# Automatic Selection of Fixture Points for Frictionless Assemblies

J.D. Wolter * and J.C. Trinkle †
Department of Computer Science
Texas A&M University, College Station, TX 77843-3112

## Abstract

During the assembly of a product, it is vital that the partially completed assembly be stable. If the assembly is unstable, then it must be fixtured to stabilize it before retrieving the next part or subassembly. This paper presents a stability test and a new approach to automatically generating the positions of a small set of fixture elements (fixels) that will stabilize an assembly. The stability test and the fixel positioning approach consider both the translational and rotational degrees of freedom of each part. Since all the relevant mechanical constraints are linear functions of the contact force magnitudes and the components of the velocities of the parts, linear programming techniques can be used with great efficiency.

## 1 Introduction

While assembling a product, it is necessary to guarantee that all the parts already inserted remain in place until the next part is inserted. To guarantee this, we must ensure that contacts among the parts and with the fixtures are sufficient to stablize the assembly. In this paper we will present an efficient method for testing if an assembly is stable, and, if it is not, generating a set of additional fixture contact points, known as "fixels," that will stablize it.

An assembly is considered stable if all possible infinitesimal motions of the parts increase the total potential energy of the system. Parts may rotate as well as translate, and different parts may simultaneously be moving in different directions. We will assume that all parts are rigid polygons with known shapes and masses, that all contacts are frictionless, and that all parts are initially at rest in known positions and orientations. In the interest of brevity, only the two-dimensional case is presented here, however the algorithm is easily extended to three-dimensional assemblies[16] and some three-dimensional results will be given. Overcoming the frictionless assumption is more difficult, but to assume the absence of friction when testing for stability is actually quite sensible in production planning applications. Any assembly that is stable without friction will certainly be stable with friction, and assemblies that are held together only by friction may well be destablized by vibrational disturbances in the workplace. Relaxing the assumption that parts are polygonal presents no theoretical difficulties and does not affect the efficiency of the stability test. It does, however, make it more difficult to generate the set of stabilizing fixels.

The stability test presented here is actually a straight-forward application of techniques previously developed for grasping and dexterous manipulation applications[13] and has been formulated, but not implemented by Palmer[12]. The test requires only the solution of a set of linear equations and inequalities and can be done quite quickly using a linear program solution algorithm, *e.g.* the simplex algorithm. Both the velocity-domain formulation and the dual force-domain formulation will be presented here.

The primary contribution of this paper is a method of finding the contact points for a set of rigid fixture elements, called fixels, which will stablize an unstable assembly. This can be viewed as a first step in the automated design of fixtures. The method involves parametrizing fixel locations along all reachable edges, and then using the simplex algorithm to minimize the sum of the magnitudes of the contact forces on those points. All fixels with zero contact forces are discarded, leaving a small set of fixels that stabilize the assembly.

### 1.1 Background

Previous work in assembly stability analysis falls roughly into two categories: approaches that rely on the Newton-Euler equations of motion[3, 4, 7, 12] and those which utilize intuitive, qualitative *ad hoc* procedures [6, 7, 8, 10]. Researchers who adopt qualitative procedures for stability testing often cite the complexity of solving the required equations as a reason for their choice. We find, however, that the relevant equations represent a linear program and can be solved very quickly using the simplex algorithm, and they model the system much more realistically and completely than most of the *ad hoc* procedures mentioned above.

This paper builds primarily on a stability test developed by Palmer[12], who studied the stability of general systems of polygons under the influence of gravity and in contact with or without friction. Palmer defined six stability classes which all collapse, in the fric-

*The research of this author was supported in part by the National Science Foundation Presidential Young Investigator grant number DDM-9057883.

†The research of this author was supported by the National Science Foundation under grant IRI-9304734.

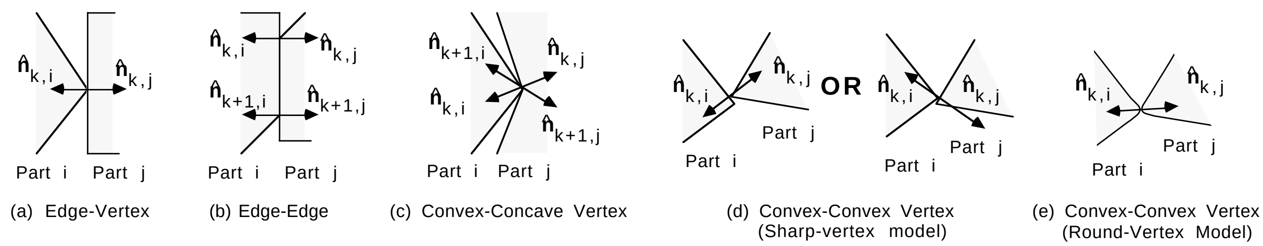



Figure 1: Definition of contact points and normal vectors for edge-vertex, edge-edge, and vertex-vertex cases

tionless case, to a pair of dual classification, referred to as "guaranteed stability" and its dual, "potential instability." In related work, Trinkle[14] derived analogous stability conditions for a single grasped frictionless object referred to as the "velocity formulation" and its dual, the "force formulation." When the velocity or force formulation is extended to multiple grasped objects and the finger velocities are set to zero, Palmer's and Trinkle's stability tests are identical. Since the test can be formulated as a linear program, stability can be determined efficiently with the simplex method.

This paper goes beyond a simple stability test. It shows how a variant of the force formulation can be used to find a small set of fixel positions that will stabilize an unstable assembly. The approach requires the solution of a single linear program in which the fixel locations (and the contact forces) appear as variables. With the exception of Fahlman's work, previous approaches to automated fixture design have been developed for only a single part (for example, see [1, 5]), and so they cannot be applied to the problem considered in this paper.

Fahlman[7] designed and implemented a "blocks world" task planner that automatically determined assembly plans for block structures. Some of the structures required temporary supports and counter balances, because not all partially completed assemblies were stable. When unstable subassemblies had to be used in order to complete the assembly, the planner estimated the fall-down motion of the blocks and identified the downward moving surfaces. Then, when possible, temporary scaffolds were placed to support those surfaces before the destabilizing block was placed.

Fahlman's stability test was not accurate, because he did not solve the required equations simultaneously. Instead, to save computation time, he used *ad hoc* rules to decouple the system of equations. As a result, his technique for finding extra supports to stabilize an assembly is not reliable. By contrast, the method we present here, solves all relevant equations simultaneously. When the equations have no solution, no stabilizing fixture exists. Otherwise, our approach will find the positions of a set of fixels which stabilizes the assembly.

## 2 Contacts

The first stage in analyzing the stability of an assembly is to discover the locations and orientations of the contacts among the parts. This is fundamentally problematical, because a geometric model of an assembly can never be a precise representation of the real assembly. It is quite possible for small inaccuracies to significantly affect the stability of the assembly. For the purposes of our analysis, however, we will assume that the locations and normals of the contacts can be derived from the geometric model with sufficient accuracy.

Parts will be labeled by indices $1 \le i \le n_b$, where $n_b$ is the number of movable parts. Unmovable parts, such as fixtures and table-tops, are not labeled. Each movable part may have a different coordinate frame, $\mathcal{B}_i$, defined for it.

Contacts will be labeled by indices $1 \le k \le n_c$, where $n_c$ is the number of contacts among all the parts and fixtures. We define $P(k)$ to be the set of movable parts involved in the the $k^{\text{th}}$ contact. This will always be a set of size one (if the contact is between a movable part and an unmovable part) or two (if the contact is between two movable parts).

Let $k$ be a contact and $i \in P(k)$ be a movable part involved in that contact. That contact will be characterized by its location $\mathbf{r}_{k,i}$ and its unit normal $\hat{\mathbf{n}}_{k,i}$. Both are expressed in part $i$'s coordinate frame, $\mathcal{B}_i$, and the normal is of unit length and directed inward with respect to part $i$. Thus, if $P(k) = \{i, j\}$, then $(\mathbf{r}_{k,i}, \hat{\mathbf{n}}_{k,i})$ and $(\mathbf{r}_{k,j}, \hat{\mathbf{n}}_{k,j})$ describe the same contact with oppositely directed normals represented in possibly different coordinate frames.

The existence of a contact means that there can be no relative motion between the contacting parts that causes interpenetration, *i.e.*, the relative linear velocity of the contact point on part $i$ with respect to the contact point on body $j$ may not have a component in the direction $\hat{\mathbf{n}}_{k,j}$ (the contact normal directed away from $i$ and into $j$).

Among two-dimensional polygons, contacts can be classified into three types: vertex-edge, edge-edge, vertex-vertex. In vertex-edge contacts, as shown in figure 1a, the normal vector would simply be perpendicular to the edge. Edge-edge contacts can be modeled by a two discrete contact points, one at either end of the contact segment as shown in figure 1b. This simplification does not allow us to represent arbitrary force distributions along the edge, but for our purposes it is only necessary that the kinematic constraints are correctly represented and that any resultant forces and moments that could arise from an arbitrary force distribution along a frictionless edge-edge contact be representable. In general, any planar contact region could be modeled as the set of vertices of its convex hull[2].

There are two kinds of two-dimensional vertex-vertex contacts: convex-concave, and convex-convex. (Two concave vertices can never be in contact.) The convex-concave case can be treated as a pair of contacts between the convex vertex and each of the two edges incident on the concave vertex, as shown in figure 1c. This is because the concave vertex is constrained by both edges.

The convex-convex case is more difficult, because the contact normal is "undetermined"[12]. Some authors assume this case does not occur, but in the idealized world of geometric models it is, in fact, quite common. There are two approachs commonly used to model convex-convex vertex contacts: the *sharp-vertex model* and the *round-vertex model.*

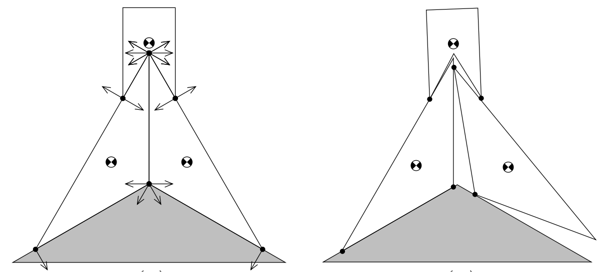
In the sharp-vertex model, it is assumed that the vertices are perfectly sharp points. This means that one vertex could either slide to the left of the other vertex, or to its right. Thus, one or the other of two possible contacts, as shown in figure 1d, will occur, but not both. To guarantee the stability of an assembly including a convex-convex contact, it must be stable when either one of these contacts, but not the other, is included in the contact set.

In the round-vertex model, we assume that each vertex is actually rounded, so the contact will actually be between two circular arcs. In this case there will be a single well-defined contact normal as shown in figure 1e.

The sharp-vertex model is superior in that it is the most conservative possible model. Any assembly stable under the sharp-vertex model will be stable under the round-vertex model. However, the either-or contacts that arise in the sharp-vertex model give rise to either-or constraints in the linear program. Such programs can be solved with mixed-integer programming techniques, but solving these is much slower than solving pure linear programs. For simplicity, we will assume the round-vertex model in the remainder of the paper.

## 3 Stability Testing

This section will describe the stability test algorithm. We are given a complete list of the contacts between a set of parts and a support structure, as well as the masses and centroids of the parts. Figure 2a shows an example of an input assembly, with contacts and centers of gravity marked. We will determine if the parts will accelerate if they begin at rest and there is no friction. Two alternative formulations of the test will be given.



(a) (b)

Figure 2: Picture of an example with three movable parts and one unmovable part(shaded). (a) Contacts among the parts. (b) Fall-down velocity found by the system applied for a short time interval.

### 3.1 Velocity Formulation

An assembly is stable (to first order) if all infinitesimal motions of the system consistent with the kinematic constraints strictly increase the potential energy of the system[14]. If a motion exists for which the potential energy decreases, then the assembly is unstable. Thus the stability of a given assembly can be assessed by determining whether the assembly's configuration defines a local minimum of its potential energy constrained by the fact that the parts may not overlap.

The stability tests developed below are derived by linearizing the constraint equations imposed on the assembly by the contacts. Therefore, they are able to assess an assembly's stability to first order. Since a first order stability test can be formulated as a linear program, stability can be determined efficiently by the simplex algorithm.

In some special situations, a first order analysis finds that no infinitesimal motion decreases the potential energy, but some motion leaves it unchanged. When such a situation occurs, the stability tests presented below cannot resolve the stability question, but they do provide an indication that second or higher order information is needed. A method for determining stability or instability using second-order information is presented in[15].

Following the approach used in [14], let $\dot{\mathbf{q}}$ be the generalized velocity vector of the bodies in the assembly. In two-dimensional assemblies, this is defined as:

$$\dot{\mathbf{q}} = [\dot{x}_1, \dot{y}_1, \dot{\theta}_1, \dot{x}_2, \dot{y}_2, \dot{\theta}_2, \ldots, \dot{x}_{n_b}, \dot{y}_{n_b}, \dot{\theta}_{n_b}]^T$$

where $\dot{x}_i$, $\dot{y}_i$ and $\dot{\theta}_i$ are the linear and angular velocities of the origin of the coordinate frame $\mathcal{B}_i$ of the $i^{\text{th}}$ part, and $n_b$ is the number of movable parts. The instantaneous change in potential energy and the instantaneous kinematic constraints can be written as functions of the generalized velocity to yield our stability test in the following form:

$$\text{Minimize:} \quad \dot{V}(\dot{\mathbf{q}}) \tag{1}$$

$$\text{Subject to:} \quad \text{Non-Penetration Constraints}(\dot{\mathbf{q}}) \tag{2}$$

where $\dot{V}$ is the time rate of change of the assembly's potential energy.

The linearized non-penetration constraint for each contact point is an inequality in the generalized velocity, so all the constraints on the motions of the $n_b$ parts arising from the set of $n_c$ distinct contacts can be written as:

$$\mathbf{W}_n^T \dot{\mathbf{q}} \geq \mathbf{0} \tag{3}$$

where $\mathbf{W}_n^T$ is the $n_c$ by $3n_b$ wrench matrix. Specifically, for each part $i \in P(k)$ involved in a contact $1 \leq k \leq n_c$, there will be three adjacent terms in $\mathbf{W}_n^T$ which are defined below:

$$\mathbf{W}_n^T[k, 3i] = (\hat{\mathbf{n}}_{k,i})_x \tag{4}$$

$$\mathbf{W}_n^T[k, 3i+1] = (\hat{\mathbf{n}}_{k,i})_y \tag{5}$$

$$\mathbf{W}_n^T[k, 3i+2] = \mathbf{r}_{k,i} \otimes \hat{\mathbf{n}}_{k,i} \tag{6}$$

Here $(\hat{\mathbf{n}}_{k,i})_x$ and $(\hat{\mathbf{n}}_{k,i})_y$ are the $x$- and $y$-components of $\hat{\mathbf{n}}_{k,i}$ and $\mathbf{r}_{k,i} \otimes \hat{\mathbf{n}}_{k,i}$ is $(\mathbf{r}_{k,i})_x(\hat{\mathbf{n}}_{k,i})_y - (\mathbf{r}_{k,i})_y(\hat{\mathbf{n}}_{k,i})_x$, the out-of-plane component of the cross product. All other elements of the $\mathbf{W}_n$ are zero. Note that the row of $\mathbf{W}_n$ corresponding to the $k^{\text{th}}$ contact will have a triplet of these values for each of the movable parts involved in that contact. In equation (3), these three elements will multiply $\dot{x}_k$, $\dot{y}_k$ and $\dot{\theta}_k$ respectively, giving rise to an expression for the separation velocity of the contact. As shown in equation (3), this must be non-negative for non-penetration to be satisfied.

The rate of gain of potential energy of the assembly can be written as $-\mathbf{g}_{ext}^T \dot{\mathbf{q}}$, where $\mathbf{g}_{ext}$ is referred to as the gravitational wrench or generalized force. Choosing the coordinate frames such that the origin of $\mathcal{B}_i$ coincides with the center of mass of the $i^{\text{th}}$ body and the $y$-axis is vertical (*i.e.*, directly opposite to the direction of the gravitational field), $\mathbf{g}_{ext}$ takes on the following simple form:

$$\mathbf{g}_{ext}^T = \left[ \begin{array}{c|c|c|c} \mathbf{g}_{ext,1}^T & \mathbf{g}_{ext,2}^T & \cdots & \mathbf{g}_{ext,n_b}^T \end{array} \right]^T \tag{7}$$

where $\mathbf{g}_{ext,i}^T = [0 \ -m_i g \ 0]$ is the gravitational wrench acting on part $i$, $m_i$ is the mass of part $i$, and $g$ is the acceleration due to gravity.

Substituting the expressions for the kinematic constraints and the rate of gain of potential energy into the optimization problem (1,2) yields a linear program in the unknown generalized velocity $\dot{\mathbf{q}}$ of the assembly. We refer to this linear program as the *velocity formulation*:

$$\text{Minimize:} \quad -\mathbf{g}_{ext}^T \dot{\mathbf{q}} \tag{8}$$

$$\text{Subject to:} \quad \mathbf{W}_n^T \dot{\mathbf{q}} \geq \mathbf{0} \quad . \tag{9}$$

When the velocity formulation has a unique solution (as do most linear programs), the velocity of the assembly will be zero, indicating stability.

When the velocity formulation is unbounded, the assembly is unstable. The kinematic constraints represent a polyhedral convex cone with its apex at the origin of the space of generalized velocities. Therefore, if a velocity of the assembly exists for which the objective function is negative, then that same velocity multiplied by a scalar satisfies the kinematic constraints and reduces the potential energy of the assembly also. Letting the scale factor go to infinity, we see that this linear program will be unbounded. However if the velocity formulation is be modified by placing arbitrary bounds on the magnitudes of the velocities (especially those in the downward direction), then a possible fall-down motion will be found by the linear programming algorithm (as shown, for example, in figure 2b). This will not typically be the actual fall-down motion, however. A more elaborate formulation based on the dynamic equations of motion[11] would be required if the intent were to simulate the motions of the parts as the assembly collapses.

### 3.2 Force Formulation

The dual linear program is referred to as the *force formulation* and is given as follows:

$$\text{Maximize:} \quad 0 \tag{10}$$

$$\text{Subject to:} \quad \mathbf{W}_n \mathbf{c} = -\mathbf{g}_{ext} \tag{11}$$

$$\mathbf{c} \geq \mathbf{0} \tag{12}$$

where the unknown, $\mathbf{c}$, is the vector of force magnitudes. Specifically, the $k^{\text{th}}$ element of $\mathbf{c}$ is the magnitude of the force at the contact point $k$.

Notice that the objective function of the force formulation is independent of the contact force magnitudes. This implies that the requirement for stability is that a $\mathbf{c}$ exist which satisfies the constraints (11) and (12). The former constraint is the set of equilibrium equations and the latter requires that the contacts be compressive. Thus, a frictionless assembly is stable only if equilibrium can be satisfied with compressive contact forces. If at least $3n_b$ elements of $\mathbf{c}_n$ are strictly positive, then the assembly is stable. Otherwise, as before, second order effects must be considered.

Since the two formulations of the stability problem are dual, either one maybe used to determine instability. When the assembly is stable, the solution of the velocity formulation, $\dot{\mathbf{q}}$, is identically zero and its the lagrange multipliers are one possible set of contact force magnitudes that stabilize the assembly. Solving the force formulation yields the same set of contact force magnitudes and its lagrange multipliers are the velocities which are all zero. When the assembly is unstable, the force formulation will be infeasible and the velocity formulation will be unbounded, so neither will give any information beyond the fact of instability.

## 4 Fixture Synthesis

Given an unstable assembly, we wish to find a small set of fixels that can be placed on the open edges of the parts to stablize the assembly. An open edge is any portion of the surface of any part that is not already in contact with another part.

One sure method of fixturing an assembly would be to cover every open edge of the assembly with contacts. This would be the equivalent of embedding the assembly in a block of concrete. This strategy would obviously stablize any assembly, but we would prefer to find a smaller set of contacts. To find this set, we take a four stage approach:

1. Add a finite number of fixels to each open edge of the assembly.
2. Construct a linear program using the force-formulation to incorporate the fixels.
3. Solve the linear program, minimizing the sum of the contact forces at the fixels. This will normally give many fixels with zero contact force.
4. Discard all fixels which have zero contact forces. The remaining fixels constitute a fixture design.

This approach suggests a modification of the force formulation presented in the previous section.

As we have observed before, any contact force distribution along a frictionless edge can replaced by a pair of frictionless point contacts at the endpoints of the contact region. Thus it will suffice in all cases to add just two fixels to each open edge. So $2n_e$ fixels will be added to an assembly with $n_e$ open edges. The positions of these added contacts will be parameterized, so that the solution procedure can move them to any point along their respective edges. These added contacts will be labeled from one to $2n_e$, while the $n_c$ known contacts are labeled from $2n_e+1$ to $2n_e+n_c$. We will use the variable $c_k$, $1 \leq k \leq 2n_e+n_c$, to represent the forces at contact $k$, and the variable $s_k$, $1 \leq k \leq 2n_e$, to represent the positions of added contact $k$.

Let $i$ be the part on which the $k^{\text{th}}$ added fixel is to be placed, and let $\mathbf{p}_{k,i}$ and $\mathbf{o}_{k,i}$ be the endpoints of the open edge of $i$ that it is to be placed on (expressed in part $i$'s frame of reference, $\mathcal{B}_i$). The variable $s_k$ will range from zero to one, and will parameterize this fixel's position. If $s_k$, is zero, then the $k^{\text{th}}$ fixel will be at the endpoint $\mathbf{p}_{i,k}$. If it is one, it will be at $\mathbf{o}_{i,k}$. Thus, in general, the $k^{\text{th}}$ fixel location on part $i$ will be:

$$\mathbf{r}_{k,i} = \mathbf{p}_{k,i} + \mathbf{e}_{k,i}\, s_k \tag{13}$$

where $\mathbf{e}_{k,i} = \mathbf{o}_{k,i} - \mathbf{p}_{k,i}$. The normal for fixel $k$ on part $i$, $\hat{\mathbf{n}}_{k,i}$, will be the inward-pointing normal of the edge. Note that $\hat{\mathbf{n}}_{k,i}$ does not vary with the fixel location.

The equilibrium constraint for moments on part $i$ will include terms for each added fixel $k$ on part $i$ of the form $(\mathbf{r}_{k,i} \otimes \hat{\mathbf{n}}_{k,i})\, c_k$. Since $\mathbf{r}_{k,i}$ is a function of the variable $s_k$, this gives a non-linear term in the constraint, since it contains the product $s_k c_k$ of two variables. However, if a dummy variable $d_k = s_k c_k$ is substituted in, all these terms become linear and take the form:

$$(\mathbf{p}_{k,i}) \otimes \hat{\mathbf{n}}_{k,i}\, c_k + (\mathbf{e}_{k,i}) \otimes \hat{\mathbf{n}}_{k,i}\, d_k \tag{14}$$

Combining these constraints with an objective function which sums the magnitudes of the fixel contact forces yields the following linear program:

$$\begin{aligned} \text{Minimize:}\quad & \mathbf{b}^T\mathbf{x} && && (15)\\ \text{Subject to:}\quad & \mathbf{A}\mathbf{x} && = -\mathbf{g}_{ext} && (16)\\ & \mathbf{B}\mathbf{x} && \geq \mathbf{0} && (17)\\ & \mathbf{x} && \geq \mathbf{0} && (18) \end{aligned}$$

where:

$$\begin{aligned} \mathbf{x} &= [c_1 \ldots c_{2n_e+n_c}\; d_1 \ldots d_{2n_e}]^T && (19)\\ \mathbf{B} &= [\mathbf{I}_{2n_e} \,|\, \mathbf{0}_{2n_e \times n_c} \,|\, -\mathbf{I}_{2n_e}] && (20)\\ \mathbf{b} &= [\underbrace{1 \ldots 1}_{2n_e}\; \underbrace{0 \ldots 0}_{n_c+2n_e}]^T && (21) \end{aligned}$$

where $\mathbf{I}_n$ is the identity matrix of size $n$ and $\mathbf{0}_{\mathbf{n}\times\mathbf{m}}$ is an $n \times m$ array of zeros. The matrix $\mathbf{A}$ is the $3n_b$ by $4n_e+n_c$ wrench matrix where for each fixel $1 \leq k \leq 2n_e$ on part $i$ we have

$$\begin{aligned} \mathbf{W}_n[3i,k] &= (\hat{\mathbf{n}}_{k,i})_x && (22)\\ \mathbf{W}_n[3i+1,k] &= (\hat{\mathbf{n}}_{k,i})_y && (23)\\ \mathbf{W}_n[3i+2,k] &= \mathbf{p}_{k,i} \otimes \hat{\mathbf{n}}_{k,i} && (24)\\ \mathbf{W}_n[3i+2,k+2n_e+n_c] &= \mathbf{e}_{k,i} \otimes \hat{\mathbf{n}}_{k,i} && (25) \end{aligned}$$

And for each part $i \in P(i)$ involved in contact $k$, $2n_e \leq k \leq 2n_e+n_c$, we have

$$\begin{aligned} \mathbf{W}_n[3i,k] &= (\hat{\mathbf{n}}_{k,i})_x\\ \mathbf{W}_n[3i+1,k] &= (\hat{\mathbf{n}}_{k,i})_y\\ \mathbf{W}_n[3i+2,k] &= \mathbf{r}_{k,i} \otimes \hat{\mathbf{n}}_{k,i} \end{aligned}$$

If this linear program is infeasible, then it is impossible to stablize the assembly with contacts on the given set of open edges. This would never occur unless some of the open edges have been eliminated from consideration.

If the linear program has a solution then the assembly can be stabilized. We eliminate all added contacts for which the contact force $c_k$ is zero. For the remainder, we can find the location of the contact on the edge by computing $s_k = d_k/c_k$. Since the simplex algorithm will usually produce many zeros, this usually gives a small set of contacts with minimum total contact force that will stablize the assembly.

## 5 Discussion

A C language implementation of the fixture generation procedure and both formulations of the stability test have been completed. The results presented here are based on implementations running on an IBM RS6000 using IMSL routines to solve the linear programs.

Figure 2 shows an example of the stability test. The two triangular bodies are assumed to have mass $m$ and the gravitational field points directly down the page.

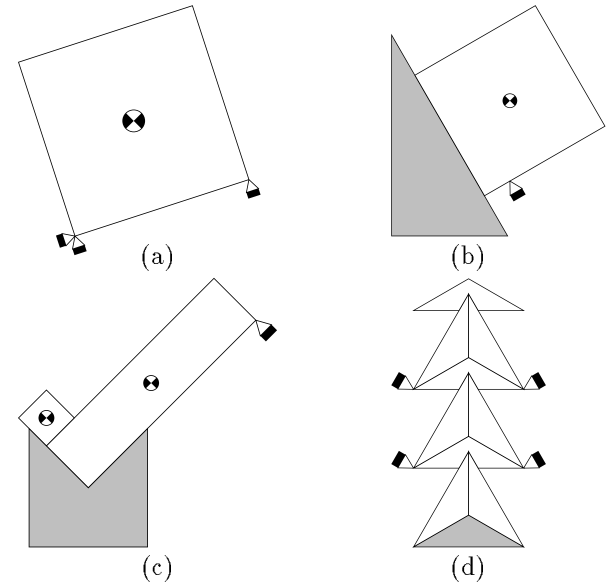



Figure 3: Fixel positions generated for four unstable assemblies. Shaded parts are unmovable and gravity is straight down.

It turns out that this assembly is stable if and only if the mass of the third body (the cap) is greater than or equal to $2.594m$. Figure 2b shows the fall-down motion found by solving the velocity formulation. Note that the cap is actually rising slightly while the triangles are descending. This result is very sensitive to the geometry. If the size of the cap is reduced by 25%, it must be 10 times as heavy as before to stablize the assembly, because its upward velocity is much smaller.

Figure 3 shows four examples of fixel positions on originally unstable assemblies. It should be noted that, because of our particular choice of objective function, the optimal solutions are non-unique in most cases, and thus the solution found may vary. In figure 3a a single free part is fixtured with two fixels on the bottom and one on the side. Since simplex always searches along the constraint surfaces, fixels tend to be placed at the extreme endpoints of the regions in which they could be placed. In figure 3b placing the fixel at either endpoint of the available lower edge would not stabilize the assembly. Instead, stability can be obtained if the fixel is placed anywhere in a particular region of that edge. The solution shown places the fixel at the lower limit of that feasible region. With this fixel position, the square is in unstable equilibrium. Any disturbance would cause the square to roll over the fixel point. We will discuss a remedy for this difficulty later. Figure 3c shows an assembly with two parts. If the square is made heavier, the assembly becomes stable, and no fixture point is added. Figure 3d shows a more complex assembly. The bottom three movable parts are instantaneously stable due to the weight of the parts above, but the others would fall down if not fixtured. The algorithm places four fixels, although two would suffice. We do not, in general get the minimum number of fixels with the current algorithm.

The run time of the algorithm is strongly dominated by the time to solve the linear program. The time to find the contacts and set up the matrices of coefficients typically accounted for less than one twenty-fifth of the total run time. Simplex algorithms typically run in time roughly proportional to the cube of the number of constraints[9], however the complexity of some of the linear programming routines tested appeared to be nearer to the fourth power of the number of constraints. Since the number of constraints is linear in the complexity of the assembly (there are $3n_b + 2n_e$ constraints in two dimensions), it appears that our method typically runs in roughly $\Theta(N^3)$ or $\Theta(N^4)$ time where $N$ is the size of the input. For example, two-dimensional assemblies with 10 to 15 parts typically had about 200 constraints and took 45 to 60 seconds of CPU time to solve.

The basic methods described here can be extended considerably. Extensions to three-dimensions have been done. Placing parameterized fixels on faces of three-dimensional polyhedra is a bit more complex, since two position parameters are needed for each fixel, and a more complex set of constraints are needed to constrain the fixels to the faces. However, if non-convex faces are decomposed into sets of convex faces, and dummy variable substitutions similar to the one in the two-dimensional case are used, then a set of linear equations can be produced.

Figure 4 shows a three-dimensional assembly consisting of a three-legged table and two boxes, one leaning against the other. The full description of this assembly leads to a linear program with 134 constraints (18 equilibrium constraints, 116 constraints keeping fixels on faces) in 274 unknowns (13 inter-part contact forces, 87 fixel contact forces, 174 fixel positions). The fixture shown in Figure 4 was designed for this assembly in 9.4 cpu seconds. Fixel 1, on the table top, is somewhat surprising. One might have expected it to stablize the table by replacing the missing leg with the alternate fixel 1 shown in grey. However since the foot print of the table legs is finite, the fixel on the tabletop actually has a slightly longer moment arm, so it requires slightly less force to stabilize the table this way. However, if the user prohibits fixels on the tabletop, then the fixel will be placed in the alternate position instead.

The primary difficulty with curved surfaces is that contacts between them often can not be precisely represented by a finite set of point contacts. For example, a cylinder standing upright on a plane has a circular contact which would have to be approximated by a polygon with a finite number of vertices. Having done this, the stability test can be extended in a straight-forward manner to handling curved surfaces, however placing fixels with parameterized positions on curved surfaces generally produces a nonlinear system of equations. One way to handle such surfaces is to use a sampling approach to selecting possible fixel positions instead of allowing them to move anywhere on

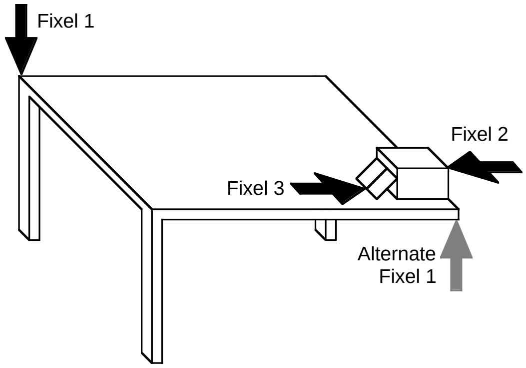



Figure 4: A three-dimensional assembly consisting of a three-legged table and two boxes standing on an unmovable floor (not shown). Contacts 1, 2 and 3 are found to stablize it. If contacts on the tabletop are excluded the alternate contact is used instead of contact 1.

the surface. We add contacts at a large number of fixed points scattered uniformly over the surface, solve the equations, and place fixels at all of the points where the forces were non-zero. This does not give the same assurance that we can stabilize any assembly as the parameterized fixel method does, but it is adequate for many assemblies.

In some cases it might be desirable to reformulate the problem to generate the minimum number of fixels. The objective would be to minimize the number of added contacts with non-zero forces. A problem with such an objective function can be solved by standard mixed-integer programming techniques.

As described above, the fixel selection algorithm tends to produce marginally stable fixturings that are easily upset. One method that has been developed to avoid such problems is to solve a series of linear programs, each with different disturbing forces added onto the gravitational forces. After each run all parameterized fixels with non-zero contact forces are added into the fixture design and taken as given for all future runs. This will lead to a fixture that will be stable for all the external forces tested, and for all positive linear combinations of those forces. Though the number of fixels placed by this method may not be minimal, tests have given good results[16].

In conclusion, we have described a basic technique for generating fixture points that is both efficient enough to be quite practical and readily extensible to a wide range of related problems. While our technique has several drawbacks, they can easily be remedied.

## References


[1] H. Asada and A. B. By. Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE Journal of Robotics and Automation*, RA-1(2):86–94, June 1985.

[2] D. Baraff. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica*, pages 292–352, October 1993.

[3] M. Blum, A. Griffith, and B. Neumann. A stability test for configurations of blocks. Memo AIM-188, Massachusetts Institute of Technology, February 1970.

[4] N. Boneschanscher, H. van der Drift, S. J. Buckley, and R. H. Taylor. Subassembly stability. *Proceedings, AAAI*, pages 780–785, 1988.

[5] M. R. Cutkosky and S. H. Lee. Fixture planning with friction for concurrent product/process planning. Technical Report TR 89-01, Stanford University Department of Mechanical Engineering, January 1989.

[6] L. S. Homem de Mello. A geometric reasoning approach to finding the stable orientations of an assembly. *IEEE Transactions on Robotics and Automation*, to appear 1994.

[7] S. E. Fahlman. A planning system for robot construction tasks. *Artificial Intelligence*, 5:1–49, 1974.

[8] B. V. Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13, 1980.

[9] Frederick S. Hillier and Gerald J. Lieberman. *Operations Research*. Holden-Day, Inc., San Francisco, second edition, 1974.

[10] S. Lee and Y. G. Shin. Assembly planning based on subassembly extraction. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1606–1611, May 1990.

[11] P. Lötstedt. Coulomb friction in two-dimensional rigid-body systems. *Zeitschrift für Angewandte Mathematik und Mechanik*, 61:605–615, 1981.

[12] R. Palmer. *Computational Complexity of Motion and Stability of Polygons*. PhD thesis, Cornell University Department of Computer Science, January 1987.

[13] J. C. Trinkle and R. P. Paul. Planning for dextrous manipulation with sliding contacts. *International Journal of Robotics Research*, 9(3):24–48, June 1990.

[14] J.C. Trinkle. On the stability and instantaneous velocity of grasped frictionless objects. *IEEE Transactions on Robotics and Automation*, 8(5):560–572, October 1992.

[15] J.C. Trinkle, A.O. Farahat, and P.F. Stiller. Second-order stability cells for frictionless rigid body systems. Technical Report TAMU-CS TR 93-020, Texas A&M University Department of Computer Science, April 1993.

[16] J. D. Wolter and J. C. Trinkle. Automatic selection of fixture points for frictionless assemblies. Technical Report 94-017, Texas A&M University, Computer Science Dept., Feb 1994.