# Buildable Objects Revisited

Martin Waßmann and Karsten Weicker

HTWK Leipzig, IMN, Postfach 301166, 04251 Leipzig, Germany
http://portal.imn.htwk-leipzig.de/fakultaet/weicker/

**Abstract.** Funes and Pollack introduced the The buildable objects experiments where LEGO® structures are evolved that may carry loads. This paper re-evaluates and extends the approach. We propose a new evaluation scheme using maximum network flow algorithms and a graph representation. The obtained structures excel previous results. Furthermore, we are now able to address problems that require more than one bearing.

**Keywords:** Buildable objects, physics simulation, evolutionary design.

## 1 Introduction

The evolution of structures with a load-carrying capacity, e.g., structural frame design or whole buildings, is an appealing part of evolutionary design. Since the work of Funes and Pollack [3–5], the evolution of LEGO® structures serves as an interesting application domain and research playground.

Although the problem is present for 15 years, there are still two major challenges:

- The stability of LEGO® structures is difficult to determine because the physics of structural frame design cannot be applied. And dynamic physics engine apply the various forces subsequently and isolated from each other which often leads to inexactness. A more precise approximation of the forces within a structure could improve the results considerably.
- Funes and Pollack [4] call their representation of the structure, a tree, underconstrained because it induces several problems, e.g., the overlapping of tree branches.

Also, if we consider problems with multiple bearings, the tree is not suited and it is unclear how well the existing evaluation of the stability works.

As a consequence, we present a graph-based representation together with a new technique for approximating the stability using flow networks based on the findings in the master's thesis [9]. However, the focus is on the representation and the results for a crane and a bridge problem with multiple bearings.

## 2 Related Work

There are two application domains of EAs using LEGO® bricks. First, Funes and Pollack [3–5] construct structures like cranes that are required to be stable in the

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 255–265, 2012.

presence of external forces. They use a tree-based representation and a repair function. However, since certain connections between bricks are not represented by the tree, the operators cannot use this information. Moreover, the approach is limited to single bearings. In [5], 3D models are constructed. Second, Petrovic [8] constructs models that mimic the shape of real-world objects. He uses a direct representation in form of a list of bricks (with the respective coordinates) – external forces are not considered.

Another relevant publication is the work of Devert et al. [2] who construct structures using "toy bricks" without considering pinches for connecting bricks.

The problem at hand appears to be closely related to structural frame design for which a comprehensive survey is available [7]. However, the very specific connection mechanism of LEGO® bricks inhibit re-using ideas from those approaches for both constructing and evaluating structures.

Related to our representation are research projects that evolve graphs, e.g. [6]. However, as we will see later, the graphs in our representation are very restricted such that knowledge transfer is difficult.

## 3 Evaluating Structures of Buildable Objects

The quality of a LEGO® structure depends on the evaluation mechanism, i.e., the computation of the forces within the structure. A graph model of the structure, e.g. in Fig. 1, enables the evaluation similarly to [3, 4].

Fig. 2 shows the forces that affect the stability of a LEGO® structure.

Mere compressive forces do not affect the stability of structures. Tensile forces might excess the adhesion of two connected bricks. But the moments are the primary cause for instable LEGO® structures when they exceed the capacity of a joint of two bricks. Basically, the forces are caused by the weight of each single brick as well as external loads.

Funes and Pollack [3–5] computed the stability by isolating forces and moments. For each force, the resulting moment is computed at each joint which leads to a flow network of moments and counteracting moments that are propagated towardsthe bearings. At each joint, the sum of all resulting moments for all forces must respect the moment capacity of the joint, i.e. it is a multi-commodity network flow problem. However, this approach overestimates the moments within the structure since the isolation of forces (and resulting moments) does not consider compensation of opposing moments.

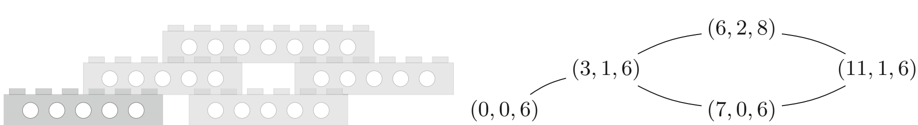


**Fig. 1.** Modelling a LEGO® structure: the placement of the bricks (left) and the corresponding graph where in each vertex the first two values are the position and the third value is the size of the brick (right)

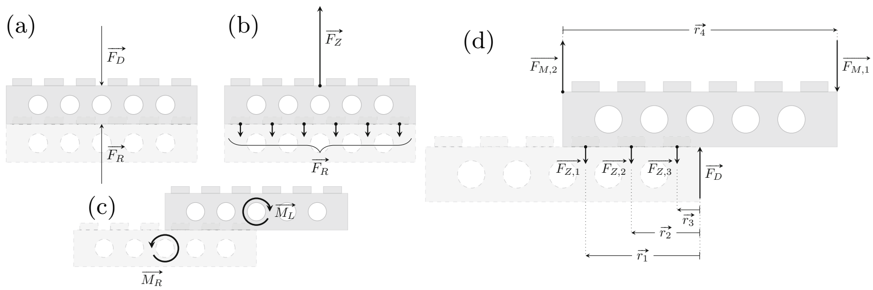


**Fig. 2.** Forces between bricks of a structure: (a) compressive force, (b) tensile force, and (c) moment. Subfigure (d) shows the moments from (c) as pairs of forces.

Our method for evaluating the stability of a structure follows the approach of classical statics. A system of equations is constructed, where the equilibrium of forces and moments for each brick is described by equations. Additional inequalities guarantee that no connection between two bricks is stressed beyond it's capacity. This system of equations can be solved if for each brick the external loads, e.g. gravitational forces or resulting moments, are counteracted by an equal and opposite reaction.

In general, such a system cannot be solved trivially as it may be underdetermined. We have investigated two techniques to solve such a system. First, constraint satisfaction solvers have been tried. However, the asymptotic runtime is exponential with the number of forces which leads to an unacceptable runtime. Instead, the system of equations is turned into maximum flow problems which can be solved in polynomial time using the push-relabel algorithm [1].

In a first phase, we consider only the compressive and tensile forces without moments – as if each brick is fixed in its orientation and cannot be tilted. For each brick the forces and counteracting forces are considered and a flow network is constructed that guarantees that all forces and counteracting forces are balanced. The maximum flow represents a possible distribution of the forces in the LEGO® structure – resulting in the effective forces for each single brick. If the tensile force exceeds the respective capacity of a joint the structure is not stable.

In a second phase, for each brick, the moments are computed from the effective forces. The moments need to be balanced too – modeled as a flow network with moments and counteracting moments. If the resulting effective moments exceed the capacity of a joint the structure is not stable.

The two phased approach allows us to use single commodity flow networks. However the approach has the disadvantage, that forces and moments are distributed successively and only one force distribution is calculated. Hence a *bad* force distribution may lead to an impossible moment distribution. The structure is declared not stable, even though there might exist a force distribution for which a moment distribution is possible. The approach is decribed in more detail in [10].

Exemplarily, Fig. 3 shows a simple LEGO® structure and the flow network model for distributing vertical forces. The source vertex $s$ supplies the forces due

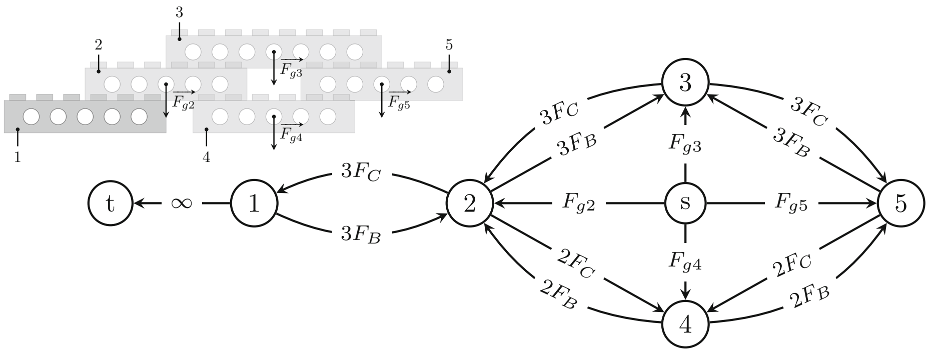


**Fig. 3.** A simple structure and the flow network modelling the forces with maximal compressive force $F_C$ and maximal tensile force $F_B$

to the bricks' weight and the target vertex $t$ is connected to the bearing. Each vertex represents a brick and each pair of edges between two vertices a joint of two bricks. Throughout the experiments, we use the maximal compressive force $F_c = 5.44$ kN and the maximal tensile force $F_B = 2.5$ N.

This approach has the advantage that we can consider problems with an arbitrary number of bearings. However, the computation of the moments is still an approximation since the distribution of forces might be unrealistic. Moreover, additional stabilizing factors, e.g., if bricks are placed side by side, are not considered. The maximal possible load for a structure is determined by binary search and iterative solution of the networks.

## 4 Concepts of the Evolutionary Algorithm

Given the expensive evaluation function described in the previous section, we decided to use a steady-state genetic algorithm to make the newly generated individual immediately available (similarly to [3]). In each generation one individual is produced with the following procedure:

1. select two parents with rank-based fitness-proportional selection
2. apply the recombination operator; if the new individual is invalid, use the first parent
3. apply the mutation operator; if the new individual is invalid, use the recombinant

The new individual replaces the worst individual in the population.

The LEGO® structures are represented directly as graphs like in Fig. 1. Such a graph $G = (B \cup L, E)$ contains the vertices $B$ (placeable bricks) and $L$ (bearings). Both are annotated by the position and the size of the brick. The edges in $E$ reflect the connections between bricks. We consider an individual to be *legal* iff the bricks do not overlap pairwise and the represented structure is stable.

For recombination, one of the following operations is applied to the parents' brick sets $B_1$ and $B_2$.

***horizontal cut:*** A brick $b \in B_1$ is chosen uniformly. The new individual contains all bricks $\{a \in B_1 | a_\mathrm{y} \leq b_\mathrm{y}\} \cup \{a \in B_2 | a_\mathrm{y} > b_\mathrm{y}\}$ where $c_\mathrm{y}$ denotes the y-coordinate of a brick $c$.

***vertical cut:*** A brick $b \in B_1$ is chosen uniformly. The new individuals contains all bricks $a \in B_1$ with $a_\mathrm{x} \leq b_\mathrm{x}$ where $c_\mathrm{x}$ denotes the x-coordinate of the left end of brick $c$. Furthermore, all bricks $a \in B_2$ are added that fulfil $a_\mathrm{x} > b_\mathrm{x}$ and do not overlap with the bricks selected from $B_1$.

***mixing:*** A random set of bricks $C \subseteq B_1$ is chosen. Let $D \subseteq B_2$ be the set of all bricks, that do not overlap with the bricks in $C$. The new individual contains $C \cup D$.

Because mixing is more disruptive, we use the probability 0.2 for mixing and 0.4 for the other two operators.

The mutation operator applies one of the following operations to the individual's brick set $B$.

***addition:*** A brick $a \in B$ with size $(t, 1)$ and a new brick $b$ with size $(t', 1)$ is chosen. The new brick is placed at position $(a_\mathrm{x} + z_1, a_\mathrm{y} + z_2)$ with random integer numbers $1 - t' \leq z_1 < t$ and $z_2 \in \{-1, 1\}$.

***deletion:*** A brick $a \in B$ is removed.

***shifting:*** A brick $a \in B$ with size $(t, 1)$ is moved to the new position $(a_\mathrm{x} + z, a_\mathrm{y})$ with random number $1 - t \leq z < t$.

***replacement:*** A brick $a \in B$ with size $(t, 1)$ is replaced by a new brick $b$ of size $(t', 1)$ where the centre of mass is kept unchanged, i.e., $b$ is positioned at $(a_\mathrm{x} + \lfloor \frac{t-t'}{2} \rfloor, a_\mathrm{y})$.

***exchange:*** Two bricks $a, b \in B$ are chosen and exchanged aligned according to the left end of the bricks.

***shifting partial structures:*** A random brick $a \in B$ is chosen as well as a random number $z \sim \mathcal{N}(0, 1)$. All bricks $b \in B$ with $b_\mathrm{y} > a_\mathrm{y}$ are moved by $\lfloor z + \frac{1}{2} \rfloor$ along the x-axis.

The operations are applied with probability 0.1 (first two) and 0.2 (rest).

In our experiments, we start with a population of empty individuals that contain only the given bearings. As a consequence, the evolution process focuses on growing functional structures during the early phases. More sophisticated initialisation procedures have not been investigated in detail.

## 5 Resulting Structures

Using our algorithm, we investigated two problems classes – the classical crane problem as in [3] and a new bridge problem, which is distinct from the bridge in [3] because we consider more than one bearing.

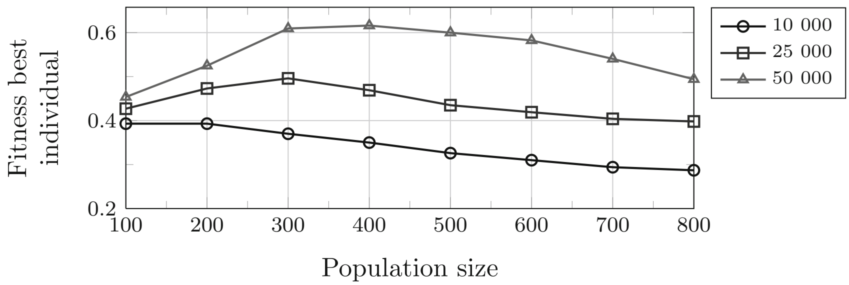


**Fig. 4.** Calibrating the population size: final fitness after $10^4$, $2.5 \cdot 10^4$, and $5 \cdot 10^4$ generations

### 5.1 Crane Problem

In the crane problem, a fixed bearing is given and the aim is to produce a structure that is able to carry a weight $load^* \in \mathbb{R}^+$ at a distance $length^* \in \mathbb{R}^+$ from the bearing. The tuple $(length^*, load^*)$ describes an instance of the problem class with which we can adjust whether we want short structures that can carry heavy loads or more outreaching structures.

The fitness of such a LEGO® structure $M$ is measured using the function

$$f_{\text{crane}}(M) = \left(\min\left\{1, \frac{length(M)}{length^*}\right\}\right)^s \cdot \min\left\{1, \frac{load(M)}{load^*}\right\} \cdot (1 - c \cdot size(M))$$

where $length(M)$ is the actual length of $M$, $size(M)$ the number of bricks, and $load(M)$ is the resulting maximum load using the physics simulation. Moreover, $s$ is a parameter to put more emphasis on the length of the structures (which seems to be more difficult to evolve); the parameter $c$ controls to what extent the number of bricks should be minimised.

All parameters have been investigated thoroughly for the crane problem. Fig. 4 shows how the final fitness changes with varying population size. A too big population size exhibits problems due to missing convergence. As a consequence, we used a population with 400 individuals and 50,000 generations.

Even more interesting are the parameters $s$ and $c$ to modify the fitness function. Fig. 5 shows how length and load capacity change for three different values of $s$. The inverse direction of the curves in the two subfigures shows that $s$ is a proper means to control the focus of the optimisation. However, as Fig. 6 demonstrates, there is not a unique scale for the different problem instances.

Parameter $c$ controls the impact of the number of bricks onto the fitness function. Fig. 7 shows that values between $10^{-4}$ and $10^{-3}$ decrease the number of bricks considerably with only little loss in length and load capacity.

Two of our results are shown in Figures 8 and 9. The crane for $(1 \text{ m}, 0.25 \text{ kg})$ was created using $s = 1.4$, $c = 10^{-9}$, and 50,000 generations. Remarkable about these crane arms is, besides achieved length and load-carrying capacity, their quality. The graph overlay in Fig. 8 shows the regular structure of the load arm, that allows the distribution of the applied load along as much as possible

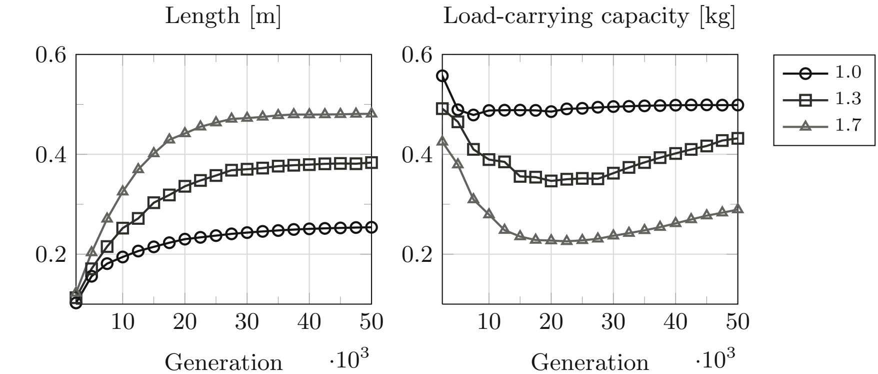


**Fig. 5.** Calibrating the factor $s$ for the problem instance $(0.5m, 0.5kg)$: length (left) and load capacity (right) for three values for $s$

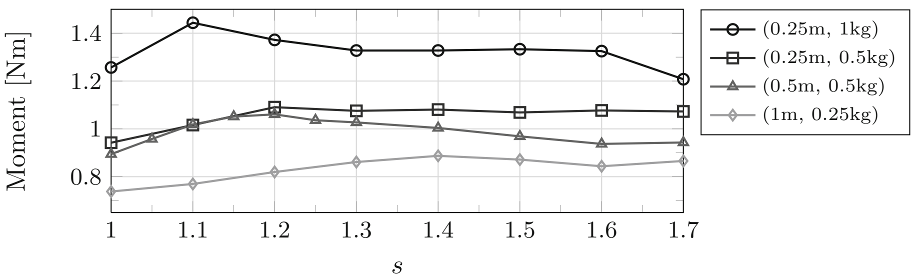


**Fig. 6.** Calibrating the factor $s$ for other problem instances: the maximum momentum for four problem instances and various values for $s$

bricks, reducing the strain per joint. Furthermore a counter balance is used to counteract the moment created by the load. These two features are reoccurring – depending on the parameter $c$, which inhibits the creation of a counter balance – and distinguish our solutions for the crane problem from those in the literature. The structure left of the bearing is randomly shaped since it only serves as a counter balance. The crane for $(0.5 \text{ m}, 0.5 \text{ kg})$ is the result of an experiment using $c = 10^{-3.5}$.

Table 1 compares the results of the Funes/Pollack approach and our results in load and length. Although a comparison is difficult, the results of the new approach appear to outperform the older results in both load and length.

### 5.2 Bridge Problem

The bridge problem requires the evolution to create a supporting structure between the bearings and the surface of the bridge where each brick is required to support a given load.

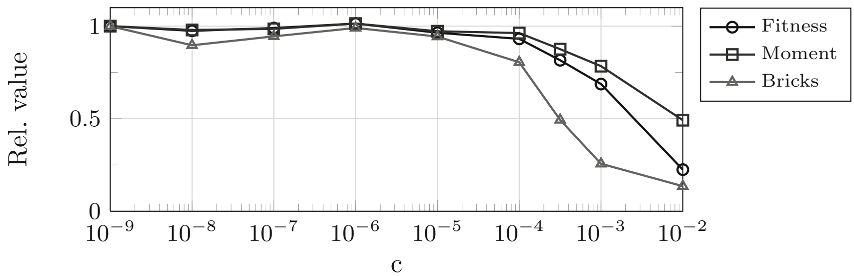


**Fig. 7.** Calibrating the factor $c$: normalised values for the number of bricks, the length, and the load capacity after $10^4$ generations

**Table 1.** Comparison of the constructed structures

| Funes/Pollack | | Waßmann/Weicker | | |
|---|---|---|---|---|
| length | load | length | load | problem instance |
| 0.5 m | 0.148 kg | 0.496 m | 0.482 kg | (0.5 m, 0.5 kg) |
| 0.16 m | 0.5 kg | 0.256 m | 1.0 kg | (0.25 m, 1 kg) |
| | | 0.664 m | 0.236 kg | (1 m, 0.25 kg) |
| | | 0.296 m | 0.41 kg | (0.5 m, 0.5 kg) |

The fitness function is designed in such a way that the evolution focusses on connecting all parts of the bridge first:

$$f_{\mathrm{bridge}}(M) = \begin{cases} \frac{1}{1+f_{\mathrm{con}}(M')}, & \text{if } M' \text{ is not connected} \\ 1 + p_{\max}(M'')^{\frac{1}{1+\delta}} \cdot c^{size(M)}, & \text{otherwise} \end{cases}$$

where $M'$ is the structure extended by the road surface as mounting, $f_{\mathrm{con}}(M')$ is the minimal distance between the connected components of the graph including the mounting bricks. $M''$ is constructed by adding iteratively those bricks of the surface for which the forces are supported by the LEGO® structure – the value $\delta$ corresponds to the number of unsupported bricks. The factor $c$ controls again the number of involved bricks – in the following experiments $c = 0.995$ was used.

A first scenario with three bearings is shown in Fig. 10 together with one of our first results. However, the bridge problem has not been analysed in the same depth as the crane problem. The structure in the lower left region serves as a counter balance and moves the centre of mass above the central bearing.

The second scenario (Fig. 11) has four bearings and requires a free space between the two central supports.

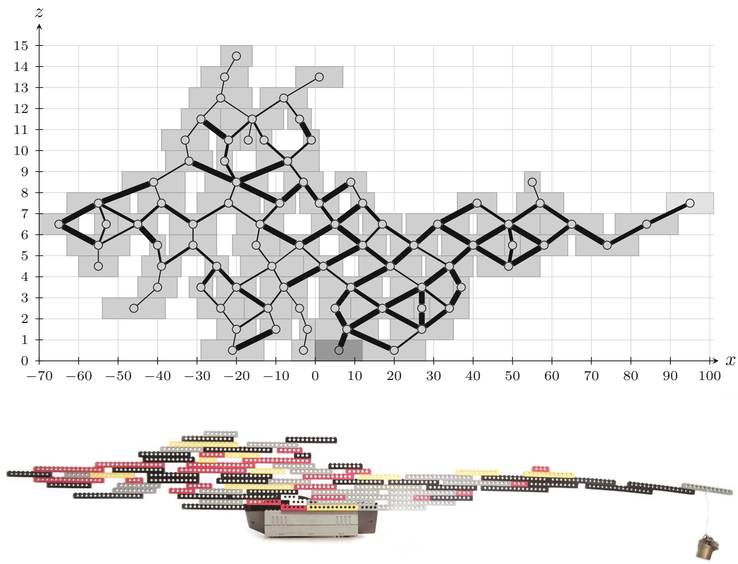

**Fig. 8.** Result for problem instance (1 m, 0.25 kg): length 0.664 m and load 0.236 kg

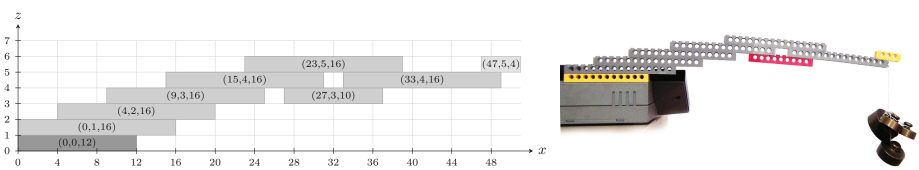


**Fig. 9.** Result for problem instance (0.5 m, 0.5 kg): length 0.296 m and load 0.41 kg

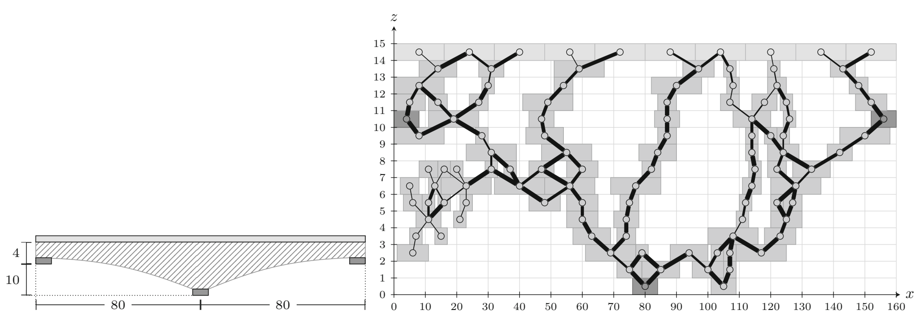

**Fig. 10.** First bridge problem

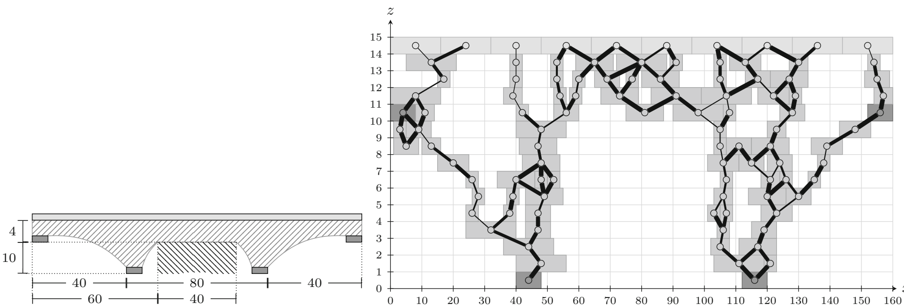


**Fig. 11.** Second bridge problem

## 6 Conclusion

The new algorithm convinces in two respects, the quality of the exemplary structures as well as the reliability with which competitive structures are produced – as the calibration results demonstrate. This is due to (a) a representation and respective operators that are designed specifically for the problem at hand and (b) a more exact evaluation mechanism for the structures.[1]

Where the crane problem has been investigated thoroughly, we presented only few preliminary results for the bridge problem with multiple bearings. Future research should focus on this problem class as well as 3D structures for which we tested our approach already successfully (not reported here). Furthermore, the computation of the stability could be extended to handle external horizontal forces like it would be necessary for tower structures.

## References


1. Cherkassky, B.V., Goldberg, A.V.: On implementing the push-relabel method for the maximum flow problem. Algorithmica 19, 390–410 (1997)
2. Devert, A., Bredeche, N., Schoenauer, M.: Evolution design of buildable objects with blind builder: an empirical study. In: Pham, T.L., Le, H.K., Nguyen, X.H. (eds.) Proc. of the Third Asian-Pacific Workshop on Genetic Programming, pp. 98–109. Military Technical Academy, Hanoi (2006)
3. Funes, P., Pollack, J.B.: Computer evolution of buildable objects. In: Husbands, P., Harvey, I. (eds.) Fourth European Conf. on Artificial Life, pp. 358–367. MIT Press, Cambridge (1997)
4. Funes, P., Pollack, J.B.: Computer evolution of buildable objects. In: Bentley, P.J. (ed.) Evolutionary Design by Computers, pp. 387–403. Morgan Kaufmann, San Francisco (1999)
5. Funes, P.J.: Buildable evolution. SIGEVOlution 2(3), 6–19 (2007)
6. Globus, A., Lawton, J., Wipke, T.: Automatic molecular design using evolutionary techniques. Tech. Rep. NAS-99-005, NASA Ames Research Center (1999)


[1] The source code is available for download at
`http://portal.imn.htwk-leipzig.de/fakultaet/weicker/forschung/download`

7. Kicinger, R., Arciszewski, T., De Jong, K.A.: Evolutionary computation and structural design: a survey of the state of the art. Computers & Structures 83(23-24), 1943–1978 (2005)
8. Petrovic, P.: Solving lego brick layout problem using evolutionary algorithms. In: Norsk Informatikkonferanse (NIK 2001), pp. 87–97 (2001)
9. Waßmann, M.: Physiksimulation und Evolution von LEGO®-Strukturen. Master's thesis, HTWK Leipzig, Leipzig, Germany (2011)
10. Waßmann, M., Weicker, K.: Maximum flow networks for stability analysis of LEGO® structures (2012), under submission (ESA 2012)