My Simulate Annealing Program (SAP) and the National Resident Matching Program (NRMP) algorithm both seek to match each of many individuals into one of many groups as optimally as possible with regard to preference of the entities involved in the match. However, there are distinct differences in their methodologies, including the way in which the initial state is generated, and the ways in which successor states are generated and evaluated.

The stochastic nature of simulated annealing means that the terminal state may be vastly different from the initial state, so the method by which the initial state is generated is trivial. All that matters is for each student to be assigned to a section and each section have 20 students assigned to it. In light of this, the SAP generates the initial state in the simplest way possible, by iterating through the list of students and assigning each to the first lab with an open slot. The NRMP, on the other hand, has a very specific initial state that must be the same for each run. That state is one in which all residents and hospitals are completely unmatched. It bears mentioning that the SAP does not allow states in which there are unmatched students or labs at all.

When it comes to generating and evaluating successors, the algorithms also differ. The SAP generates successors totally randomly and evaluates them on the basis of trying to minimize an evaluation function. The evaluation function gives the highest bonus for students assigned to their highest preference section and gives a penalty for students assigned to a section they have excluded. The lab sections themselves do not impact the evaluation function. Successive states are generated by first choosing two lab sections at random, and from each removing one student at random. Each removed student is then added to the other lab. This randomness means that there can be wide fluctuations in the value of the evaluation function between successive states. In order to maintain the benefits of the sharp upswings while mitigating the risk of a significant down turn, the SAP utilizes a temperature construct. Successor states with better evaluations than the current state are always accepted. Successors with worse evaluations can still be accepted, but only probabilistically as a function of the temperature and how much worse the evaluation is.

Alternatively, the NRMP generates successor states sequentially and accepts any successor state that is valid based on its rules of successor generation. To generate successors the NRMP iterates through the list of unmatched residents and attempts to match each with a hospital in order of the resident’s preference. In contrast to the SAP, here the hospital has some say in the matter, and successor states in which a resident would be assigned to a hospital that has not ranked that resident are not accepted. If all the slots at a particular hospital are filled, a new resident will be assigned there only if they are more highly preferred by the hospital than a resident already assigned there. The lower preference resident will be displaced and returned to the unmatched queue to eventually be further evaluated for a match as described above. In this way, the NRPM only accepts successor states that improve the overall evaluation. Since successors that worsen the evaluation are never accepted, there is no need for a construct to manage how much of a decrease is allowed; another difference from the SAP. Additionally, since the NRMP generates successor states sequentially it is deterministic, whereas the SAP is stochastic.

While the SAP and the NRMP are designed to accomplish similar goals, they do so in noticeably different ways. They generate their initial states differently. They generate and evaluate successor states differently. And one is stochastic while the other is deterministic.