INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

On-line Class Constraint Bin Packing

E. C. Xavier *F. K. Miyazawa*

Technical Report - IC-05-16 - Relatório Técnico

July - 2005 - Julho

On-line Class Constraint Bin Packing

E. C. Xavier[†] F. K. Miyazawa[†]

July 5, 2005

Abstract

In this paper we present approximation results for the on-line class constrained bin packing problem. In this problem we are given bins of capacity 1 with C compartments, and n items of Q different classes, each item $i \in \{1, \ldots, n\}$ with class $c(i)$ and size $s(i)$. The problem consists to pack the items into bins in an on-line way, where each bin contains at most C different classes and has total items size at most 1. We show that the bounded space version of this problem does not have an algorithm with constant competitive ratio. If each item have size at least $\varepsilon < 1$, we show that the problem does not admit an algorithm with competitive ratio better than $O(1/C\varepsilon)$. In the unbounded case we show that the First-Fit algorithm has competitive ratio in $[2.7, 3]$ and we present an algorithm with competitive ratio in $[2.66, 2.75]$.

1 Introduction

In this paper we study the class constrained version of the well known on-line bin packing problem, which we denote by on-line CCBP (On-Line Class Constrained Bin Packing). In this problem we are given a tuple $I = (L, s, c, Q, C)$ where $L = (a_1, \ldots, a_n)$ is a list of items, each item $a_i \in L$ with size $0 < s(a_i) \leq 1$ and class $c(a_i) \in \{1, \ldots, Q\}$, and a set of bins with capacity 1 and C compartments. A packing $\mathcal{P}$ of L is a partition of the items into bins, where each part has total items size at most 1 and the number of classes is at most C. The problem consists in find a packing of L into the minimum number of bins. In the on-line version of the CCBP problem the items must be packed in the order $(a_1, \ldots, a_n)$, where each item a_i must be packed without knowledge of further items. This problem has many applications in resource allocation in computer and manufacturing systems [9, 3, 8, 2].

The bins used to pack the items are classified as *open* or *closed*. An empty bin is declared open when it receives its first item, and remains so until it is declared closed. Only open bins may receive items. Once a bin is closed, it cannot be declared open again. We consider the bounded and unbounded space versions for the on-line CCBP problem. In the k-bounded space problem an algorithm must keep at any time during its execution at most k open bins. In the unbounded version an algorithm may keep an unbounded number of open bins.

Given an algorithm $\mathcal{A}$ for the CCBP problem and an instance I, we denote by $\mathcal{A}(I)$ the number of bins used by the algorithm to pack this instance. We denote by $\mathrm{OPT}(I)$ the number of bins used by

[2]Instituto de Computação — Universidade Estadual de Campinas, Caixa Postal 6176 — 13084–971 — Campinas–SP — Brazil, {eduardo.xavier,fkm}@ic.unicamp.br.

an optimum (off-line) solution to pack the instance I. An on-line algorithm $\mathcal{A}$ for a minimization problem is said to have a competitive ratio α if there exists a constant β such that $\mathcal{A}(I) \leq \alpha \cdot \text{OPT}(I) + \beta$ for any instance I.

Related Work: The on-line bin packing is a well studied problem. There are many on-line algorithms presented in the literature for the bin-packing problem. The best on-line algorithm for the bin packing problem is due to Seiden [6] with a competitive ratio of 1.58889 and the best lower bound for this problem is 1.54014 due to van Vliet [10]. Recently the class-constrained versions of packing problems have obtained attention. In [1], Dawande et al presented approximation schemes for a class constrained version of the offline bin packing problem. They consider the problem where bins can have different sizes and each bin can pack items of at most K different classes. In [7], Schachnai and Tamir presented a dual polynomial time approximation scheme for the offline class constrained bin packing problem (CCBP). In [11] and [12], Xavier and Miyazawa study packing problems where items have classes and items of different classes must be packed in different shelves inside bins, such that shelves have capacity constraints and are separated by non null width shelf divisors. The total size of shelves must satisfy the capacity of the bin. In [9], Schachnai and Tamir presented algorithms for the on-line CCBP problem, but they consider that all items have equal size. In this case, where all items have equal size, they provide a lower bound of 2 to the problem and also algorithms that get a competitive ratio of 2.

Results: In this paper we generalize the work presented by Schachnai and Tamir [9], since we consider the on-line CCBP problem where items can have different sizes. We show that the bounded space on-line CCBP problem cannot have a constant competitive ratio. If any item of the instance have size at least $\varepsilon < 1$ we show that any algorithm cannot have a competitive ratio better than $O(1/C\varepsilon)$. For the unbounded space case we show that the First-Fit algorithm is 3-competitive, but it cannot have competitive ratio better than 2.7. We also present an algorithm with competitive ratio in $[2.666, 2.75]$.

Organization: In Section 2, we present a basic notation used in this paper. In Section 3, we present lower bounds for the competitive ratio of any algorithm for the bounded space on-line CCBP problem. In Sections 4 we consider the First-Fit algorithm and in section 5, we present an algorithm with competitive ratio in $[2.666, 2.75]$.

2 Notation

In this section we present a basic notation used in this paper. We refer to an instance $I = (L, s, c, Q, C)$ only by I and L the sequence of incoming items. We may say that an item $a \in I$ by meaning that $a \in L$. We use $s(I) = s(L) = \sum_{a \in L} s(a)$.

Given two sequences $L_a = (a_1, \ldots, a_n)$ and $L_b = (b_1, \ldots, b_m)$, we denote the concatenation of these two lists by $L_a \| L_b$, i.e, $L_a \| L_b = (a_1, \ldots, a_n, b_1, \ldots, b_m)$. Given a packing $\mathcal{P}$ we denote by $|\mathcal{P}|$ the number of bins in $\mathcal{P}$.

During the text we use the terms color and class with the same meaning. We say that a bin is *colored* if it contains items of C different classes. In this case, this bin cannot pack any other item of a different class. We say that a bin is *full* if the total size of the items packed inside it is 1.

3 Lower bounds for bounded space algorithms

In this section we present inapproximability results for the bounded space on-line CCBP problem. In this case, the basic strategy is to compare the result obtained by the algorithm with the optimum off-line packing.

Theorem 3.1 *The K-bounded space on-line* CCBP *problem does not have algorithms with constant competitive ratio.*

Proof. Let $\mathcal{A}$ be an algorithm for the K-bounded space on-line CCBP problem. Consider an instance I, such that $|L| = n^2K$, $Q = nK$, and n is divisible by C. The list L have nK different classes and all items have size $1/Cn$. Consider that $L = L_1\|\ldots\|L_n$, where each $L_i = (a_1, \ldots, a_{nK})$ is a sequence of nK items where each a_j has class j.

Let t_i the time immediately after the algorithm has packed the list L_i. At time t_1 the algorithm $\mathcal{A}$ can have at most K open bins. Since each item of the first sequence is of a different class, the algorithm uses at least nK/C bins to pack L_1, where at least $nK/C - K$ of these bins are closed. When the packing of the list L_2 starts, the algorithm has at most K open bins that can pack at most KC items of the sequence L_2. To pack this sequence, the algorithm uses at least $(Kn - KC)/C$ bins. This is valid for the other sequences $L_3, \ldots, L_n$.

Therefore, to pack the list L, the algorithm $\mathcal{A}$ uses at least

$$n(nK/C) - (n-1)K = n^2K/C - (n-1)K$$

bins.

Since all items have size $1/Cn$, an optimal off-line solution can use at most Kn/C bins, by packing Cn items in each bin. Therefore, the competitive ratio must be at least

$$\lim_{n\to\infty} \frac{n^2K/C - (n-1)K}{nK/C} = n.$$

$\square$

In Theorem 3.1, items may have arbitrary small sizes. If all items must have size at least some constant ε we may also obtain an inapproximability result using similar arguments. Notice that in this case, any simple algorithm has a competitive ratio of $1/\varepsilon$.

Theorem 3.2 *There is no algorithm for the K-bounded space on-line* CCBP *problem with competitive ratio better than $O(1/C\varepsilon)$ when all items have size at least $\varepsilon < 1$.*

Proof. Suppose that $1/\varepsilon$ divides n and we have the same instance as in Theorem 3.1, but every item has size equal to ε. In this case any algorithm uses at least $n^2K/C - (n-1)K$ bins. An optimal off-line packing can have items of a given class in $n\varepsilon$ bins. Then to pack L an optimal off-line algorithm uses at most $n^2K\varepsilon$ bins.

Therefore, the competitive ratio is at least

$$\lim_{n\to\infty} \frac{n^2K/C}{n^2K\varepsilon} - \frac{nK-K}{n^2K\varepsilon} = \frac{1}{C\varepsilon}.$$

$\square$

For the remaining of this paper we consider the unbounded space on-line CCBP problem.

4 The First-Fit Algorithm

Given an on-line algorithm $\mathcal{A}$ for the bin-packing problem, we can obtain an on-line algorithm $\mathcal{A}^*$ for the on-line CCBP problem in a straightforward manner. To pack the next item e, the algorithm $\mathcal{A}^*$ packs as follows: Let c_e be the class of the item e, $\mathcal{B}$ be the list of bins in the order they were opened. Let $\mathcal{B}_e$ be the list of bins of $\mathcal{B}$, in the same order of $\mathcal{B}$, where each bin has at least one item of class c_e or has items of at most $C-1$ different classes. The item e is packed with algorithm $\mathcal{A}$ into the bins of $\mathcal{B}_e$.

One of the most famous algorithm for the bin-packing problem is the First-Fit algorithm. This algorithm packs the next item into the first bin, in the order they were open, that has sufficient room for it.

In this section we show that the competitive ratio of the algorithm FF^* is in $[2.7, 3]$. Notice that the algorithm FF^* is on-line, since it only looks for the item it is packing and it is unbounded since it keeps all bins opened. In fact it closes a bin only if the bin is full.

Lemma 4.1 *Let I be an instance for the on-line CCBP problem such that every item has size at most ε. Let $\mathcal{P}$ be the set of bins generated by the algorithm FF^*, applied over the instance I, that are filled by less than $1-\varepsilon$. Then: (i) Each bin in $\mathcal{P}$, which is not the last generated bin, is colored. (ii) There is no items of a same color in two different bins of $\mathcal{P}$.*

Proof. Let B_1 be a bin in $\mathcal{P}$, B_l the last bin created by the algorithm FF^* and a_l an item packed in B_l. Since B_1 is filled with less than $1-\varepsilon$ and $s(a_l) \leq \varepsilon$, a_l was not packed in B_1 because it must be colored.

Now suppose there are two different bins B_1 and B_2 in $\mathcal{P}$ that are filled with less than $1-\varepsilon$ and there are items $a_i \in B_i$, $i = 1, 2$ with the same class. Without loss of generality, consider that B_1 was opened first. Since the maximum size of a_2 is ε and the algorithm FF^* tries to pack an item into the bins in the order they were opened, satisfying the size and class constraints, the item a_2 would be packed in the bin B_1. That is, a contradiction. □

Theorem 4.2 *The algorithm FF^* has a competitive ratio 3 for the on-line CCBP problem.*

Proof. Let I be an instance for the on-line CCBP problem. Let $\mathcal{P}$ be the packing generated by the algorithm FF^* applied over the instance I. Let $\mathcal{P}_1$ be the list of bins in $\mathcal{P}$ filled by at least $1/2$ and $\mathcal{P}_2$ the list of the remaining bins.

Since each bin in $\mathcal{P}_1$ is filled by at least $1/2$ we have

$$|\mathcal{P}_1|(1/2) \leq s(I) \leq \mathrm{OPT}(I). \tag{1}$$

Notice that each item in $\mathcal{P}_2$ has size at most $1/2$. By Lemma 4.1, each bin of $\mathcal{P}_2$, except perhaps the last, is colored. Therefore,

$$(|P_2| - 1)C \leq Q \leq C\,\mathrm{OPT}(I). \tag{2}$$

The last inequality is valid since an optimal packing uses at least $\lceil Q/C \rceil$ bins. From inequalities (1) and (2) we have

$$|\mathcal{P}| = |\mathcal{P}_1| + |\mathcal{P}_2| \leq 3\mathrm{OPT}(I) + 1.$$

□

Now, we show that the algorithm FF* cannot have a competitive ratio better than 2.7. We first give an intuitive lower bound of 2.666 and then we present the lower bound of 2.7.

Theorem 4.3 *There is an instance I_n, $n \geq 1$, for the on-line* CCBP *problem such that* $\frac{\mathrm{FF}^*(I_n)}{OPT(I_n)} \rightarrow 2.666$ *as* $n \rightarrow \infty$.

Proof. Let I be an instance with an input list of items $L = L_a \| L_b \| L_c \| L_d$. Let C be the maximum number of classes allowable in each bin. The list $L_a = (a_1, \ldots, a_{(C-1)6N})$ is such that each item a_i has class i, $i = 1, \ldots, (C-1)6N$ and each item has size α, which is a very small value. This list is followed by a list $L_b = (b_1, \ldots, b_{6N})$, where each item b_i has class $r = 6NC$, and size $1/7 + \varepsilon$. In the list $L_c = (c_1, \ldots, c_{6N})$ each item c_i has size $1/3 + \varepsilon$ and class r. Finally, in the list $L_d = (d_1, \ldots, d_{6N})$ each item d_i has size $1/2 + \varepsilon$ and class r.

The FF* algorithm packs the list L_a in $\frac{6N(C-1)}{C}$ bins, the list L_b in N bins, the list L_c in $3N$ bins and the list of L_d in $6N$ bins. The Figure 1 presents the different bins in the packing generated by the FF* algorithm.

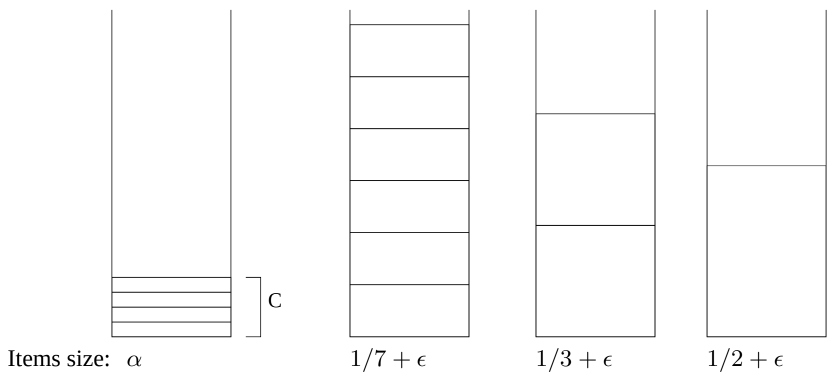

Figure 1: The bins generated by the FF* algorithm.

An optimal (off-line) solution uses at most $6N$ bins. This packing is obtained by packing one item of L_d, one item of L_c, one item of L_b and $C - 1$ items of the list L_a in only one bin.

This gives a lower bound of

$$\lim_{N,C \to \infty} \frac{\frac{(C-1)6N}{C} + 10N}{6N} = 2.666.$$

□

The previous lower bound can be improved using an intricate instance presented by Johnson et al. [4] that provides a lower bound of 1.7 for the FF algorithm in the bin packing problem.

Theorem 4.4 *The competitive ratio of the algorithm* FF* *is at least* 2.7.

Proof. To prove this theorem, we consider an instance I such that each bin can pack at most C different classes. The input list L is the concatenation of four lists: $L = L_a \| L_b \| L_c \| L_d$. In the list $L_a = (a_1, \ldots, a_{5N(C-1)})$, each item a_i has class i, for $i = 1, \ldots, 5N(C-1)$, and each item has size α, which is a very small value. The list L_a is followed by an instance similar to the one presented by Johnson et al. [4] that provides a lower bound of 1.7 for the FF algorithm in the bin packing problem. In the list $L_b = (b_1, \ldots, b_{5N})$ each item b_i has size $1/7 + y_i$, where $y_i \in \mathcal{R}$, for $i = 1, \ldots, 5N$. In the list $L_c = (c_1, \ldots, c_{5N})$ each item c_i has size $1/3 + w_i$, where $w_i \in \mathcal{R}$, for $i = 1, \ldots, 5N$. In the list $L_d = (d_1, \ldots, d_{5N})$ each item d_i has size $1/2 + \varepsilon$. All items in the lists L_b, L_c and L_d have class $5NC$.

The algorithm FF* generates a packing as the one presented in the proof of the Theorem 4.3, except that it packs only five items of the list L_b per bin. That is,

$$\mathrm{FF}^*(I) \geq \frac{5N(C-1)}{C} + N + 2.5N + 5N.$$

An optimal solution can use $5N + 2$ bins (see [4]), packing one item of each list L_b, L_c and L_d and $C - 1$ items of the list L_a.

Therefore, the competitive ratio of the algorithm FF* is at least

$$\lim_{N,C\to\infty} \frac{5N(C-1)/C + 8.5N}{5N + 2} = 2.7.$$

□

5 A 2.75-competitive algorithm

In this section we present an algorithm, which we denote by $\mathcal{A}_C$, with competitive ratio in the interval $(2.666, 2.75]$

ALGORITHM $\mathcal{A}_C(L, s, c, Q, C)$

1. Let $\mathcal{P}_i \leftarrow \emptyset$, for $i = 1, 2, 3$.
2. For each $e \in L$ do
3. if $s(e) \in (\frac{1}{2}, 1]$ then $k \leftarrow 1$.
4. if $s(e) \in (\frac{1}{3}, \frac{1}{2}]$ then $k \leftarrow 2$.
5. if $s(e) \in (0, \frac{1}{3}]$ then $k \leftarrow 3$.
6. Let $\mathcal{P}'_k$ the sublist of bins in $\mathcal{P}_k$ having items of class $c(e)$ or with at most $C - 1$ classes, preserving the order of the bins in $\mathcal{P}_k$.
7. If possible pack the item e into the bins $\mathcal{P}'_k$ using the algorithm FF*. Otherwise, pack e into a new empty bin in $\mathcal{P}_k$.
8. Return $\mathcal{P}_1 \| \mathcal{P}_2 \| \mathcal{P}_3$.

Figure 2: Algorithm $\mathcal{A}_C$.

To prove the competitive ratio of the algorithm $\mathcal{A}_C$, we use the following lemma, which proof can be found in [5].

Lemma 5.1 *Suppose X, Y, x, y are real numbers such that $x > 0$ and $0 < X < Y < 1$. Then*

$$\frac{x+y}{\max\{x,\ X\,x + Y\,y\}} \leq 1 + \frac{1-X}{Y}.$$

Theorem 5.2 *Algorithm $\mathcal{A}_C$ has a competitive ratio of 2.75.*

Proof. We divide the proof in two cases: when $C = 1$ and when $C \geq 2$.

If $C = 1$, the class information is irrelevant and the problem is exactly the bin packing problem. In this case each bin in the packing $\mathcal{P}_i$ if filled by at least $\frac{1}{2}$, except perhaps in the last bin of the list $\mathcal{P}_i$, for $i = 1, 2, 3$. Since the volume occupation is a lower bound for the size of an optimum packing, we have

$$(\mathcal{A}_C(I) - 3)\frac{1}{2} \leq s(I) \leq \mathrm{OPT}(I).$$

The proof of this case can be concluded with the following inequality

$$\mathcal{A}_C(I) \leq 2\mathrm{OPT}(I) + 3.$$

Now, consider the case when $C \geq 2$. Let L_i the list of items packed in $\mathcal{P}_i$, for $i = 1, 2, 3$.

Note that all bins of $\mathcal{P}_1$ have exactly one item with size greater than $\frac{1}{2}$. In fact we cannot pack more than one item of L_1 per bin. Therefore,

$$|\mathcal{P}_1| \leq \mathrm{OPT}(I) \tag{3}$$

$$\frac{1}{2}|\mathcal{P}_1| \leq s(L_1). \tag{4}$$

The packing $\mathcal{P}_2$ has exactly two items per bin, except perhaps the last, each item with size at least $\frac{1}{3}$. Therefore,

$$(|\mathcal{P}_2| - 1)\frac{2}{3} \leq s(L_2). \tag{5}$$

Let $\mathcal{P}_3'$ the set of bins in $\mathcal{P}_3$ that are filled by at least $\frac{2}{3}$ and $\mathcal{P}_3''$ the remaining bins (i.e., $\mathcal{P}_3'' = \mathcal{P}_3 \setminus \mathcal{P}_3'$). Using the same analysis performed for the packing $\mathcal{P}_2$, the following is valid

$$(|\mathcal{P}_3'| - 1)\frac{2}{3} \leq s(L_3'). \tag{6}$$

where L_3' is the set of items packed in $\mathcal{P}_3'$. Let $N_A = |\mathcal{P}_1|$ and $N_B = |\mathcal{P}_2| + |\mathcal{P}_3'| - 2$. Since $\mathrm{OPT}(I) \geq s(I) \geq s(L_1) + s(L_2 \| L_3')$ from inequalities (4)–(6) we have

$$\begin{aligned} \mathrm{OPT}(I) &\geq s(I) \geq s(L_1) + s(L_2 \| L_3') \\ &\geq \frac{1}{2}N_A + \frac{2}{3}N_B. \end{aligned} \tag{7}$$

From inequalities (3) and (7) we have

$$\mathrm{OPT}(I) \geq \max\{N_A, \frac{1}{2}N_A + \frac{2}{3}N_B\}. \tag{8}$$

From Lemma 5.1 we have that

$$|\mathcal{P}_1| + |\mathcal{P}_2| + |\mathcal{P}'_3| \leq \frac{N_A + N_B}{\max\{N_A, \frac{1}{2}N_A + \frac{2}{3}N_B\}}\mathrm{OPT}(I) + 2 \quad (9)$$
$$\leq 1.75\,\mathrm{OPT}(I) + 2. \quad (10)$$

Now, consider the packing $\mathcal{P}''_3$. Using a similar argument used in Lemma 4.1, we have

$$|\mathcal{P}''_3| - 1 \leq \frac{Q}{C} \leq \mathrm{OPT}(I). \quad (11)$$

The proof can be completed summing the inequalities (10) and (11).

$$\mathcal{A}_C(I) = |\mathcal{P}_1| + |\mathcal{P}_2| + |\mathcal{P}'_3| + |\mathcal{P}''_3|$$
$$\leq 1.75\,\mathrm{OPT}(I) + \mathrm{OPT}(I) = 2.75\,\mathrm{OPT}(I).$$

□

The same instance used to prove a lower bound for the algorithm FF^* in Theorem 4.3 is also valid for the $\mathcal{A}_C$ algorithm.

Theorem 5.3 *There is an instance I for the on-line* CCBP *problem such that* $\mathcal{A}_C(I)/OPT(I) \geq 2.666$.

6 Conclusions

In this paper we analyze the on-line class-constrained bin packing problem. We provide lower bounds for the bounded space version and two algorithms for the unbounded problem.

References

[1] M. Dawande, J. Kalagnanam, and J. Sethuranam. Variable sized bin packing with color constraints. *Eletronic Notes in Discrete Mathematics*, 7:4, 2001.

[2] S. Ghandeharizadeh and R. R. Muntz. Design and implementation of scalable continous media servers. *Parallel Computing Journal*, 24(1):91–122, 1998.

[3] P. S. Wu J. L. Wolf and H. Shachnai. Disk load balancing for video-on-demand-systems. *ACM Multimedia Systems Journal*, 5:358–370, 1997.

[4] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3:299–325, 1974.

[5] F. K. Miyazawa and Y. Wakabayashi. Parametric on-line algorithms for packing rectangles and boxes. *European Journal on Operational Research*, 150:281–292, 2003.

[6] S. S. Seiden. On the online bin packing problem. *Journal of ACM*, 49(5):640–671, 2002.

[7] H. Shachnai and T. Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6):313–338, 2001.

[8] H. Shachnai and T. Tamir. Multiprocessor scheduling with machine allotment and parallelism constraints. *Algorithmica*, 32(4):651–678, 2002.

[9] H. Shachnai and T. Tamir. Tight bounds for online class-constrained packing. *Theoretical Computer Science*, 321(1):103–123, 2004.

[10] A. van Vliet. An improved lower bound for online bin packing algorithms. *Inform. Process. Lett.*, 43:277–284, 1992.

[11] E. C. Xavier and F. K. Miyazawa. Approximation schemes for a class-constrained knapsack problem. Technical Report 12, Institute of Computing, State University of Campinas, april 2003.

[12] E. C. Xavier and F. K. Miyazawa. An approximation scheme for a one-dimensional bin packing problem with shelf divisions. Technical Report 8, Institute of Computing, State University of Campinas, august 2004.