Séminaire Lotharingien de Combinatoire XX (2021)
Article #YY, 12 pp.

Proceedings of the 33rd Conference on Formal Power Series and Algebraic Combinatorics (Ramat Gan)

Box-ball systems and RSK tableaux

Ben Drucker[1], Eli Garcia[2], Emily Gunawan[3], and Rose Silver[4]

[1] *Swarthmore College, Swarthmore, PA, USA*
[2] *Massachusetts Institute of Technology, Cambridge, MA, USA*
[3] *Department of Mathematics, University of Oklahoma, Norman, OK, USA*
[4] *Northeastern University, Boston, MA, USA*

Abstract. A box-ball system is a collection of discrete time states. At each state, we have a collection of countably many boxes with each integer from 1 to n assigned to a unique box; the remaining boxes are considered empty. A permutation on n objects gives a box-ball system state by assigning the permutation in one-line notation to the first n boxes. After a finite number of steps, the system will reach a so-called soliton decomposition which has an integer partition shape. We prove the following: if the soliton decomposition of a permutation is a standard Young tableau or if its shape coincides with its Robinson–Schensted (RS) partition, then its soliton decomposition and its RS insertion tableau are equal. We study the time required for a box-ball system to reach a steady state. We also generalize Fukuda's single-carrier algorithm to algorithms with more than one carrier.

Keywords: box-ball system, RSK correspondence, tableaux, Greene's theorem

1 Introduction

A *box-ball system* is a collection of discrete time states. At each state, we have a collection of countably many boxes with each integer from 1 to n assigned to a unique box; the remaining boxes are considered "empty." A permutation $\pi \in S_n$ gives a box-ball system state by assigning the permutation in one-line notation to the first n boxes. We apply a BBS move in the forward direction (letting time t increase by 1) by moving each integer from smallest to largest to its nearest empty space to the right. See Figure 1.

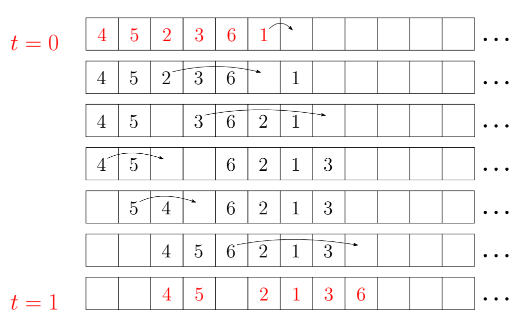

Figure 1: Performing a forward BBS move on $\pi = 452361$

A *soliton* is a consecutive increasing sequence which is preserved by all subsequent BBS moves. After a finite number of BBS moves, a box-ball system containing a configuration π will reach a steady state, decomposing into solitons whose sizes are weakly increasing from left to right, that is, forming an integer partition shape [8]. See Figure 2.

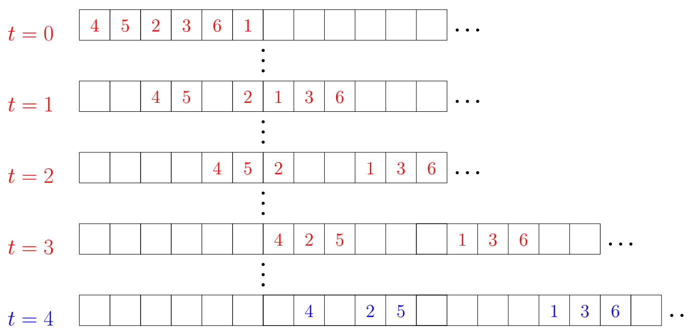

Figure 2: Forward BBS moves for $\pi = 452361$. Steady-state is achieved at $t = 3$.

From such a state, we can construct the *soliton decomposition* of a permutation, denoted SD, by stacking solitons. We obtain a tableau where each row is increasing but which may or may not be standard. For example, the soliton decomposition of the box-ball system containing $\pi = 452361$ shown in Figure 2 is

$$\mathrm{SD}(\pi) = \begin{array}{|c|c|c|}\hline 1 & 3 & 6 \\ \hline 2 & 5 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}.$$

The celebrated Robinson–Schensted (RS) insertion algorithm is a bijection $\pi \mapsto (\mathrm{P}(\pi), Q(\pi))$ from S_n onto pairs of standard Young tableaux of size n [7]. The tableau $\mathrm{P}(\pi)$ is called the *insertion tableau* or *P-tableau* of π, and the tableau $Q(\pi)$ is called the *recording tableau* or *Q-tableau* of π. For example, if $\pi = 452361$, then

$$\mathrm{P}(\pi) = \begin{array}{|c|c|c|}\hline 1 & 3 & 6 \\ \hline 2 & 5 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}, \quad \mathrm{Q}(\pi) = \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 4 \\ \cline{1-2} 6 \\ \cline{1-1}\end{array}$$

The *reading word* of a Young tableau is the permutation formed by concatenating the rows of the tableau from bottom to top. E.g., the tableau $\mathrm{P}(\pi)$ has reading word 425136.

1.1 BBS soliton partition and localized version of Greene's theorem

Greene famously showed that the RS partition of a permutation and its conjugate record the numbers of disjoint unions of increasing and decreasing sequences of the permutation [3, Theorem 3.1]. Lewis, Lyu, Pylyavskyy, and Sen recently showed that the partition shape of the soliton decomposition of a permutation and its conjugate record a pair of similar collections of permutation statistics [6, Lemma 2.1]. They studied an alternate version of the box-ball system, so we reframe their result to match our box-ball convention.

Definition 1.1. For $\pi = \pi_1 \cdots \pi_n \in S_n$ and $k \geq 1$, we define

$$I_k = \max_{\pi = u_1 | \cdots | u_k} \sum_{j=1}^{k} i(u_j),$$

where $i(u_j)$ is the length of the longest increasing subsequence in u_j and the maximum is taken over ways of writing π as a concatenation $u_1 \mid \cdots \mid u_k$ of consecutive subsequences. That is, we consider all ways to break π into k consecutive subsequences, sum the $i(u_j)$ values for each way, and let I_k be the maximum sum. We also define

$$D_k = \max_{\pi = u_1 \sqcup \cdots \sqcup u_k} \sum_{j=1}^{k} d(u_j),$$

where $d(u_j) = 1 + |\{\text{descents in } u_j\}|$ and the maximum is taken over ways to write π as the union of disjoint subsequences u_i of π. Notice that we only require $u_1, \ldots, u_k$ to be disjoint, *not* consecutive. We then form the sequences

$$\lambda_{BBS}(\pi) = (I_1, I_2 - I_1, I_3 - I_2, \ldots) \text{ and } \mu_{BBS}(\pi) = (D_1, D_2 - D_1, D_3 - D_2, \ldots).$$

Lemma 1.2 (Corollary of [6, Lemma 2.1])**.** *If $\pi \in S_n$, then $\mathrm{SD}(\pi)$ has the shape of a partition and this shape $\mathrm{sh}\,\mathrm{SD}(\pi)$ is equal to $\lambda_{BBS}(\pi)$. Furthermore, the conjugate of $\mathrm{sh}\,\mathrm{SD}(\pi)$ is equal to $\mu_{BBS}(\pi)$.*

1.2 When the soliton decomposition and RS insertion tableau coincide

In general, the soliton decomposition and the RS insertion tableau of a permutation do not coincide. We study the SD-equivalence classes of S_n using the notion of reading words, standard tableaux, and Knuth moves.

First, we show that reading words of standard tableaux have well-behaved soliton decomposition.

Theorem 1.3. *A permutation r reaches its soliton decomposition at time $t = 0$ if and only if r is the reading word of a standard Young tableau.*

In particular, if r is the reading word of a tableau T, then $\mathrm{SD}(r) = T$ and so $\mathrm{SD}(r) = T = \mathrm{P}(r)$. This shows that there are at least as many SD-equivalence classes as standard Young tableaux.

In Section 2.1, we generalize Theorem 1.3 to standard skew tableaux.

Surprisingly, having a standard tableau for a soliton decomposition or having a soliton decomposition shape which equals the RS partition shape is enough to guarantee that the soliton decomposition and the RS insertion tableau coincide.

Theorem 1.4. *Suppose π is a permutation. Then the following are equivalent:*

1. $\mathrm{SD}(\pi) = \mathrm{P}(\pi)$.
2. $\mathrm{SD}(\pi)$ *is a standard tableau.*
3. *The shape of* $\mathrm{SD}(\pi)$ *equals the shape of* $\mathrm{P}(\pi)$.

See Section 3 for a proof of Theorem 1.4. The proof that (3) implies (2) was suggested to the authors by Darij Grinberg.

1.3 Three types of Knuth moves

The RS insertion tableau is preserved under any Knuth move [4]. In contrast, the soliton decomposition is only preserved under certain types of Knuth moves.

Definition 1.5 (Knuth Moves)**.** Suppose $\pi, \sigma \in S_n$ and $x < y < z$.

- π and σ differ by a Knuth relation of the **first kind** (K_1) if
$$\pi = \pi_1 \dots yxz \dots \pi_n \text{ and } \sigma = \pi_1 \dots yzx \dots \pi_n$$
- π and σ differ by a Knuth relation of the **second kind** (K_2) if
$$\pi = x_1 \dots xzy \dots x_n \text{ and } \sigma = x_1 \dots zxy \dots x_n$$
- π and σ differ by Knuth relations of **both kinds** (K_B) if
$$\pi = x_1 \dots y_1 xzy_2 \dots x_n \text{ and } \sigma = x_1 \dots y_1 zxy_2 \dots x_n$$
where $x < y_1 < z$ and $x < y_2 < z$.

Using the localized version of Greene's Theorem given in Section 1.1, we prove a partial characterization of the shape of SD in terms of types of Knuth moves.

Theorem 1.6. *If π and w are related by a sequence of K_1 or K_2 moves (but not K_B), then* $\mathrm{sh}\,\mathrm{SD}(\pi) = \mathrm{sh}\,\mathrm{SD}(w)$. *If π and w are related by a sequence of Knuth moves containing an odd number of K_B moves, then* $\mathrm{sh}\,\mathrm{SD}(\pi) \neq \mathrm{sh}\,\mathrm{SD}(w)$.

This allows us to use Knuth moves to find more permutations whose soliton decomposition and RS insertion tableau coincide.

Corollary 1.7 (Corollary of Theorem 1.4 and Theorem 1.6)**.** *Suppose $\pi \in S_n$ is a sequence of K_1 or K_2 moves (but not K_B) away from the reading word of a standard tableau T. Then* $\mathrm{SD}(\pi) = \mathrm{P}(\pi) = T$. *If $\pi \in S_n$ is related to the reading word of a standard tableau by a sequence of Knuth moves such that an odd number of the moves are K_B moves, then* $\mathrm{SD}(\pi) \neq \mathrm{P}(\pi) = T$.

Proposition 1.8. *Suppose $\pi \in S_n$ is the reading word of a standard tableau. Let π' be a permutation one K_1 or K_2 (but not K_B) move away from π. Then π' reaches its steady-state after one BBS move.*

2 Steady states

We study the steady-state configurations and the minimum time-steps to go from a permutation to its soliton decomposition.

2.1 Standard tableaux of skew shapes

A BBS state can be represented as an array containing the integers from 1 to n as follows: scanning the boxes from right to left, each string of increasing integers becomes a row in the array. A string of k empty boxes indicates that the next row below should be shifted k steps to the left. Note that this array has increasing rows but not necessarily increasing columns; it also may not have a valid skew shape. The following is a generalization of Theorem 1.3.

Theorem 2.1. *A BBS configuration C is in steady-state if and only if the associated array has rows of weakly decreasing length, and has increasing columns.*

Note that in this case, the array will be a standard skew tableau.

Example 2.2 (of Theorem 2.1). Let $\pi = 521643$. The soliton decomposition $\mathrm{SD}(\pi)$ is the tableau given in Figure 3. Note that $C = \ldots e52e6413ee$ is a steady-state configuration, where we represent empty boxes with the symbol e. The configuration C yields the standard skew tableau in Figure 4. Conversely, if given the skew tableau in Figure 4 (with no knowledge of the original permutation), we may conclude the corresponding BBS configuration, $52e6413$, is in steady-state.

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 4 \\ \cline{1-1} 6 \\ \cline{1-1} 2 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array}$$

Figure 3: $\mathrm{SD}(\pi)$

$$\begin{array}{ccc} & \boxed{1} & \boxed{3} \\ & \boxed{4} & \\ & \boxed{6} & \\ \boxed{2} & & \\ \boxed{5} & & \end{array}$$

Figure 4: Resultant skew tableau

2.2 Permutations with n–3 time steps

We use the RS correspondence to associate a permutation in S_n to each standard tableau of shape $(n-3, 2, 1)$ and show that its box-ball steady-state value is $n-3$. We conjecture that all other permutations in S_n have steady-state value smaller than $n-3$.

Definition 2.3. If $n \geq 5$, let $Q_0 := Q_0(n)$ denote the tableau

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline n & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ldots\begin{array}{|c|c|}\hline n-2 & n-1 \\ \hline\end{array}.$$

Let $S_n(Q_0)$ be the set of permutations $\pi \in S_n$ such that its recording tableau $Q(\pi)$ is equal to Q_0.

Example 2.4. For $n = 5$, the five permutations of this set are the following.

45132 25143 35142 45231 35241

For $n = 6$, the sixteen permutations of this set are as follows.

451362	351462	352461	261354	461253	261453	461352	362451
251463	452361	561243	361254	561342	361452	562341	462351

Remark 2.5. It follows from Definition 2.3 that the RS algorithm induces a bijection from $S_n(Q_0)$ to the set of standard tableaux of shape $(n-3, 2, 1)$, see [10].

Proposition 2.6. *Every permutation in $S_n(Q_0)$ has steady-state value of $n-3$.*

The following conjecture has been computationally verified up to $n = 10$.

Conjecture 2.7. *A permutation not in $S_n(Q_0)$ has steady-state value smaller than $n-3$.*

3 Proof of Theorem 1.4

3.1 Fukuda's carrier algorithm as a sequence of Knuth moves

Some of our proofs use an algorithm called the *carrier algorithm* which was first introduced in [9] and generalized in [2, Section 3.3]. The carrier algorithm is used to calculate the $t = k + 1$ state of a BBS given the $t = k$ state. In section 4.1, we introduce a multi-carrier generalization of the carrier algorithm called the M-carrier algorithm (Algorithm 4.1). When restricted to $M = 1$, our algorithm coincides with the original carrier algorithm.

Example 3.1 (Carrier Algorithm [2]). We compute the $t = 3$ configuration of the box-ball system from Figure 2 by applying the carrier algorithm to the $t = 2$ configuration. Following Algorithm 4.1 for $M := 1$, we set $B := eeee452ee136\ldots$. The carrier algorithm then proceeds as follows:

begin Process 1: insertion process

$$\underbracket{eeeeee}\,eeee452ee136$$
$$e\,\underbracket{eeeeee}\,eee452ee136$$
$$\vdots$$
$$eeee\,\underbracket{eeeeee}\,452ee136$$
$$eeeee\,\underbracket{4eeeee}\,52ee136$$
$$eeeeee\,\underbracket{45eeee}\,2ee136$$
$$eeeeee4\,\underbracket{25eeee}\,ee136$$
$$eeeeee42\,\underbracket{5eeeee}\,e136$$
$$eeeeee425\,\underbracket{eeeeee}\,136$$
$$\vdots$$
$$eeeeee425eee\,\underbracket{136eee}$$

end insertion process

begin Process 2: flushing process

$$eeeeee425eee\,\underbracket{136eee} \leftarrow e$$
$$eeeeee425eee1\,\underbracket{36eeee} \leftarrow e$$
$$eeeeee425eee13\,\underbracket{6eeeee} \leftarrow e$$
$$eeeeee425eee136\,\underbracket{eeeeee}$$

end flushing process

After each insertion, the sequence in the carrier is weakly increasing.

Remark 3.2 ([2, Remark 4]). The carrier algorithm can be viewed as a sequence of Knuth moves (if we think of the elements in and to the left of the carrier as a single sequence.) Consider the insertion of p into the carrier. If the carrier contains a number greater than p, then the insertion process is equivalent to applying a sequence of K_1 moves

$$\underbracket{\cdots C_p z_1 \cdots z_{l-1} z_l}\ p$$
$$\underbracket{\cdots C_p z_1 \cdots z_{l-1} p z_l}$$
$$\vdots$$
$$\underbracket{\cdots C_p p z_1 \cdots z_{l-1} z_l}$$

followed by a sequence of K_2 moves:

$$\underbracket{x_1 \cdots x_{m-1} x_m C_p p \cdots}$$
$$\underbracket{x_1 \cdots x_{m-1} C_p x_m p \cdots}$$
$$\vdots$$
$$C_p\,\underbracket{x_1 \cdots x_{m-1} x_m p \cdots}.$$

Otherwise, if p is greater than or equal to every element in the carrier, we apply the trivial transformation:

$$\underbrace{x \cdots}\ p$$
$$x\ \underbrace{\cdots p}\,.$$

Lemma 3.3 ([2, Theorem 3.1])**.** *The RS insertion tableau is a conserved quantity under the time evolution of the BBS, that is, it is preserved under each BBS move.*

3.2 Soliton decompositions and RSK tableaux

The following gives a characterization of permutations whose soliton decompositions are equal to their RS insertion tableaux.

Theorem 1.4. *Let π be a permutation. Then the following are equivalent:*

1. $\mathrm{SD}(\pi) = \mathrm{P}(\pi)$.
2. $\mathrm{SD}(\pi)$ *is a standard tableau.*
3. *the shape of* $\mathrm{SD}(\pi)$ *equals the shape of* $\mathrm{P}(\pi)$.

Lemma 3.4 (Due to Darij Grinberg)**.** *Suppose S is a row-strict tableau of a partition, that is, every row is increasing (with no restrictions on the columns). Let r be the reading word of the tableau S. Let $\mathrm{P}(r)$ be the RS insertion tableau of r. If the shape of S equals the shape of $\mathrm{P}(r)$, then S is standard.*

Proof of Theorem 1.4. Certainly (1) implies (2) and (3). First, we show that (2) implies (1). Suppose that $\mathrm{SD}(\pi)$ is a standard tableau. Let r denote the reading word of $\mathrm{SD}(\pi)$. We know that r is the order in which the elements of π are configured once we reach a steady state. By Lemma 3.3, $\mathrm{P}(\pi) = \mathrm{P}(r)$. Since r is the reading word of $\mathrm{SD}(\pi)$, we have $\mathrm{P}(r) = \mathrm{SD}(\pi)$ by Theorem 1.3. Therefore $\mathrm{P}(\pi) = \mathrm{SD}(\pi)$.

Next, we show that (3) implies (2). Suppose that the shape of $\mathrm{SD}(\pi)$ equals the shape of $\mathrm{P}(\pi)$. Let r be the reading word of $\mathrm{SD}(\pi)$. Lemma 3.3 tells us that the RSK insertion tableau is preserved under a sequence of box-ball moves, so $\mathrm{P}(\pi) = \mathrm{P}(r)$ and, in particular, $\mathrm{sh}\,\mathrm{P}(\pi) = \mathrm{sh}\,\mathrm{P}(r)$. By assumption, we have $\mathrm{sh}\,\mathrm{SD}(\pi) = \mathrm{sh}\,\mathrm{P}(r)$. Since $\mathrm{SD}(\pi)$ is a row-strict tableau and r is the reading word of $\mathrm{SD}(\pi)$, Lemma 3.4 tells us that $\mathrm{SD}(\pi)$ is standard. □

4 Multi-carrier algorithms

In this section, we give insertion algorithms that can help us study steady-states and soliton decompositions.

4.1 M-carrier algorithm

In this section, we define the *M-carrier algorithm* which is equivalent to performing the carrier algorithm M times (Proposition 4.3). In addition to improving the efficiency of the box-ball system calculations, the M-carrier algorithm enables us to compare the RSK-insertion algorithm and the box-ball system more directly. Given a large enough M, the M-carrier algorithm gives us an RSK-like insertion algorithm which sends a permutation to its soliton decomposition.

Algorithm 4.1 (The M-carrier algorithm)**.**

begin M-carrier algorithm
  Set $e := n+1$
  Set $B :=$ the $t = k$ configuration of the BBS, replacing empty boxes with e's, so that the first (leftmost) element of B is the integer in the first (leftmost) non-empty box in the configuration and the last (rightmost) element of B is the integer in the last (rightmost) non-empty box of the configuration at time k.
  Denote B_i as the i^{th} leftmost element of B and let there be ℓ elements of B.
  Fill M adjacent "carriers"—depicted $\sqcup$—with n copies of e.
  Denote this string of carriers $\mathcal{C}$
  Denote the rightmost carrier c_1, and in general, the j^{th} rightmost carrier c_j.
  Write B to the right of $\mathcal{C}$
  begin Process 1: insertion process
    for all i in $\{1, 2, \ldots, \ell\}$ **do**
      Set $p := B_i$
      begin element ejection process
        for all j in $\{1, 2, \ldots, M\}$ **do**
          if an element in c_j is larger than p **then**
            Set $s :=$ the smallest element in c_j larger than p
            Eject s by replacing it with p and setting $p := s$
          else
            Set $s :=$ the smallest element in c_j.
            Remove s from c_j
              ▶ Note: There are now $n-1$ elements in c_j.
            Place p in the rightmost location in c_j.
              ▶ Note: There are now n elements in c_j.
            Set $p := s$
          end if
          if $j = M$ **then**
            Place p to the left of $\mathcal{C}$
          end if

```
|   |   |   |   end for
|   |   |   end element ejection process
|   |   end for
|   end Process 1: insertion process
|   begin Process 2: flushing process
|   |   while there are non-e elements in C do
|   |   |   Set p := e. Perform the element ejection process
|   |   end while
|   end Process 2: flushing process
|      ▶ Note: The elements to the left of C correspond to the t = k + 1 state of the BBS
end M-carrier algorithm
```

Example 4.2. We apply the M-carrier algorithm (with $M = 3$) to $\pi = 361425$.

begin Process 1: insertion process

$eeeeee\ eeeeee\ eeeeee\ 361425$

$e\ eeeeee\ eeeeee\ 3eeeee\ 61425$

$ee\ eeeeee\ eeeeee\ 36eeee\ 1425$

$eee\ eeeeee\ 3eeeee\ 16eeee\ 425$

$eeee\ eeeeee\ 36eeee\ 14eeee\ 25$

$eeeee\ 6eeeee\ 34eeee\ 12eeee\ 5$

$eeeee6\ 3eeeee\ 4eeee\ 125eee$

end insertion process

begin Process 2: flushing process

$eeeee6\ 3eeeee\ 4eeeee\ 125eee \leftarrow e$

$eeeee6e\ 34eeee\ 1eeeee\ 25eeee \leftarrow e$

$eeeee6e3\ 4eeeee\ 12eeee\ 5eeeee \leftarrow e$

$eeeee6e34\ eeeeee\ 125eee\ eeeeee \leftarrow e$

$eeeee6e34e\ 1eeeee\ 25eeee\ eeeeee \leftarrow e$

$eeeee6e34ee\ 12eeee\ 5eeeee\ eeeeee \leftarrow e$

$eeeee6e34eee\ 125eee\ eeeeee\ eeeeee \leftarrow e$

$eeeee6e34eee1\ 25eeee\ eeeeee\ eeeeee \leftarrow e$

$eeeee6e34eee12\ 5eeeee\ eeeeee\ eeeeee \leftarrow e$

$eeeee6e34eee125\ eeeeee\ eeeeee\ eeeeee$

end flushing process

Proposition 4.3. *Performing the M-carrier algorithm (with M carriers) is equivalent to performing the 1-carrier algorithm M times. In particular, if $\pi \in S_n$, applying algorithm 4.1 to π yields the box-ball configuration of π at $t = M$.*

Proof. Ejecting an element from a carrier c_i and then immediately inserting it into the next carrier c_{i+1} is equivalent to ejecting all the elements from c_i, forming a sequence and then inserting that sequence into c_{i+1}. □

4.2 Infinite-carrier algorithm

We define the *infinite-carrier algorithm* to be the same as Algorithm 4.1, but with an infinite number of carriers, so an entry is always in some carrier at every step. (This is in contrast to the M-carrier algorithm, where an entry may be ejected to the left of the carriers.) Unfortunately, it's not always possible to obtain a soliton decomposition this way.

Theorem 4.4. *Let w be a permutation and let $\sigma_1, \sigma_2, \ldots \sigma_\ell$ be the solitons of a box-ball system containing w as a configuration.*
(1) In the infinite carrier algorithm, for each soliton σ_i, there exists a smallest positive number r_i such that, after inserting all the elements of w and r_i copies of e, the soliton σ_i is completely and solely contained in a carrier.
(2) Let $s_1, s_2, \ldots, s_\ell$ be the lengths of the respective solitons. If $gcd(s_i, s_j)$ divides $r_i - r_j$ for all i and j, then there exists a unique number of e's (mod $lcm\{s_1, s_2, \ldots, s_\ell\}$) such that the infinite carrier algorithm puts the solitons of a permutation in separate carriers (i.e., the infinite-carrier algorithm yields the box-ball soliton decomposition of w).

Example 4.5. Let $\pi = 24513$. The box-ball system containing π has solitons 135 and 24, which have lengths 3 and 2 respectively (with $\mathrm{lcm}\{3, 2\} = 6$). Since all the (pairwise) greatest common divisors of the soliton lengths are 1, there exists a unique number of e's (mod 6) such that the infinite carrier algorithm puts the solitons of a permutation in separate carriers. When one completes the infinite-carrier algorithm, after all entries of π and $0 + 6k$ of e's are inserted, the solitons of π are sorted into separate carriers:

begin Process 1: insertion process

... *eeeee* *eeeee* 24513
... *eeeee* 2*eeee* 4513
... *eeeee* 24*eee* 513
... *eeeee* 245*ee* 13
... *eeeee* 2*eeee* 145*ee* 3
... *eeeee* **24***eee* **135***ee*

end insertion process

begin Process 2: flushing process

... *eeeee* *eeeee* **24***eee* **135***ee* ← *e* #1
... *eeeee* 2*eeee* 14*eee* 35*eee* ← *e* #2
... *eeeee* 24*eee* 13*eee* 5*eeee* ← *e* #3
... *eeeee* 2*eeee* 4*eeee* 135*ee* *eeeee* ← *e* #4
... *eeeee* 24*eee* 1*eeee* 35*eee* *eeeee* ← *e* #5
... *eeeee* 2*eeee* 4*eeee* 13*eee* 5*eeee* *eeeee* ← *e* #6
... *eeeee* **24***eee* *eeeee* **135***ee* *eeeee* *eeeee* ← *e* #7
⋮ ⋮

end flushing process

Acknowledgements

This research started during the University of Connecticut 2020 math REU which was supported by NSF (DMS-1950543). We thank Aubrey Rumbolt for working with us during the REU. Our project was inspired by a blog post [5] for the University of Minnesota's Open Problems in Algebraic Combinatorics (OPAC) conference and conversations with Joel Lewis. We thank Ian Whitehead for serving as a faculty mentor to the first author's research course in Fall 2020, for careful reading of this draft, and for many helpful suggestions. We also thank Rei Inoue for answering our questions about box-ball system literature. Special thanks to Darij Grinberg for proving one of our conjectures. This work also benefited from computation using SageMath [1] and the High Performance Computing (HPC) facility at University of Connecticut.

References

[1] The Sage Developers. *Sage Mathematics Software (Version 9.1)*. The Sage Development Team. 2020. URL: `http://www.sagemath.org`.

[2] K. Fukuda. "Box-ball systems and Robinson-Schensted-Knuth correspondence". In: *Journal of Algebraic Combinatorics* 19.1 (2004), pp. 67–89.

[3] C. Greene. "An extension of Schensted's theorem". In: *Advances in Math.* 14 (1974), pp. 254–265.

[4] D. E. Knuth. "Permutations, matrices, and generalized Young tableaux". In: *Pacific J. Math.* 34 (1970), pp. 709–727.

[5] J. Lewis. *A localized version of Greene's theorem*. `https://realopacblog.wordpress.com/2019/11/24/a-localized-version-of-greenes-theorem/`. [Online; accessed 1-November-2020].

[6] J. Lewis et al. *Scaling limit of soliton lengths in a multicolor box-ball system*. Preprint `arXiv:1911.04458`. 2019.

[7] C. Schensted. "Longest increasing and decreasing subsequences". In: *Canadian J. Math.* 13 (1961), pp. 179–191.

[8] D. Takahashi. "On some soliton systems defined by using boxes and balls". In: *Proceedings of the international symposium on nonlinear theory and its applications (NOLTA'93)*. 1993, pp. 555–558.

[9] D. Takahashi and J. Matsukidaira. "Box and ball system with a carrier and ultradiscrete modified KdV equation". In: *Journal of Physics A: Mathematical and General* 30.21 (1997), p. L733.

[10] *The On-Line Encyclopedia of Integer Sequences*. `http://oeis.org/A077415`. Number of standard tableaux of shape (n-1,2,1) [Online; accessed 1-November-2020].