Everything I know about Subset Sum Reconfiguration

Robin Houston · July 16, 2013

This is a brief summary of everything I know about the subset sum reconfiguration problem as of four o'clock in the afternoon of Monday the 15th of July. It is brief by necessity, because I know so little. **Update on Tuesday**: I found out another thing.

Proposition 1. *Subset sum reconfiguration is weakly NP-hard.* (*Demaine–Ito*)

Proof. The ordinary subset sum problem may be reduced to subset sum reconfiguration. Given elements $a_1, \dots, a_m$ with target sum t, build an SSR instance with elements $a_1, \dots, a_m$, t_1 and t_2, threshold values $k = t$ and $c = 2t$, starting configuration t_1 and ending configuration t_2. At some point element t_1 must move from inside to outside: the first time it does, the inside set during the move must consist of some collection of a elements that sum to t. □

Proposition 2. *For all $n > 1$, there is an n-element SSR instance whose configuration graph has diameter $2n - 2$.*

Proof. We exhibit in terms of a parameter m:

- an instance with $2m + 2$ elements and a shortest solution with $4m + 2$ steps;
- if $m > 0$, an instance with $2m + 1$ elements and a shortest solution with $4m + 1$ steps.

The instance with $2m + 2$ elements has elements $a_1, \dots, a_m$, $b_1, \dots, b_m$, x and y. For the weights, let $w(a_i) = m + 1$ and $w(b_i) = m + 2$ for all i. Let $w(x) = w(y) = m(m + 2)$. The starting configuration is $a_1, \dots, a_m, x$ and the target configuration is $a_1, \dots, a_m, y$. For threshold values let $k = m(m + 2)$ and $c = 2m(m + 2)$.

The first time x moves, the weights of the remaining elements inside must sum to $m(m + 2)$. In particular, these weights must sum to m modulo $m + 1$, hence the inside elements at this point must include all the b's. Since the weight of x plus the b's equals the capacity c, the elements inside at this point must therefore be precisely all the b's. So all the a's and b's have to move twice, and x and y once each, for a total of $4m + 2$ moves.

To obtain the instance with $2m + 1$ elements, remove the element a_1. □

Remark 3. *The lower bound exhibited in Proposition 2 is optimal for $n < 8$. I have verified this by exhaustive enumeration using the program `ssr_graphs.py`.*

Proposition 4. *For all m, there is a $(5m + 3)$-element SSR instance that has an element that must move at least $2m + 1$ times in any solution sequence.*

Proof. The elements are $x_1, \dots, x_{2m+1}$ of weight 3, $a_1, \dots, a_{3m+1}$ of weight 2, and b of weight 1. Let $k = 6m$ and $c = 6m + 3$. The starting configuration is $x_1, \dots, x_{2m+1}$, and the target is $a_1, \dots, a_{3m+1}, b$. For $i = 0, 1, \dots, 2m$, consider the first move after which there are precisely $2m - i$ of the x elements inside. That means the a's and possibly b inside at this point must sum to $3i$. If i is odd then, since b is the only element with odd weight, b must be inside at this point; similarly if i is even then b must be outside. So element b moves at least $2m$ times. Since it must end up inside, it must move an odd number of times altogether, so at least $2m + 1$ times. □

New finding

The construction of Proposition 4 can be modified to give a quadratic lower bound for the diameter of the reconfiguration graph.

Proposition 5. *For all m, there is a* $(6m+2)$*-element SSR instance that requires* $2m^2+6m+2$ *moves.*

Proof. The elements are $x_1, \ldots, x_{2m+1}$ of weight $3m$, $a_1, \ldots, a_{3m+1}$ of weight $2m$, and $b_1, \ldots, b_m$ of weight 1. Let $k = 6m^2$ and $c = 6m^2 + 3m$. The starting configuration is $x_1, \ldots, x_{2m+1}$, and the target is $a_1, \ldots, a_{3m+1}, b_1, \ldots, b_m$. For $i = 0, 1, \ldots, 2m$, consider the first move after which there are precisely $2m - i$ of the x elements inside. That means the a's and b's inside at this point must sum to $3im$. If i is odd then, since the a's all weigh $2m$, all the b's must be inside at this point; similarly if i is even then the b's must all be outside. So the b elements each move at least $2m$ times, and in fact $2m+1$ times since they move. Therefore the total number of moves must be at least $(2m+1)+(3m+1)+(2m+1)m$, which is equal to $2m^2 + 6m + 2$ as required. □