MIT/LCS/TM-145

A SPACE BOUND FOR ONE-TAPE MULTIDIMENSIONAL TURING MACHINES

Michael C. Loui

November 1979

A Space Bound for One-Tape Multidimensional Turing Machines

Michael C. Loui

November 1979

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract. Let L be a language recognized by a nondeterministic Turing machine with one d-dimensional worktape of time complexity $T(n)$. Then L can be recognized by a deterministic Turing machine of space complexity $(T(n) \log T(n))^{d/d+1}$. The proof employs a generalized crossing sequence argument.

Key Words: multidimensional Turing machine, nondeterminism, time-space tradeoff, time complexity, tape complexity, computational complexity.

This report was prepared with the support of the National Science Foundation Grant No. MCS-77-19754 and the Fannie and John Hertz Foundation.

1. Introduction

Time and space are fundamental resources for computation. It is generally believed that these resources can be exchanged for each other. For instance, a program that saves space (storage) by compressing data spends extra time encoding the data and decoding the stored representation. Some data structures use minimum space, but require long access times; others reduce access times by occupying large amounts of memory. Quantitative tradeoffs have been established between time and space for multitape Turing machines [5] and for straight-line programs [10], [12], [14].

Recently, Paul and Reischuk [9] proved that the tradeoff of [5] is not an artifact of the linearity of the Turing machine tapes: every deterministic multitape multidimensional Turing machine of time complexity $T(n)$ can be simulated by a deterministic Turing machine of space complexity $T(n) \log \log T(n)/\log T(n)$. We obtain a space bound for a restricted class of multidimensional Turing machines: for every nondeterministic machine M with one d-dimensional worktape that runs in time $T(n)$, there is a deterministic Turing machine M' such that M' accepts the same language as M in space $(T(n) \log T(n))^{d/d+1}$, provided that $T(n)$ is constructible in space $(T(n) \log T(n))^{d/d+1}$.

Previous studies of multidimensional Turing machines have concerned only their time complexity. Let $DTIME^d(T(n))$ denote the class of languages recognized by deterministic multitape d-dimensional Turing machines of time complexity $T(n)$ on inputs of length n. For all positive integers d and e,

(i) $DTIME^{d}(T(n)) \subseteq DTIME^{1}(T(n)^{2 - 1/d})$ [11];

(ii) $DTIME^{d+1}(T(n)) \subseteq DTIME^{d}(T(n)^{1 + 1/d})$ [2];

(iii) $DTIME^{e}(T(n)) \subseteq DTIME^{d+1}(T(n)^{1 + 1/d})$ [2];

(iv) $DTIME^{e}(T(n)) \not\subseteq DTIME^{d}(T(n)^{1 - 1/e + 1/d - \epsilon})$ for all $\epsilon > 0$ [2], [4];

Furthermore, every d-dimensional Turing machine with multiple heads on some of its tapes can be simulated in real time by a d-dimensional Turing machine having more tapes [7]. Every multitape multidimensional Turing machine can be simulated in real time by a storage modification machine [13]. Every d-dimensional Turing machine of time complexity $T(n) \geq n \log n$ can be simulated by a unit-cost random access machine in time $O(T(n)/(\log T(n))^{1/d})$ [1].

A multidimensional Turing machine that uses space $S(n)$ can be simulated by a conventional Turing machine in space $S(n)$. (This fact is implicit in [3].) Thus, the space measure is the same for all dimensions.

Section 2 introduces definitions, including a generalization of crossing sequences. Section 3 describes a deterministic simulation of a nondeterministic machine M with one d-dimensional worktape, and Section 4 proves that this simulation uses space $(T(n) \log T(n))^{d/d+1}$ when M runs in time $T(n)$. (All logarithms in this paper are taken to base 2.) A familiarity with [8] will be helpful, but not necessary.

2. Definitions

Fix a finite alphabet Σ and a positive integer d. A worktape over Σ is a set of cells, each of which can contain a symbol in Σ. A worktape is d-dimensional if its cells are in bijective correspondence with $\mathbb{Z}^d$, the set of d-tuples of integers. For every $\mathbf{x}$ in $\mathbb{Z}^d$ there is a unique worktape cell $C(\mathbf{x})$ at location $\mathbf{x}$. In $\mathbb{Z}^d$ let

$$\mathbf{e}_0 =_{\mathrm{df}} (0, 0, 0, \ldots, 0), \qquad \mathbf{e}_1 =_{\mathrm{df}} (1, 0, 0, \ldots, 0),$$

$$G_i(a) =_{\mathrm{df}} \{(x_1, \ldots, x_d): x_i \geq a\}, \qquad L_i(a) =_{\mathrm{df}} \{(x_1, \ldots, x_d): x_i < a\}.$$

(We use $=_{\mathrm{df}}$ for equality by definition.) A box is a subset of $\mathbb{Z}^d$ comprising the d-tuples

$$[a_1,b_1] \times [a_2,b_2] \times \ldots \times [a_d,b_d]$$

for some integers $a_1, b_1, \ldots, a_d, b_d$. The volume of a box B, denoted $|B|$, is the number of d-tuples that it comprises. A content function for a box B is a map from B to Σ; such a function specifies the contents of cells whose locations are in B. If φ is a content function for B_1 and B_2 is a subset of B_1, then the restriction of φ to B_2 is written $\varphi \backslash B_2$.

A (one-tape) d-dimensional Turing machine (with alphabet Σ) has a d-dimensional worktape on which the head can move one cell along any of the d orthogonal dimensions in either positive or negative direction at each step; if the head reads cell $C(x_1, x_2, \ldots, x_d)$ at step s, then at step $s + 1$ it reads cell $C(x_1, x_2, \ldots, x_d)$, $C(x_1 \pm 1, x_2, \ldots, x_d)$, $C(x_1, x_2 \pm 1, \ldots, x_d)$, or $C(x_1, x_2, \ldots, x_d \pm 1)$. In each cell the machine can write a symbol from Σ. The input to the machine is presented on a two-way read-only input tape. Initially, at step 0, the worktape is completely blank, the input head is positioned on the leftmost symbol of the input word, and the worktape head reads cell $C(\mathbf{e}_0)$. The machine accepts an input word by entering a designated accepting state and halting.

Let M be a one-tape nondeterministic d-dimensional Turing machine that runs in time $T(n)$ on inputs of length n: for every word of length n that M accepts there is an accepting computation of at most $T(n)$ steps. Assume that M reads all of its input -- $T(n) \geq n$ -- and that $T(n)$ is constructible in space $(T(n) \log T(n))^{d/d+1}$.

We may assume without loss of generality that M halts with its worktape blank and its worktape head positioned on $C(\mathbf{e}_0)$. Furthermore, the worktape head, which starts on $C(\mathbf{e}_0)$, moves immediately to $C(e_1)$ in state $\hat{q}_0$ and remains on cells whose locations are in the box

$$B_0(n) =_{df} [1,T(n)] \times [0,T(n)] \times \dots \times [0,T(n)]$$

until the final step. To accept the input word, M moves its worktape head from $C(e_1)$ to $C(e_0)$ in state $\hat{q}_1$ while the head on the input tape scans the leftmost symbol of the input word (at tape position 0). For the remainder of this paper we consider the computation(s) of M on a fixed input word of length n.

A <u>partial configuration</u> of M is a triple

$$\pi = (q, x, p)$$

where q is a state, x is a worktape cell location, and p is a position on the input tape. If M is in state q with the worktape head reading cell $C(x)$ and the input head at position p at some step during some computation, then it is in partial configuration π. Let

$$\hat{\pi}_0 =_{df} (\hat{q}_0, e_1, 0);$$

at step 1 this is the partial configuration. The <u>restriction</u> of a partial configuration π to a box B, written $\pi \backslash B$, is π if the worktape cell location of π is in B, null ($\emptyset$) otherwise.

Crossing sequences [6] have been employed to study computations of conventional one-dimensional Turing machines. We use a more general notion.

A <u>crossing event</u> from box B_1 to box B_2 occurs at step s if the worktape head moves from a cell $C(x_1)$ at location x_1 in B_1 to a cell $C(x_2)$ at location x_2 in B_2 at the end of step s. This crossing event <u>exits</u> B_1 and <u>enters</u> B_2. The <u>crossing record</u> for a crossing event is a 5-tuple

$$(q, x_1, x_2, s, p),$$

where q is the state of the machine as it moves from cell $C(x_1)$ to cell $C(x_2)$ at the end of step s with the input head at position p. Call s the <u>event time</u> of the crossing record. This record <u>exits</u> B_1 and <u>enters</u> B_2. Let R be a set of crossing records. The <u>earliest</u> record of R has the smallest event time;

the latest record has the largest event time. The restriction of R to a box B, written $R\backslash B$, comprises precisely the records of R for which $x_1 \in B$ or $x_2 \in B$; the restriction of R to an interval of steps $[s_1,s_2]$, written $R\backslash[s_1,s_2]$, is the subset of records for which $s \in [s_1,s_2]$. Define the predicate **Between** (R, B_1, B_2) to be true if and only if R is a set of crossing records for crossing events between B_1 and B_2 – each event is either from B_1 to B_2 or from B_2 to B_1.

We also generalize the computation diagrams of [8]. Let B be a box and φ_1, φ_2 be content functions for B. Let π_1 and π_2 be partial configurations such that for $i = 1,2$, either the worktape cell location of π_i is in B or π_i is null. Let $[s_1,s_2]$ be a nonempty interval of steps and R be a set of crossing records that enter or exit B. The 8-tuple $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ is compatible if assertions (i) below together imply that assertions (ii) hold for some computation of M:

(i) (a) the contents of B at step s_1 are specified by φ_1;

(b) if the worktape cell location of π_1 is in B, then the partial configuration of M at step s_1 is π_1; otherwise, if π_1 is null, then at step s_1 the worktape head does not read a cell whose location is in B;

(c) the set of records of R that en[illegible] the set of crossing records for crossing events that enter B during $[s_1,s_2]$;

(ii) (a) the contents of B at step $s_2 + 1$ are specified by φ_2;

(b) if the worktape cell location of π_2 is in B, then the partial configuration of M at step $s_2 + 1$ is π_2; otherwise, if π_2 is null, then at step $s_2 + 1$ the worktape head does not read a cell whose location is in B; and

(c) the set of records of R that exit B is the set of crossing records for crossing events that exit B during $[s_1,s_2]$;

The predicate **Comp** $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ is true if and only if $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ is compatible. Define a special trivial case of Comp:

$$\text{Triv}\,(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2) =_{\text{df}} \text{Comp}\,(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2) \textit{ and } |B| = 1.$$

Let B be a box and R be a set of crossing records that enter or exit B. Let π_1 and π_2 be partial configurations such that for $i = 1,2$, either the worktape cell location of π_i is in B or π_i is null ($\emptyset$). The quadruple (B, π_1, π_2, R) is consistent if either (i) or (ii) holds:

(i) $R = \emptyset$ and either $\pi_1 = \pi_2 = \emptyset$ or the worktape cell location of π_1 and the worktape cell location of π_2 are both in B;

(ii) (a) $R \neq \emptyset$;

(b) the records R, when ordered by event time, alternate between records that enter B and records that exit B;

(c) if the worktape cell location of π_1 is in B, then the earliest record in R exits B; if π_1 is null, then the earliest record in R enters B; and

(d) if the worktape cell location of π_2 is in B, then the latest record in R enters B; if π_2 is null, then the latest record in R exits B.

When $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ is compatible, (B, π_1, π_2, R) is necessarily consistent. The predicate Cons (B, π_1, π_2, R) is true if and only if (B, π_1, π_2, R) is consistent.

For each box B let β_B be the content function that assigns a blank to every location in B. Define a predicate for a box B and a set of crossing records R:

$$\text{Bridge}\,(B, R) =_{\text{df}} \text{Comp}\,(B, \hat{\pi}_0 \backslash B, \beta_B, \emptyset, \beta_B, R, 1, T(n)).$$

For $1 \leq \tau \leq T(n)$ put

$$R_\tau =_{\text{df}} \{(\hat{q}_1, e_1, e_0, \tau, 0)\}.$$

Machine M accepts the input word if and only if Bridge $(B_0(n), R_\tau)$ is true for some τ.

3. Simulation

To determine whether M accepts its input word, deterministic Turing machine M' will check whether **Bridge** $(B_0(n), R_\tau)$ is true by repeatedly partitioning the box $B_0(n)$ and the time interval $[1,T]$. Predicates **Bridge** and **Comp** evidently satisfy the following two properties, which justify this recursive strategy.

Bridge (B, R) *if and only if* **Comp** $(B, \hat{\pi}_0\backslash B, \beta_B, \varnothing, \beta_B, R, 1, T)$ *or*

[**Cons** $(B, \hat{\pi}_0\backslash B, \varnothing, R)$ *and*

$(\exists\, i,a,R',B_1,B_2)$ $(B_1 = B \cap G_i(a)$ *and* $B_2 = B \cap L_i(a)$ *and*

Between (R', B_1, B_2) *and*

Bridge $(B_1, (R \cup R')\backslash B_1)$ *and*

Bridge $(B_2, (R \cup R')\backslash B_2))]$.

Comp $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ *if and only if* **Cons** (B, π_1, π_2, R) *and*

[**Triv** $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$ *or*

$(\exists\, s',\pi',\varphi')$ (**Comp** $(B, \pi_1, \varphi_1, \pi'\backslash B, \varphi', R\backslash[s_1,s'], s_1, s')$ *and*

Comp $(B, \pi'\backslash B, \varphi', \pi_2, \varphi_2, R\backslash[s'+1,s_2], s'+1, s_2))$ *or*

$(\exists\, i,a,R',B_1,B_2)$ $(B_1 = B \cap G_i(a)$ *and* $B_2 = B \cap L_i(a)$ *and*

Between (R', B_1, B_2) *and*

Comp $(B_1, \pi_1\backslash B_1, \varphi_1\backslash B_1, \pi_2\backslash B_1, \varphi_2\backslash B_1, (R \cup R')\backslash B_1, s_1, s_2)$ *and*

Comp $(B_2, \pi_1\backslash B_2, \varphi_1\backslash B_2, \pi_2\backslash B_2, \varphi_2\backslash B_2, (R \cup R')\backslash B_2, s_1, s_2))]$.

Lemma A guarantees that for each box, there is some partition into two boxes that induces a small set of crossing records. To simplify our arguments, we neglect to distinguish between ρ, $\lfloor\rho\rfloor$, and $\lceil\rho\rceil$ for real numbers ρ; one can justify this simplification routinely.

Lemma A. Let B be a box with volume $v = |B|$. Let $s_2 \geq s_1$ be steps and $t = s_2 - s_1 + 1$. There is a coordinate i and an integer a such that

(i) the number of crossing events between $B \cap G_i(a)$ and $B \cap L_i(a)$ during $[s_1,s_2]$ is at most $3t/v^{1/d}$, and

(ii) the boxes $B \cap G_i(a)$ and $B \cap L_i(a)$ have volumes between $v/3$ and $2v/3$.

Proof. Let $B = [a_1,b_1] \times ... \times [a_d,b_d]$. Identify the longest side of B: select i for which $b_i - a_i$ is largest. Then $b_i - a_i \geq v^{1/d}$. Among the $(b_i - a_i)/3 + 1$ integers in

$$[a_i + (b_i - a_i)/3, a_i + 2(b_i - a_i)/3]$$

there is some a for which the number of crossing events between $B \cap G_i(a)$ and $B \cap L_i(a)$ during $[s_1,s_2]$ is at most $(s_2 - s_1)/((b_i - a_i)/3 + 1) \leq 3t/v^{1/d}$. $\square$

We describe the simulating machine M' informally. Fix $\delta =_{\text{df}} d/(d+1)$ and

$$c_1 =_{\text{df}} 2 \times 18^{\delta} \tag{1}$$

MAIN PROGRAM FOR M'

For $\tau = 1, ..., T(n)$ calculate BRIDGE $(B_0(n),R_\tau)$. If BRIDGE $(B_0(n),R_\tau)$ is true for some τ, then accept the input word. Otherwise, reject the input word.

Subroutine BRIDGE (B, R)

Inputs: box B, set of crossing records R that enter or exit B.

Output: the value of Bridge (B,R).

Procedure: Let $v = |B|$.

Case 1: $v \leq 3(T \log T)^{\delta}$. Return the value of COMP $(B, \pi_0\backslash B, \beta_B, \emptyset, \beta_B, R, 1, T)$.

Case 2: $v > 3(T \log T)^{\delta}$. If $(B, \pi_0\backslash B, \emptyset, R)$ is not consistent, then return *false*. Iterating through all partitions of B into two boxes $B_1 = B \cap G_i(a)$ and $B_2 = B \cap L_i(a)$ such that $v/3 \leq |B_1| \leq 2v/3$ and all sets R' of at most $3T/v^{1/d}$ crossing records for crossing events between B_1 and B_2, search for B_1, B_2, and R' for which BRIDGE (B_1, $(R \cup R')\backslash B_1$) and BRIDGE ($B_2$, $(R \cup R')\backslash B_2$) are true. If suitable B_1, B_2, and R' are found, then return *true*; otherwise, return *false*.

Subroutine COMP ($B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2$)

Inputs: box B, partial configurations π_1 and π_2, content functions φ_1 and φ_2 on B, set of crossing records R, steps s_1 and s_2.

Output: the value of Comp ($B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2$).

Assumptions: the records in R either enter or exit B; for $i = 1,2$, either the worktape cell location of π_i is in B or π_i is null.

Procedure: Let $v = |B|$, $r = |R|$, and $t = s_2 - s_1 + 1$. Verify that (B, π_1, π_2, R) is consistent; if it is not, then return *false*. If $v = 1$, then return *true* if ($B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2$) is compatible on the one cell whose location is in B, *false* if not.

Case 1: $v \leq (r + 1) \log T$ and $r \geq 1$. Determine a step s' at which $|R\backslash[s_1,s']| = r/2$. Enumerating all content functions φ' on B and partial configurations π', search for φ' and π' such that COMP ($B, \pi_1, \varphi_1, \pi'\backslash B, \varphi', R\backslash[s_1,s'], s_1, s'$) and COMP ($B, \pi'\backslash B, \varphi', \pi_2, \varphi_2, R\backslash[s'+1,s_2], s'+1, s_2$) are true. Return *true* if appropriate φ' and π' are found, *false* if not.

Case 2: Either $v \leq \log T$ and $r = 0$ or $v + (r + 1) \log T \leq c_2(t \log T)^{\delta}$. Set $s' = (s_1 + s_2)/2$. As in Case 1, search for φ' and π' such that COMP ($B, \pi_1, \varphi_1, \pi', \varphi', R\backslash[s_1,s'], s_1, s'$) and COMP ($B, \pi', \varphi', \pi_2, \varphi_2, R\backslash[s'+1,s_2], s'+1, s_2$) are true.

Case 3: $v \geq (r + 1) \log T$ and $v + (r + 1) \log T \geq c_1(t \log T)^{\delta}$. Enumerating all partitions of B into two boxes $B_1 = B \cap G_i(a)$ and $B_2 = B \cap L_i(a)$ such that $v/3 \leq |B| \leq 2v/3$ and al. .ets R' of at most $3t/v^{1/d}$ crossing records for crossing events between B_1 and B_2, search for B_1, B_2, and R' such that both COMP $(B_1, \pi_1\backslash B_1, \varphi_1\backslash B_1, \pi_2\backslash B_1, \varphi_2\backslash B_1, (R \cup R')\backslash B_1, s_1, s_2)$ and COMP $(B_2, \pi_1\backslash B_2, \varphi_1\backslash B_2, \pi_2\backslash B_2, \varphi_2\backslash B_2, (R \cup R')\backslash B_2, s_1, s_2)$ are true. If suitable B_1, B_2, and R' are found, then return *true*; otherwise, return *false*.

4. Analysis of the Simulation

We show that M' uses space $O((T(n) \log T(n))^{\delta})$. The amount of space used by subroutines COMP and BRIDGE is dominated by the storage required for the actual parameters in their subroutine calls.

Since every location of the d-dimensional worktape can be specified by a list of d integers written in binary, each box B in $B_0(n)$ can be specified in space $O(\log T(n))$. Each partial configuration π and each crossing record can be stored in space $O(\log T(n))$. Thus, a set R of r crossing records can be stored in space linearly proportional to $r \log T(n)$. A content function φ on a box of volume v requires space proportional to v to store. Let k be a constant so large that M' needs at most

$k(r + 1) \log T(n)$ space to specify a pair (B, R),

$k(v + (r + 1) \log T(n))$ space to specify an 8-tuple $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$.

Choose constants c_2, c_3, c_4, c_5 (depending only on d and k) such that

$$c_2 \geq 20k, \tag{2}$$

$$c_2 \geq c_2/2^{\delta} + 2kc_1, \tag{3}$$

$$c_4 \geq (k + c_2)(c_3 + 3) + c_2 + c_5, \tag{4}$$

$$c_3 \geq (2/3)^{1/d}(c_3 + 3), \tag{5}$$

$$c_5 \le (3/2)^{1/d} c_5 - 2k(c_3 + 3). \tag{6}$$

Lemma B. On input $(B, \pi_1, \varphi_1, \pi_2, \varphi_2, R, s_1, s_2)$, subroutine COMP uses at most space $c_2\Psi(|B|, |R|, s_2 - s_1 + 1)$, where

$$\Psi(v, r, t) =_{\mathrm{df}} v + (r + \log t) \log T + (t \log T)^{\delta}.$$

Proof. We proceed by induction on the depth of recursion in COMP. Set $v = |B|$, $r = |R|$, $t = s_2 - s_1 + 1$. If $v = 1$, then COMP uses no extra space. Otherwise, there are three cases.

Case 1: $v \le (r + 1) \log T$ and $r \ge 1$. The space required to store the arguments for the two recursive calls is $2k(v + (r/2 + 1) \log T)$. In this case, COMP uses space

$$\begin{aligned} &2k(v + (r/2 + 1) \log T) + c_2\Psi(v, r/2, t) \\ &\quad \le c_2 v + ((3k + c_2/2)r + 4k) \log T + c_2 \log t \log T + c_2(t \log T)^{\delta} \\ &\quad \le c_2 v + ((13c_2/20)r + 5c_2/20) \log T + c_2 \log t \log T + c_2(t \log T)^{\delta} \\ &\quad \le c_2\Psi(v, r, t) \end{aligned}$$

because c_2 satisfies (2) and $r \ge 1$.

Case 2: Either $v \le \log T$ and $r = 0$ or $v + (r + 1) \log T \le c_1(t \log T)^{\delta}$. If $v \le \log T$ and $r = 0$, then COMP uses space

$$\begin{aligned} &2k(v + \log T) + c_2\Psi(v, 0, t/2) \\ &\quad \le c_2 v + 4k \log T + c_2 \log (t/2) \log T + c_2(t \log T)^{\delta} \\ &\quad \le c_2\Psi(v, r, t) \end{aligned}$$

by (2). If $v + (r + 1) \log T \le c_1(t \log T)^{\delta}$, then COMP uses space

$$\begin{aligned} &2k(v + (r + 1) \log T) + c_2\Psi(v, r, t/2) \\ &\quad \le c_2 v + c_2(r + \log (t/2)) \log T + (2kc_1 + c_2/2^{\delta})(t \log T)^{\delta} \\ &\quad \le c_2\Psi(v, r, t) \end{aligned}$$

because of (3).

Case 3: $v \geq (r+1)\log T$ and $v + (r+1)\log T \geq c_1(t \log T)^{\delta}$. In this case,

$$c_1(t \log T)^{\delta} \leq 2v,$$

hence since $\delta = d/(d+1)$,

$$(t \log T)/v^{1/d} \leq (2/c_1)^{1/\delta} v. \tag{7}$$

Boxes B_1 and B_2 of volumes $v_1 = |B_1|$ and $v_2 = |B_2|$ are defined, where $v_1 + v_2 = v$ and max $\{v_1, v_2\} \leq 2v/3$. Furthermore, at most $3t/v^{1/d}$ new crossing records are introduced. The space used by COMP is at most

$$\begin{aligned}
&k(v_1 + (r + 3t/v^{1/d} + 1)\log T) + k(v_2 + (r + 3t/v^{1/d} + 1)\log T) + c_2\Psi(2v/3,\ r + 3t/v^{1/d},\ t) \\
&\leq kv + 2k(r+1)\log T + 6k(t \log T)/v^{1/d} + 2c_2 v/3 + c_2(r + \log t)\log T \\
&\qquad + 3c_2(t \log T)/v^{1/d} + c_2(t \log T)^{\delta} \\
&\leq (k + 2k + 6k(2/c_1)^{1/\delta} + 2c_2/3 + 3c_2(2/c_1)^{1/\delta})v + c_2(r + \log t)\log T + c_2(t \log T)^{\delta} \\
&\leq c_2\Psi(v, r, t)
\end{aligned}$$

by (7), (1), and (2). □

Lemma C. Let B be a box with volume $v = |B|$. Let R be a set of r crossing records that either enter or exit B. If

$$(T \log T)^{\delta} \leq v, \tag{8}$$

$$r + 1 \leq c_3 T/v^{1/d}, \tag{9}$$

then on input (B, R) subroutine BRIDGE uses at most space

$$\Phi(v) =_{\mathrm{df}} c_4(T \log T)^{\delta} + c_2(\log T)^2 - c_5(T \log T)/v^{1/d}.$$

Proof. We proceed by induction on the depth of recursion. There are two cases.

<u>Case 1</u>: $v \leq 3(T \log T)^{\delta}$. The space for the arguments passed to COMP is $k(v + (r + 1) \log T)$. According to (8) and (9),

$$(r + 1) \log T \leq c_3(T \log T)/(T \log T)^{\delta/d} = c_3(T \log T)^{\delta}.$$

Thus, Lemma B and (4) imply that BRIDGE uses space

$$\begin{aligned} & k(v + (r + 1) \log T) + c_2\Psi(v, r, T) \\ & \quad \leq (k + c_2)v + (k + c_2)(r + 1) \log T + c_2(\log T)^2 + c_2(T \log T)^{\delta} \\ & \quad \leq (3(k + c_2) + (k + c_2)c_3 + c_2)(T \log T)^{\delta} + c_2(\log T)^2 \\ & \quad \leq \Phi(v). \end{aligned}$$

<u>Case 2</u>: $v > 3(T \log T)^{\delta}$. Boxes B_1 and B_2 of volumes $v_1 = |B_1|$ and $v_2 = |B_2|$ are defined. Let r_1 and r_2 be the numbers of crossing records passed to the corresponding recursive calls of BRIDGE. By definition, (9), and (5), for $i = 1,2$,

$$(T \log T)^{\delta} < v/3 \leq v_i \leq 2v/3,$$

$$r_i + 1 \leq r + 1 + 3T/v^{1/d} \leq (c_3 + 3)T/v^{1/d} \leq (2/3)^{1/d}(c_3 + 3)T/v_i^{1/d} \leq c_3T/v_i^{1/d}.$$

Thus, by induction, BRIDGE uses at most space

$$\begin{aligned} & k(r_1 + 1) \log T + k(r_2 + 1) \log T + \max \{\Phi(v_1), \Phi(v_2)\} \\ & \quad \leq k(r_1 + 1 + r_2 + 1) \log T + c_4(T \log T)^{\delta} + c_2(\log T)^2 - c_5(T \log T)/(2v/3)^{1/d} \\ & \quad \leq c_4(T \log T)^{\delta} + c_2(\log T)^2 + (2k(c_3 + 3) - (3/2)^{1/d}c_5)(T \log T)/v^{1/d} \\ & \quad \leq \Phi(v) \end{aligned}$$

because c_5 satisfies (6). □

<u>Theorem</u>. For all $T(n) \geq n$ that can be constructed in space $(T(n) \log T(n))^{d/d+1}$, a nondeterministic machine M with one d-dimensional worktape that runs in time $T(n)$ can be simulated by a deterministic Turing machine in space $(T(n) \log T(n))^{d/d+1}$.

Proof. Section 3 presented a method for simulating M by a deterministic machine M'. The main program for M' calls BRIDGE with actual parameters $B_0(n)$, R_τ. Since $|R_\tau| = 1$ and $|B_0(n)| \geq T(n)^d \geq (T(n) \log T(n))^\delta$, Lemma C implies that M' uses space $O((T(n) \log T(n))^\delta)$. Apply constant-factor tape reduction to decrease the space to $(T(n) \log T(n))^\delta$. □

Acknowledgment. Andrea LaPaugh suggested substantial improvements to an earlier draft of this paper.

References

[1] P.W. Dymond, "Complexity relationships among some models of computation." Tech Rep. 113, Dept. Comp. Sci., Univ. Toronto, Dec. 1977.

[2] D.Ju. Grigor'ev, "Imbedding theorems for Turing machines of different dimensions and Kolmogorov's algorithms." *Soviet Math. Dokl.* 18 (1977) 588-592.

[3] J. Hartmanis and R.E. Stearns, "On the computational complexity of algorithms," *Trans. Amer. Math. Soc.* 117 (1965) 285-306.

[4] F.C. Hennie, "On-line Turing machine computations." *IEEE Trans. Elec. Comp.* EC-15 (1966) 35-44.

[5] J. Hopcroft, W. Paul, and L. Valiant, "On time versus space." *J. ACM* 24 (1977) 332-337.

[6] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata.* Addison-Wesley, Reading, Mass., 1969.

[7] B. Leong and J. Seiferas, "New real-time simulations of multihead tape units." *Proc. 9th Ann. ACM Symp. on Theory of Computing*, 1977, pp. 239-248.

[8] M.S. Paterson, "Tape bounds for time-bounded Turing machines." *J. Comp. Sys. Sci.* 6 (1972) 116-124.

[9] W.J. Paul and R. Reischuk, "On time versus space II." Fakultät für Mathematik, Univ. Bielefeld, W. Germany, April 1979.

[10] N. Pippenger, "A time-space trade-off." *J. ACM* 25 (1978) 509-515.

[11] N. Pippenger and M.J. Fischer, "Relations among complexity measures." *J. ACM* 26 (1979) 361-381.

[12] J.E. Savage and S. Swamy, "Space-time trade-offs on the FFT algorithm." *IEEE Trans. Info. Th.* IT-24 (1978) 563-568.

[13] A. Schönhage, "Real-time simulation of multidimensional Turing machines by storage modification machines." Tech. Memo. 37, Project MAC, Mass. Inst. Tech., Dec. 1973.

[14] M. Tompa, "Time-space tradeoffs for computing functions, using connectivity properties of their circuits." *Proc. 10th Ann. ACM Symp. on Theory of Computing*, 1978, pp. 196-204.