Saitou, K., Jakiela, M. J., "Automated Optimal Design of Mechanical Conformational Switches," *Artificial Life* **2**(2):129–156, 1995.

Automated Optimal Design of Mechanical Conformational Switches

Kazuhiro Saitou **Mark J. Jakiela** [*]
Computer-Aided Design Laboratory
Massachusetts Institute of Technology

Abstract

Bacteriophage viruses spontaneously self-assemble in the presence of their component parts (certain protein molecules). It is believed that conformational switches, interacting chemical bonding sites that allow tentative incorrect bonds, facilitate this randomized assembly process. A one-dimensional conformational switch is proposed and used to study the randomized assembly of mechanical parts. A genetic algorithm is used to search the space of parameterized switch designs to maximize the rate of a desired assembly.

Keywords: randomized assembly, mechanical conformational switches, genetic optimization

1 Introduction

1.1 Viruses and conformational switching

Viruses are one of the simplest known life forms. An interesting characteristic of some viruses is that they can spontaneously self-assemble [1]. If the correct components (various proteins molecules) of a virus are brought into proximity, chemical bonds will form in appropriate locations to create the virus structure. The organism arises, therefore, from disorganized numbers of its components in its environment.

Over the past several years, Thompson, Goel, and associated researchers have developed and documented a computer model that simulates the assembly and operation of a particular type of virus, the T4 bacteriophage [11, 12, 3]. A simulation builds a simplified model of a virus from building blocks which are abstractions of the protein molecules. These building blocks bond to one another at prespecified bond sites with prespecified bond strengths. As molecular bonding is a thermodynamic process, the bonding among building blocks, and hence the simulation, is governed by the minimization of free energy in the system. With an appropriately designed (various sizes and aspect ratios, bond site locations on each building block, and bond strengths) set of building blocks, Thompson and Goel have developed two- [11] and three- [12] dimensional augmented cellular automata models of viral assembly and operation.

A key aspect of the simulation scheme is the concept of *conformational switching.* If a building block has several bond sites, a conformational switch causes the bond at one site to influence a bond at another site. A bond made at one site can weaken or strengthen the bond potential at another site, or a bond at one site may destroy the bond at another site, in the manner of a latch.

Figure 1 shows an example of such conformational switching in the inhibition of protein enzymes by specific end-product inhibitor molecules. The catalytic activity of many protein enzymes are activated by binding their substrates, and forming enzyme-substrate complexes (Figure 1-a). The end-product inhibitor blocks the enzyme activity by reversibly binding to the enzyme at a site other than the active site (the region that binds the substrates). The binding of the inhibitor causes a conformational change at the active site on the enzyme, which prevents the enzyme from combining with its substrate (Figure 1-b). The chemical forces binding a specific end-product inhibitor to an enzyme are weak secondary forces such as hydrogen bonds, salt linkages, and van der Waals forces and do not involve covalent bonding. Hence, inhibition can be quickly reversed once the end-product concentration is reduced to a low level.

[*] Author to whom correspondence should be addressed: Massachusetts Institute of Technology, Department of Mechanical Engineering, 77 Massachusetts Avenue, Room 3-461a, Cambridge, MA 02139; Phone: (617) 253-6552; FAX:(617) 258-9346; email: {jakiela, kazu}@mit.edu.

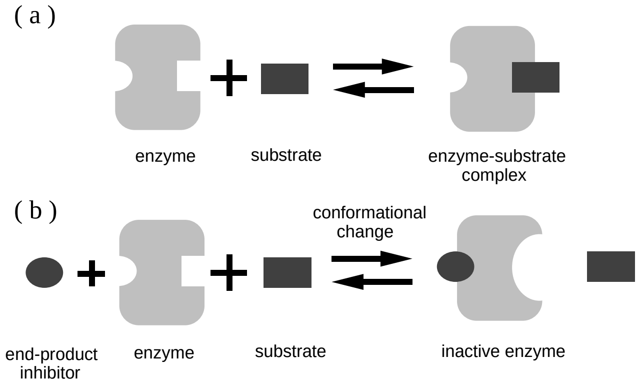


Figure 1: A biological example of conformational switching: inhibition of protein enzymes by specific end-product inhibitor molecules (abstracted from Figure 4-21 of [13])

In the simulation of the viral assembly, conformational switches allow the tentative assembly of subunits that may not be assembled together in the final complete correct virus. In the work of Thompson and Goel, the conformational switches are simply predetermined numerical relationships. We note, in contrast, the work of Penrose [9], who suggests several designs for mechanical conformational switches that are used in devices that self-reproduce. These conformational switches are binary, in the sense that a bond at one location breaks a bond existing at another location or prevents a bond from occurring at another location. The self-reproducing devices attach to some number of other "neutral" sub-devices. When the correct number and arrangement of sub-devices are linked, conformational switches cause the entire chain to cleave into two copies of the original self-reproducing device in a process akin to cell division.

Our interest in this paper is the design of such mechanical conformational switches. In particular, we will describe the optimization of a specific parameterized mechanical conformational switch design. The objective of this optimization is to maximize the assembly rate of a set of parts held together with these conformational switches. The assembly process considered is a randomized juxtaposition of the parts which is loosely similar to the randomized interaction of bacteriophage components prior to the viral assembly. We will first describe the switch design and how it functions. Then we will describe how the design is parameterized and how it can be represented with a genetic algorithm chromosome. An evolutionary (genetic) algorithm is used to search the space of possible switch designs to maximize the rate of a desired assembly. Several examples are presented and discussed. Before proceeding to these issues, however, we discuss potential practical uses for mechanical conformational switches and previous work in the area.

1.2 Practical applications of conformational switching

In the biological assembly of bacteriophage viruses, conformational switches are important in allowing the assembly of randomly arranged and randomly moving components. We will refer to this as a *randomized* assembly process. Similarly, in domains other than the biological, we feel that conformational switches will be important in any type of randomized assembly process. Practical applications of conformational switches will follow from practical applications of randomized assembly. What are some practical applications of randomized assembly?

- **Assembly at very high rates.** The common image of assembly is a robotic or human hand grasping one part at a time, assembling it into a product held on a fixture. There are natural limits to the speed with which this process can be carried out. Adding more human or robot hands is often not cost effective to achieve very high rates of assembly. Working in the domain of high volume assembly of consumer electromechanical products (cameras, videocassette recorders, *etc.*), we developed the concept of mass aggregate assembly [8, 7]. This process employs part presentation devices (feeding and orienting machines) that transport bulk quantities of parts to actually assemble large quantities of parts in parallel simultaneously. In the implementation developed by Moncevicz [6], parts are "palletized" by using vibration to convey them over a plastic "pallet" into which are carved an array of relief shapes that trap and orient the flowing parts. To achieve parallel assembly of two parts, a quantity of the first part is palletized. This first part is designed such that once held in the pallet relief forms, it becomes integral with the pallet for the purpose of palletizing a quantity of the second part. The second part palletization actually assembles a quantity of the second part to the first part. The mated part pair is removed as a subassembly unit from the pallet. Since many part insertions occur simultaneously, a very high assembly rate can be achieved.

We believe that this layered palletization process (and other similar processes) could be employed in more cases if temporary nonfunctional shape features could be added to the pallet or the palletized parts. Between part layers, for example, a "primer layer" of nonfunctional "pseudo parts" could be palletized. The sole purpose of this primer layer would be to cause geometric shapes that would facilitate the subsequent palletization of the next part layer. If the pseudo parts are not needed for (or worse prevented) the functioning of the assembled device, they must be removed before the device is used. This could be accomplished if their design involved the use of a conformational switch that detached them upon the arrival and assembly of the part whose assembly they were intended to facilitate.

- **Assembly of very small things.** Another problem with the common "part grasping" image of assembly is that some parts are simply too small to grasp. A randomized process might be the only possible approach. Yeh and Smith [14] used a (non-layered) palletization technique similar to [8, 7], to assemble microstructures. They fabricated trapezoidal galium arsenite (GaAs) blocks and a Si wafer with trapezoidal holes. Assembly is then done by releasing the GaAs blocks in a carrier fluid (ethanol) and dispensing the fluid over the Si wafer. Cohn *et al.* [2] experimented with the self-assembly of a small hexagonal lattice (1 *mm* in diameter) by placing a quantity of them on a slightly concave diaphragm that was agitated with a loudspeaker. Incorporating conformational switching to such micro-scale randomized assembly processes might facilitate the non-trivial assembly of very small parts.

- **Assembly very far away.** Consider trying to remotely assemble a set of components that are very far away. An example might be the construction of housing units on a distant planet. Teleoperation control issues arise along with the problem of the cost of transmitting control signals back and forth to the remote site. An alternate approach might be to send quantities of the component parts to the site, and outfit them with an autonomous means of locomotion. The parts could roam around until they find each other and correctly assemble. Again, the similarities with the viral assembly process are clear, and conformational switches could be similarly useful.

1.3 Previous work

All three practical applications just described would require the optimization of the assembly rate achieved with the parts that could conformationally assemble. This is to say, for example, that the best design of remote planet housing is the one that will assemble most quickly in the randomized assembly process. The objective of the design optimization for parts and related conformational switches is therefore the maximization of assembly rate. This is the approach we have taken in this work.

In this context, the work most closely related to our efforts is that of Hosokawa *et al.* [5]. They propose a chemical kinetics analogy to randomized assembly. Chemical reactions are analogous to the assembly and/or disassembly of certain subsets of parts. The amount of each intermediate subassembly is predicted given the initial quantities of each part and a relation describing the reaction dynamics. A physical experiment was devised by using a quantity of triangular parts that were fitted with conformational switches made from movable magnets. These parts were assembled in a rotating box randomizer. Empirical results agreed reasonably well with predicted values. The important distinction with our work

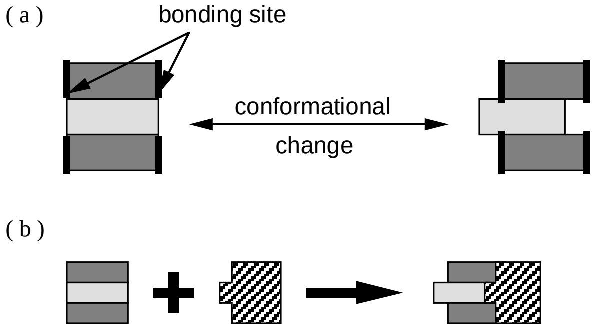


Figure 2: one-dimensional conformational switch

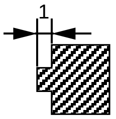
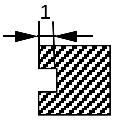
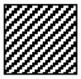
bond site			
bond configuration	+1	-1	0

Figure 3: bond configurations

is that they do not address the design optimization of the part and switch design. Their emphasis is on verifying the chemical kinetics analogy.

2 Conformational switch model

2.1 One-dimensional conformational switch

Our design of a conformational switch for one-dimensional assembly is motivated by the "counting device" appearing in [9]. We extended this counting device model so that a part can *form* and *destroy* a bond with another part.

As shown in Figure 2-a, a part has two bonding sites and a conformational switch is realized with a sliding bar mechanism that connects the two bonding sites. Conformational change is triggered by interaction with another part (see Figure 2-b). Note that parts can be assembled in only one direction, say horizontally. In Figure 2, therefore, one can place a part on the left or the right of another part, but *not* on the top or bottom. However, the model could easily be extended to the two-dimensional case.

A *bond configuration* is a variable which describes the shape of a bonding site. It takes a positive value if the corresponding site has convex shape, a negative value if the site is concave, and zero if the site is flat. Examples of bond configurations and the corresponding shape of bonding sites are shown in Figure 3.

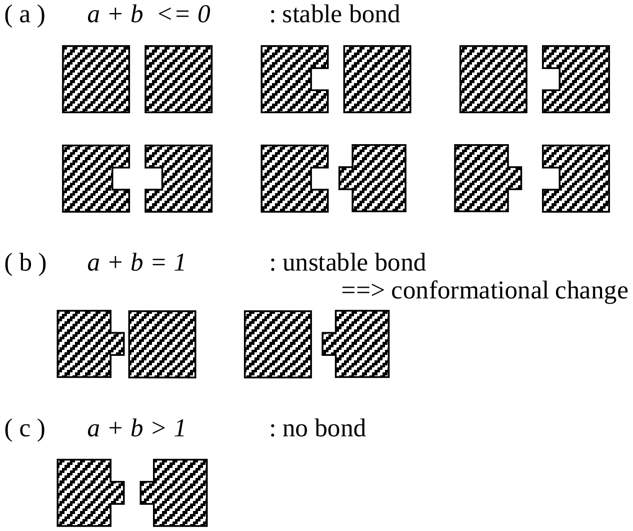

(a) a + b <= 0 : stable bond
(b) a + b = 1 : unstable bond
==> conformational change
(c) a + b > 1 : no bond

Figure 4: three types of bonding

When bonding sites of two parts meet, they form either 1) a stable bond, 2) an unstable bond, or 3) no bond. The occurrences of these cases depend on the shape of the two bonding sites, or equivalently, the bond configurations of the sites. Let a and b be bond configurations of two bonding sites contacting each other. These sites form a stable bond if $a + b \leq 0$ (complementary), form an unstable bond if $a + b = 1$ (fairly complementary), and form no bond if $a + b > 1$ (not complementary). Figure 4 shows examples of each of the three cases.

An unstable bond induces conformational change of the involved bonding sites, which can propagate over the connected parts via the sliding bar mechanism. After the conformational changes, a stable bond is formed if $a + b \leq 0$ and no bond is formed if $a + b \geq 1$. Also, an existing bond is destroyed if $a + b \geq 1$ after the conformational changes, which results in detaching of the corresponding parts. Figure 5 illustrates an example of such propagation and detaching.

Ambiguous situations may arise when conformational change can propagate in both directions or in no direction, such as the cases shown in Figure 6-a. To resolve such ambiguity, we assume an *upstream*

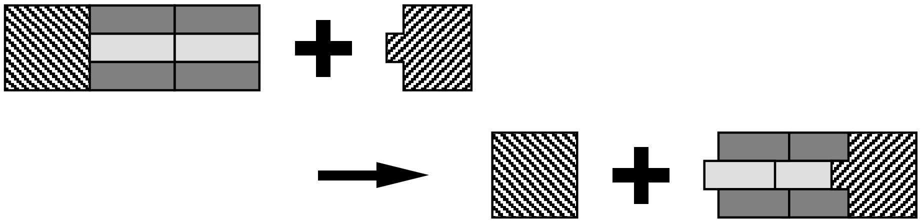


Figure 5: propagation of conformational changes and detaching

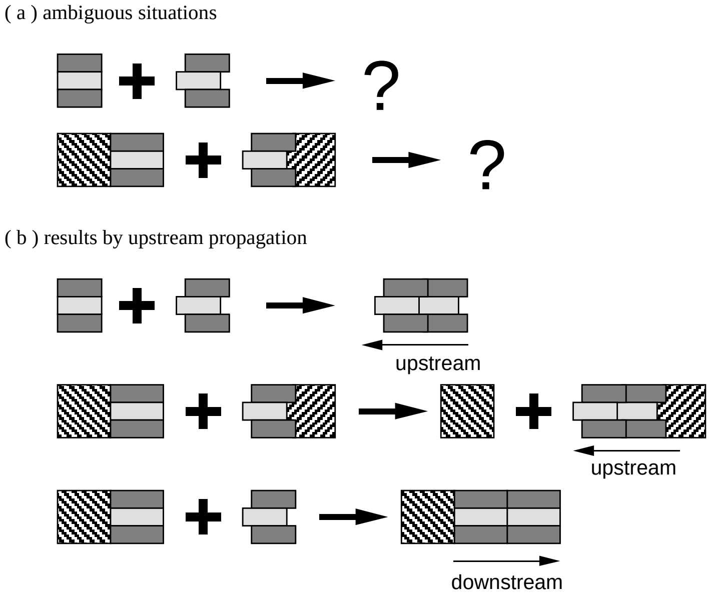

(a) ambiguous situations
?
?
(b) results by upstream propagation
upstream
upstream
downstream

Figure 6: priority to upstream propagation

propagation priority. As shown in Figure 6-b, conformational change propagates downstream (as defined in Figure 6), *only* when the upstream direction has a rigid end and downstream has a free end (the bottom picture of Figure 6-b). Otherwise, propagation goes upstream (the top and middle picture of Figure 6-b).

2.2 One-dimensional randomized assembly

In the computer implementation of one-dimensional randomized assembly, we assumed that the process is sequential. In other words, one pairwise mating of two parts (possibly subassemblies of parts) occurs at a given time. Given this assumption, a randomized assembly can be achieved in a straightforward way by using a *robot bin-picking metaphor* described as follows:

Assume a random assortment of parts in a (one-dimensional) bin (Figure 7-a).

Step 1: The robot arm #1 randomly picks up a part from the bin. Then, robot arm #2 randomly picks up another part from the bin (Figure 7-b).

Step 2: The two parts are pushed against each other, possibly causing formation and destruction of bonds (Figure 7-c).

Step 3: The parts are randomly returned to the bin (Figure 7-d), possibly as an assembly.

The steps 1-3 are repeated until pre-specified conditions are satisfied (*e.g.* repeat for a specified number of iterations, repeat until the number of parts decreases below a limit, *etc.*). It is assumed that the parts do not change their orientations, so in general, AB and BA are two distinct assemblies.

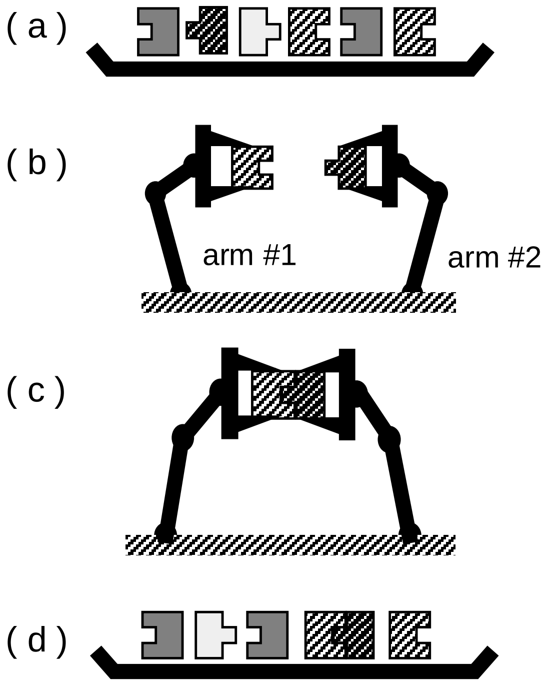


Figure 7: robot bin-picking metaphor

3 Genetic algorithm optimization

3.1 Genetic algorithms

Genetic algorithms (GAs) are an optimization technique in which points in the design space are analogous to organisms subject to a process of natural selection, or "survival of the fittest" [4]. The quality of a design (or a point in the design space) is measured by a fitness function, which takes as input an encoded representation of the point, or a genetic "chromosome" of the design. GAs model reproduction in populations of such chromosomes over generations. In a given generation, designs with higher fitness have a higher probability to be selected for reproduction. During reproduction, two "parent" chromosomes are mated through genetic crossover, which results in two offspring chromosomes. Many generations of such selection and mating will produce a highly-fit population of chromosomes, *i.e.* better designs.

3.2 Formulation of GA search

We are interested in designing conformational switches that, when randomly assembled, maximize the rate of a desired assembly. More precisely, the problem is stated as follows:

Given: the total number of parts, and the number of each kind of part in the bin (in other words, the initial state of the bin)

Find: the optimal design of conformational switches that maximizes the assembly rate (*i.e.* yield) of a desired set of parts in the process of randomized assembly.

Note that we are designing not only the bar mechanisms, but also the initial bond configurations. The above problem is formulated by parameterizing the design of the conformational switches, with genetic algorithms used to search the parameter space of possible switch designs.

The one-dimensional conformational switch of a part is uniquely specified in terms of four parameters: *left_config*, *right_config*, *link*, and *bar_length*. *Left_config* and *right_config* are the initial bond configurations of the left bond site and the right bond site, respectively. *Link* is a Boolean variable that specifies the existence of the conformational link (a bar mechanism) in a part. If *link* is TRUE, there is a conformational link between the two bond sites, so they can undergo conformational change. If *link* is FALSE, there is no conformational link, so the bond configurations do not change from their initial values (*i.e.* a solid part). *Bar_length* is an upper limit on *right_config* and *left_config*. In order for a design to be valid, we need $left\_config, right\_config \leq bar\_length$. Note that if $left\_config = right\_config = bar\_length$, the bar cannot move at all. Also, *bar_length* is ignored if *link* is FALSE.

A chromosome used in genetic search is a binary string that encodes the above design parameters for all kinds of parts in the bin. For the examples in the next section (unless otherwise stated), two bits are assigned for each of *left_config* and *right_config*, and one bit for *link* and *bar_length*[1]. The location of these bits on a chromosome is shown in Figure 8.

For *left_config* and *right_config*, the first bit corresponds to the absolute value and the second bit corresponds to sign, with minus being 0 and plus being 1[2]. If $left\_config = -1$, for example, the corresponding two bits are 10. Since six bits are necessary for one part, a chromosome that encodes n kinds of parts has length $6n$.

The fitness of a chromosome (*i.e.* the suitability of conformational switch designs and initial configurations) is based upon the number of desired assemblies after some numbers of iterations of the robot bin-picking simulation described earlier. The actual fitness is measured as an average of a pre-specified number of such bin-picking runs. We take an average value because the picking process is stochastic, and our interest is in maximizing the overall yield, not the yield of a particular run.

[1] This implies *bar_length* can only take values {0, 1}

[2] This implies that *left_config* and *right_config* can only take values $\{-1, 0, 1\}$.

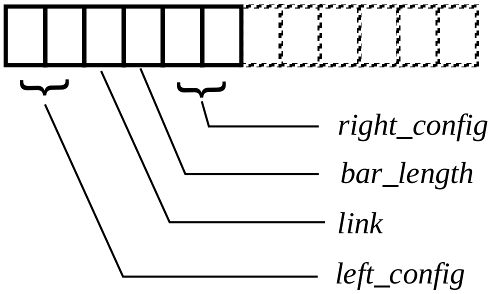


Figure 8: bit assignment of a chromosome

4 Examples

This section describes several examples of genetic optimization of one-dimensional conformational switches. The GA in the following examples uses fitness proportionate (roulette wheel) selection, linear fitness scaling with scaling coefficient = 2.0, an elitist selection scheme, crossover probability = 0.8, and mutation probability = 0.03.

4.1 Two part non-randomized assembly

As an initial example, a two part non-randomized assembly is studied, where bin-picking is done deterministically. The assumption here is as follows: two parts, part A and part B, come in two separate bins, and robot arm #2 is picking a part from each bin in the fixed sequence $ABAB\cdots$. The robot arm #1 is initially holding a base part (we will call it part Z). The assembly is done by pushing parts against each other, followed by release of the part held by arm #2. Since this pushing possibly causes formation and breakage of the existing bond between parts, the assembled parts either stay on arm #1 (no parts detached), or fall off (detaching occurs). The fallen parts are sent to a bin for assembled parts. Arm #2 then goes to the next bin and the process iterates. The process is illustrated in Figure 9.

Our objective is to maximize the yield of the assembly AB[3]. Since the assembly process is deterministic, it suggests the following scenario for the maximum yield:

Step 1: Part A is picked and forms a bond to Z (*i.e.* $Z + A \rightarrow ZA$).

Step 2: Part B is picked and forms a bond to A. Conformational change destroys the bond between A and Z, releasing the assembly AB (*i.e.* $ZA + B \rightarrow Z + AB$)

The evaluation of the switch designs, therefore, needs only two pickings (part A followed by part B). Note that we do not need to take an average of several runs since the process is deterministic. The fitness function is as follows:

$$fitness = \begin{cases} 0 & \text{if } ZA \text{ is not formed at Step 1} \\ 1 & \text{if } ZA \text{ is formed at Step 1 but } AB \text{ is not formed at Step 2} \\ 2 & \text{otherwise} \end{cases}$$

The base part Z is assumed to have only the right bond site and no conformational link, so the size of a chromosome is $6 \times 2 + 2 = 14$ bits. This means that there are 2^{14} possible designs. Figure 10 illustrates the best designs obtained by three GA runs with population size = 10 and number of generations = 5. The numbers below each designs are the corresponding parameters (*left_config*, *link*, *bar_length*,

[3] Note that AB and BA are distinct.

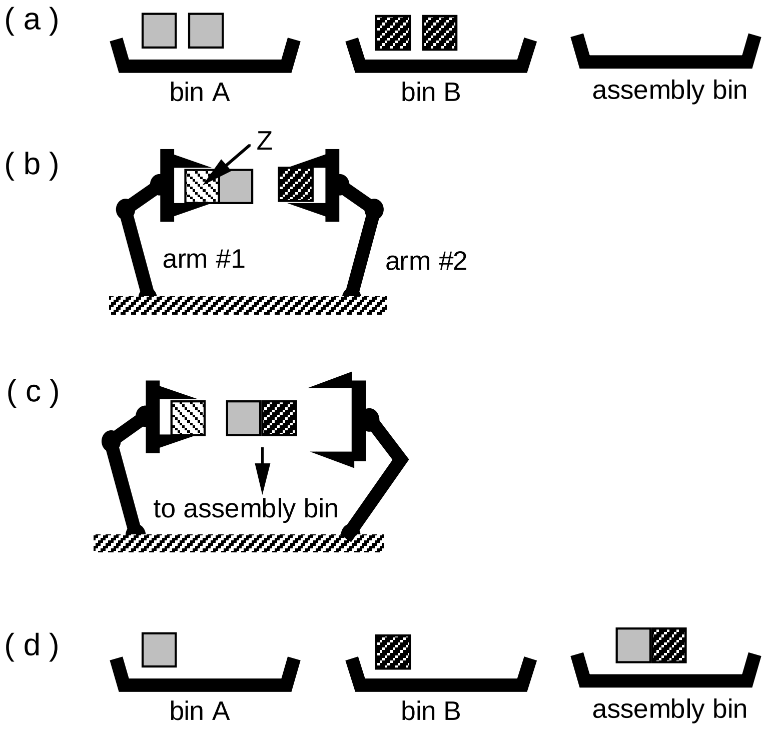


Figure 9: deterministic robot bin-picking

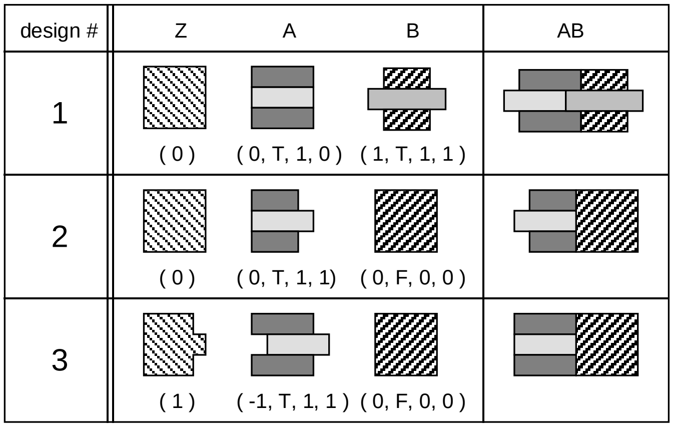


Figure 10: best designs for two part, non-randomized assembly, found by GA.

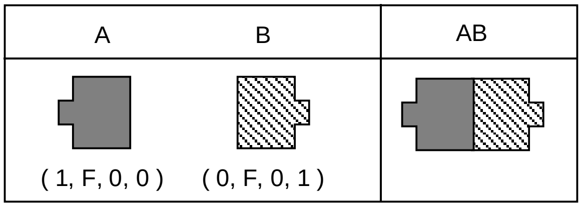


Figure 11: best design (part A : part $B = 1 : 1$) : Design I

right_config)[4]. Only *right_config* is shown for the base part Z. Note that all three designs scored the maximum fitness = 2. All the possible designs that score the maximum fitness, found using a depth-first search, are listed in Appendix A.

4.2 Two part randomized assembly

The second example is a two part randomized assembly as described in Section 2.2. The initial bin contains a random mixture of two types of parts, part A and part B, and the design objective is to maximize the yield of the assembly AB. The total number of parts in the initial bin is fixed at 50 and the number of AB's in the bin is counted after 50 iterations of Steps 1-3 in Section 2.2. The fitness of a chromosome is the average count of AB's over 50 such runs. In the GA runs described below, the population size is 30 and the number of generations is 10.

Figure 11 shows the best design (fitness = 9.76) when the initial fraction of part A's and part B's is 1:1 (*i.e.* 25 part A's and 25 part B's). Let us call it Design I. As easily seen from the figure, the result is as expected: only $A + B \rightarrow AB$[5] occurs.

Figure 12-a shows the best design (fitness = 6.7) when the initial fraction of part A's and part B's is 4:1 (*i.e.* 40 part A's and 10 part B's). Let us call it Design II. All possible assemblies with this design are illustrated in Figure 12-b, where A' represents part A after conformational change. Note that not

[4] Note that part B in design #1 is equivalent to a solid part $(1, F, 0, 1)$

[5] It is implicitly assumed the operator '+' is not commutative. In particular, $B + A$ is not possible in this case.

only $A + B \to A'B$ (reaction 2), but also $A + A \to A'A$ (reaction 1) and $A' + A \to A'A$ (reaction 3), are possible. Once $A'A$ is formed, it can also be bound to B to produce $A'B$ (*i.e.* $A'A + B \to A' + A'B$: reaction 7). The reactions 1, 3 and 5 help to decrease the total number of part A's in the bin, and in turn, to increase the chance of part B's being picked. If a part B is picked, it can bind to any of A, A', and $A'A$ (by reactions 2, 4, and 7). Also, once $A'B$ is formed, it can never be destroyed[6]. The overall yield of $A'B$'s, therefore, is better than that from only $A + B \to AB$. For comparison, the typical fitness of the design in Figure 11, when applied to the same situation (*i.e.* part A : part B = 4 : 1), is 5.5. The next section describes a comparison of these two designs based on expected yield.

4.3 Rate equation analysis of two part randomized assembly

Given the part designs and the their possible reactions, one can formulate a rate equation analysis similar to the ones found in [10, 5]. The following recurrence describes the randomized assembly process described in Section 2.2:

$$\mathbf{n}(t+1) = \mathbf{n}(t) + \mathbf{A}\mathbf{p}(t) \tag{1}$$

where $\mathbf{n}(t)$ is a vector of the numbers of each possible subassembly at iteration t, $\mathbf{p}(t)$ is a vector of probabilities for each possible reaction at iteration t, and $\mathbf{A}$ is a matrix of stoichiometric coefficients. Since only one reaction $A + B \to AB$ is possible for Design I, $\mathbf{n}(t)$, $\mathbf{A}$ and $p(t)$ (scalar in this case) are defined as follows:

$$\mathbf{n}(t) = \begin{pmatrix} n_A(t) \\ n_B(t) \\ n_{AB}(t) \end{pmatrix}, \quad \mathrm{A} = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, \quad p(t) = \frac{n_A(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}}$$

where $n_A(t)$, $n_B(t)$ and $n_{AB}(t)$ are the numbers of part A, part B and assemblies AB at iteration t, respectively, and $s(t) = n_A(t) + n_B(t) + n_{AB}(t)$.

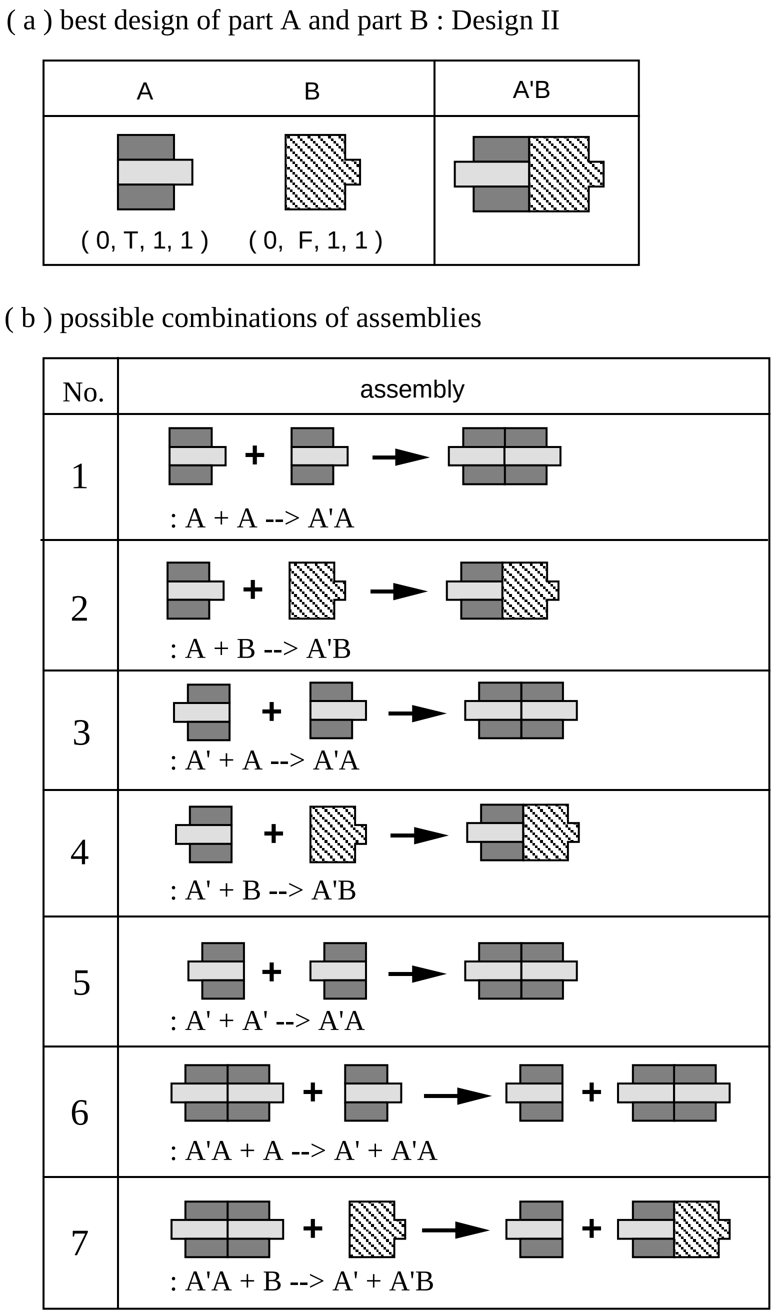
In the case of Design II, the possible reactions are:

$$\begin{array}{ll} A + A \to A'A & A' + A' \to A'A \\ A + B \to A'B & A'A + A \to A' + A'A \\ A' + A \to A'A & A'A + B \to A' + A'B \\ A' + B \to A'B & \end{array}$$

Corresponding $\mathbf{n}(t)$, A and $\mathbf{p}(t)$ are, therefore, defined as follows:

$$\mathbf{n}(t) = \begin{pmatrix} n_A(t) \\ n_B(t) \\ n_{A'}(t) \\ n_{A'A}(t) \\ n_{A'B}(t) \end{pmatrix}, \quad \mathrm{A} = \begin{pmatrix} -2 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & -2 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

[6] Note that reverse of reaction 7 $A' + A'B \to A'A + B$ is *not* possible due to the upstream priority rule discussed in Section 2.1.


(a) best design of part A and part B : Design II

(b) possible combinations of assemblies

Figure 12: best design (part A : part $B = 4 : 1$)

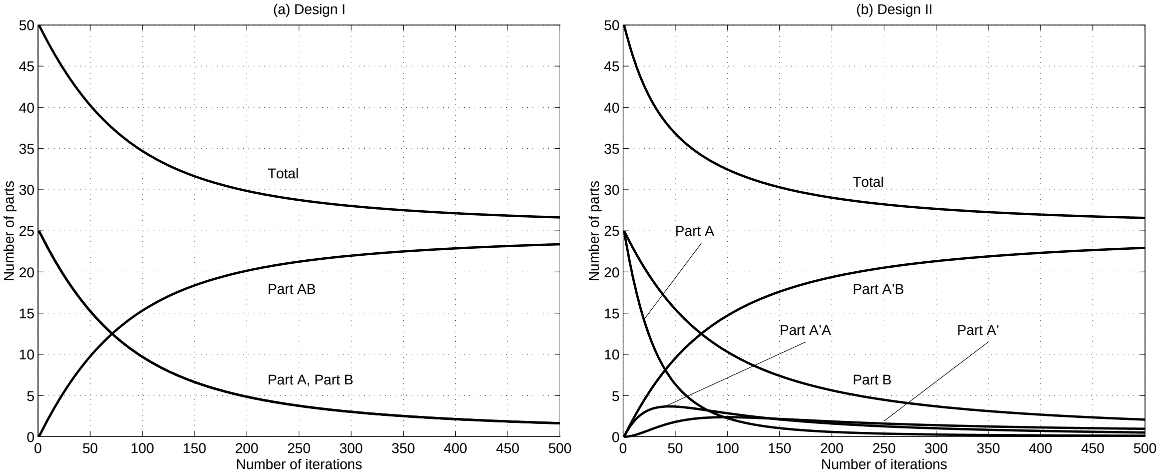


Figure 13: solution of equation (1) (part A : part $B = 1 : 1$)

$$\mathbf{p}(t) = \begin{pmatrix} \dfrac{n_A(t)\cdot\{n_A(t)-1\}}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_A(t)\cdot n_B(t)}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_{A'}(t)\cdot n_A(t)}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_{A'}(t)\cdot n_B(t)}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_{A'}(t)\cdot\{n_{A'}(t)-1\}}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_{A'A}(t)\cdot n_A(t)}{s(t)\cdot\{s(t)-1\}} \\[2ex] \dfrac{n_{A'A}(t)\cdot n_B(t)}{s(t)\cdot\{s(t)-1\}} \end{pmatrix}$$

For Design I and Design II, equation 1 is solved numerically with the two initial conditions discussed in Section 4.2. Figure 13-a and Figure 13-b are the solution for Design I and Design II with initial condition $\mathbf{n}(0) = (25, 25, 0)'$, and $\mathbf{n}(0) = (25, 25, 0, 0, 0)'$, respectively. Note that for *both* Design I and Design II, $n_A, n_B \rightarrow 0$ and n_{AB}[7]$\rightarrow 25$ as $t \rightarrow \infty$. The yield of the desired assembly AB is compared in Figure 14. It shows Design I is consistently better than Design II for $t \in \{0, 500\}$, and the difference in the yield is maximum at $t \simeq 150$. At $t = 50$, $n_{AB}(t)$ for Design I and Design II are 9.8914 and 9.6590, respectively.

Similar analysis is done with the initial conditions $\mathbf{n}(0) = (40, 10, 0)'$ for Design I, and $\mathbf{n}(0) = (40, 10, 0, 0, 0)'$ for Design II. As shown in Figures 15-a and 15-b, Asymptotic behaviors of $n_B(t)$ and $n_{AB}(t)$ are similar in Design I and Design II. In Design II, however, $n_A(t) \rightarrow 0$ quickly whereas in Design I, $n_A(t) \rightarrow 30$ as $t \rightarrow \infty$. This drop of $n_A(t)$ and increase of $n_{A'A}(t)$ results in decrease of total number

[7] denoted $n_{A'B}$ in Design II.

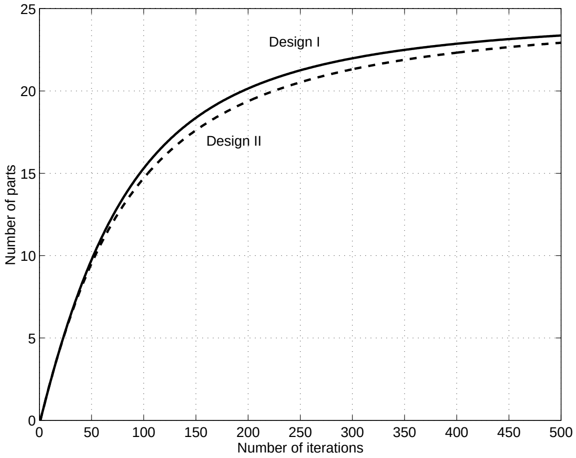


Figure 14: yield computed by equation (1) (part A : part $B = 1 : 1$)

of parts in the bin $s(t)$, which is clearly shown in Figure 15-b. This supports the qualitative argument in Section 4.2. Comparison of the yield in Figure 16 indicates Design II is consistently better than Design I for $t \in \{0, 500\}$, and the difference in the yield is maximum at $t \simeq 100$. At $t = 50$, $n_{AB}(t)$ for Design I and Design II are 5.7961 and 6.3230, respectively.

4.4 Two part randomized assembly with a dummy part

This example is the same as the previous example except that a dummy part C is mixed in the initial bin. The dummy part C is not involved in the desired assembly AB, but is expected somehow to help part A's and part B's assemble to AB (as an enzyme does in viral assembly).

Figure 17 shows the best design (fitness = 2.74) where initially $A : B : C = 8 : 1 : 1$ (*i.e.* 40 part A's, 5 part B's and 5 part C's). Let us call this Design I. The designs of part A and part B are identical to the designs of part A' and part B in Figure 12, respectively, and part C cannot bond to any of the possible assemblies. In other words, part C does not play any role in the assembly of AB.

This, however, is not the case if the fraction of part C's is larger. Figure 18 shows Design II, the best design (fitness = 2.84) where $A : B : C$ is $11 : 3 : 11$ (*i.e.* 22 part A's, 6 part B's and 22 part C's). Part A and part B are the same as in the design of Figure 12. This time, however, part C can bind to part A, part B, and other assemblies. During the GA run, it was observed that part A's and part C's form a long chain such as $A'C \cdots CA$, which seems to help to increase the chance of part B's being picked.

The rate equations for Design I are relatively simple. For Design II, however, the number of possible subassemblies and the number of possible reactions can be very large since part C's can form a subassembly C_n (an n concatenation of part C's) for any positive integer n. To get around this problem, we simply do not distinguish part C_n and part C_m for any positive integers n and m. This reduces the number of subassemblies down to 11. The resulting rate equations are still useful since the desired assembly $A'B$ does not contain part C's. Also, there is no need to keep track of the number of each C_n's. The derivation of the rate equations is found in Appendix B.

Figure 19 shows the comparison of the yield by Design I and by Design II with the initial conditions $\mathbf{n}(0) = (40, 5, 5, 0, 0)'$ and $\mathbf{n}(0) = (40, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0)'$, respectively. It is observed that as $t \to \infty$, the yield of Design I converges to 5 (maximum possible) and the yield of Design II goes to 4.5. At $t = 50$, however, the yield of Design I is 3.1260, whereas it is 3.1551 for Design II. In other words, the numerical analysis indicates that at $t = 50$, Design II is slightly better in yield, in contrast to the simulation results. A similar trend is observed for the second case where $\mathbf{n}(0) = (22, 6, 22, 0, 0)'$ for Design I and $\mathbf{n}(0) = (22, 6, 22, 0, 0, 0, 0, 0, 0, 0, 0)'$ for Design II (Figure 20). Even though Design I is asymptotically better in yield, Design II barely wins at $t = 50$: $n_{A'B}(50) = 2.1148$ for Design I and $n_{A'B}(50) = 2.1709$

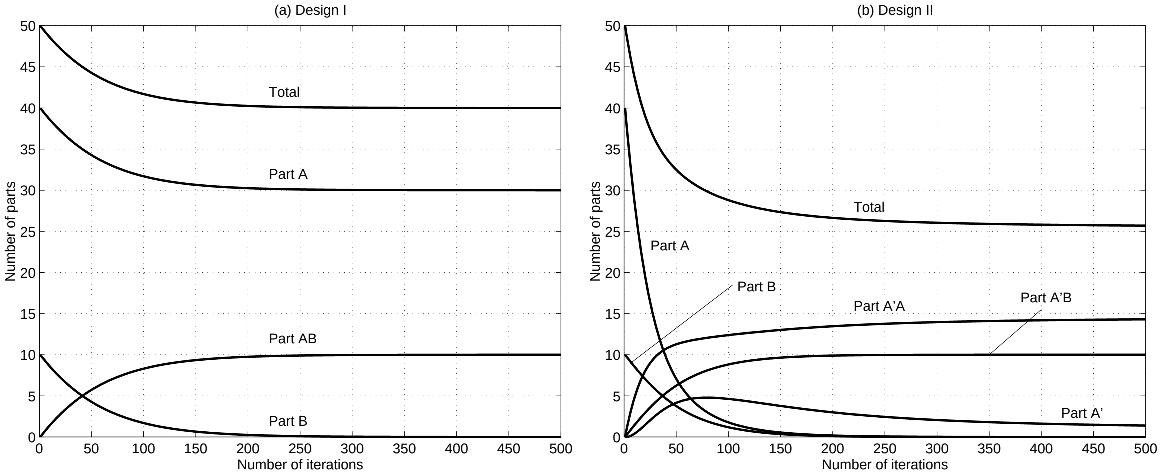


Figure 15: solution of equation (1) (part A : part $B = 4 : 1$)

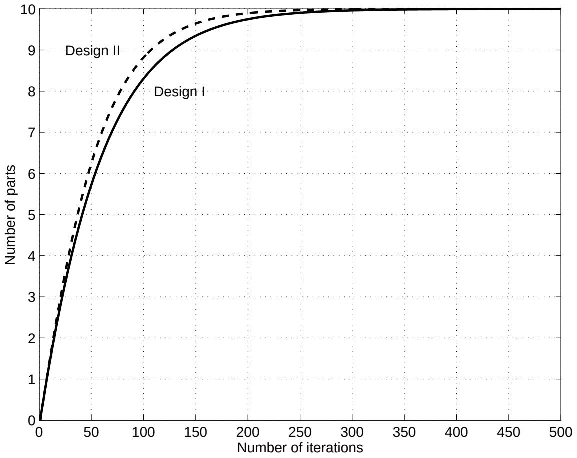


Figure 16: yield computed by equation (1) (part A : part $B = 4 : 1$)

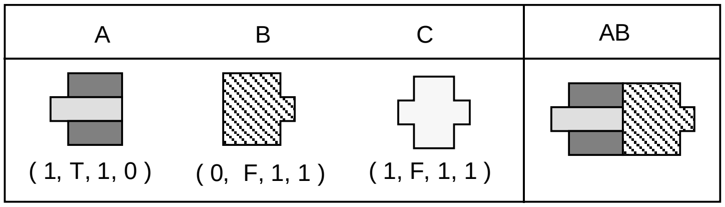


Figure 17: best design (part A : part B : part $C = 8 : 1 : 1$) : Design I.

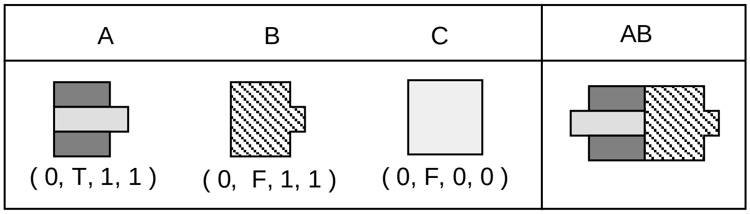


Figure 18: best design (part A : part B : part $C = 11 : 3 : 11$) : Design II.

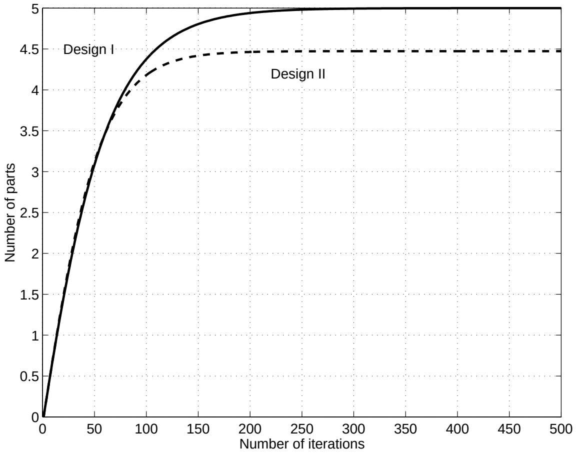


Figure 19: yield computed by equation (1) (part A : part B : part $C = 8 : 1 : 1$)

for Design II. This matches the simulation result, but the difference in yield is marginal.

Overall, the above results indicate there are not great advantages in having dummy parts. These examples with dummy parts, however, suggests two important issues:

1. **Importance of rate equation analysis:** due to the stochastic nature of randomized assembly simulation (and the GA itself), designs found by the GA may not be optimal in a statistical sense. Also, since it is impossible to run the simulation of each design for $t \to \infty$, the fitness of a design is evaluated based on the "reaction rate" at the time of simulation termination, not the yield at $t \to \infty$. Rate equation analysis would give statistically reliable predictions of the behavior of parts during randomized assembly for a long period of time. It must be noted, however, that the rate equation analysis is possible *only* after a design of parts and their possible reactions are specified.

2. **Difficulty of rate equation analysis:** as shown in appendix B, the number of possible reactions for a design of parts can be *very* large even in simplest cases, since one must consider reactions between parts and all possible assemblies. In the case of Design II above, we avoid combinatorial explosion by not distinguishing part C_n and part C_m for any positive integers n and m. This kind of trick, however, is not always possible. In general, it becomes increasingly difficult to enumerate all possible reactions and subassemblies, as the number of distinct parts increases.

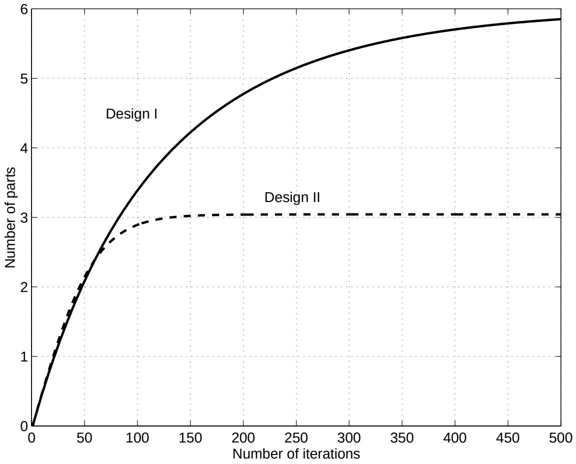


Figure 20: yield computed by equation (1) (part A : part B : part $C = 11 : 3 : 11$)

5 Discussion and future work

Our examples in the previous section indicate that conformational switches can be used to temporarily change the concentration of parts in the bin. If an AB assembly is desired, for example, and there are many more part A than part B in the bin, conformational assemblies containing only part A (*e.g.* AA), will decrease the overall part count and increase the probability of randomly picking part B. This behavior was evident in the results shown in Figure 12-b. Such emergent changes in concentration would be useful in ensuring that "scarce" parts would be found and assembled in an efficient and timely manner. Again, consider a bin in which there are relatively many A's and relatively few B's, and the assembly task requires the construction of AB subassemblies before the final assembly tasks. An early "clumping" of A's would aid the process.

This paper described our preliminary efforts toward the automated design of mechanical conformational switches. We assumed an idealized (yet physically meaningful) switch design and assembly process. Our next step will be to continue with further experiments with this and other idealized switch designs. We plan to increase the number of parts, and explore issues related to assembly task planning. As discussed above, for example, how can we impose constraints that ensure that subassemblies are built first? Also, we hope to evolve conformational switches that embody assembly precedence relationships. A container cannot be closed, for example, until all of its contents are assembled into it. Some of these succeeding efforts may require two-dimensional idealized conformational switches, which we plan to implement.

Following these efforts with idealized switch designs, we hope to experiment with more realistic embodiments of parts and switches. This could involve, for example, simulations of the dynamic interaction of parts outfitted with conformational switches (*e.g.* a dynamic simulation of the physical experiment performed by Hosokawa *et al.* [5]), or the design of parts that have implicit conformational switches. Certain characteristics of their physical shape would cause them to conformationally assemble during bulk dynamic interaction.

6 Acknowledgments

This work is supported by the National Science Foundation with a Presidential Young Investigator's grant (DDM-9058415). Matchable funds for this grant have been provided by Schlumberger Inc.. These sources of support are gratefully acknowledged. Additionally, the work was carried out using the computational facilities of the Computer-Aided Design Laboratory at the Massachusetts Institute of Technology, Department of Mechanical Engineering. This support is also gratefully acknowledged.

References

[1] S. Casjens. and J. King. Virus assembly. *Annual Review of Biochemistry*, 44:555–611, 1975.

[2] M. B. Cohn, C. J. Kim, and A. P. Pisano. Self-assembling electrical networks: an application of micromachining technology. In *Transducers '91: 1991 Sixth International Conference on Solid-State Sensors and Actuators*, pages 490–493, New York, New York, 1991. IEEE.

[3] N. S. Goel and R. L. Thompson. *Computer Simulations of Self-organization in Biological Systems.* Croom Helm, London, England, 1988.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[5] K. Hosokawa, I. Shimoyama, and H. Miura. Dynamics of self-assembling systems: Analogy with chemical kinetics. *Artificial Life*, 1(4):413–427, 1994.

[6] P. H. Moncevicz. Orientation and insertion of randomly presented parts using vibratory agitation. Master's thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1991.

[7] P. H. Moncevicz and M. J. Jakiela. Method and appratus for automatic parts assembly. United States Patent 5,155,895, October 20 1992.

[8] P. H. Moncevicz, M. J. Jakiela, and K. T. Ulrich. Orientation and insertion of randomly presented parts using vibratory agitation. In A. H. Soni, editor, *Proceedings of the ASME 3rd Conference on Flexible Assembly Systems*, pages 41–47, Miami, Florida, September 1991. The American Society of Mechanical Engineers. DE-Vol. 33.

[9] L. S. Penrose. Self-reproducing machines. *Scientific American*, 200:105–114, June 1959.

[10] J. I. Steinfeld, J. S. Francisco, and W. L. Hase. *Chemical Kinetics and Dynamics.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[11] R. L. Thompson and N. S. Goel. A simulation of T4 bacteriophage assembly and operation. *BioSystems*, 18:23–45, 1985.

[12] R. L. Thompson and N. S. Goel. Movable finite automata (MFA) models for biological systems I: Bacteriophage assembly and operation. *Journal of Theoretical Biology*, 131:351–385, 1988.

[13] J. D. Watson, Nancy H. Hopkins, Jeffrey W. Roberts, Joan A. Steitz, and Alan M. Weiner. *Molecular Biology of the Gene.* Benjamin/Cummings, Menlo Park, California, 1987.

[14] H. J. Yeh and J. S. Smith. Fluidic self-assembly of GaAs microstructures on Si substrates. *Sensors and Materials*, 6(6):319–332, 1994.

A Optimal designs of two part non-randomized assembly

This appendix lists all the possible designs that score the maximum fitness (fitness = 2) in the two part non-randomized assembly discussed in Section 4.1. The length of the chromosome in this example is 14 (6 for part A and part B, and 2 for part Z), so there are $2^{14} = 16384$ possible chromosomes. Since the assembly process is deterministic, depth first search can find all the optimal solutions with little enumeration. They are listed in Figures 21 and 22. Due to the degeneracy of parameter coding (see Section 3.2), more than one chromosome maps to a design. Also, some designs are functionally equivalent. Figure 22 shows such designs equivalent to the two designs of part B appearing in Figure 21. Since (0, F, 0, 0) has 4 equivalent part designs and (1, F, 0, 0) has 3 equivalent designs, there are $4 \times 5 + 4 \times 4 = 36$ optimal designs.

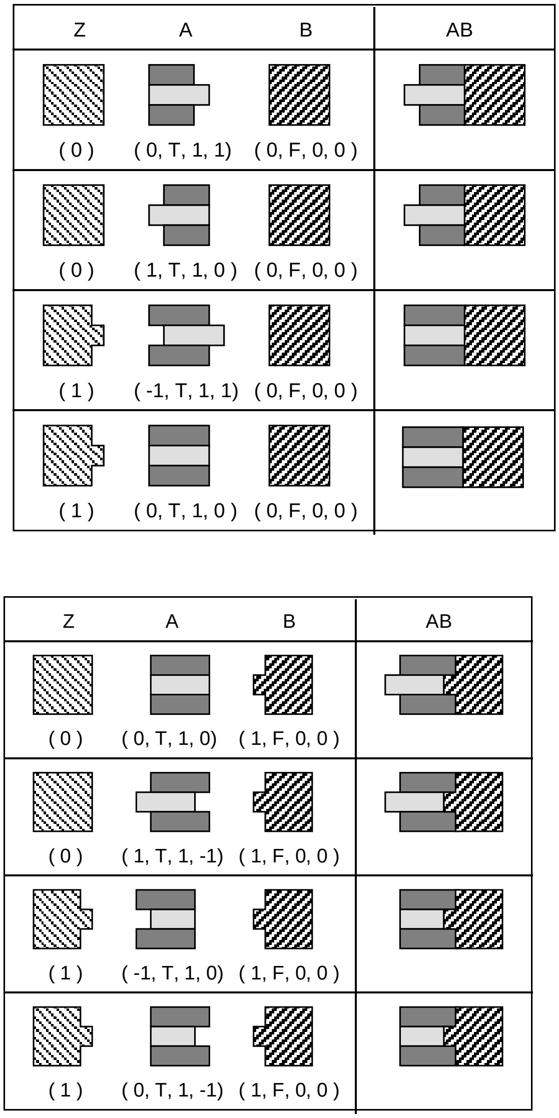


Figure 21: optimal designs for two-part non-randomized assembly

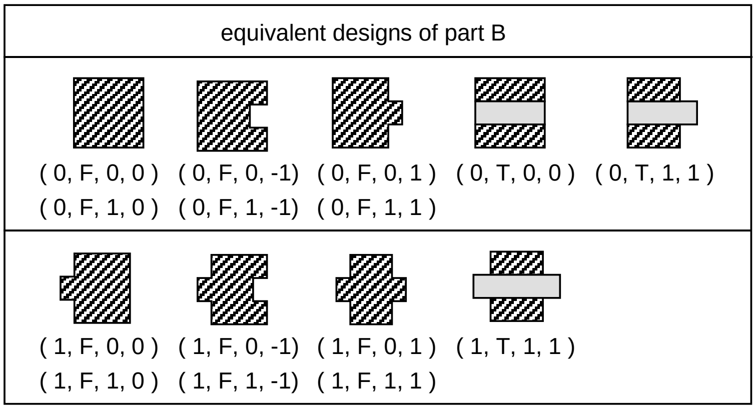


Figure 22: equivalent designs of part B

B Rate equation analysis of two part randomized assembly with a dummy part

This appendix explains details of rate equation analysis of two part randomized assembly with a dummy part discussed in Section 4.4. In the following, Design I and Design II refer to the part designs described in Section 4.4, unless otherwise specified. Derivation of the rate equations of these part designs follows the same steps found in Section 4.3.

The possible reactions for Design I are:

$$A + A \rightarrow AA' \qquad A + B \rightarrow AB \qquad AA' + B \rightarrow A + AB$$

where A' denotes a part A after conformational change. And $\mathbf{n}(t)$, A and $\mathbf{p}(t)$ are:

$$\mathbf{n}(t) = \begin{pmatrix} n_A(t) \\ n_B(t) \\ n_C(t) \\ n_{AA'}(t) \\ n_{AB}(t) \end{pmatrix}, \quad \mathrm{A} = \begin{pmatrix} -2 & -1 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}, \quad \mathbf{p}(t) = \begin{pmatrix} \dfrac{n_A(t) \cdot \{n_A(t) - 1\}}{s(t) \cdot \{s(t) - 1\}} \\ \\ \dfrac{n_A(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \\ \\ \dfrac{n_{AA'}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \end{pmatrix}$$

As easily seen in Figure 18, the number of possible reactions (and the number of possible subassemblies) can be very large for Design II since part C's can form a subassembly C_n (an n concatenation of part C's) up to 22 elements long. Fortunately, C is a solid part so there are no conformational changes in C: once a C binds to another C, the bond will never be destroyed. The resulting assembly CC will then behave *exactly* as a single C. This leads to the idea of not distinguishing part C_n and part C_m for any positive integer n and m. This reduces the number of subassemblies down to 11. The resulting rate equations are still useful since the desired assembly $A'B$ does not contain part C's. Therefore there is no need to keep track of the number of each C_n's. Under the above assumption, there are 36 possible reactions of Design II:

$$\begin{array}{lll}
A + A \rightarrow A'A & A' + B \rightarrow A'B & A'C_n + C_nA \rightarrow A'C_nA \\
A + B \rightarrow A'B & A' + C_n \rightarrow A'C_n & A'C_n + C_nB \rightarrow A'C_nB \\
A + C_n \rightarrow A'C_n & A' + A' \rightarrow A'A & C_nA + A \rightarrow C_n + A'A \\
A + C_nA \rightarrow A'C_nA & A'A + A \rightarrow A' + A'A & C_nA + B \rightarrow C_n + A'B \\
A + C_nB \rightarrow A'C_nB & A'A + B \rightarrow A' + A'B & C_nA + C_n \rightarrow C_n + A'C_n \\
C_n + A \rightarrow C_nA & A'A + C_n \rightarrow A' + A'C_n & C_nA + C_nA \rightarrow C_n + A'C_nA \\
C_n + B \rightarrow C_nB & A'A + C_nA \rightarrow A' + A'C_nA & C_nA + C_nB \rightarrow C_n + A'C_nB \\
C_n + C_n \rightarrow C_n & A'A + C_nB \rightarrow A' + A'C_nB & A'C_nA + A \rightarrow A'C_n + A'A \\
C_n + A' \rightarrow C_nA & A'C_n + A \rightarrow A'C_nA & A'C_nA + B \rightarrow A'C_n + A'B \\
C_n + C_nA \rightarrow C_nA & A'C_n + B \rightarrow A'C_nB & A'C_nA + C_n \rightarrow A'C_n + A'C_n \\
C_n + C_nB \rightarrow C_nB & A'C_n + C_n \rightarrow A'C_n & A'C_nA + C_nA \rightarrow A'C_n + A'C_nA \\
A' + A \rightarrow A'A & A'C_n + A' \rightarrow A'C_nA & A'C_nA + C_nB \rightarrow A'C_n + A'C_nB
\end{array}$$

where A' denotes a part A after conformational change, and C_n denotes a concatenation of *some* number of part C's. Corresponding $\mathbf{n}(t)$, A and $\mathbf{p}(t)$ are, therefore, defined as follows:

$$\mathbf{n}(t) = (n_A(t), n_B(t), n_{C_n}(t), n_{A'}(t), n_{A'A}(t), n_{A'B}(t), n_{A'C_n}(t), n_{C_nA}(t), n_{C_nB}(t), n_{A'C_nA}(t), n_{A'C_nB}(t))'$$

$$\mathrm{A} = \begin{pmatrix}
-2 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 & -2 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \\
-1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & -2 & -1 & 0 & 0 & 0 & -1 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & 0 & -1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$p_1(t) = \frac{n_A(t) \cdot \{n_A(t) - 1\}}{s(t) \cdot \{s(t) - 1\}} \qquad p_2(t) = \frac{n_A(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_3(t) = \frac{n_A(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_4(t) = \frac{n_A(t) \cdot n_{C_nA}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_5(t) = \frac{n_A(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_6(t) = \frac{n_{C_n}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_7(t) = \frac{n_{C_n}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_8(t) = \frac{n_{C_n}(t) \cdot \{n_{C_n}(t) - 1\}}{s(t) \cdot \{s(t) - 1\}} \qquad p_9(t) = \frac{n_{C_n}(t) \cdot n_{A'}(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{10}(t) = \frac{n_{C_n}(t) \cdot n_{C_nA}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{11}(t) = \frac{n_{C_n}(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{12}(t) = \frac{n_{A'}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{13}(t) = \frac{n_{A'}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{14}(t) = \frac{n_{A'}(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{15}(t) = \frac{n_{A'}(t) \cdot \{n_{A'}(t) - 1\}}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{16}(t) = \frac{n_{A'A}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{17}(t) = \frac{n_{A'A}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{18}(t) = \frac{n_{A'A}(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{19}(t) = \frac{n_{A'A}(t) \cdot n_{C_nA}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{20}(t) = \frac{n_{A'A}(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{21}(t) = \frac{n_{A'C_n}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{22}(t) = \frac{n_{A'C_n}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{23}(t) = \frac{n_{A'C_n}(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{24}(t) = \frac{n_{A'C_n}(t) \cdot n_{A'}(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{25}(t) = \frac{n_{A'C_n}(t) \cdot n_{C_nA}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{26}(t) = \frac{n_{A'C_n}(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{27}(t) = \frac{n_{C_nA}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{28}(t) = \frac{n_{C_nA}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{29}(t) = \frac{n_{C_nA}(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{30}(t) = \frac{n_{C_nA}(t) \cdot \{n_{C_nA}(t) - 1\}}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{31}(t) = \frac{n_{C_nA}(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{32}(t) = \frac{n_{A'C_nA}(t) \cdot n_A(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{33}(t) = \frac{n_{A'C_nA}(t) \cdot n_B(t)}{s(t) \cdot \{s(t) - 1\}}$$

$$p_{34}(t) = \frac{n_{A'C_nA}(t) \cdot n_{C_n}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{35}(t) = \frac{n_{A'C_nA}(t) \cdot n_{C_nA}(t)}{s(t) \cdot \{s(t) - 1\}} \qquad p_{36}(t) = \frac{n_{A'C_nA}(t) \cdot n_{C_nB}(t)}{s(t) \cdot \{s(t) - 1\}}$$

For Design I and Design II, equation 1 is solved numerically with the two initial conditions discussed in Section 4.4. Figure 23 and Figure 24 are the solutions for Design I and Design II with initial condition $\mathbf{n}(0) = (40, 5, 5, 0, 0)'$, and $\mathbf{n}(0) = (40, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0)'$, respectively. The same analysis is done with initial condition $\mathbf{n}(0) = (22, 6, 22, 0, 0)'$ for Design I, and $\mathbf{n}(0) = (22, 6, 22, 0, 0, 0, 0, 0, 0, 0, 0)'$ for Design II. These results are shown in Figure 25 and Figure 26.

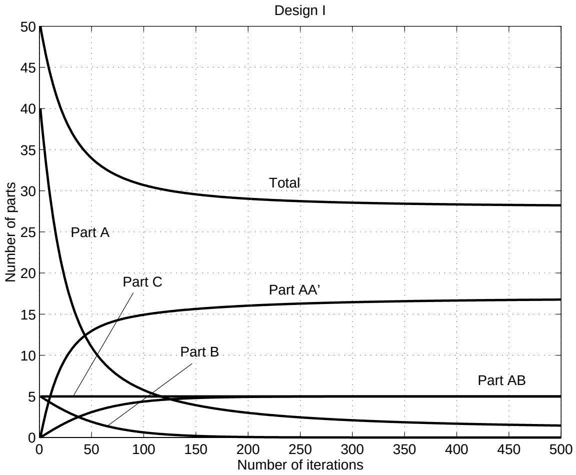


Figure 23: solution of equation (1) for Design I (part A : part B : part $C = 8 : 1 : 1$)

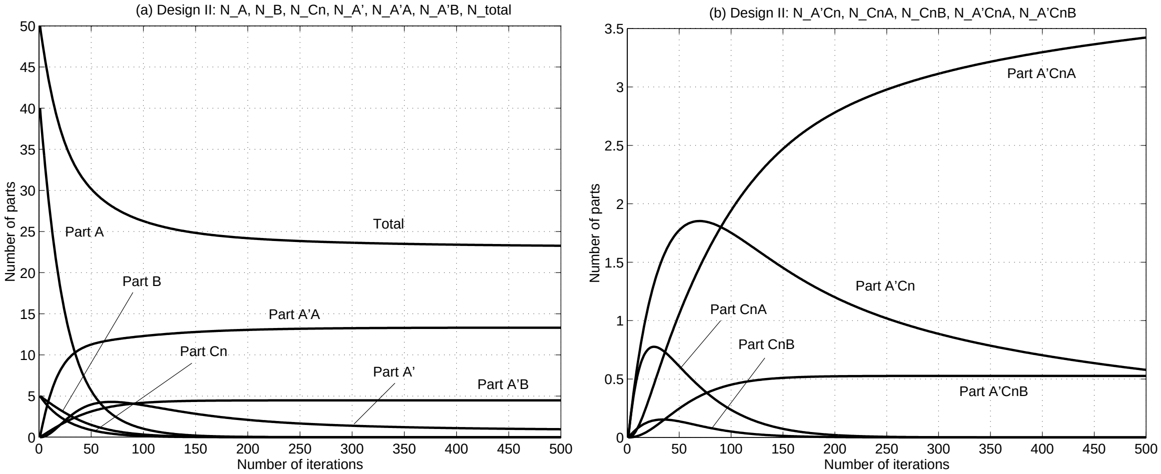


Figure 24: solution of equation (1) for Design II (part A : part B : part $C = 8 : 1 : 1$)

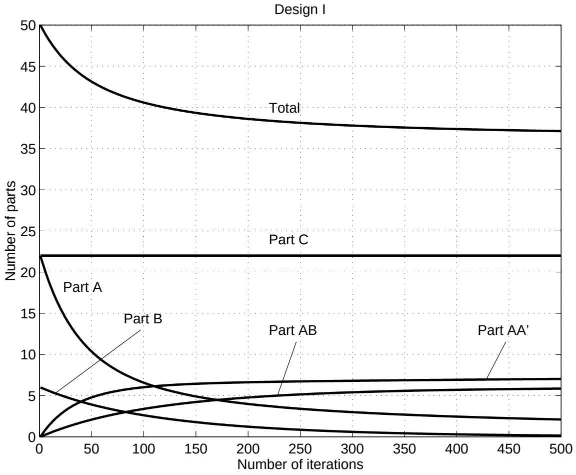


Figure 25: solution of equation (1) for Design I (part A : part B : part $C = 11 : 3 : 11$)

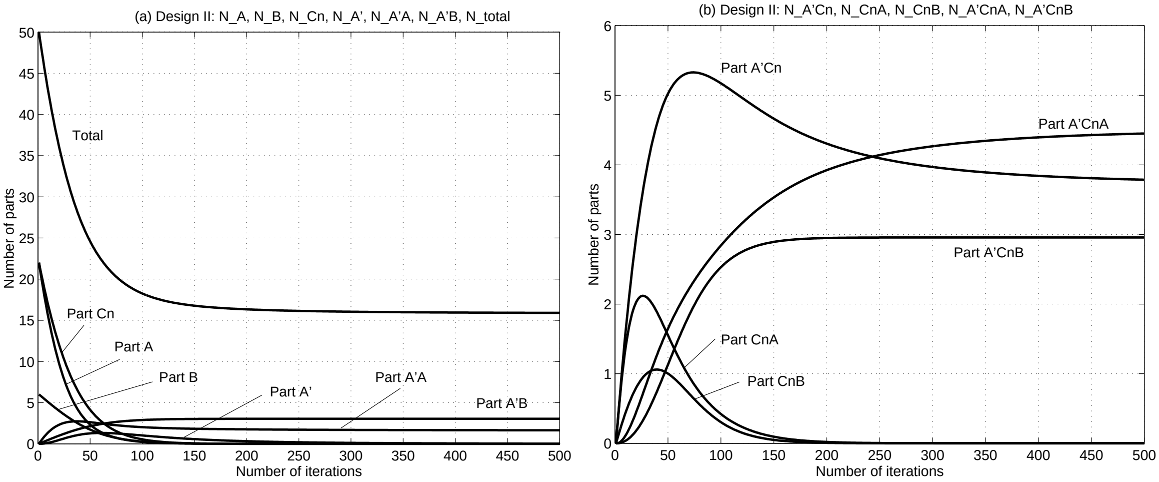


Figure 26: solution of equation (1) for Design II (part A : part B : part $C = 11 : 3 : 11$)