Componential Structural Simulator

TECHNICAL REPORT CS-98-198

BRANDEIS UNIVERSITY DEPARTMENT OF COMPUTER SCIENCE

Pablo J. Funes and Jordan B. Pollack

1 A procedure to predict stability of certain structures

Our componential structural simulator procedure provides an approximate simulation that predicts resistance of structures made of modular components. The simulation focuses on torque strains and is able to predict stability of a structure whose breakage depends on torque stress.

Structures that can be described in this fashion include those made out of building toy bricks such as Lego[1] bricks, a well-known type of snap-on toy bricks, which we have used in our initial applications.

The model could be applied to many other kinds of structures made out of modular components. It is a prediction tool that can be programmed in a computer and used to test the stability of a structure before proceeding to its construction.

1.1 Networks of Torque Propagation in 2D

We begin considering two-dimensional systems, then generalize to three dimensions.

1.1.1 Two dimensional modular structures

We call ‘two-dimensional’ structures those made out of sets of component elements of the same width, assembled in a plane. Thus the structure can be fully described in a two dimensional plane (Figs. 1 and 2 are an example).

1. Lego is a trademark of The Lego Group.

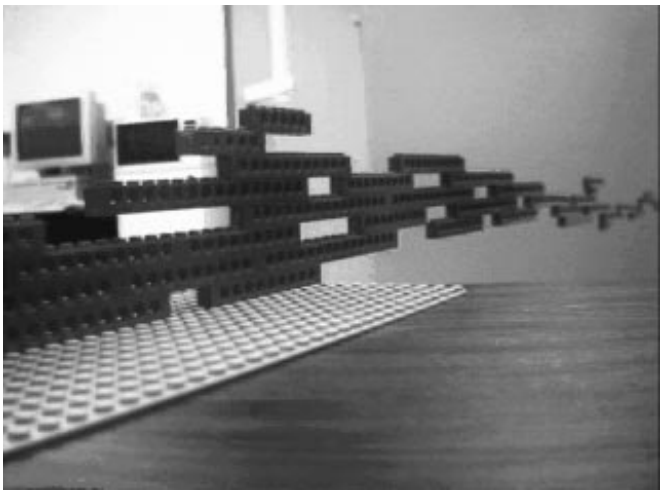

fig. 1. A ‘two dimensional’ structure made of Lego bricks. All components are the same width and arranged in a plane so the structure can be fully described in a two-dimensional scheme (fig. 2)

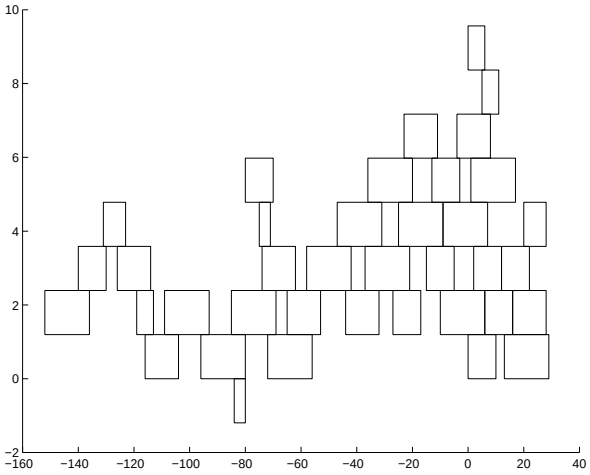

fig. 2. A two-dimensional schematic representation of the structure depicted in fig. 1. (The *x* and *y* axis are in different scales)

1.1.2 Requirements for using the componential structural simulator

Our method requires that

- The torque resistance of any union between a pair of components be predictable.
- The torque resistance of any union be smaller than its resistance to radial forces.
- The bricks or components themselves be stronger than the force required to separate them, so in a stress situation the bricks will separate rather than break.
- The entire structure be supported (grounded) at one or more points. The unions between the elements and the 'ground' must be of the same nature as those between elements.

1.1.3 Lego bricks torque resistance

The previous assumptions are satisfied by structures made out of building brick toys such as Lego bricks, with snap-on unions. The resistance of the plastic material of Lego bricks and similar building toys far surpasses the force necessary to break their unions. This allows us to apply a model that ignores the resistance of the material and evaluates the strain forces over a group of bricks only at their union areas. If a Lego structure fails, it will generally do so at the joints, but the actual bricks will not be damaged.

In the example of Lego bricks, our two-dimensional structures are composed of "1×n" bricks (Fig. 3).

fig. 3. Types of Lego bricks used in our 2D applications.

The amount of stress that different linear (1×1, 2×1, 3×1, etc., as in fig. 1) unions of brick pairs can support has been tentatively measured (table 1). We disregard radial forces such as vertical

pulls, and describe the system of static forces inside a complex structure of Lego bricks as a network of “rotational” joints located at each union between brick pairs and subject to loads coming from the weight of each brick.

Joint size(knobs)	Approximate torque capacity ($N\text{-}m \times 10^{-6}$)
1	10.4
2	50.2
3	89.6
4	157.3
5	281.6
6	339.2
7	364.5

Table 1. Estimated minimal torque capacities of the basic types of joints

In our examples we consider the ‘ground’ to have knobs to stick to bricks in the same fashion as bricks stick to one another. The structure can rest, for example, in a big Lego plate that is affixed to a table.

1.2 Building a Network of Torque Propagation

Given a two-dimensional structure formed by a combination of bricks, our model builds a network with joints of different capacities and external forces that must be in static equilibrium if the structure is not going to collapse. Each idealized joint is located at the center of the area of contact between a pair of bricks. A network of torque propagation (NTP) is composed of:

- A List of bodies
- A List of joints ‘JOINTLIST’
- A List of forces ‘FORCELIST’
- A symbol *G* - the ‘ground’

The network building procedure is as follows:

1.2.1 List of Joints

First all the joints between bricks are calculated and stored in a list.

1. JOINTLIST = ∅
2. For each unique pair *(b,b')* of bodies attached to each other (*)
3. Create Joint $J = J_{b,b'}$ with the following properties:
4. Position(*J*) = Center of area of contact between *b* and *b'*
5. Strength(*J*) = (obtained from table 1) (**)
6. add *J* to JOINTLIST
7. For each brick *b* that is attached to the ground
8. Create joint $J = J_{b,G}$ with the following properties:
9. Position(*J*) = Center of area of contact between *b* and the ground
10. Strength(*J*) = (obtained from table 1) (**)
11. add *J* to JOINTLIST

NOTES:

(*) *Unique pair* indicates that *(b',b)* should not be considered whenever *(b,b')* has already been processed.

(**) If the size is greater than the maximum entry in the table (7 in the example table 1), compute strength=size*strength(maximum entry tabulated)/size(maximum entry tabulated)

1.2.2 List of forces

Once the network is computed, a list of forces is calculated and submitted to it. Each brick in the structure has a weight and thus must be computed as one of the forces:

1. FORCELIST=∅
2. For each brick *b* in the structure compute
3. Position(*F*) = center of *b (*)*
4. Vector(*F*) = (0,-1)*weight of *b* (**)
5. Target(*F*) = *b*
6. add *F* to FORCELIST
7. For each external force *f* acting on *b*
8. Create new *F* such that:
9. Position(*F)* = point of application of *f*
10. Vector(*F*) = vectorial representation of *f*
11. Target(*F*) = *b*
12. add *F* to FORCELIST

NOTES:

(*) The center of the brick is a reasonable approximation of its center of mass

(**) Gravitational forces point down.

(***) The weight of a size *n* brick can be approximated by *n* * (weight of size 1 brick)

1.3 Solutions of a Network of torque propagation

1.3.1 An NTP generates a set of equations for structure stability

Each force applied to a component, either its own weight or an external load, has to be cancelled by one or more reaction forces if the brick is stable — otherwise it would be falling. Such reaction forces can originate in any of the joints that connect it to neighbor elements. In that case the force is transmitted through the joint to a connected component. Thus a load is propagated through the network until finally absorbed by a fixed body — the ‘ground’.

The structure is stable if and only if all forces can flow to the ground down the network to be absorbed by the ground. This is represented, for each force F, by a flow ϕ_F:BODIES $\cup$ $\{G\}\rightarrow$[-1,1] such that

$$\phi_F(b, c) = -\phi_F(c, b) \text{ for all } c, b \in \text{BODIES} \cup \{G\} \quad (1)$$

$$\phi_F(b, \text{BODIES} \cup \{G\}) = 0 \text{ for all } b \in \text{BODIES-}\{\text{target}(F)\}(*) \quad (2)$$

$$\phi_F(\text{target}(F), \text{BODIES} \cup \{G\}) = 1 \quad (3)$$

NOTE: (*) We use the abbreviated notation $\phi(b, X) = \sum_{x \in X} \phi(b, x)$ for a set X.

Thus ϕ_F constitutes a flow network (Cormen *et al.*, 1989, page 580) where the flow only occurs at the joints:

$$\phi_F(b, c) = 0 \text{ if neither } J_{b,c} \text{ nor } J_{c,b} \text{ exist in JOINTLIST} \quad (4)$$

A flow ϕ_F indicates that the force or weight is supported throughout the structure. A structure will be considered to be table if and only if a flow with a value of 1 (eq. 3) exists for each and every force such that the added torque for all flows at every joint respects the capacity of the joint:

$$|\tau(J)| \leq \text{Strength}(J) \quad (5)$$

where the total torque τ at a joint is defined by

$$\tau(J_{b, b'}) = \sum_{F \in \text{FORCELIST}} \tau(J_{b, b'}, F)\, \phi_F(b, b') \quad (6)$$

where $\tau(J_{b, b'}, F)$ is the torque exerted by a force of magnitude Vector(F) acting at Position(F) with axis of rotation located at Position($J_{b,b'}$).

If a solution to this network exists, it means that there is a way to distribute all the forces along the structure. The operating principle of the componential structural simulator is this: As long as there is a way to distribute the weights among the network of bricks such that no joint is stressed beyond its maximum capacity, the structure will not break.

1.3.2 Finding solutions for a NTP

From this strategy of modelling an algorithmic problem arises. Where nature simply distributes work dynamically through small deformations along the structure, our model needs an algorithm to determine the existence of solutions. We have not found a complete algorithm for this problem, but a "greedy" technique, not always capable of finding the solution when there is one, guarantees the stability of the structure in the numerous cases when a solution is actually found.

In the case where only one force is present, the problem is describable as a network flow algorithm (NFA) for maximum flow (Goldberg, 1989, Cormen *et al.*, 1989, ch. 12) and can be solved by known methods. The complete problem is not reducible, however, to a maximum flow problem, due to the fact that there are multiple forces to be applied at different points, and the capacity of each joint relative to each one varies with the magnitude of the force and the orthogonal distance between force and joint.

The greedy technique analyzes one force at a time. Once a solution has been found for the distribution of the first force, it is fixed, and a remaining capacity for each joint is computed that will conform a reduced network that must support the next force, and so on.

While there may be a better algorithm for solving the weight distribution for a stable Lego structure, an incomplete algorithm could be enough for many applications. Any structure that is approved as "gravitationally correct" by our simulation possesses a load distribution that does not overstress any joint, and thus will not fall under its own weight. Our model is thus conservative. It

might be wrong in predicting the breakage of a structure, but any shape approved by it is guaranteed to resist the required loads.

The possibility of adapting generalized versions of the NFA problem (Iusem and Zenios, 1995; Leighton *et al.*, 1995) to our problem remains to be explored.

The greedy algorithm is this:

1. Define TORQUE(*J)* = 0 for all $J \in$ JOINTLIST
2. For every $F \in$ FORCELIST
3. For every joint $J = J_{b,b'}$ define
4. $$\text{cap}(b, b') = \text{MIN}\{\frac{\text{Strength}(J) - \text{TORQUE}(J)}{\tau(J, F)}, 1\}$$
5. $$\text{cap}(b', b) = \text{MIN}\{\frac{\text{Strength}(J) + \text{TORQUE}(J)}{\tau(J, F)}, 1\}$$
6. Use a maximum flow algorithm to calculate a maximum flow ϕ_F from Target(F) to G as defined by equations (1 through 4) and capacities as per cap just computed.
7. If the value of the resulting maximum flow ϕ_F is not one
8. Then exit returning FAIL: a solution was not found for the NTP
9. Else for every joint $J = J_{b,b}$ compute
10. TORQUE(J) = TORQUE(J) + $\phi_F(b,b')\cdot\tau(J,F)$
11. Next F
12. Return SUCCEED: A solution exists.

This procedure uses a standard maximum network flow algorithm (Goldberg 1989) to obtain a partial solution for every force, ensuring at each point the compatibility with the capacities of the joints given the sum of the previous forces.

1.4 From 2- to 3-dimensional networks

To extend our model of networks of torque propagation to cover three-dimensional brick structures, our definition of joint needs to be extended.

Our definition of 3D joint is as follows: A union that can be described as a combination of two two-dimensional joints, one with an axis of rotation along the *x* axis and the other along the *y* axis. This combination allows us to evaluate a 3D structure as two independent 2D projections, one in the *x*-*z* plane and the other in the *y*-*z* plane.

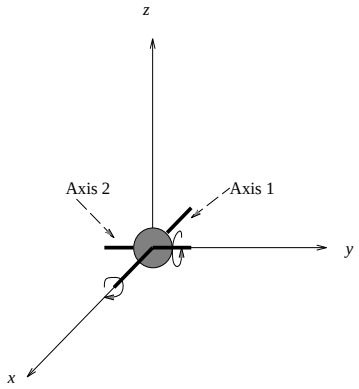

fig. 4. A 3D joint is considered to be composed of two independent joints, one with axis along *x* and the other along *y*.

1.4.1 Properties of 3D joints of snap-on bricks

Where before all brick unions could be described with one integer quantity, the number of knobs that join two bricks, in the three dimensional case these unions will be *n*-by-*m* rectangles. Two 2×4 bricks for example can be stuck together in 8 different types of joints: 1×1, 1×2, 1×3, 1×4, 2×1, 2×2, 2×3, 2×4. We know already, from the one dimensional case, how *n*×1 unions respond to forces acting along the *x* axis alone. A 2×1 union supports more than double the torque admitted by a 1×1, the reason being that the brick itself acts as a fulcrum (fig. 1). The distance from the border to the first knob is shorter than the distance to the second knob, resulting in a lower multiplication of the force for the second knob. This fulcrum effect does not happen when the force is orthogonal to the line of knobs. A 2×1 union can be considered as two 1×1 unions, or as one joint with double the strength of a 1×1 (fig. 2).

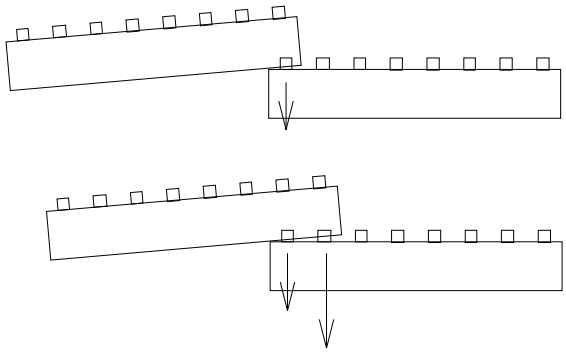

fig. 5. Fulcrum effect: a 1×2 union resists more than twice the load of a 1×1 because the second knob is farther away from the axis of rotation.

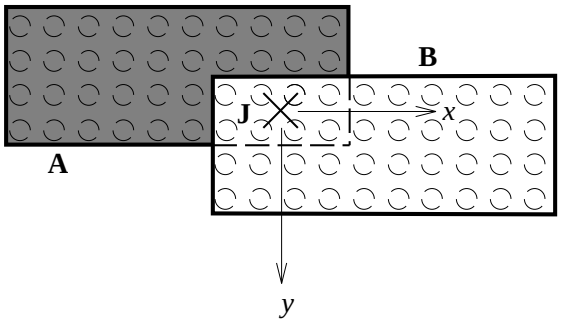

fig. 6. Two-dimensional brick joint: bricks A and B overlap in a 4×2 joint J. Along *x* the joint is a double 4×1 joint. Along the *y* axis it is a quadruple 2×1-joint.

Following these ideas we enunciate the following rule: Two bricks united by $n \times m$ overlapping knobs will form a joint with a capacity *Kx* along the *x* axis equal to *m* times the capacity of one *n*-joint and *Ky* along the *y* axis equal to *n* times the capacity of an *m*-joint.

To test the resistance of this composite joint to any spatial force *f* we have to separate it into its two components, *fx* on the *xz* plane and *fy* on the *yz* plane. These components induce two torques τx, τy. To break the union either τx must be larger than *Kx* or τy larger than *Ky.* With this procedure we induce, from a 3-dimensional brick structure, two separate 2-dimensional systems of joints, one for the *x* components of torques and joints and the other for the *y* components. The structure is stable if and only if both 2-dimensional projections are stable networks (figs. 4 and 5).

We are assuming that a dimensional independence hypothesis is true; it could be the case, however, that a force exerted along one axis will either weaken or strengthen the resistance in the orthogonal dimension. We made some exploratory experiments that suggested that the presence of stress along one axis does not modify the resistance along the other. It is probably the case that the rectangular shape of the joint makes it stronger for diagonal forces, justifying this simplification.

1.4.2 Solving a model

By separating each 3D joint into two orthogonal and independent 2D joints, which receive the *x* and *y* components of each external force, we can project an entire 3D network model of a bricks and joints structure into two orthogonal planes, *xz* and *yz*, generating two 2D networks that can be solved separately (figs. 4 and 5). Thus the problem of solving a 3D network does not add any more complexity to the existing problem of solving 2D networks of bricks and joints.

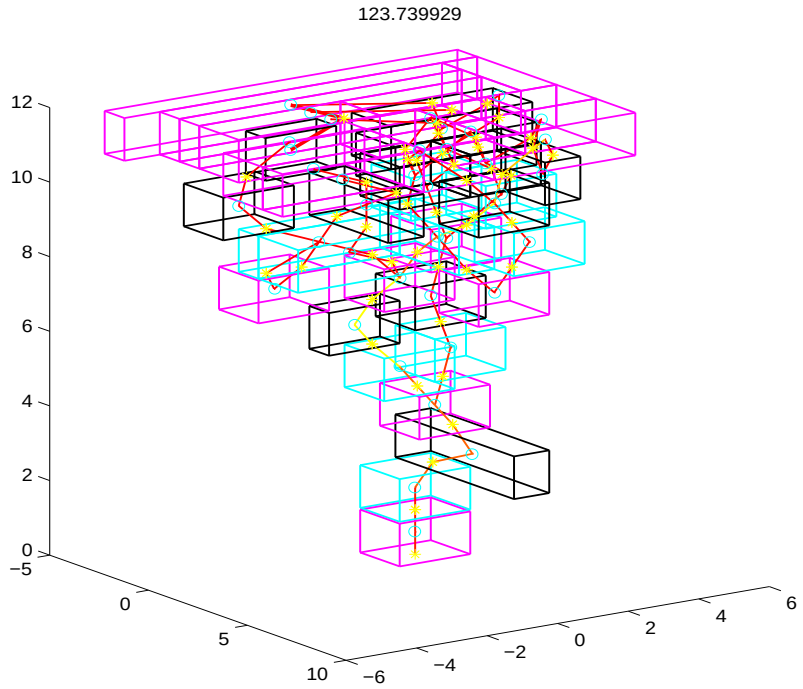

fig. 7. Computer-generated structure of Lego bricks.The underlying network model is shown.

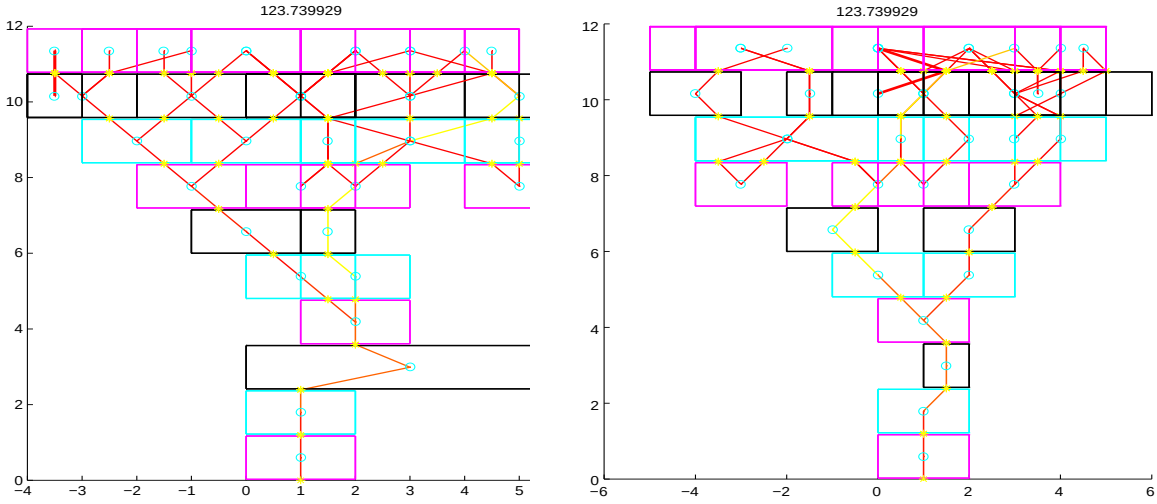

fig. 8. Projecting the 3D structure of fig. 4 to the *xz* and *yz* planes two 2D networks are obtained that can be solved independently.

2 Sample applications

2.1 Lego crane arm example

This application used the two-dimensional version of our structural simulator in a computer simulation that designs the arm of a crane. The crane needs to support its own structure plus an external load of up to 500 grams.

The computer was able to design the crane arm by itself and produce a crane arm meeting — according to our model — the specifications given. The crane was build afterwards following the computer design and tested to conclude the correctness of the design.

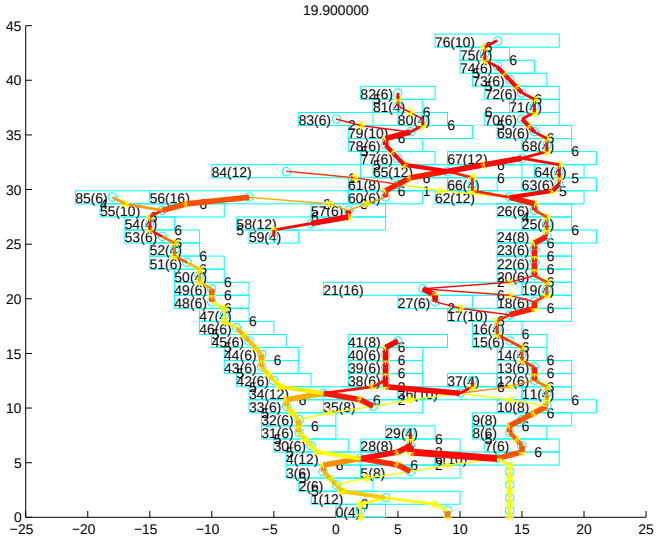

fig. 9. Computer-generated design for a crane arm made of Lego bricks

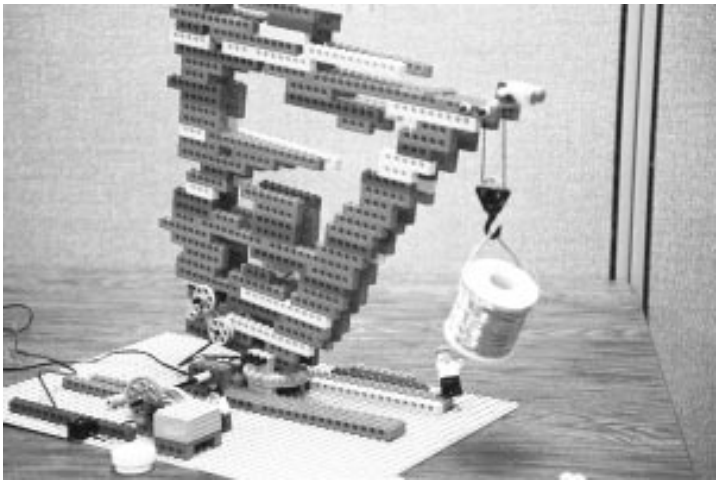

fig. 10. Lego crane built with the arm described by fig. 6 is shown supporting a 500g roll of solder as predicted by our model

2.2 Lego table example

This application is also an example of computed-generated design of structures using Lego bricks.

The 3D version of our componential structural simulator is used this time to generate the design for a table that has to support a weight of 50g all over its support surface.

The design produced by the computer and tested using the componential structural simulator was shown above (fig. 4). As in the earlier case, we have built the table using real bricks and tested it to prove that it behaves as predicted by our model.

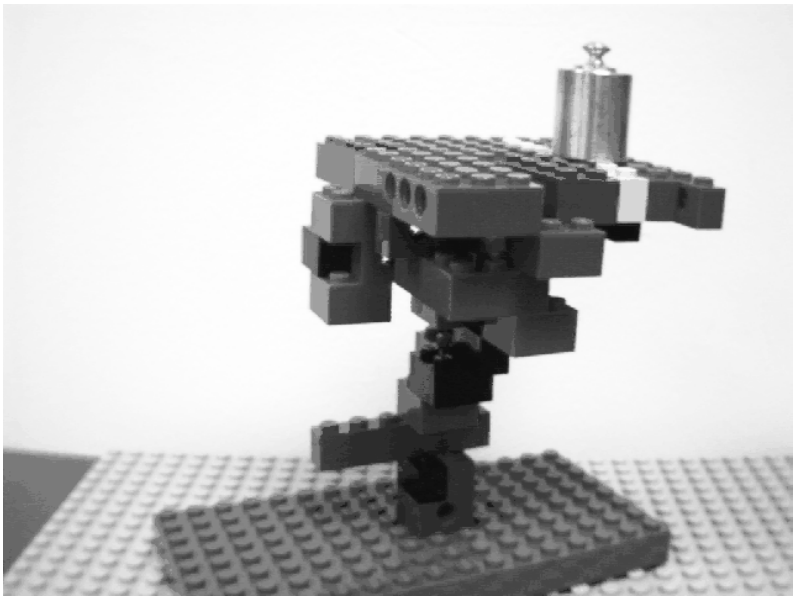

fig. 11. Computer-designed table. The stability of the structure was calculated with the componential structural simulator.

References

Cormen, T. H., Leiserson, C. E. and Rivest, R. L. (1989). *Introduction to Algorithms*. MIT press - McGraw Hill.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Iusem, A. and Zenios, S. (1995). Interval Underrelaxed Bregman's method with an application. In *Optimization*, vol. 35, iss. 3, p. 227.

Leighton, T., Makedon, F., Plotkin, S., Stein, C., Tardos, E. and Tragoudas, S. (1995). Fast Approximation Algorithms for Muticommodity Flow Problems. *Journal of Computer and Syst. Sciences* 50. p. 228-243.