Approximating Scheduling Problems in Parallel*

Maria Serna and Fatos Xhafa

Department of LSI, UPC Campus Nord, C6, Jordi Girona Salgado, 1-3 08034-Barcelona Email: {mjserna,fatos}@lsi.upc.es

Abstract. We show how to approximate in NC the problem of Scheduling Unrelated Parallel Machines, for a fixed number of machines. We develop a $(2 + \varepsilon)$ -approximate parallel algorithm for the problem. Our approach shows how to relate the linear program obtained by relaxing the integer programming formulation of the problem with a linear program formulation that is positive and in the packing/covering form. The relationship established enables us to transfer approximate fractional solutions from the later formulation that is known to be approximable in NC. Then, we show how to obtain an integer approximate solution, i.e. a schedule, from the fractional one, using the randomized rounding technique. Finally, we show that the same technique can be applied to the General Assignment Problem of fixed number of machines and a given makespan T, thus yielding a schedule whose cost is at most $(2+\varepsilon)$ times the minimum cost and has makespan at most 2T.

1 Introduction

Linear Programming plays an important role in designing sequential approximation algorithms for many NP-hard optimization problems. All of them start from (or make use of) a fractional solution of the relaxed linear program for the problem in hand. In some cases, the feasible fractional solution is directly transformed into an integer feasible solution to the problem, whose measure is within some bound (constant) of the optimum value. For example, Potts [14] shows how to transform the (basic) optimal fractional solution of a linear program for Scheduling Unrelated Parallel Machines into an integer feasible solution whose measure is within 2 of the optimum schedule, where the number of machines is supposed a fixed constant. For the same problem, but without the restriction on the number of machines to be fixed, Lenstra et al. [9] show how to transform a vertex of the fractional polyhedron into a vertex of the integer polyhedron which is a schedule with completion time at most twice the optimum one. Similar ideas are employed also in [17, 4]. (For other sequential results on this type of scheduling see also [6].) In some other cases, transforming the fractional feasible solution into an integer one uses a rounding procedure, e.g., the 2-approximation

^{*} This research was partially supported by the ESPRIT Long Term Research Project No. 20244 - ALCOM IT.

for Vertex Cover [5], and more generally by means of Randomized Rounding technique [15] or the same technique with more refined probability choices [3].

If we would like to employ similar ideas to obtain parallel approximation algorithms, a lot of difficulties are encountered. Indeed, to begin with, we cannot solve in NC a given LP, unless P=NC [8, 2]. Even we cannot find an approximate solution of a given LP in NC [16, 12]. The only known kind of parallel algorithm for LP is the Luby and Nisan's approximation scheme [11] for a subclass of LP, namely an LP with non-negative coefficients and in packing/covering form, referred to as Positive Linear Programming (PLP). Even if the LP program that models a given problem is in PLP form we cannot optimally solve it in NC, unless P=NC [19]. Thus, for example, the technique of [14] fails in this case. Maybe, from this viewpoint can be explained somehow why the NC approximation algorithms based on LP technique are scarce. Very recently, Trevisan [18], used LP technique to obtain (improved) parallel approximation algorithms for various optimization problems, including MAX SAT, MAX DICUT, etc. This is the first case where parallel approximation results were obtained for problems whose original LP model was not in PLP form.

In this paper, following the work of [18], we develop a $(2 + \varepsilon)$ -approximate parallel algorithm for the problem of scheduling a fixed number of unrelated parallel machines. This problem is known to be NP-hard. Karp [7] proved that even for the case of two identical machines the problem is NP-hard.

Here is a brief description of the proof. The linear program for Scheduling Unrelated Parallel Machines obtained by relaxing its integer programming formulation is not a Positive Linear Program in packing/covering form so it is not appropriate for Luby and Nisan's algorithm. Our strategy, is then, to transform the linear program into a positive linear program in packing/covering form. In the sequential case, transforming a linear program with linear restrictions that are equalities into a packing/covering program is easily done (see, e.g., [20, pages 141-143) since we only need to assure that both linear programs – the original and the transformed one – have the same optimal solution. It is, however, more complicated in the parallel case. Because Luby and Nisan's algorithm only approximately solves a PLP, we have to assure that not only the optimal solutions of both linear programs coincide but also we can recover near optimal fractional solutions of the original program from those of the transformed one. The final step in our proof is to randomly round the near optimal fractional solution into an integer feasible solution whose measure is within $(2 + \varepsilon)$ of the minimum schedule. As another application of our technique, we show that given an instance of the General Assignment Problem of fixed number of machines and makespan T, there is an NC algorithm that finds a schedule whose cost is at most $(2 + \varepsilon)$ times the minimum cost and has makespan at most 2T. In both cases, solving in parallel a linear program is the caveat of our scheme.

Preliminaries

Positive LP in packing form is $\max_{x} \{c^{T}x : Ax \leq b; x \geq 0\}$, and in covering form $\min_{y} \{by : A^{T}y \geq c; y \geq 0\}$, where A, b and c have non-negative entries. We will refer to the following theorem by Luby and Nisan, which gives an NC

approximation scheme for Positive LP in packing/covering form. Let N be the number of non-zero coefficients associated to an instance of PLP.

Theorem 1. [11] Given an instance of PLP in packing form (resp. in covering form), and a rational $\varepsilon > 0$, there is an NC algorithm that finds an $(1 - \varepsilon)$ -approximate solution (resp. an $(1 + \varepsilon)$ -approximate solution) to that instance and has running time polynomial in $\log N/\varepsilon$ and uses O(N) processors.

2 Scheduling Unrelated Parallel Machines

This problem is stated as follows (see, e.g., [14]). Each of n jobs is to be processed without interruption on one of m machines. At any time, each machine can process at most one job. Job j (j = 1, ..., n) requires a positive processing time p_{ij} on machine i (i = 1, ..., m). The objective is to schedule the jobs on machines so that the maximum completion time is minimized. Here after the number of machines m is supposed a fixed constant, independent of the instance.

The problem can be casted by an integer program by letting variables x_{ij} , $x_{ij} = 1$, if job j is assigned to machine i, and $x_{ij} = 0$ otherwise, and the variable z which stands for the maximum completion time of machines. By relaxing the integrality conditions i.e., $0 \le x_{ij} \le 1$, we obtain the following linear program,

minimize
$$z$$

subject to

$$\sum_{j=1}^{n} p_{ij} x_{ij} \le z \ (i = 1, ..., m)$$
(1)

$$\sum_{i=1}^{m} x_{ij} = 1 \ (j = 1, ..., n)$$
(2)
 $0 \le z, 0 \le x_{ij} \ (i = 1, ..., m; \ j = 1, ..., n)$

(LP1)

The (LP1) program is not a Positive Linear Program since some of its coefficients are negative (see constraints (1)) and also it is not in the packing form (see constraints (2)). Our first objective is then to transform it into a PLP without changing the optimal solution. To this aim, we let $p = \max_{1 \le i \le m} \sum_{j=1}^{n} p_{ij}$ and we introduce the variable $y \ge 0$ such that z + y = p, and we obtain a linear program (LP2) equivalent to (LP1), but with all the coefficients non-negative:

maximize
$$y$$

subject to
$$\sum_{j=1}^{n} p_{ij} x_{ij} + y \le p \ (i = 1, \dots, m)$$
(3)
$$\sum_{i=1}^{m} x_{ij} = 1 \qquad (j = 1, \dots, n)$$
(4)

$$0 \le y, 0 \le x_{ij}$$
 $(i = 1, ..., m; j = 1, ..., n)$

(LP2)

Notice that the relation z + y = p is not actually needed as a constraint as far as we keep the relationship between y and z as an external change of variables. Next, we transform (LP2) into a packing form by adding adequately some terms in the objective function which allow us to relax the equality constraints into inequalities. We choose the coefficients of those terms in such a way that not only the optimal solution remains unchanged but also we can recover approximate solutions of ² (LP1). We let, for any j = 1, ..., n, $p_j = \sum_{i=1}^{n} p_{ij}$ and consider,

maximize
$$y + \sum_{j=1}^{n} \frac{p_j}{m} \sum_{i=1}^{m} x_{ij}$$

subject to
 $\sum_{j=1}^{n} p_{ij} x_{ij} + y \le p \quad (i = 1, \dots, m)$ (5)

$$\sum_{i=1}^{m} x_{ij} \le 1 \qquad (j = 1, \dots, n)$$
(6)

(LP3)

3 Finding an $(1 + \epsilon)$ -Approximate Fractional Solution for (LP1) from an $(1 - \epsilon)$ -Approximate One for (LP3)

The following proposition relates the solutions (and their respective measures) of (LP1) with those of (LP2). Its proof is clear from the definition of (LP2).

Proposition 2. For any feasible solution (y, \mathbf{x}) of (LP2) of measure k, $(p-y, \mathbf{x})$ is feasible to (LP1) of measure p - k. For any feasible solution (z, \mathbf{x}) for (LP1) of measure p - k, $(p - z, \mathbf{x})$ is feasible of (LP2) and has measure k.

The relation between solutions of (LP2) and (LP3) is given as follows:

Proposition 3. The following hold:

- a) Any feasible solution for (LP2) of measure k is also a feasible solution for the (LP3) and has measure $k + \sum_{j=1}^{n} p_j/m$.
- b) Given any feasible solution for the (LP3) of measure $k + \sum_{j=1}^{n} p_j/m$, we can compute in NC a feasible solution for (LP2) of measure at least k.

Proof. The part a) is immediate from the definitions of (LP2) and (LP3) programs. For the part b), given a feasible solution (y, \mathbf{x}) for (LP3) whose measure is $k + \sum_{j=1}^{n} p_j/m$, we define a solution (y', \mathbf{x}') for (LP2) as follows:

$$y' := \min\{y, \min_i\{p - \sum_{j=1}^n p_{ij}x'_{ij}\}\}, \quad x'_{ij} := x_{ij} + (1 - \sum_{r=1}^m x_{rj})/m, \quad \forall i, j$$
.

Notice that (y', \mathbf{x}') is feasible for (LP2). This is the desired solution. Indeed, we have that, for any $i, y \leq p - \sum_{j=1}^{n} p_{ij}x_{ij}$, and on the other hand, a) y' = y or b) $y' = p - \sum_{j=1}^{n} p_{ioj}x'_{ioj}$, for some i_0 . If y' = y then we are done since $y \geq k$. In the second case, let us denote $\alpha_j = (1 - \sum_{r=1}^{m} x_{rj})/m$. We have

² This construction is, in some sense, similar to Lagrangean multipliers

 $y' = p - \sum_{j=1}^{n} p_{i_0j} x_{i_0j} - \sum_{j=1}^{n} p_{i_0j} \alpha_j$, or equivalently, $y' + \sum_{j=1}^{n} p_{i_0j} \alpha_j = p - \sum_{j=1}^{n} p_{i_0j} x_{i_0j} \ge y$ from which we deduce

$$y \leq p - \sum_{j=1}^{n} p_{i_0j} x_{i_0j} = y' + \sum_{j=1}^{n} p_{i_0j} \alpha_j$$

$$\leq y' + \sum_{j=1}^{n} \frac{p_j}{m} (1 - \sum_{r=1}^{m} x_{rj})$$

$$= y' + \sum_{j=1}^{n} \frac{p_j}{m} - \sum_{j=1}^{n} \frac{p_j}{m} \sum_{i=1}^{m} x_{ij}$$
(7)

where in (7) we have bounded p_{i_0j} by p_j , and thus, $y + \sum_{j=1}^n (p_j/m) \sum_{i=1}^m x_{ij} \le y' + \sum_{j=1}^n p_j/m$. From the hypothesis and the last inequality it is easily derived $y' \ge k$. Finally the solution (y', \mathbf{x}') can be constructed in NC.

From Propositions 2 and 3, we will prove that, from any fractional $(1 - \varepsilon)$ -approximate solution to (LP3), we can construct a fractional $(1+\varepsilon)$ -approximate solution to (LP1). As a first step, we will state and prove the result with an additional condition, that we call *Bounded Scheduling Condition*. Then, we prove it for the general case. Let us define $q = (1/m) \sum_{j=1}^{n} \min_{1 \le i \le m} p_{ij}$ and $p = \max_{1 \le i \le m} \sum_{j=1}^{n} p_{ij}$, as before.

Bounded Scheduling Condition: There is a constant B such that $p \leq Bq$. Let us consider an instance of the scheduling that satisfies the above condition with constant B, and let also (LP1), (LP2) and (LP3) be the corresponding linear programs to this problem. We prove the following theorem.

Theorem 4. Assume that there is a constant B such that $p \leq Bq$. Suppose that (y, \mathbf{x}) is a $(1 - 1/(2B) \cdot \varepsilon)$ -approximate solution for (LP3), and let (z', \mathbf{x}') be the feasible solution of (LP1) obtained from Proposition 3, part b) and Proposition 2. Then (z', \mathbf{x}') is a $(1 + \varepsilon)$ -approximate solution for (LP1).

Proof. Let us denote by C_1^* , C_2^* and C_3^* the optimal values of the (LP1), (LP2) and (LP3), respectively. We have the following relations: (1) $C_1^* \ge q$, (2) $C_2^* = p - C_1^*$, and (3) $C_3^* \le 2 \cdot p$. The relation (1) comes from the definition of q, relation (2) is obvious (see Prop. 2), and relation (3) results from upper bounding any term in the objective function of (LP3). Notice that, for relation (3) we use

$$\sum_{j=1}^{n} p_j/m = (1/m) \sum_{j=1}^{n} \sum_{i=1}^{m} p_{ij} = (1/m) \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \le mp/m = p$$

From the above relations we deduce $C_1^* \ge q \ge p/B \ge 1/(2B) \cdot C_3^*$. From the supposition, (y, \mathbf{x}) is feasible to (LP3) of cost3 such that $(C_3^* - \cos t3)/C_3^* \le (1/2B)\varepsilon$. Further, we apply Prop. 3, part b) to obtain a feasible solution (y', \mathbf{x}') for (LP2) and then the corresponding feasible solution (z', \mathbf{x}') to (LP1) of cost1. Now we show that (z', \mathbf{x}') is $(1 + \varepsilon)$ -approximate for (LP1). Indeed,

$$\begin{aligned} \cos\texttt{t1}/C_1^* &= 1 + (\texttt{cost1} - C_1^*)/C_1^* \stackrel{(*)}{\leq} 1 + (C_3^* - \texttt{cost3})/C_1^* \\ &\leq 1 + 2B \cdot (C_3^* - \texttt{cost3})/C_3^* \leq 1 + 2B \cdot (1/2B)\varepsilon = 1 + \varepsilon \end{aligned}$$

The inequality (*) holds because of:

$$cost1 - C_1^* = (p - cost2) - C_1^* = C_2^* - cost2 \le (C_3^* - cost3)$$
(8)

where in (8) we have used Proposition 3, and thus the theorem follows. \Box

We can apply Luby and Nisan's algorithm [11] to find a feasible solution (y, \mathbf{x}) to (LP3) of cost3 such that $(C_3^* - \text{cost3})/C_3^* \leq (1/2B)\varepsilon$. Therefore we have

Theorem 5. Assume that there is a constant B such that $p \leq Bq$. Given any rational $\varepsilon > 0$, we can construct in NC a feasible solution of (LP1) (z', \mathbf{x}') that is a $(1 + \varepsilon)$ -approximate solution for (LP1).

Following we prove that an instance of (general) scheduling problem can be reduced to an instance satisfying the Bounded Scheduling Condition. Therefore, from Theorem 5, for any instance of Unrelated Parallel Machines we can find in NC a near optimal fractional solution to (LP1) from (LP3).

General Case: Scaling of the Processing Times

Given a positive constant K > 1, we can transform the instance of scheduling $\{p_{ij}, i = 1, \ldots, m; j = 1, \ldots, n\}$ into an instance $\{p'_{ij}, i = 1, \ldots, m; j = 1, \ldots, n\}$ such that $1 \le p'_{ij} \le K, \forall i, \forall j$. Indeed, this can be done by a linear transformation ax + b, such that $\{ap_{i,j_r} + b = 1, ap_{i,j_r} + b = K\}$ where $p_{i,j_r} = \min_{i,j} p_{ij}$ and $p_{i,j_s} = \max_{i,j} p_{ij}$. Also, the variable y becomes $\overline{y} = ay + b$. Further, we see that for the new instance,

$$p' = \max_{1 \le i \le m} \sum_{j=1}^{n} p'_{ij} = \sum_{j=1}^{n} p'_{i_1j} , \text{ for some } i_1 ,$$

$$\leq \sum_{j=1}^{n} K \cdot \min_{1 \le i \le m} p'_{ij} = K \cdot \sum_{j=1}^{n} \min_{1 \le i \le m} p'_{ij} = Kmq' ,$$

that is $p' \leq Bq'$, with B = Km. Notice that once we have the solution \overline{y} of the new instance, we can recover the value of y by taking $y = (\overline{y} - b)/a$ from the known linear transformation ax + b. Therefore, concluding, we have,

Theorem 6. Assuming that m is a constant, for any $\varepsilon > 0$, we can construct in NC a solution (y, \mathbf{x}) for (LP3) that is a $(1 - 1/(2B) \cdot \varepsilon)$ -approximation, where B = K m, for some constant K > 1. And therefore, we can construct in NC a feasible solution of $(LP1)(z', \mathbf{x}')$ that is a $(1 + \varepsilon)$ -approximate for (LP1).

4 Rounding of the Fractional Solution

We use the Randomized Rounding technique of [15] to obtain a $(2+\varepsilon)$ -approximate integer solution to (LP1). Let \tilde{x}_{ij} , $(i = 1, \ldots, m; j = 1, \ldots, n)$ be the fractional solution of (LP1) that is $(1+\varepsilon)$ -approximate (see Sect. 3). Let \tilde{C}_1 be the completion time corresponding to this solution and C_1^* be the optimal fractional value and finally C_{\max}^* the minimum completion time. We have that

$$(1-\varepsilon)C_1 \le C_1^* \le C_{\max}^* . \tag{9}$$

We proceed as follows. For any (fixed) constraint j of (2), set $x_{ij} = 1$ according to the probability distribution $\{\tilde{x}_{ij}\}$ for the fixed j, and the choices are done independently, for all j. Next, we apply Theorem 2.1 of [15] that estimates the expected value of the objective function in the integer solution defined above.

Theorem 7. [15] Let ε be a positive real such that $0 < \varepsilon < 1$. Provided that

$$C_1^* \ge 3\ln(2n(n-1)/\varepsilon) \quad , \tag{10}$$

the corresponding value of the solution constructed above does not exceed

$$C_1^* + (3 \cdot C_1^* \ln(2n(n-1)/\varepsilon))^{1/2}$$
(11)

with probability at least $1 - \varepsilon$.

Based on this theorem we obtain the following result.

Theorem 8. Given an instance of Scheduling Unrelated Parallel Machines with a fixed number of machines where we want to minimize the maximum completion time, there exist an NC algorithm that finds a $(2 + \varepsilon)$ -approximate solution to that instance.

Proof. We find a fractional solution \tilde{x}_{ij} according to Theorem 6 and then, in NC, we generate randomly an integer solution x_{ij} as described above. From (9), (10) and (11) it is easy to see that the integer solution is a 2-approximate solution (schedule). But, since we apply this theorem with the near optimal value \tilde{C}_1 corresponding to the $(1 + \varepsilon)$ -approximate solution of (LP1), it implies a $(2 + \varepsilon)$ -approximate solution for (LP1). The integer solution found randomly in the above process can also be found deterministically in NC. Notice that the variables of constraint j do not appear in the rest of constraints (2), and furthermore any probability constraint (event) on the random variables involves at most m of them (the number of variables of any constraint (2)). Therefore, m-wise independence of random variables would suffice for the analysis. The derandomized technique of Luby [10] is then applicable.

5 Extension

Following we show that the technique described above can be used to obtain parallel approximation algorithms also for other scheduling models.

5.1 The Generalized Assignment Problem

The Generalized Assignment Problem is viewed as a problem of scheduling unrelated parallel machines with costs. (We are referred to [17] for the definition of the problem and its LP formulation.) Each of n jobs is to be processed by exactly one of m unrelated parallel machines. Job j takes p_{ij} times units when processed in machine i and acquires a cost c_{ij} . Also there is supposed that the load of any machine is at most T. This problem can be modeled by an integer program by letting variables x_{ij} be 1 if job j is processed in machine i and 0 otherwise. Here is the relaxed linear program $(0 \le x_{ij})$, referred to as (GA):

minimize
$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$
subject to
$$\sum_{i=1}^{m} x_{ij} = 1 \quad (j = 1, \dots, n)$$
$$\sum_{i=1}^{n} p_{ij} x_{ij} \leq T \quad (i = 1, \dots, m)$$

We show that the Generalized Assignment Problem fits in the schema we described for Scheduling Unrelated Parallel Machines. To this aim we will give the linear programs (GA1), (GA2) and (GA3) (analogs to (LP1), (LP2) and (LP3)) and also we will mention some points that are more relevants.

The program (GA) can be written equivalently as follows, denoted (GA1)

minimize
$$z$$

subject to $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \leq z$
 $\sum_{i=1}^{m} x_{ij} = 1$ $(j = 1, ..., n)$
 $\sum_{j=1}^{n} p_{ij} x_{ij} \leq T$ $(i = 1, ..., m)$ (12)

Let us denote by $C = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}$, the total cost of the instance and for any $j = 1, \ldots, n, c_j = \sum_{i=1}^{m} c_{ij}$. Also we let the variable y such that z + y = C. Then, transforming (GA1) into a positive linear program in packing form goes through the following two programs (GA2) and (GA3):

maximize
$$y$$

subject to $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + y \leq C$
 $\sum_{i=1}^{m} x_{ij} = 1$ $(j = 1, \dots, n)$
 $\sum_{j=1}^{n} p_{ij} x_{ij} + u \leq T$ $(i = 1, \dots, m)$
 $0 \leq u, 0 \leq x_{ij}$ $(i = 1, \dots, m; j = 1, \dots, n)$

maximize $y + \sum_{j=1}^{n} c_j \sum_{i=1}^{m} x_{ij}$ subject to

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + y \le C$$

$$\sum_{i=1}^{m} x_{ij} \le 1 \qquad (j = 1, ..., n)$$

$$\sum_{j=1}^{n} p_{ij} x_{ij} + u \le T \qquad (i = 1, ..., m)$$

$$0 \le u, 0 \le x_{ij} \qquad (i = 1, ..., m; j = 1, ..., n)$$

where u is a "dummy" variable that we add for the sake of the analysis. Having the linear programs (GA1), (GA2) and (GA3) we can follow the schema of the Sect(s). 2, 3 and 4 quite similarly. Here we give the analog of Proposition 3.

Proposition 9. The following hold:

- a) If (y, u, \mathbf{x}) is a feasible solution for (GA2) of measure k, then it is also a feasible solution for the (GA3) and has measure $k + \sum_{j=1}^{n} c_j$.
- b) Given any feasible solution (y, u, \mathbf{x}) for the (GA3) of measure $k + \sum_{j=1}^{n} c_j$, we can compute in NC a feasible solution for (GA2) of measure at least k.

Proof. We give a sketched idea of part b). Given a feasible solution (y, u, \mathbf{x}) for (GA3) of measure $k + \sum_{j=1}^{n} c_j$, we define (y', u', \mathbf{x}') as follows:

$$x'_{ij} := x_{ij} + \frac{1 - \sum_{r=1}^{m} x_{rj}}{m}, \quad y' := \min\{y, C - \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x'_{ij}\},$$
$$u' := \min\{u, \min_i\{T - \sum_{j=1}^{n} p_{ij} x'_{ij}\}\}.$$
(13)

From the above definition it is easy to prove that (y', u', \mathbf{x}') is feasible for (GA2) and whose measure y' is $y' \ge k$. Notice that in (13) is where we use the "dummy" variable u so that $\{x'_{ij}\}$ satisfy also (12).

We give (proof omitted) the following theorem which shows that it is possible to recover in NC near optimal fractional solutions of (GA1) from those of (GA3).

Theorem 10. Assuming that m is a constant, for any $\varepsilon > 0$, we can find in NC a feasible solution of (GA1) (z', \mathbf{x}') that is $(1 + \varepsilon)$ -approximate solution.

Another point to be mentioned is that regarding the Randomized Rounding applied to the fractional solution. Because of (12), the schedule obtained by randomly rounding the fractional solution will have cost at most $(2 + \varepsilon)$ times the minimum cost and makespan at most 2T.

Theorem 11. Given an instance of Generalized Assignment, where m is a fixed constant, and we want to find a schedule that minimize the maximum completion time, and of makespan at most T, then there exist an RNC algorithm that finds a schedule whose cost is at most $(2 + \varepsilon)$ times the minimum cost and makespan at most 2T. The schedule can be found also deterministically in NC.

6 Conclusions

We have shown how to obtain a parallel approximation for Scheduling Unrelated Parallel Machines, when the number of machines is fixed. The condition for the number of machines to be fixed comes from the Luby & Nisan's schema. In fact, this condition can be relaxed up to $m = O(\log n)$ since in this case the running time of the the Luby & Nisan's algorithm is still polylogarithmic in n and we can still perform in NC the randomized rounding as well as the derandomization [13, 1]. Finally, we have shown how to apply the same technique to the General Assignment Problem of fixed number of machines and a given makespan T, to obtain a schedule whose cost is at most $(2 + \varepsilon)$ times the minimum cost and has makespan at most 2T.

Acknowledgment

We thank the anonymous referees for their helpful comments.

References

- Berger, B. and Rompel, J.: Simulating (log^c n)-wise Independence in NC. J. of the ACM, 38 (1991) 1026–1046
- Dobkin, D., Lipton, J.R. and Reiss, S.: Linear Programming is Logspace Hard for P. IPL, 8 (1979) 96-97
- Goemans, M.X. and Williamson, D.P.: New 3/4-approximation Algorithms for the Maximum Satisfability Problem. SIAM J. Disc. Math., 7 (1994) 656-666
- Hall, L.A.: Approximability of Flow Shop Scheduling. In Proc. of 36th IEEE Symp. on Found. of Comp. Sc, (1995) 82-91
- 5. Hochbaum, D.S.: Approximation Algorithms for the Set Cover and the Vertex Cover. SIAM J. of Comp., 11(3) (1982) 555-556
- Horowitz, E. and Sahni, S.: Exact and approximate algorithms for scheduling nonidentical processors. J. ACM, 23 (1976) 317–327
- Karp, R.M.: Reducibility Among Combinatorial Problems. Complexity of Computer Computations, eds. R.E. Miller and J.W. Thatcher, Plenum Press, 1972.
- Khachyan, L.G.: A Polynomial Algorithm in Linear Programming. Trans. in Sov. Math. Dok., 20 (1979) 191–194
- Lenstra, J.K., Shmoys, D.B. and Tardos, É.: Approximation Algorithms for Scheduling Unrelated Parallel Machines. Math. Prog., 46 (1990) 259-271
- Luby, M.: Removing Randomness in Parallel Computation without a Processor Penalty. In Proc. of 29th IEEE Symp. on Found. of Comp. Sci., (1988) 162–173
- Luby, M. and Nisan, N.: A Parallel Approximation Algorithm for Positive Linear Programming. In Proc. of 25th ACM Symp. on Th. of Comp., (1993) 448-457
- 12. Megiddo, N.: A Note on Approximate Linear Programming. IPL, 42 (1992) 53
- Motwani, R. and Naor, J.S. and Naor, M.: The Probabilistic Method Yields Deterministic Parallel Algorithms. J. of Comp. and Sys. Sci., 49 (1994) 478-516
- Potts, C.N.: Analysis of a Linear Programming Heuristic for Scheduling Unrelated Parallel Machines. Disc. App. Math., 10 (1985) 155-164
- Raghavan, R. and Thompson, C.: Randomized Rounding: a Technique for Provably Good Algorithms and Algorithmic Proofs. Combinatorica, 7 (1987) 365-374
- Serna, M.J.: Approximating Linear Programming is Logspace Complete for P. IPL, 37 (1991) 233-236
- Shmoys, D.B. and Tardos, É.: An Approximation Algorithms for the Generalized Assignment Problem. Math. Prog., 62 (1993) 461-474
- Trevisan, L.: Positive Linear Programming, Parallel Approximation and PCP's. ESA'96, LNCS 1136 (1996) 62-75
- Trevisan, L. and Xhafa, F.: The Parallel Complexity of Positive Linear Programming. Tech. Rep. Dept. of LSI, (1996) LSI-96-52-R
- 20. Walukiewicz, S.: Integer Programming, Kluwer Acad. Publ., (1990)