NP-completeness of Dynamic Remapping

Ulrich Kremer

CRPC-TR93330
July 1993

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

NP-completeness of Dynamic Remapping *

D NEWSLETTER # 8

Ulrich Kremer

e-mail: kremer@cs.rice.edu

July 30, 1993

Department of Computer Science
Rice University
P.O. Box 1892
Houston, Texas 77251

1 Problem Statement

The data layout problem is formulated as an optimization problem over the phase control flow graph. We assume that the input program has a linear phase control flow graph.

Let V denote the set of variables in the program, $V = \{v_1, ...v_r\}$. The linear phase control flow graph has n nodes, $P_1, ... P_n$, one node for each phase. Let p_i denote the variables referenced in the i-th phase P_i, i.e. $p_i \subseteq 2^V, 1 \leq i \leq n$. For each p_i there is a set of candidate data layouts $D_i = \{d_i^1, ... d_i^m\}$. A single candidate data layout $d_i^k = \{d_{ij_1}^k, ...d_{ij_q}^k\}, 1 \leq k \leq m$, is a set of layouts, one layout for each variable $v \in p_i = \{v_{j_1}, ...v_{j_q}\}$.

The cost of executing phase P_i under the data layout $d_i^k \in D_i$ is denoted by $c(P_i, d_i^k)$. The remapping cost from one data layout scheme to another can be defined based on the remapping costs of each individual variable common to both schemes. Let d_α and d_β be two candidate data layouts for phase P_α and phase P_β, respectively. The remapping cost is given below:

$$c(d_\alpha, d_\beta) = \sum_{v_i \in p_\alpha \cap p_\beta} c(d_{\alpha i}, d_{\beta i}),$$

where $c(d_{\alpha i}, d_{\beta i})$ is the cost for remapping the single variable v_i.

*This research was supported by the Center for Research on Parallel Computation (CRPC), a Science and Technology Center funded by NSF through Cooperative Agreement Number CCR-9120008. This work was also sponsored by DARPA under contract #DABT63-92-C-0038, and the IBM corporation. The content of this paper does not necessarily reflect the position or the policy of the U.S. Government and no official endorsement should be inferred.

Let $f_i : p_i \rightarrow \{1, ...n\}$ be a mapping that determines for each variable $v \in p_i$ the phase that most recently referenced v. If no such phase exists, then $f_i(v)$ has the value i. A data remapping of v may occur between phase $p_{f_i(v)}$ and phase p_i.

Definition 1 *An instance of the dynamic data layout problem consists of a linear phase control flow graph with n phases, a set of program variables $V = \{v_1, ...v_r\}$, sets p_i and D_i for each phase P_i, and cost functions $c(P_i, d_i)$, $d_i \in D_i$, and $c(d_{ij}, d_{f_i(v_j)j})$ for each $v_j \in p_i$ and $d_i \in D_i$, with $1 \leq i \leq n$ and $1 \leq j \leq r$.*

Definition 2 *A solution of an instance of the dynamic data layout problem is a sequence $d_1, d_2, ...d_n$ of data layout schemes $d_i \in D_i, 1 \leq i \leq n$, such that*

$$\sum_{i=1}^{n} c(P_i, d_i) \quad + \quad \sum_{i=1}^{n} \sum_{v_j \in p_i} c(d_{ij}, d_{f_i(v_j)j})$$

is minimized, where $c(d_{ij}, d_{f_i(v_j)j})$ is 0, if $i = f_i(v_j)$, i.e. an initial data layout is not associated with any cost.

Definition 2 is an optimization problem. We will show in Section 2 that the related decision problem DYN-REMAP(k) is NP-complete.

Definition 3 *DYN-REMAP(k) represents a decision problem defined as follows:*

DYN-REMAP(k) := set of all instances of the dynamic data layout problem such that there exists a sequence of data layouts $d_1, d_2, ...d_n$, $d_i \in D_i, 1 \leq i \leq n$, with a cost less or equal to k, where k is a non-negative integer.

2 NP-completeness Proof

Definition 4 *An instance of the 3 Conjunctive Normal Form Satisfiability Problem consists of a boolean expression B in conjunctive normal form,*

$$B = \bigwedge_{i=1}^{t} F_i,$$

where $F_i = l_i^1 \vee l_i^2 \vee l_i^3$, $1 \leq i \leq t$, and each literal is a variable or its negation in the set of variables $V = \{v_1, ...v_r\}$.
The decision problem 3SAT is represented as follows:

3SAT := set of all instances of the 3 Conjunctive Normal Form Satisfiability Problem for which there exists a truth value assignment $w : V \rightarrow \{\texttt{true}, \texttt{false}\}$ such that B evaluates to `true` *under w.*

Theorem 1 *DYN-REMAP(k) is NP-complete.*

Proof: The proof consists of two parts. First we show in Lemma 1 that DYN-REMAP(k) is in NP. Lemma 2 states that 3SAT can be reduced to DYN-REMAP(k) in polynomial time. Since 3SAT is NP-complete, DYN-REMAP(k) has to be NP-complete.
□

Lemma 1 *DYN-REMAP(k) is in NP.*

Proof: Let $d_1, d_2, ...d_n$, $d_i \in D_i, 1 \leq i \leq n$, be a sequence of data layouts for an instance of the dynamic data layout problem, one data layout for each phase in the program. The overall cost of this sequence can be computed in polynomial time as described in Definition 2. Therefore it can be verified in polynomial time whether a given sequence of data layouts has a cost smaller or equal to a given cost k. Hence, DYN-REMAP(k) is in NP.
□

Lemma 2 *3SAT can be reduced in polynomial time to DYN-REMAP(k).*

Proof: Let g be a function that maps an instance B of the 3SAT problem onto an instance $g(B)$ of the DYN-REMAP(0) problem such that $B \in$ 3SAT $\Leftrightarrow g(B) \in$ DYN-REMAP(0). We will define the function g in Part 1 of the proof. In Part 2, we will prove the required property of the function. Finally, in Part 3, we will show that g can be computed in polynomial time.

Part 1: Let B be an arbitrary instance of the 3 Conjunctive Normal Form Satisfiability Problem, $B = \bigwedge_{i=1}^{t}(\ l_i^1 \vee l_i^2 \vee l_i^3 \ )$. g maps the instance B to an instance of the dynamic remapping problem as follows:

- $V = \{v_1, ...v_r\}$, i.e. the sets of variables are the same.
- Each F_i is represented by a distinct phase P_i. The linear order of the nodes in the phase control flow graph corresponds to the numbering of the F_i terms, i.e. the phase control flow graph has edges (P_i, P_{i+1}) for each i, $1 \leq i \leq t-1$.
- $p_i = \{v_j \mid l_i^k \textit{ is a literal of variable } v_j, \ 1 \leq k \leq 3\}$, where $1 \leq i \leq t$. Note that $|p_i| \leq 3$.
- Each variable $v \in p_i$ has 2 possible data layouts, called `T` and `F`. D_i contains $2^{|p_i|}$ candidate layouts, one layout for each possible combination of the single variable layouts. In other words, each $d_i \in D_i$ represents a truth value assignment w_i for all variables in p_i:

$$w_i(v_j) = \begin{cases} \texttt{true} & \text{if } d_{ij} = \texttt{T} \\ \texttt{false} & \text{if } d_{ij} = \texttt{F} \end{cases}$$

- Assume $D_i = \{d_i^1, ... \ d_i^m\}$.

$$c(P_i, d_i^k) = \begin{cases} 0 & \text{if } F_i \text{ is true under the truth value assignment represented by } d_i^k \\ 1 & \text{otherwise} \end{cases},$$

where $1 \leq i \leq t$ and $1 \leq k \leq m$.

- Assume $d_{ij}^k \in d_i^k \in D_i$ and $d_{i'j}^{k'} \in d_{i'}^{k'} \in D_{i'}$, where $i' = f_i(v_j)$.

$$c(d_{ij}^k, d_{i'j}^{k'}) = \begin{cases} 0 & \text{if both data layouts are identical} \\ 1 & \text{otherwise} \end{cases}$$

  In other words, $c(d_{ij}^k, d_{i'j}^{k'}) = 0$ if and only if no remapping of v_j is required between the two data layouts.

An example of the application of g to an instance of 3SAT is given in Section 3.

Part 2a: Claim: $B \in$ 3SAT $\Rightarrow g(B) \in$ DYN-REMAP(0).

Proof: Let $w : V \rightarrow \{\texttt{true}, \texttt{false}\}$ be a truth value assignment that satisfies the problem instance B. There is exactly one data layout scheme in each phase P_i of $g(B)$ that represents w restricted to the variables in p_i. Call this data layout scheme d'_i. For all i, $1 \leq i \leq t$, $c(d'_i) = 0$. Since w specifies a unique data layout for each single program variable $v_j \in V$, redistribution between the sequence of data layouts $d'_1, d'_2, \ldots d'_t$ does not occur. Therefore the sequence has an overall cost of 0. Hence $g(B) \in$ DYN-REMAP(0).

Part 2b: Claim: $g(B) \in$ DYN-REMAP(0) $\Rightarrow B \in$ 3SAT.

Proof: Let $d_1, d_2, \ldots d_t$ be a sequence of data layouts, one data layout for each phase P_i, with an overall cost of 0. Therefore no remapping can occur between the data layouts and each data layout d_i has to represent a truth value assignment that satisfies F_i. Hence, there exists a unique truth value assignment w that satisfies all $F_i, 1 \leq i \leq t$. The existence of such a truth value assignment means that B is in 3SAT.

Part 3: Claim: $g(B)$ can be computed in polynomial time.

Proof: The collection of functions f_i can be computed in $\mathcal{O}(t * r)$, where t and r are the number of phases and program variables, respectively.

There are at most 8 data layouts per phase. Therefore there are at most $t * 8$ cost functions for all phases in the program. For each data layout, at most $3 * 8$ remapping cost functions for individual variables have to be computed per phase, resulting in $t * 3 * 8^2$ cost functions for the entire program. Hence, g can be computed in polynomial time.
□

3 Example Reduction

The function g maps the instance $B = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_4) \wedge (v_1 \vee v_3 \vee \neg v_4)$ of the 3SAT problem into an instance of the decision problem DYN-REMAP(0) as follows:

- $V = \{v_1, v_2, v_3, v_4\}$
- There are three phases, P_1, P_2,and P_3. The ordering of the phases is given by their indices.
- $p_1 = \{v_1, v_2, v_3\}$, $p_2 = \{v_1, v_2, v_4\}$, and $p_3 = \{v_1, v_3, v_4\}$.

- $D_1 = \{ \{(v_1, F), (v_2, F), (v_3, F)\}, \{(v_1, F), (v_2, F), (v_3, T)\}, \{(v_1, F), (v_2, T), (v_3, F)\}, \{(v_1, F), (v_2, T), (v_3, T)\}, \{(v_1, T), (v_2, F), (v_3, F)\}, \{(v_1, T), (v_2, F), (v_3, T)\}, \{(v_1, T), (v_2, T), (v_3, F)\}, \{(v_1, T), (v_2, T), (v_3, T)\} \},$

  $D_2 = \{ \{(v_1, F), (v_2, F), (v_4, F)\}, \{(v_1, F), (v_2, F), (v_4, T)\}, \{(v_1, F), (v_2, T), (v_4, F)\}, \{(v_1, F), (v_2, T), (v_4, T)\}, \{(v_1, T), (v_2, F), (v_4, F)\}, \{(v_1, T), (v_2, F), (v_4, T)\}, \{(v_1, T), (v_2, T), (v_4, F)\}, \{(v_1, T), (v_2, T), (v_4, T)\} \},$ and

  $D_3 = \{ \{(v_1, F), (v_3, F), (v_4, F)\}, \{(v_1, F), (v_3, F), (v_4, T)\}, \{(v_1, F), (v_3, T), (v_4, F)\}, \{(v_1, F), (v_3, T), (v_4, T)\}, \{(v_1, T), (v_3, F), (v_4, F)\}, \{(v_1, T), (v_3, F), (v_4, T)\}, \{(v_1, T), (v_3, T), (v_4, F)\}, \{(v_1, T), (v_3, T), (v_4, T)\} \}.$

- The cost functions for the phases and the cost functions for remapping of individual variables is shown in Figure 1. Individual remapping functions are only shown for $d_3^5 \in D_3$, $d_3^5 = \{ (v_1, T), (v_3, F), (v_4, F) \}$. Each edge in the graph represents a cost function $c(d_{3j}^5, d_{f_3(v_j)j}^k)$, $j \in \{1, 3, 4\}$.

Figure 2 shows a solution s to the example dynamic data layout problem, $s = d_1^7, d_2^7, d_3^5 \in$ DYN-REMAP(0). Note that all cost functions evaluate to 0. The corresponding truth value assignment is $\{(v_1, \texttt{true}), (v_2, \texttt{true}), (v_3, \texttt{false}), (v_4, \texttt{false})\}$. This truth value assignment satisfies B.

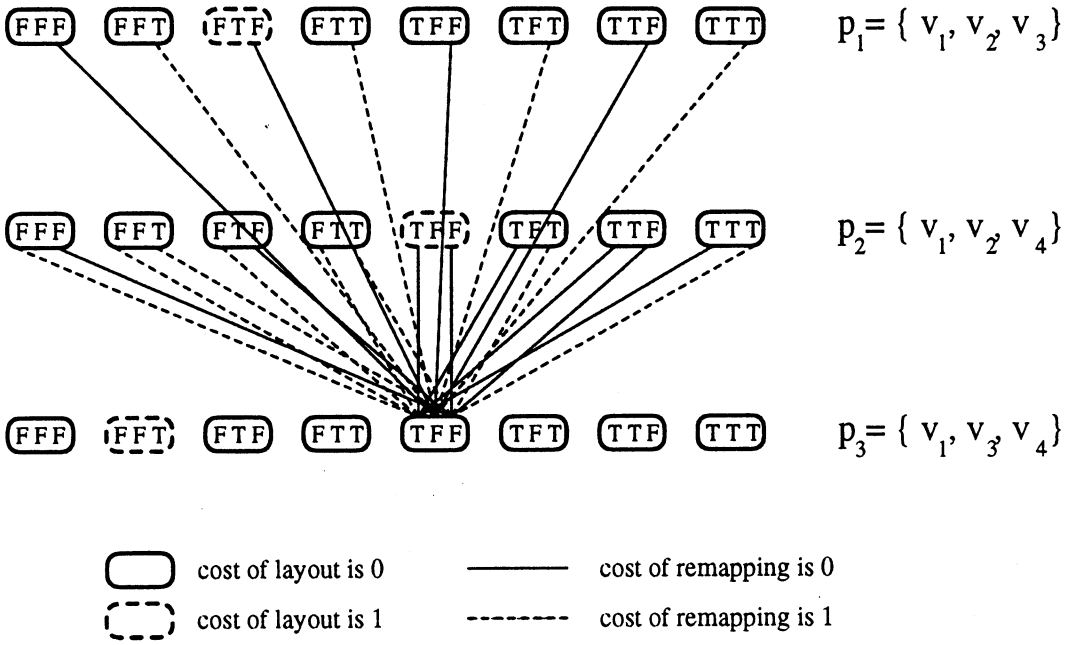

Figure 1: Sample cost functions for $g(B)$, $B = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_4) \wedge (v_1 \vee v_3 \vee \neg v_4)$

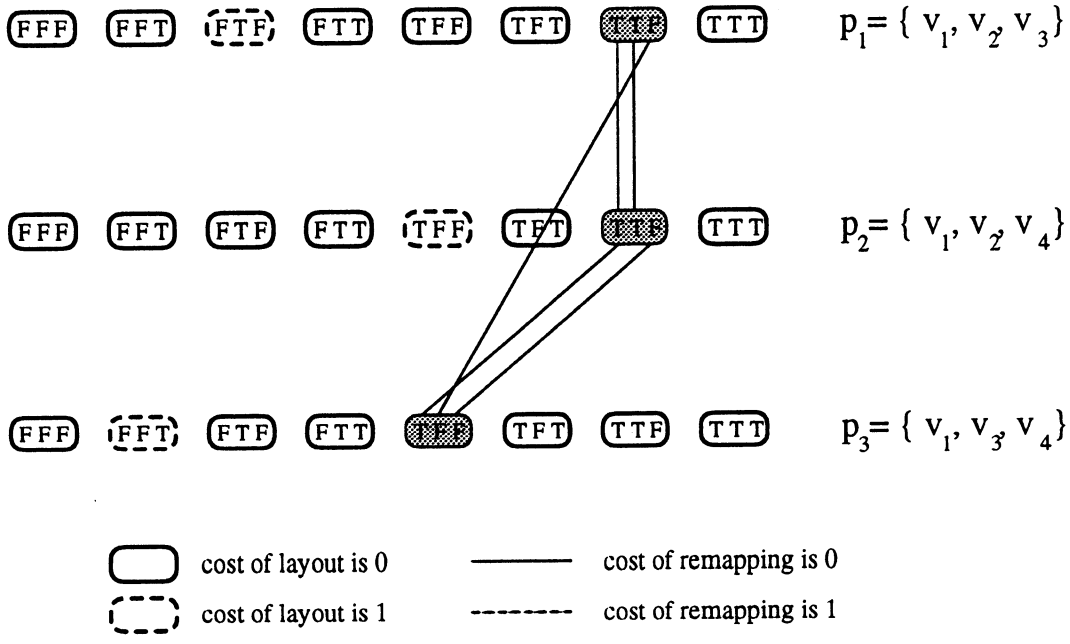

Figure 2: Solution for $g(B)$, $B = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_4) \wedge (v_1 \vee v_3 \vee \neg v_4)$