ELSEVIER

Theoretical Computer Science 178 (1997) 265–273

Theoretical Computer Science

Note

An approximation scheme for scheduling independent jobs into subcubes of a hypercube of fixed dimension

Y. Kopidakis*, V. Zissimopoulos

Université de Paris Sud, L.R.I, CNRS-URA 410, Centre d'Orsay, 91405 Orsay, France

Received October 1995; revised October 1996
Communicated by M. Nivat

Abstract

We study the problem of scheduling independent jobs in a hypercube where jobs are executed in subcubes of various dimensions. The problem being NP-complete, several approximation algorithms based on list scheduling have been proposed, having approximation ratio of order of 2. In this paper, a linear time ε-approximation algorithm for the problem is provided when the size of the hypercube is fixed. We use a reduction to a special strip-packing (or two-dimensional packing) problem with bounded number of distinct pieces. Then, we transform the strip-packing solution into a feasible one for the initial scheduling problem with a small loss in performance. Finally, we provide an improvement which leads to significant reduction of the size of the strip-packing problem.

1. Introduction

We consider the problem of scheduling n independent jobs in an m-dimensional hypercube, where each job requires a set of processors which form a subcube of dimension at most m [10]. Let $T = \{T_i,\ i = 1,\dots,n\}$ a set of n tasks. Task T_i may be executed in any subcube of the hypercube of dimension d_i, $0 \leqslant d_i \leqslant m$, i.e., in a set of 2^{d_i} processors where each processor communicates with d_i neighbors. Task T_i requires any d_i-cube in the hypercube for t_i units of time. We are searching for a non-preemptive schedule with minimum finish time. The problem is NP-complete since the well-known NP-complete problem of multiprocessor scheduling [5] reduces to it. In fact, multiprocessor scheduling is the special case of hypercube scheduling where each job requires exactly one processor ($\forall i,\ d_i = 0$). List-scheduling approximation algorithms have been proposed for the hypercube scheduling problem. In [1], Chen and Lai presented LDLPT (largest dimension largest processing time) with an absolute

* Corresponding author. E-mail: kopidaki@lri.lri.fr.

PII S0304-3975(96)00282-4

bound $2-(1/2^{m-1})$ and in [10], Zhu and Ahuja proposed LDF (largest dimension first) with an absolute bound $2-(1/2^m)$.

In this paper, we present a polynomial-time approximation scheme for the hypercube scheduling problem when the hypercube dimension is fixed. Let $\mathscr{I}$ be the set of instances of a NP-complete minimization problem. Let $I \in \mathscr{I}$ and $OPT(I)$ the value of the optimal solution for I. We say that an algorithm A is a polynomial-time approximation scheme for $\mathscr{I}$, if given any $\varepsilon > 0$: $A(I) \leqslant (1+\varepsilon)OPT(I)$, $\forall I \in \mathscr{I}$, where $A(I)$ is the value of the solution for I returned by A in $O(p(n))$ time and $p(n)$ a polynomial depending on n. In addition, if algorithm A runs in $O(p(n,\varepsilon))$ time, where $p(n,\varepsilon)$ is a polynomial on both n and ε, we say that algorithm A is a fully polynomial-time approximation scheme.

The defined hypercube scheduling problem can be viewed as a strip-packing or a two-dimensional packing problem. Strip packing is a well known NP-complete problem [5], where we search to place n rectangles in a single bin of width 1, so that the total height utilized is minimum. Let (h_i, l_i) with $l_i \leqslant 1$, $i=1,\dots,n$, represent the height and width of the ith piece, respectively. The hypercube scheduling problem reduces to strip-packing where pieces have dimensions:

$$h_i = t_i, \qquad l_i = \frac{2^{d_i}}{2^m}, \tag{1}$$

Clearly,

$$\frac{1}{2^m} \leqslant l_i \leqslant 1. \tag{2}$$

The above reduction makes use of a linear representation of the hypercube in a straight line, where subintervals correspond to nested subcubes [1]. Even though, this representation does not include all the possible subcubes, it guarantees the availability of an entire subcube of proper size, when allowed by the hypercube load. Related results, along with a discussion on subcube allocation strategies in general, can be found in [2, 8].

Using the strip-packing model, we will present a polynomial-time ε-approximation algorithm for scheduling n jobs to an m-dimensional hypercube for fixed m. In Section 2, we claim that strip-packing with a finite number of piece-types can be solved within $1+\varepsilon$ in constant time. In Section 3, we reduce the hypercube scheduling problem to this special strip-packing formulation and use linear grouping to obtain different piece-types. The ε-approximation scheme is described in Section 4. Finally, in Section 5, we use geometric grouping, a more sophisticated grouping technique, in order to achieve an important reduction of the size of the strip-packing problem.

2. Strip-packing with a finite number of types of rectangles

In the present section, we claim that strip-packing can be almost optimally solved in constant time when rectangles belong to a set of finite piece-types.

Proposition 1. *Given any $\varepsilon_1 > 0$, there is an algorithm that packs in constant time within $1+\varepsilon_1$ of the optimal height in a bin of width 1, n rectangles with dimensions (h_i, l_i), when $h_i \in H$ and $l_i \in L$ and $|H| \leqslant a$ and $|L| \leqslant b$, where a and b are constants.*

Proof. The proposition is proved in [4]. The proof is based on a reduction of the strip-packing formulation into a two-dimensional bin-packing formulation. The corresponding optimal solutions differ at most by a factor of $1+\varepsilon_1$. Furthermore, the two-dimensional bin-packing problem can be optimally solved in constant time when the number of distinct piece-types is bounded. Essentially, for a constant number of distinct pieces, we can enumerate all possible bin-types [3, 7]. However, the constant time of the enumeration has an exponential dependence on the number of piece types. □

3. Reduction to strip-packing

In order to reduce the initial scheduling problem to a strip-packing with a bounded number of distinct rectangles, we consider $t_{\max} = \max\{t_i,\ i = 1,\dots,n\}$ and we first divide each t_i by $t_{\max}$ in order to normalize piece heights into the interval $(0,1]$. Then, we apply transformation (1) of Section 1.

Let I the strip-packing formulation produced:

$$I\text{: strip pack } (h_i, l_i),\ i = 1,\dots,n, \quad h_i = \frac{t_i}{t_{\max}}, \quad l_i = \frac{2^{d_i}}{2^m}.$$

Next, we use a grouping technique, called linear grouping (see [3, 7]), in order to deal with a finite number of piece types. We distribute normalized piece heights in a constant number of types of heights. Let k an integer constant. We partition the height interval into k equal subintervals and we define h'_i as follows:

$$h'_i = \frac{j}{k} \quad \text{if } \frac{j-1}{k} < h_i \leqslant \frac{j}{k},\ j = 1,\dots,k,\ i = 1,\dots,n. \tag{3}$$

Clearly,

$$0 \leqslant h'_i - h_i < \frac{1}{k}, \quad i = 1,\dots,n. \tag{4}$$

Let

$$I'\text{: strip pack } (h'_i, l_i),\ i = 1,\dots,n, \quad h'_i \in \left\{\frac{j}{k},\ j = 1,\dots,k\right\}.$$

Proposition 2. *When the hypercube dimension is fixed, I' can be solved within $1+\varepsilon_1$ in constant time, for any $\varepsilon_1 > 0$.*

Proof. In I', $h'_i \in H = \{1/k, 2/k, \dots, k-1/k, 1\}$ and $l_i \in L = \{1/2^m, \dots, 2^{d_i}/2^m, \dots, 1\}$. Obviously, $|H| \leqslant k$ and $|L| \leqslant m$ where k and m are constants. So I', is a strip-packing problem where the number of distinct types of rectangles is bounded. From Proposition 1, it follows that I' can be almost optimally solved in constant time. □

In addition, we will prove that the optimal cost of the transformed problem is very close to the optimal cost for the initial one.

Proposition 3. *For the optimal solutions of* I *and* I', $OPT(I)$ *and* $OPT(I')$, *respectively, we have*: $OPT(I) \leqslant OPT(I') \leqslant OPT(I) + n/k$.

Proof. The inequality $OPT(I) \leqslant OPT(I')$ is straightforward, since $h_i \leqslant h'_i$, $\forall i = 1,\dots,n$. To prove the second one, in solution $OPT(I)$, we transform each h_i into h'_i. As $h'_i - h_i \leqslant 1/k$ (from (4)), we have added at $OPT(I)$ at most n times $\frac{1}{k}$. However, the new packing obtained corresponds to a feasible solution of I', say $FEAS(I')$. Thus

$$FEAS(I') \leqslant OPT(I) + \frac{n}{k}.$$

Obviously, any feasible solution of I' has a total height superior to $OPT(I')$:

$$OPT(I') \leqslant FEAS(I')$$

From the above inequality the proposition is proved. □

4. The approximation scheme

In this section, we describe the ε-approximation algorithm for the problem of scheduling n jobs each requiring a subcube of dimension d_i for t_i units of time in order to minimize completion time:

1. Transform the hypercube scheduling instance into a strip-packing instance I: (h_i, l_i), $i = 1,\dots,n$, $h_i = t_i/t_{\max}$, $l_i = 2^{d_i}/2^m$.
2. For fixed k, get I': (h'_i, l_i), $i = 1,\dots,n$, $h'_i \in \{j/k,\ j = 1,\dots,k\}$ by grouping piece heights into k distinct types as indicated by transformation (3) (linear grouping).
3. Using Proposition 2, find a solution of I' within $1 + \varepsilon_1$ of the optimal, where $\varepsilon_1 < \varepsilon$.
4. Transform the determined solution into a feasible one for I by changing h'_i into initial h_i and return total height.

Proposition 4. *For* $k \geqslant \lceil[(1+\varepsilon_1)/(\varepsilon - \varepsilon_1)h_{\min} l_{\min}]\rceil$, *where* $h_{\min}$ *and* $l_{\min}$ *denote the minimum piece height and width, the above algorithm is a polynomial time approximation scheme for hypercube scheduling of fixed hypercube dimension.*

Proof. We will show that the above algorithm is an ε-approximation scheme for the equivalent strip-packing problem. Let A be the height of the final solution for I. From

(4), $h'_i \geqslant h_i$ and thus total height is reduced from $OPT(I')$ to A. Combining this with Propositions 2 and 3, we have

$$A \leqslant (1+\varepsilon_1)\ OPT(I') \leqslant (1+\varepsilon_1)\left(OPT(I)+\frac{n}{k}\right) \tag{5}$$

and also

$$\frac{A}{OPT(I)} \leqslant 1+\varepsilon_1+\frac{n}{k}\ \frac{1+\varepsilon_1}{OPT(I)}. \tag{6}$$

Let $S(I)=\sum_{i=1}^{n} h_i l_i$, the total surface of pieces of I. Clearly, the height of any solution of I is larger than $S(I)$ and, consequently, $OPT(I) \geqslant S(I)$ and from (6)

$$\frac{A}{OPT(I)} \leqslant 1+\varepsilon_1+\frac{n}{S(I)}\ \frac{1+\varepsilon_1}{k}. \tag{7}$$

In order to get an ε-approximation, we want

$$\varepsilon_1+\frac{n}{S(I)}\ \frac{1+\varepsilon_1}{k} \leqslant \varepsilon \ \Leftrightarrow\ k \geqslant \frac{n}{S(I)}\frac{1+\varepsilon_1}{\varepsilon-\varepsilon_1}. \tag{8}$$

However, $S(I) \geqslant n h_{\min} l_{\min}$, where $h_{\min}$ and $l_{\min}$ are the minimum piece height and width, respectively. Thus, the quantity $n/S(I)$ is bounded above by $1/h_{\min} l_{\min}$ and consequently, since k is an integer, (8) is equivalent to

$$k \geqslant \left\lceil \frac{1+\varepsilon_1}{(\varepsilon-\varepsilon_1) h_{\min} l_{\min}} \right\rceil, \tag{9}$$

which proves that the algorithm provides a solution within $1+\varepsilon$ of the optimal. □

In Proposition 4, k grows with the decrease of $h_{\min}$ and $l_{\min}$. Since the hypercube dimension is fixed, $l_{\min}$ is bounded below by the quantity $1/2^m$. By the normalization procedure $h_{\min}=t_{\min}/t_{\max}$ and the above ratio could take arbitrarily small values. However, even though the number of jobs n grows to infinity, we consider that the execution time of each job is bounded for the scheduling problem. Thus, the increase of k cannot be arbitrarily large.

The above algorithm runs in linear time on n, since step 1 can be executed in $O(n)$ time, step 2 in $O(n \log k) = O(n \log[(1+\varepsilon_1)/(\varepsilon-\varepsilon_1) h_{\min} l_{\min}])$ time, step 3 in constant time (exponential on k) and step 4 in $O(n)$ time.

5. Scheme improvement

In the ε-approximation algorithm presented in the previous section, the number of different types of piece height using linear grouping (k of Proposition 4) increases linearly with the decrement of ε. In order to improve the rate of increment of k and consequently reduce the size of the corresponding strip-packing problem, we propose the application of a slightly more sophisticated grouping technique, called geometric grouping, exploited in a similar way in [4, 9].

Consider the initial strip-packing formulation:

$$I\text{: strip pack } (h_i, l_i),\ i = 1,\dots,n, \quad h_i = \frac{t_i}{t_{\max}}, \quad l_i = \frac{2^{d_i}}{2^m}$$

Instead of distributing piece heights into equal subintervals, we consider intervals of the form $((1+\varepsilon_1)^{-q}, (1+\varepsilon_1)^{-(q-1)}]$, q integer. Thus, we partition $[h_{\min}, 1]$ into k subintervals:

$$[h_{\min} = (1+\varepsilon_1)^{-k}, (1+\varepsilon_1)^{-(k-1)}],\ ((1+\varepsilon_1)^{-(k-1)}, (1+\varepsilon_1)^{-(k-2)}], \dots,$$
$$((1+\varepsilon_1)^{-q}, (1+\varepsilon_1)^{-(q-1)}], \dots, ((1+\varepsilon_1)^{-1}, 1].$$

Clearly,

$$k = \left\lceil \frac{-\log h_{\min}}{\log(1+\varepsilon_1)} \right\rceil. \tag{10}$$

We define h_i^* as follows:

$$h_i^* = (1+\varepsilon_1)^{-(q-1)} \quad \text{if } (1+\varepsilon_1)^{-q} < h_i \leqslant (1+\varepsilon_1)^{-(q-1)},$$
$$q = 1, \dots, \left\lceil \frac{-\log h_{\min}}{\log(1+\varepsilon_1)} \right\rceil. \tag{11}$$

Let

$$I^*\text{: strip pack } (h_i^*, l_i), \quad i = 1,\dots,n$$

Working in exactly the same way as in Proposition 2, we can trivially prove the following:

Proposition 5. *When the hypercube dimension is fixed, I^* can be solved within $1+\varepsilon_2$ in constant time, for any $\varepsilon_2 > 0$.*

The geometric grouping used to produce I^* guarantees that the optimal costs for the initial and the transformed problem can differ by a factor of $1+\varepsilon_1$ at most:

Proposition 6. *For the optimal solutions of I and I^*, $OPT(I)$ and $OPT(I^*)$, respectively, we have $OPT(I) \leqslant OPT(I^*) < (1+\varepsilon_1)\ OPT(I)$.*

Proof. For the first part, notice that $h_i \leqslant h_i^*$, $\forall i = 1,\dots,n$. For the second part, in solution $OPT(I)$, we transform each h_i into h_i^*. Let $FEAS(I^*)$ be the corresponding solution for I^*. By the grouping procedure, when $h_i^* = (1+\varepsilon_1)^{-q}$, then $h_i > (1+\varepsilon_1)^{-(q+1)}$. Consequently,

$$\frac{h_i^*}{h_i} < \frac{(1+\varepsilon_1)^{-q}}{(1+\varepsilon_1)^{-(q+1)}} \quad \forall i = 1,\dots,n$$

and thus,

$$h_i^* < (1+\varepsilon_1)h_i \quad \forall i = 1,\dots,n.$$

The above relation binds the size growth of each transformed piece height and consequently the size growth of the corresponding solution. Thus,

$$FEAS(I^*) < (1+\varepsilon_1)OPT(I).$$

For the optimal solution of I^*, $OPT(I^*)$, we have $OPT(I^*) \leqslant FEAS(I^*)$ and, finally,

$$OPT(I^*) \leqslant FEAS(I^*) < (1+\varepsilon_1)OPT(I),$$

which proves the second part and the proposition. □

The improved ε-approximation algorithm differs from the algorithm of the previous section in the grouping technique used for the transformation of the strip-packing problem to a formulation with a finite number of types of pieces. Its description follows:

1. Transform the hypercube scheduling instance into a strip-packing instance I: (h_i, l_i), $i = 1, \ldots, n$, $h_i = t_i/t_{\max}$, $l_i = 2^{d_i}/2^m$.
2. For $k = \lceil -\log h_{\min}/\log(1+\varepsilon_1)\rceil$, get I^*: (h_i^*, l_i), $i = 1, \ldots, n$, $h_i^* = (1+\varepsilon_1)^{-(q-1)}$, $q = 1, \ldots, k$ by grouping piece heights into k distinct-types as indicated by Eq. (11) (geometric grouping).
3. Find a solution of I^* within $1+\varepsilon_2$ (Proposition 5).
4. Transform the determined solution into a feasible one for I by changing h_i^* into initial h_i and return total height A.

Since $h_i \leqslant h_i^*$, A is no greater than its corresponding solution for I^* and using Proposition 5, clearly, $A \leqslant (1+\varepsilon_2)OPT(I^*)$. In addition, by Proposition 6

$$A \leqslant (1+\varepsilon_1)\ (1+\varepsilon_2)\ OPT(I)$$

and

$$\frac{A}{OPT} \leqslant 1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2.$$

In order to get an ε-approximation scheme, $\forall \varepsilon > 0$, it should be

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2 \leqslant \varepsilon. \tag{12}$$

But, for any $\varepsilon > 0$, we can trivially choose $\varepsilon_1, \varepsilon_2$ to satisfy (12).

In the above algorithm, step 2 is executed in $O(n \log k) = O(n \log(-\log h_{\min}/\log(1+\varepsilon_1)))$. Thus the algorithm remains linear on n.

6. Conclusions

We have presented an approximation scheme for scheduling independent jobs on subcubes of a hypercube of fixed dimension. Initially, we have used a reduction to the well-known strip-packing problem. By performing linear grouping of piece height, we get a strip-packing formulation with a finite number of piece-types which can be almost optimally solved. Furthermore, using geometric grouping, we have achieved an important reduction of the size of the strip-packing formulation.

The above algorithm automatically provides an approximation scheme for the classical multiprocessor scheduling problem (which is the special case with $d_i = 0, \ \forall i$) when the number of processors is fixed. Even though approximation schemes already exist for multiprocessor scheduling [6], the modelization of the problem through strip-packing provides a much simpler approximation algorithm which runs in linear time.

It must be noted that the complexity of the approximation schemes presented in this work, depends on the minimum piece height $h_{\min}$ and, consequently, on the minimum task processing time. This is due to the grouping technique used to distribute the tasks into a fixed number of task types. In the extreme case where we deal with very small processing times, the growth of the running time of the algorithm could be important. In order to avoid this growth, we could consider that processing times are bounded by constants, which is reasonable for any scheduling problem. However, if the restriction of processing times is not desired, very small tasks could be assigned separately after the application of the scheme, in order to preserve the load balance of the determined makespan. With the addition of a separate procedure for the allocation of small tasks, the complexity of the approximation schemes would be data independent.

It would be very interesting to extend the techniques and results of this paper in the case of hypercube scheduling without fixing hypercube dimension. This would require grouping of both height and width of the pieces and would lead to different strip-packing formulations. However, it must be pointed out that there exists no approximation scheme for the general strip-packing problem, even though such a scheme is provided in [4] for the special case where piece dimensions are bounded from below. The probable nonapproximability of general strip-packing implies that the extension in question is a nontrivial task.

References

[1] G.I. Chen and T.H. Lai, Preemptive scheduling of independent jobs on a hypercube, *Inform. Process. Lett.* **28** (1988) 201–206.

[2] M. Chen and K. Shin, Subcube allocation and task migration in hypercube multiprocessors, *IEEE Trans. Comput.* **39**(9) (1990) 1146–1155.

[3] W. Fernandez de la Wega and G. Lueker, Bin packing can be solved in $1 + \varepsilon$ in linear time, *Combinatorica* **1**(4) (1981) 349–355.

[4] W. Fernandez de la Wega and V. Zissimopoulos, An approximation scheme for strip-packing of rectangles with bounded dimensions, Technical Report 713, LRI Universite de Paris-Sud, submitted, 1996.

[5] M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of* NP-*Completeness* (Freeman, San Fransisco, CA, 1979).

[6] D.S. Hochbaum and D.B. Shmoys, Using dual approximation algorithms for scheduling problems: theoretical and practical results, *J. ACM* **34** (1987) 144–162.

[7] N. Karmarkar and R. Karp, An efficient approximation scheme for the one-dimentional bin packing problem, In: FOCS (1982) 312–320.

[8] J. Kim, C. Das and W. Lin, A top-down processor allocation scheme for hypercube computers, *IEEE Trans. Parallel Distrib. Systems* **2**(1) (1991).

[9] J. Leung, Bin packing with restricted piece sizes, *Inform. Process. Lett.* **31** (1989) 145–149.

[10] Y. Zhu and M. Ahuja, On job scheduling on a hypercube, *IEEE Trans. Parallel Distrib. Systems* **4** (1993) 62–69.