] Orlin, J.B., "Algorithms for Problems on Dynamic/Periodic Graphs", in Progress in Combinatorial Optimization, William Pulleyblank, ed., Academic Press, Toronto, 1984.
- [P] Papadimitriou, C.A., "Games Against Nature", Proceedings of the 24th FOCS, (1983), pp.446-450.
- [PS] Papadimitriou, C.A. and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, 1981.
- [R] Reif, J.H., "Complexity of the Movers Problem and Generalizations", Proceedings of the 20th FOCS, (1979), pp.421-427.
- [Sa] Savitch, W.J., "Relationship Between Non-Deterministic and Deterministic Tape Complexities", J. Comput. Systems Science 4, (1970), pp.177-192.
- [Sc] Schaefer, T., "The Complexity of Some Two-Person Perfect Information Games", J. Comput. Systems Science, 16, (1978), pp. 185-225.
- [Si] Simpson, R.W., "Scheduling and Routing Models for Airline Systems", Technical Report 1268-3, Flight Transportation Laboratory, MIT, 1969.