Parallel computation with molecular-motor-propelled agents in nanofabricated networks

Dan V. Nicolau Jr.[a,b,1], Mercy Lard[c,1], Till Korten[d,e,1], Falco C. M. J. M. van Delft[f,2], Malin Persson[g], Elina Bengtsson[g], Alf Månsson[g], Stefan Diez[d,e], Heiner Linke[c,3], and Dan V. Nicolau[h,i,3]

[a]Department of Integrative Biology, University of California, Berkeley, CA 94720-3140; [b]Molecular Sense, Ltd., Wallasey CH44 1AJ, United Kingdom; [c]NanoLund and Solid State Physics, Lund University, S-22100 Lund, Sweden; [d]Center for Molecular Bioengineering (B CUBE) and Center for Advancing Electronics Dresden (cfaed), Technische Universität Dresden, 01069 Dresden, Germany; [e]Max Planck Institute of Molecular Cell Biology and Genetics, 01307 Dresden, Germany; [f]Philips Research (MiPlaza) and Philips Innovation Services, 5656 AE, Eindhoven, The Netherlands; [g]Department of Chemistry and Biomedical Sciences, Linnaeus University, SE-39182 Kalmar, Sweden; [h]Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool L69 3GJ, United Kingdom; and [i]Department of Bioengineering, McGill University, Montreal, QC, Canada H3A 0C3

Edited by Hillel Kugler, Microsoft Research, Cambridge, United Kingdom, and accepted by the Editorial Board January 18, 2016 (received for review June 5, 2015)

The combinatorial nature of many important mathematical problems, including nondeterministic-polynomial-time (NP)-complete problems, places a severe limitation on the problem size that can be solved with conventional, sequentially operating electronic computers. There have been significant efforts in conceiving parallel-computation approaches in the past, for example: DNA computation, quantum computation, and microfluidics-based computation. However, these approaches have not proven, so far, to be scalable and practical from a fabrication and operational perspective. Here, we report the foundations of an alternative parallel-computation system in which a given combinatorial problem is encoded into a graphical, modular network that is embedded in a nanofabricated planar device. Exploring the network in a parallel fashion using a large number of independent, molecular-motor-propelled agents then solves the mathematical problem. This approach uses orders of magnitude less energy than conventional computers, thus addressing issues related to power consumption and heat dissipation. We provide a proof-of-concept demonstration of such a device by solving, in a parallel fashion, the small instance {2, 5, 9} of the subset sum problem, which is a benchmark NP-complete problem. Finally, we discuss the technical advances necessary to make our system scalable with presently available technology.

parallel computing | molecular motors | NP complete | biocomputation | nanotechnology

Many combinatorial problems of practical importance, such as the design and verification of circuits (1), the folding (2) and design (3) of proteins, and optimal network routing (4), require that a large number of possible candidate solutions are explored in a brute-force manner to discover the actual solution. Because the time required for solving these problems grows exponentially with their size, they are intractable for conventional electronic computers, which operate sequentially, leading to impractical computing times even for medium-sized problems. Solving such problems therefore requires efficient parallel-computation approaches (5). However, the approaches proposed so far suffer from drawbacks that have prevented their implementation. For example, DNA computation, which generates mathematical solutions by recombining DNA strands (6, 7), or DNA static (8) or dynamic (9) nanostructures, is limited by the need for impractically large amounts of DNA (10–13). Quantum computation is limited in scale by decoherence and by the small number of qubits that can be integrated (14). Microfluidics-based parallel computation (15) is difficult to scale up in practice due to rapidly diverging physical size and complexity of the computation devices with the size of the problem, as well as the need for impractically large external pressure.

Here, we propose a parallel-computation approach, which is based on encoding combinatorial problems into the geometry of a physical network of lithographically defined channels, followed by exploration of the network in a parallel fashion using a large number of independent agents, with very high energy efficiency. To demonstrate operational functionality, we applied it to a small instance of a benchmark classical nondeterministic-polynomial-time complete (NP-complete) problem (16), the subset sum problem (SSP) (Fig. 1). This problem asks whether, given a set $S = \{s_1, s_2, ..., s_N\}$ of N integers, there exists a subset of S whose elements sum to a target sum, T. More formally, the question is whether there is a solution $\sum_{i=1}^{N} w_i s_i$ where $w_i \in \{0, 1\}$, for any given T from 0 to $\sum_{i=1}^{N} s_i$. To find all possible subset sums by exploring all possible subsets requires the testing of 2^N different combinations, which—even for modest values of N—is impractical on electronic computers because of exponentially increasing time requirements (*SI Appendix*, section S1). Although more sophisticated algorithms exist (17–19), none of these avoids the exponentially growing exploration time, a property that is harnessed in some cryptography systems to generate encoded messages (20).

Significance

Electronic computers are extremely powerful at performing a high number of operations at very high speeds, sequentially. However, they struggle with combinatorial tasks that can be solved faster if many operations are performed in parallel. Here, we present proof-of-concept of a parallel computer by solving the specific instance {2, 5, 9} of a classical nondeterministic-polynomial-time complete ("NP-complete") problem, the subset sum problem. The computer consists of a specifically designed, nanostructured network explored by a large number of molecular-motor-driven, protein filaments. This system is highly energy efficient, thus avoiding the heating issues limiting electronic computers. We discuss the technical advances necessary to solve larger combinatorial problems than existing computation devices, potentially leading to a new way to tackle difficult mathematical problems.

Author contributions: Dan V. Nicolau Jr. and Dan V. Nicolau conceived the calculation method and designed the overall network; F.C.M.J.M.v.D designed the junctions; M.L., T.K., F.C.M.J.M.v.D, A.M., S.D., and H.L., designed the device layouts; M.L. and F.C.M.J.M.v.D fabricated the devices; M.L., T.K., M.P., and E.B. ran motility experiments and analyzed motility data; Dan V. Nicolau Jr., T.K., and A.M. carried out numerical simulations; Dan V. Nicolau initiated the project; Dan V. Nicolau and H.L. coordinated the project; and Dan V. Nicolau Jr., M.L., T.K., F.C.M.J.M.v.D., M.P., E.B., A.M., S.D., H.L., and Dan V. Nicolau contributed to planning the work, to data interpretation, and to writing the manuscript.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission. H.K. is a guest editor invited by the Editorial Board.

Freely available online through the PNAS open access option.

[1]Dan V. Nicolau Jr., M.L., and T.K. contributed equally to this work.

[2]Present address: Molecular Sense, Ltd., Wallasey CH44 1AJ, United Kingdom.

[3]To whom correspondence may be addressed. Email: heiner.linke@ftf.lth.se or dan.nicolau@mcgill.ca.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1510825113/-/DCSupplemental.

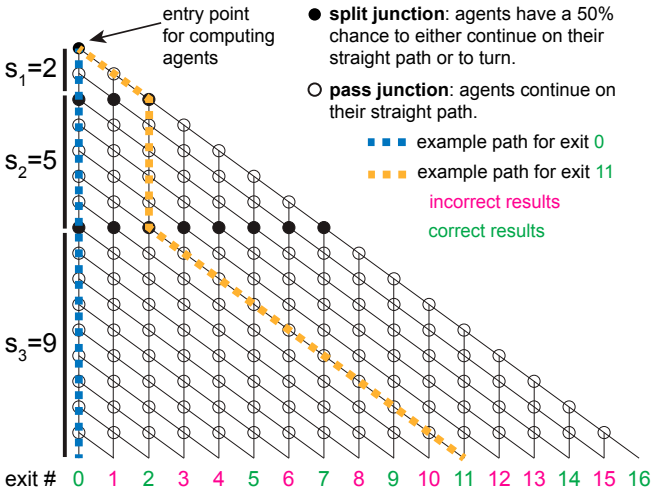


Fig. 1. Computation network for the SSP {2, 5, 9}. The agents enter the network from the top-left corner. Filled circles represent split junctions where it is equally probable that agents continue straight ahead or turn. Empty circles represent pass junctions where agents continue straight ahead. Moving diagonally down at a split junction corresponds to adding that integer (numbers 2 and 9 for the yellow example path). The actual value of the integer potentially added at a split junction is determined by the number of rows of junctions until the next split junction. The exit numbers correspond to the target sums *T* (potential solutions) represented by each exit; correct results for this particular set {2, 5, 9} are labeled in green, and incorrect results (where no agents will arrive) are labeled in magenta. The working principle is also detailed in Movie S1.

Our approach replaces the requirement for exponentially growing time needed by traditional, electronic computers to solve NP-complete problems, with the requirement for an exponentially growing number of independent computing agents. We use a proof-of-concept device to successfully solve the specific three-variable instance {2, 5, 9} of the SSP. Key technical advancements necessary to scale up our approach to be of practical relevance include the need to reduce error rates and to supply sufficiently many computing agents. We identify several possible approaches to address these requirements.

Results

In our network encoding of the SSP, the channel-guided unidirectional motions of agents are equivalent to elementary operations of addition, and their spatial positions in the network are equivalent to "running sums." Starting from an entrance point at one corner of the network (Fig. 1, *Top Left*), agents are guided unidirectionally downward by the channels in vertical or diagonal directions. Two types of junctions were designed to regulate the motion of agents in the network: (*i*) "split junctions," where agents are randomly distributed between two forward paths, and (*ii*) "pass junctions," where agents are guided onward to the next junction along the initial direction. The vertical distance (measured as the number of junctions) between two subsequent rows of split junctions represents an integer from the set *S*. The process of an agent moving straight downward from a given split junction is equivalent to excluding the corresponding s_i from the summation, whereas traveling diagonally downward is equivalent to including that s_i in the subset sum. A solution $\Sigma w_i s_i = T_J$ to the SSP is represented by an agent choosing a path to one of the exit nodes in the network (bottom row in Fig. 1). If a sufficiently large number of agents is used, all possible paths are explored, and therefore all possible subset sums of *S* are generated, simultaneously.

We implemented the proposed computational approach with biological agents that satisfy the following requirements: The agents (*i*) are available in large numbers at negligible cost; (*ii*) are self-propelled and thus do not require a global, external driving force; (*iii*) operate independently of each other to ensure parallel exploration; (*iv*) have small dimensions to enable use in high-density networks with high computing power per unit area; (*v*) move rapidly to maximize computational speed; and (*vi*) move in a predominantly forward direction (to ensure low error rates). In particular, we used cytoskeletal filaments (actin filaments and microtubules), which are propelled by molecular motors (myosin II and kinesin-1, respectively) along a surface in gliding motility assays (21, 22). Both kinds of cytoskeletal filaments have small diameters (~10 nm for actin filaments and ~25 nm for microtubules) and move at high speeds (5–10 μm s^{-1} for actin filaments driven by fast myosin II from skeletal muscle, and ~0.5–1 μm s^{-1} for microtubules driven by kinesin-1). The filaments are guided unidirectionally (23, 24) along lithographically defined channels, which are functionalized with molecular motors, and whose roofs are open to supply the motors with biochemical fuel (adenosine 5′-triphosphate, ATP) by diffusion from the surrounding buffer fluid (25, 26), allowing for a distributed energy supply. The width of the channels was set to below 200 nm and 250 nm, for actin filaments and microtubules, respectively, which were observed to reliably guide cytoskeletal filaments (27–29).

The general layouts of the networks used in our devices for microtubule–kinesin- and actin–myosin–based computations are shown in Fig. 2 and *SI Appendix*, Fig. S2.1, respectively. To start the computation, the filaments are collected from the bulk solution and guided to enter the network through loading zones

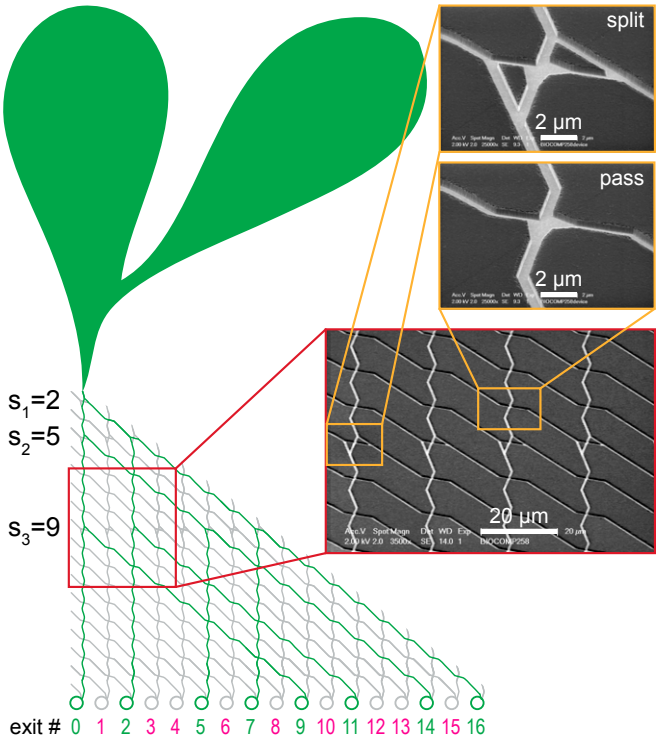


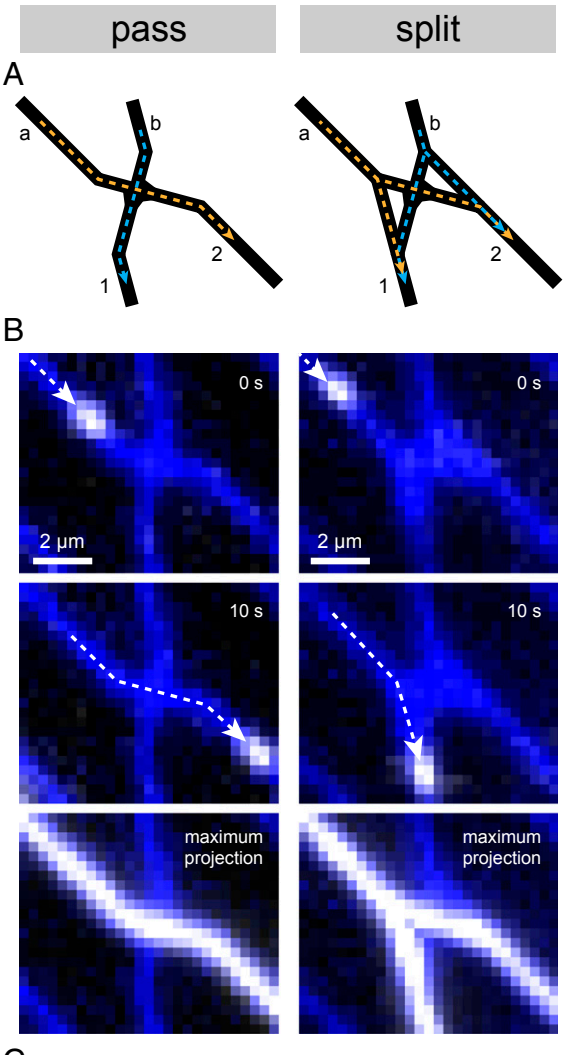
Fig. 2. Device layout of a computation device for the SSP {2, 5, 9}. Schematic of the actual device layout used for microtubules, including the loading zones for the microtubules (green balloon-like areas), the channels traversed by the microtubules during calculation (green lines), and the channels that should not be traversed (gray lines). Exit numbers corresponding to correct results are shown in green; numbers corresponding to incorrect results are shown in magenta. The circles at each exit are designed to store filaments for easy readout. (*Insets*) Scanning electron micrographs of parts of the network used for microtubules, showing a split- and a pass junction. See *SI Appendix*, section S2 for a corresponding schematic layout of the device that was used with actin filaments, including details of the rectifiers used for that device.

(large tear-shaped areas in Fig. 2 and *SI Appendix*, Fig. S2.1). The computational networks comprise a set of standardized rectangular lattices, each containing two isomorphic unit cells representing the split junctions and pass junctions (Fig. 2, *Inset* and *SI Appendix*, Fig. S2.1). This standardized structure facilitates the future encoding of problems of any size and the scalability of fabrication, e.g., through step-and-repeat deep-UV lithography, for which the fabrication of channels with widths of 200 nm (as reported here), or below, is readily achievable.

To account for different stiffness and size of actin filaments compared with microtubules, we optimized the device design and the junction geometries individually for each filament system with the assistance of numerical modeling and simulation (*SI Appendix*, section S4). After traversing the network, the filaments emerge at exits corresponding to the target sums and are either recycled back to the entrance point (actin–myosin device; *SI Appendix*, section S2) or collected (microtubule–kinesin device; Fig. 2). The networks were fabricated by electron-beam lithography on SiO_2 substrates to obtain the required resolution and fidelity (see *Materials and Methods Summary* for fabrication details).

The minimization of computation errors requires that the error rates of pass junctions are as low as possible, i.e., that filaments do not progress along erroneous paths and emerge at exit nodes not corresponding to target sums (*SI Appendix*, section S1). In contrast, the error rates of split junctions (designed to yield a 50:50 split) are less critical to computational performance because these junctions mainly serve to distribute agents across the network, thus ensuring that no solution is missed. Experiments (see *Materials and Methods Summary* for experimental procedures and imaging details) confirmed that the junction designs in our devices fulfill these performance requirements (Fig. 3). Fluorescently marked cytoskeletal filaments traversing individual split junctions and pass junctions (Fig. 3*A*) were tracked using fluorescence microscopy (Fig. 3*B*). Statistical analysis of the motion of actin filaments and the microtubules showed that 97.9% and 99.7%, respectively, took the correct (straight) paths through pass junctions, whereas split junctions distributed filaments approximately evenly (Fig. 3*C* and *SI Appendix*, section S5).

Our proof-of-concept experiments on the set {2, 5, 9} show that both the actin–myosin system and the microtubule–kinesin system can be used to solve a combinatorial problem by parallel computation (Fig. 4). Superimposed fluorescence micrographs show time-integrated paths of the fluorescently labeled filaments (Fig. 4*A*). These images demonstrate that the filaments traversed the network from the entrance point to the exit nodes that represent correct results (Fig. 4*A*, green exit numbers). Statistical analysis of the number of filaments leaving each exit (obtained by counting the filaments in image sequences tracking each filament) confirms that both types of agents found all of the correct results and that significantly more agents ($P < 0.002$; unpaired two-tailed t test) of both types exited nodes corresponding to correct results than incorrect results (Fig. 4*B*). The experimental data are in good agreement with those obtained by Monte Carlo simulations (Fig. 4*C* and *SI Appendix*, section S6), which are based on the experimentally measured error rates at each junction type (obtained separately for actin filaments and microtubules; Fig. 3*C* and *SI Appendix*, section S5).



	a2+b1	a1+b2	n
Actin	97.9%	2.1%	3669
MT	99.7%	0.3%	1801

a1	a2	b1	b2	n
50%	50%	42%	58%	1492
49%	51%	51%	49%	705

Fig. 3. Performance of pass junctions (*Left*) and split junctions (*Right*) with actin filaments and microtubules as agents. (*A*) Schematic drawing of a pass junction (*Left*) and a split junction (*Right*). Entrance channels for agents moving diagonally in the network are labeled a, whereas entrance channels for agents moving straight downward are labeled b. Exit channels are labeled 1 for straight downward and 2 for diagonally moving agents, respectively. Intended paths through the junctions are indicated by yellow (agents entering diagonally) and blue (agents entering straight downward) dotted lines. (*B*) Fluorescence micrographs of microtubules moving in a pass- (*Left*) or a split junction (*Right*). Paths of the microtubules in previous frames and direction of movement are indicated by white dotted lines and arrows, respectively. The bottom row of images shows maximum projections of several agents moving through the respective junctions. (*C*) Analysis of the error rates. For pass junctions, agents moving from entrance a to exit 2 and agents moving from b to 1 were behaving correctly (column a2 + b1). For split junctions, agents from each entrance were intended to be split evenly between exits (ideally, both would be 50%). Columns headed *n* denote the number of filaments analyzed for each junction type. MT, microtubules. See *SI Appendix*, section S6 for details.

Discussion

We developed a parallel-computation approach based on encoding combinatorial problems into the geometry of physical networks. We showed that these networks can be manufactured lithographically and explored using independent agents. Using such a device, we demonstrated the solution of one particular three-variable instance of the SSP.

Notably, once the device is loaded with the required number of agents, the effective computational time for NP-complete problems grows only polynomially, e.g., as N^2 if the elements of S are approximately equidistantly spaced. This is in contrast to traditional, sequentially operating, electronic computers, where the time required to explore every possible

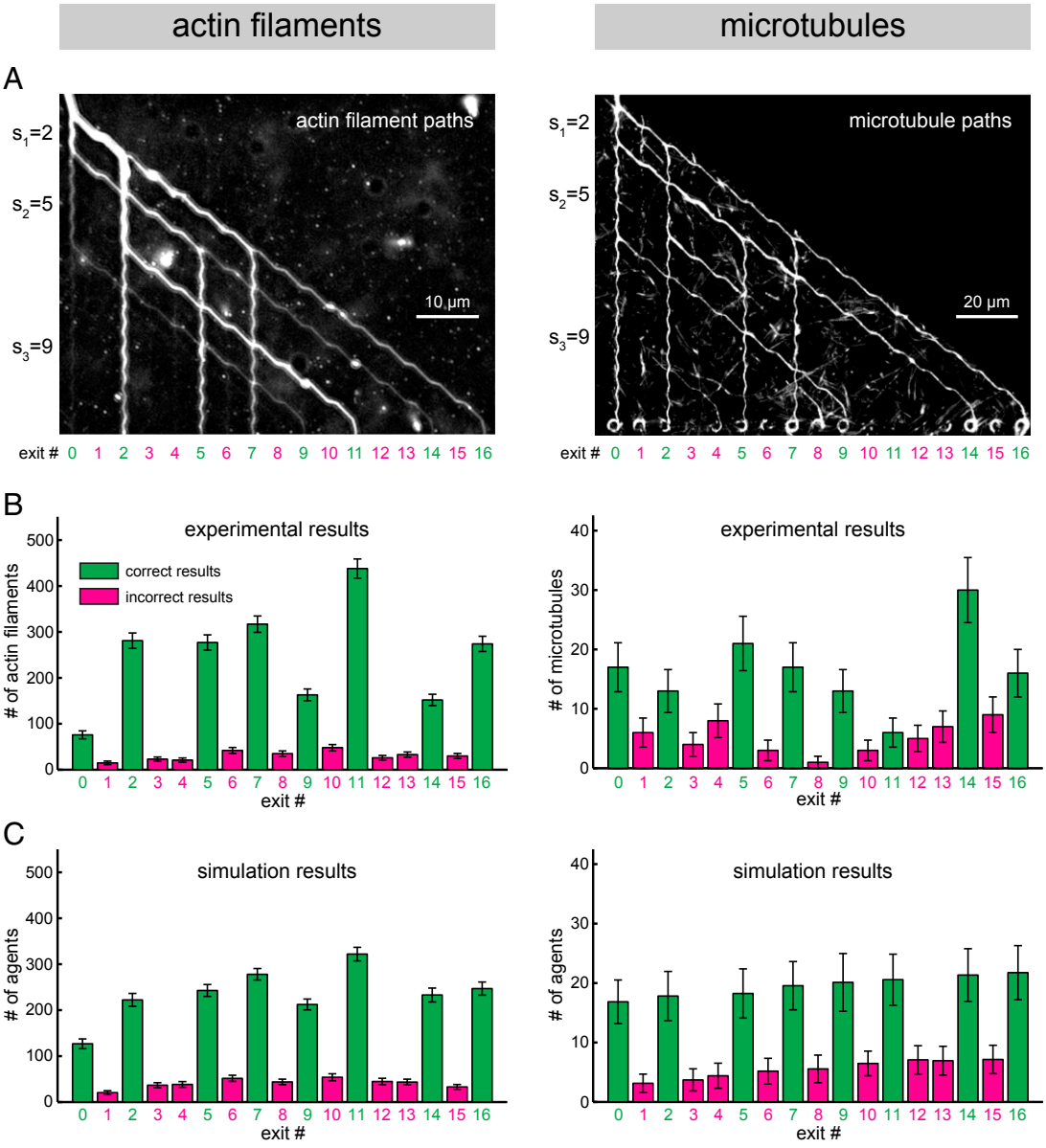


Fig. 4. Solving the SSP {2, 5, 9} by actin filaments (*Left*) and microtubules (*Right*). (*A*) Average (*Left*) and maximum (*Right*) projections of several hundred typical fluorescence micrographs of actin filaments (*Left*) and microtubules (*Right*) moving through a {2, 5, 9} device. An example of the computation (for a device using actin filaments) is presented in Movie S2. (*B*) Experimental results obtained from 2,251 actin filaments (*Left*; total experiment time: 26 min) and 179 microtubules (*Right*; total experiment time: 180 min). Error bars represent the counting error ($\sqrt{n}$). Total experiment time refers to the time required for the given number of agents to enter and traverse the network (see also *SI Appendix*, section S1). We note that the overall performance of the microtubule device was to some extent inferior to the actin device due to an accidental obstruction in a channel leading to exit 11 (causing a lower number of filaments reaching this exit), and due to a number of filaments landing at random points of the network in the channels where they were transported with high probability by the processive kinesin motors (increasing the number of filaments reaching the wrong exits). Both issues will be remedied in a next generation of devices by avoidance of fabrication errors, working in a cleanroom environment, and microfluidic focusing of the filaments in solution to the landing zones, respectively (see also *SI Appendix*, section S6 for more details on these sources of error). (*C*) Monte Carlo simulation results (mean ± SD of 100 simulations; see *SI Appendix*, section S6 for simulation details) for actin filaments (*Left*) and microtubules (*Right*) using the actually measured error rates of the pass junctions and measured splitting ratios of the split junctions (Fig. 3*C*). In *A*–*C*, green numbers and bars represent correct results, and magenta numbers and bars represent incorrect results.

solution sequentially would scale exponentially as 2^N. However, it is inherent to combinatorial and NP-complete problems (assuming P! = NP) that the exploration of the entire solution space requires the use of exponentially increasing amounts of some resource, such as time, space, or material. In the present case this fundamental requirement manifests itself in the number of agents needed, which grows exponentially with 2^N. Effectively we are trading the need of time for the need of molecular mass.

The error rates of this first device are too large for scaling up to problems containing more than ∼10 variables (see *SI Appendix*, section S1 for a more detailed scaling analysis).

Nevertheless, we argue that our approach has the potential to be more scalable in practice than other approaches because it offers several advantages: (*i*) Myosin II and kinesin-1 molecular motors use a distributed energy supply (ATP in the surrounding solution), thus eliminating the need for external forces (such as pressure or an electric potential) to drive the computation. This

need inherently prevents, for example, microfluidic approaches from scaling up, because in these devices the pressures needed to pump fluid through the network become prohibitively large for large N (30). (*ii*) The molecular motors operate in a highly energy-efficient manner. As a result, the approach demonstrated here consumes orders of magnitude less energy per operation compared with both electronic and microfluidic computers, eliminating issues related to the dissipation of heat. Specifically, we estimate an energy cost of 2–5 $\times$ 10^{-14} J per operation for a molecular-motor-based device compared with about 3–6 $\times$ 10^{-10} J per operation for the most advanced electronic computers, or an estimated minimum of 10^{-12} J per operation for microfluidics-based computers (*SI Appendix*, section S7). (*iii*) The networks in which the problems are encoded in our approach are planar and comprise standardized modules, therefore being fully scalable with existing technology (see *SI Appendix*, section S1 for a more detailed discussion). Thus, they avoid potential engineering challenges associated with building large-scale 3D microfluidics devices.

Most importantly, however, we foresee several practical solutions to managing the need for exponentially increasing numbers of agents that we highlighted above. (*i*) The number of agents can self-adjust to the problem size. Specifically, cytoskeletal filaments can self-replicate as they traverse the network by enzymatic splitting and simultaneous elongation (31, 32). Alternatively, self-propelled, dividing microorganisms can be used as agents (33–35). Thus, the larger the network, the more the agents will multiply, in an exponential fashion. Any kind of agent multiplication scheme will also solve potential problems with sequential feeding of the agents into the network through a single entrance, which represents a bottleneck for large N. Moreover, for cytoskeletal filaments, the splitting and elongation rates will be limited by the global concentrations of enzymes and filament subunits, respectively. Thus, multiplication will be negatively regulated in parts of the network where the agent density is high (i.e., above the density needed for successful computation), consequently counteracting the risk of channel clogging. (*ii*) The NP-complete problem is encoded into a planar, physical network and not into the agents themselves. This simplifies fabrication and encoding because the network that encodes the problem grows polynomially, whereas the exponentially scaling number of cytoskeletal filaments can be fabricated in bulk. This is in contrast to DNA computing, where large amounts of DNA need to be specifically synthesized for each problem. Furthermore, because of their generic nature, agents can continue to explore the network as long as ATP is available, which means they can be by recirculated and used more than once (*SI Appendix*, Fig. S2.1). (*iii*) Because our device implements basic addition operations, it can benefit from existing optimized algorithms and can be easily combined with a conventional computer to form a hybrid device. For example, an electronic computer could be used to, first, solve a subset of the largest numbers in the SSP. Then, the solutions calculated by the computer would be passed on to a set of biological computers described here, drastically reducing the number of agents needed because the biological computer solves only that part of the problem that overwhelms the electronic computer. Furthermore, the time needed to feed the filaments into the network would be further reduced because many entrances can be used in parallel. An extended analysis of scaling and energy considerations is presented in *SI Appendix*, sections S1 and S7, respectively.

Summarizing, the technical advances that would be necessary for a future device able to challenge an electronic computer are (*i*) scaling up of the physical network size from currently ~100 $\times$ 100 μm^2 to wafer size, which is achievable by current patterning technology. (*ii*) Reduction of the filament feeding time, which can be achieved by using networks with multiple entrances, or by self-replicating filaments, see above. (*iii*) Reduction of pass-junction error rates. We expect that this can be realized by simulation-driven design (such as described in *SI Appendix*, section S4), by evolutionary algorithms for designing the junction geometries (36, 37), or by using 3D geometries such as bridges or tunnels (38) which would offer zero error rates at pass junctions. (*iv*) To circumvent the inherent difficulties of tracking large numbers of individual filaments, automatic readout schemes at exits of interest can be used (39). (*v*) Programmable devices which can flexibly encode different problems could be achieved by using heat-controlled (40) or electrostatic (41) gates in only one programmable type of junction instead of the two (isomorphic) static junctions. (*vi*) Finally, filaments can be prevented from attaching to or detaching from the network by using closed channels with porous openings for allowing the supply of ATP (42).

The potential practical relevance of our approach goes beyond solving SSP, because all NP-complete problems can be converted to one another using a polynomial-time conversion (7, 43) and due to the general nature of our SSP computation network (namely a computer that can perform addition, and, by using right-to-left diagonals, also subtraction). Therefore, our approach has the potential to be general and to be developed further to enable the efficient encoding and solving of a wide range of large-scale problems. Accomplishing this would move forward (but not remove) the limit of the size of combinatorial problems that can be solved.

Materials and Methods Summary

Please see the *SI Appendix* for a detailed description of the materials and methods.

Fabrication of Computational Networks for Use with the Actin–Myosin System. Electron-beam lithography (EBL) was used for pattern formation in a poly(methyl methacrylate) (PMMA) resist on a SiO_2-coated Si substrate. After development and O_2-plasma-ashing [to ensure that the PMMA was hydrophilic and therefore unable to support motility (27)], the sample was silanized with trimethylchlorosilane to promote motility on the floor of the exposed SiO_2 substrate (44). Wetting of the surface was performed to reduce the possibility of air bubbles forming in the channels (45).

Fabrication of Computational Networks for Use with the Microtubule–Kinesin System. A silicon wafer was sputter-deposited with Au, sandwiched between two Ti adhesion layers. Next, a quartz layer was deposited, followed by a TiW layer and a ZEP520 positive-tone electron-beam resist layer. After exposure in an EBL system and development, the TiW, the quartz, and the upper Ti layers were etched by reactive ion etching down to the Au layer. Finally, the resist residue and the TiW were removed.

Actin–Myosin in Vitro Motility Assays. The in vitro motility assays were performed at 26–29 °C, as described previously (46). Briefly, the flow cell was preincubated with (*i*) heavy meromyosin (47) (120 $\mu g\ mL^{-1}$) for 4 min; (*ii*) 1 mg mL^{-1} bovine serum albumin for 1 min; (*iii*) rhodamine-phalloidin–labeled actin (48) filaments (10-nM monomeric concentration) for 1 min. The flow cell was washed both before and after actin filament incubation. Next, the flow cell was incubated with rigor solution (without ATP) for initial observations. Motility was initiated by introducing a MgAdenosine-5′-triphosphate (MgATP)-containing assay solution.

Microtubule–Kinesin in Vitro Motility Assays. Microtubule–kinesin gliding assays were performed using full-length kinesin-1 (kinesin) from *Drosophila* (49) and rhodamine-labeled tubulin (50) by following a procedure (51) that was upgraded for motility in nanochannels (52). The SiO_2 surface of the computational chip was passivated with 2-[Methoxy(poly-ethyleneoxy) propyl] trimethoxysilane] to prevent protein binding anywhere except on the gold bottom of the channels. Flow cells were perfused with (*i*) casein-containing solution (0.5 mg ml^{-1}, 5 min); (*ii*) kinesin solution (2 nM, 5 min); and (*iii*) motility solution containing 1 mM ATP and rhodamine-labeled, taxol-stabilized.

Imaging Methods. Rhodamine-labeled cytoskeletal filaments were observed using inverted fluorescence microscopes. Images were recorded with electron-multiplying charge-coupled device cameras and analyzed with ImageJ (imagej.nih.gov/ij/). Microtubule paths were tracked with software developed in-house (53).

ACKNOWLEDGMENTS. Dan V. Nicolau Jr. thanks Grant Shoffner and Yue Shark Yu from University of California, Berkeley, for help in running stochastic simulations. Financially supported by the European Union Seventh Framework

Programme (FP7/2007-2011) under Grant Agreements 228971 [Molecular Nano Devices (MONAD)] and 613044 [Parallel computing based on designed networks explored by self-propelled, biological agents (ABACUS)]; Defense Advanced Research Projects Agency under Grant Agreement N66001-03-1-8913; by NanoLund; by the Miller Foundation; by the Swedish Research Council (Projects 621-2010-5146 and 2010-4527); The Carl Trygger Foundation, German Research Foundation within the Cluster of Excellence Center for Advancing Electronics Dresden and the Heisenberg Program; and by Linnaeus University.

1. Nam GJ, Sakallah KA, Rutenbar RA (2002) A new FPGA detailed routing approach via search-based Boolean satisfiability. *Ieee T Comput Aid D* 21(6):674–684.
2. Fraenkel AS (1993) Complexity of protein folding. *Bull Math Biol* 55(6):1199–1210.
3. Pierce NA, Winfree E (2002) Protein design is NP-hard. *Protein Eng* 15(10):779–782.
4. Hopfield JJ, Tank DW (1985) "Neural" computation of decisions in optimization problems. *Biol Cybern* 52(3):141–152.
5. Nicolau DV, et al. (2006) Molecular motors-based micro- and nano-biocomputation devices. *Microelectron Eng* 83(4-9 Spec. Iss.):1582-1588.
6. Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266(5187):1021–1024.
7. Lipton RJ (1995) DNA solution of hard computational problems. *Science* 268(5210): 542–545.
8. Mao C, LaBean TH, Relf JH, Seeman NC (2000) Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* 407(6803):493–496.
9. Qian L, Winfree E (2011) Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332(6034):1196–1201.
10. Beaver D (1995) Computing with DNA. *J Comput Biol* 2(1):1–7.
11. Braich RS, Chelyapov N, Johnson C, Rothemund PWK, Adleman L (2002) Solution of a 20-variable 3-SAT problem on a DNA computer. *Science* 296(5567):499–502.
12. Ouyang Q, Kaplan PD, Liu S, Libchaber A (1997) DNA solution of the maximal clique problem. *Science* 278(5337):446–449.
13. Reif J, LaBean T, Sahu S, Yan H, Yin P (2005) Design, simulation, and experimental demonstration of self-assembled DNA nanostructures and motors. *Unconventional Programming Paradigms*, eds Banâtre J-P, Fradet P, Giavitto J-L, Michel O, Lecture Notes in Computer Science (Springer, Berlin), Vol 3566, pp 173–187.
14. Ladd TD, et al. (2010) Quantum computers. *Nature* 464(7285):45–53.
15. Chiu DT, Pezzoli E, Wu H, Stroock AD, Whitesides GM (2001) Using three-dimensional microfluidic networks for solving computationally hard problems. *Proc Natl Acad Sci USA* 98(6):2961–2966.
16. Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York), p 338.
17. Caprara A, Kellerer H, Pferschy U (2001) The multiple subset sum problem. *SIAM J Optim* 11(2):308–319.
18. Kellerer H, Mansini R, Pferschy U, Speranza MG (2003) An efficient fully polynomial approximation scheme for the Subset-Sum Problem. *J Comput Syst Sci* 66(2):349–370.
19. Schnorr CP, Euchner M (1994) Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math Program* 66(2):181–199.
20. Kate A, Goldberg I (2011) Generalizing cryptosystems based on the subset sum problem. *Int J Inf Secur* 10(3):189–199.
21. Howard J, Hudspeth AJ, Vale RD (1989) Movement of microtubules by single kinesin molecules. *Nature* 342(6246):154–158.
22. Kron SJ, Spudich JA (1986) Fluorescent actin filaments move on myosin fixed to a glass surface. *Proc Natl Acad Sci USA* 83(17):6272–6276.
23. Hiratsuka Y, Tada T, Oiwa K, Kanayama T, Uyeda TQ (2001) Controlling the direction of kinesin-driven microtubule movements along microlithographic tracks. *Biophys J* 81(3):1555–1561.
24. Vikhorev PG, et al. (2008) Diffusion dynamics of motor-driven transport: Gradient production and self-organization of surfaces. *Langmuir* 24(23):13509–13517.
25. Hess H, Clemmens J, Qin D, Howard J, Vogel V (2001) Light-controlled molecular shuttles made from motor proteins carrying cargo on engineered surfaces. *Nano Lett* 1(5):235–239.
26. Nicolau DV, Suzuki H, Mashiko S, Taguchi T, Yoshikawa S (1999) Actin motion on microlithographically functionalized myosin surfaces and tracks. *Biophys J* 77(2): 1126–1134.
27. Sundberg M, et al. (2006) Actin filament guidance on a chip: Toward high-throughput assays and lab-on-a-chip applications. *Langmuir* 22(17):7286–7295.
28. Clemmens J, Hess H, Howard J, Vogel V (2003) Analysis of microtubule guidance in open microfabricated channels coated with the motor protein kinesin. *Langmuir* 19(5):1738–1744.
29. Reuther C, Hajdo L, Tucker R, Kasprzak AA, Diez S (2006) Biotemplated nanopatterning of planar surfaces with molecular motors. *Nano Lett* 6(10):2177–2183.
30. Fuerstman MJ, et al. (2003) Solving mazes using microfluidic networks. *Langmuir* 19(11):4714–4722.
31. Orlova A, Prochniewicz E, Egelman EH (1995) Structural dynamics of F-actin: II. Cooperativity in structural transitions. *J Mol Biol* 245(5):598–607.
32. Sun HQ, Yamamoto M, Mejillano M, Yin HL (1999) Gelsolin, a multifunctional actin regulatory protein. *J Biol Chem* 274(47):33179–33182.
33. Cho H, et al. (2007) Self-organization in high-density bacterial colonies: Efficient crowd control. *PLoS Biol* 5(11):e302.
34. Hanson KL, et al. (2006) Fungi use efficient algorithms for the exploration of microfluidic networks. *Small* 2(10):1212–1220.
35. Tero A, et al. (2010) Rules for biologically inspired adaptive network design. *Science* 327(5964):439–442.
36. Rupp B, Nédélec F (2012) Patterns of molecular motors that guide and sort filaments. *Lab Chip* 12(22):4903–4910.
37. Sunagawa T, Tanahashi A, Downs ME, Hess H, Nitta T (2013) In silico evolution of guiding track designs for molecular shuttles powered by kinesin motors. *Lab Chip* 13(14):2827–2833.
38. Lard M, ten Siethoff L, Generosi J, Månsson A, Linke H (2014) Molecular motor transport through hollow nanowires. *Nano Lett* 14(6):3041–3046.
39. Lard M, ten Siethoff L, Månsson A, Linke H (2013) Tracking actomyosin at fluorescence check points. *Sci Rep* 3:1092.
40. Schroeder V, Korten T, Linke H, Diez S, Maximov I (2013) Dynamic guiding of motor-driven microtubules on electrically heated, smart polymer tracks. *Nano Lett* 13(7): 3434–3438.
41. van den Heuvel MG, de Graaff MP, Dekker C (2006) Molecular sorting by electrical steering of microtubules in kinesin-coated channels. *Science* 312(5775):910–914.
42. Graczyk M, Balaz M, Kvennefors A, Linke H, Maximov I (2012) Optimization of a self-closing effect to produce nanochannels with top slits in fused silica. *J Vac Sci Technol, B* 30(6):06FF09-1–06FF09-4.
43. Karp RM (1972) Reducibility among combinatorial problems. *Complexity of Computer Computations*, eds Miller RE, Thatcher JW (Plenum, New York), pp 85–103.
44. Sundberg M, et al. (2003) Silanized surfaces for in vitro studies of actomyosin function and nanotechnology applications. *Anal Biochem* 323(1):127–138.
45. Bunk R, et al. (2005) Guiding motor-propelled molecules with nanoscale precision through silanized bi-channel structures. *Nanotechnology* 16(6):710–717.
46. Sundberg M, et al. (2006) Selective spatial localization of actomyosin motor function by chemical surface patterning. *Langmuir* 22(17):7302–7312.
47. Kron SJ, Toyoshima YY, Uyeda TQ, Spudich JA (1991) Assays for actin sliding movement over myosin-coated surfaces. *Methods Enzymol* 196:399–416.
48. Pardee JD, Spudich JA (1982) Purification of muscle actin. *Methods Cell Biol* 24: 271–289.
49. Coy DL, Wagenbach M, Howard J (1999) Kinesin takes one 8-nm step for each ATP that it hydrolyzes. *J Biol Chem* 274(6):3667–3671.
50. Hyman A, et al. (1991) Preparation of modified tubulins. *Methods in Enzymology*, ed Vallee RB (Academic, Cambridge, MA), Vol 196, pp 478–485.
51. Nitzsche B, et al. (2010) Studying kinesin motors by optical 3D-nanometry in gliding motility assays. *Methods Cell Biol* 95:247–271.
52. van den Heuvel MGL, Butcher CT, Smeets RMM, Diez S, Dekker C (2005) High rectifying efficiencies of microtubule motility on kinesin-coated gold nanostructures. *Nano Lett* 5(6):1117–1122.
53. Ruhnow F, Zwicker D, Diez S (2011) Tracking single particles and elongated filaments with nanometer precision. *Biophys J* 100(11):2820–2828.

SUPPLEMENTARY INFORMATION

Parallel computation with molecular motor-propelled agents in nanofabricated networks

Dan V. Nicolau Jr., Mercy Lard, Till Korten, Falco van Delft, Malin Persson, Elina Bengtsson, Alf Månsson, Stefan Diez, Heiner Linke, Dan V. Nicolau

DETAILED MATERIALS AND METHODS

Device fabrication

Computational networks for use with the actin–myosin system. A Si substrate was subjected to wet thermal oxidation to form a SiO_2 layer of 104 nm and spin-coated with poly(methyl methacrylate) (PMMA 950A5; Microchem Corp., Newton, MA, USA) at 6000 rpm for 60 s, followed by baking at 160°C for 15 min. Electron-beam lithography (EBL; Raith 150, Dortmund, Germany) was used for pattern formation. The PMMA was developed with methyl isobutyl ketone and isopropanol (MIBK:IPA; Merck KGaA, Darmstadt, Germany) at a ratio of 1:3 for 1 min, followed by IPA rinsing for 30 s. The sample was O_2-plasma-ashed (Plasma Preen II-862, Plasmatic Systems, Inc, North Brunswick, NJ) at 5 mbar for 30 s to remove resist residues from the substrate and to ensure that the PMMA was hydrophilic and therefore unable to support motility (1). Silanization, using chemical vapor-phase deposition (CVD) of 98 % (gas chromatography, GC) trimethylchlorosilane (TMCS; Sigma-Aldrich Sweden AB, Stockholm, Sweden), was performed to promote motility on the floor of the exposed SiO2 substrate (2). Wetting of the surface was performed to reduce the possibility of air bubbles forming in the channels (3).

Computational networks for use with the microtubule–kinesin system. The network was formed by etching of a quartz layer, with an Au "floor" at the channel bottom only. A 6 in silicon wafer was sputter-deposited with 100 nm-thick Au, sandwiched between two 10 nm-thick Ti adhesion layers. Next, a 500 nm-thick CVD quartz layer was deposited, followed by a 100 nm-thick TiW layer and a 200 nm-thick ZEP520 positive-tone electron-beam resist layer. After exposure in a JEOL9300FS EBL system (100 kV, 4 nA beam current, 25 nm spot size, and 20 nm beam step size), the resist was developed in MIBK and rinsed in IPA. Next, the TiW, the quartz, and the upper Ti layers were etched by reactive ion etching in a low-pressure fluorine containing plasma down to the Au layer. Finally, the resist residue and the TiW were removed by oxygen barrel etching and reactive ion etching, respectively.

In vitro motility assays

Actin–myosin system. Myosin II was isolated from rabbit fast leg muscle (4) followed by α-chymotrypsin digestion to produce heavy meromyosin (HMM) (5). Actin was obtained from rabbit back muscle (6). Flow cells were constructed from a coverslip with a nanostructured chip on top, with double-sided sticky tape used as spacers (1). The in vitro motility assays were performed at 26–29 °C, as described previously (7). All solutions were based on buffer A (1 mM $MgCl_2$, 10 mM 3-(N-morpholino)propanesulfonic acid (MOPS), 0.1 mM K_2-ethylene glycol tetra-acetic acid (EGTA), pH 7.4), and all proteins were diluted in buffer B (buffer A

with 1 mM dithiothreitol (DTT) and 50 mM KCl). Briefly, the flow cell was pre-incubated with (i) HMM (120 μg mL^{-1}) for 4 min; (ii) 1 mg mL^{-1} bovine serum albumin for 1 min; (iii) rhodamine-phalloidin–labeled actin filaments (10 nM monomeric concentration) for 1 min. The flow cell was washed both before and after actin filament incubation with buffer B. Next, the flow cell was incubated with rigor solution (r60) for initial observations. This solution was composed of buffer A with 10 mM DTT, and 45 mM KCl (giving an ionic strength of 60 mM), and an anti-bleach mixture of 3 mg mL^{-1} glucose, 460 U mL^{-1} glucose oxidase, and 870 U mL^{-1} catalase. Motility was initiated by introducing a MgAdenosine-5′-triphosphate (MgATP)-containing assay solution (r60 with 1 mM MgATP and an ATP regenerating system: 2.5 mM creatine phosphate and 3.5 U mL^{-1} creatine phosphokinase).

Microtubule–kinesin system. Full-length kinesin-1 (kinesin) from *Drosophila* was expressed in bacteria and purified as described previously (8). Tubulin was isolated from porcine brain and subsequently labeled with rhodamine as described previously (9). Microtubules were polymerized from 5 μl rhodamine-labeled tubulin in BRB80 buffer (80 mM PIPES/KOH, pH 6.8, 1 mM EGTA, 1 mM $MgCl_2$; unless stated otherwise, all chemicals used for microtubule-kinesin *in vitro* motility assays were purchased from Sigma) with 4 mM $MgCl_2$, 1 mM Mg-GTP, and 5% DMSO at 37°C for 60 min. Afterward, microtubules were stabilized and diluted 100-fold in BRB80 containing 10 μM Taxol at room temperature. Microtubule-kinesin gliding assays were performed by following a procedure (10) that was upgraded for motility in nanochannels (11). The SiO_2 surface of the computational chip was incubated with 2-[Methoxy(poly-ethyleneoxy) propyl] trimethoxysilane] 90% (ABCR, SIM4492.7; 0.23% v/v in toluene·HCl) overnight at room temperature to prevent protein binding anywhere except on the gold bottom of the channels. Flow cells were constructed by placing stretched stripes of Parafilm on the chips next to the structures. The channels were closed with a glass coverslip (Menzel, 18×18 mm^2) silanized with PEG as described for the structures above. Flow cells were perfused with casein-containing solution (0.5 mg ml^{-1}) in BRB80 and left to adsorb for 5 min. Next, 50 μl of kinesin solution (2 nM full-length kinesin), was perfused into the flow cells and incubated for another 5 min. Thereafter, a motility solution (1 mM ATP, 20 mM D-glucose, 20 μg ml^{-1} glucose oxidase, 10 μg ml^{-1} catalase, 10 mM DTT, 10 μM taxol in BRB80) containing rhodamine-labeled, taxol-stabilized microtubules was applied.

Imaging methods

Actin–myosin system. Rhodamine-phalloidin–labeled filaments were observed with an inverted fluorescence microscope (Nikon, Eclipse TE300) equipped with a 100× oil immersion objective (Nikon, NA 1.4) and a Tetra methyl Rhodamine Iso-Thiocyanate filter set (Ex 540/25, DM 565, BA 605/25). Images were recorded with an electron-multiplying charge-coupled device camera (Hamamatsu C9100) and analyzed with Image J (Rasband, W.S ImageJ, U.S. National Institutes of Health, Bethesda, Maryland, USA, http://imagej.nih.gov/ij/, 1997–2012).

Microtubule–kinesin system. Fluorescence time-lapse movies were recorded with an Axiovert 200M inverted optical microscope (Zeiss) using a Tetra methyl Rhodamine Iso-Thiocyanate filter set (Chroma Technology; Ex 535/50, DM 565, BA 610/75). Images were acquired with a back-illuminated charge-coupled device camera (MicroMax 512 BFT, Roper Scientific) in conjunction with Metamorph imaging software (Universal Imaging Corp.). Microtubule paths were tracked with software developed in-house (12).

SUPPLEMENTARY TEXT S1 – S7

S1. Scaling considerations

Computing time. The lateral size of the network designed to solve the Subset Sum Problem (SSP) shown in Fig. 1 (main text) grows at a rate that depends upon the structure of the gaps between successive integers in the set of numbers used. For normal cases, in which the gaps grow less than exponentially, e.g. if the set comprises consecutive prime numbers, the horizontal network length grows polynomially. In the specific case of successive primes, it grows as $\sim N^2$, where N is the number of primes in the set. However, for all cases, the size of the potential solution set grows as 2^N, which is the total number of subsets of N. Therefore, our design implemented in a computation device potentially enables the exploration of an exponentially large solution space in polynomial *computing* time, subject to the availability of a sufficiently large number of agents and low error rates in the junctions, in particular the pass junctions.

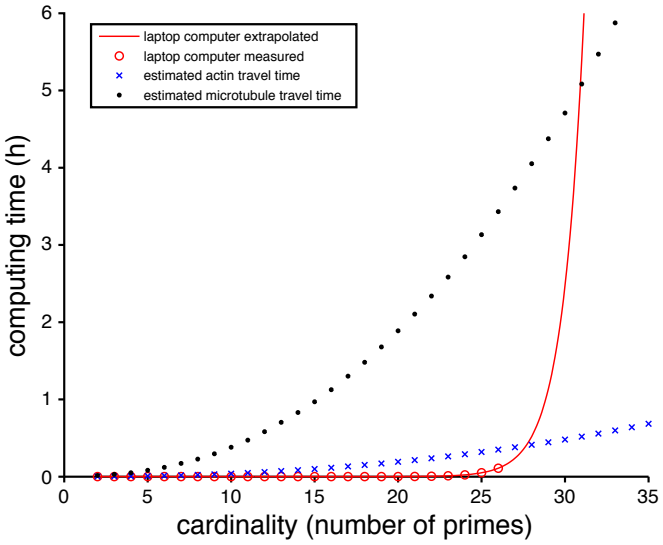


Fig. S1.1. Scaling of computing time. *Calculated and simulated computing times for sets containing the given number of prime numbers (e.g., {2, 5, 7} has a cardinality of 3). The times that actin filaments and microtubules take to travel the longest path through the networks were estimated from their speed and the dimensions of a unit cell of the network. The time that a laptop (MacBook Pro, 2.6 GHz core i5 CPU) would take to solve the SSP by brute force was measured up to the first 26 primes and then extrapolated with an exponential function. The measurement of the computing times for the SSP beyond the first 26 primes was not possible because the limitations of both the CPU and the memory of the computer resulted in computing times that increased more than exponentially.*

Fig. S1.1 presents a comparison of the estimated computing time required by a home computer and by biological agents exploring our proposed computing device to solve sets of various numbers of primes. The time taken to complete the computation is a function of the longest traversal time of a motile agent, e.g., actin filament, determined by the length of the longest diagonal arc in the network. In the case explored above, with the set comprising consecutive primes, this longest arc length grows as $\sim N^2$. For small sets, the agents-based computation is slower than the home computer, but as their computation time scales polynomially rather than exponentially, biological agents are able, in principle, to solve the network faster than a computer when the set contains more than $N = 27$ and $N = 30$ numbers for actin- or microtubule-based devices, respectively. **In the following, we will therefore use $N = 30$ for quantitative scaling estimates.**

Device size and computation time. To estimate the sizes of required devices, we again assume a set consisting of the first prime numbers. For a device that computes a problem with $N = 30$ (average numerical value of the first 30 primes: 53), and unit cells of 5 µm (for actin) and 8 µm (for microtubules) across, the corresponding networks would be ~ 5x5 mm^2 and 8x8 mm^2 in size, respectively. The time to solve such a network (determined by the length of the longest diagonal arc in the network) will be approximately 0.5 hours for actin filaments (speeds of ~5 µm s^{-1}) and 4.5 hours for microtubules (speeds of 0.5 µm/s).

Scaling of required agent mass. The time required for our device to solve SSPs scales with $\sim N^2$. However (assuming P!=NP), the fundamental issue that solving NP complete problems requires exponentially increasing resources remains. In our case this issue is represented by the amount of cytoskeletal filaments required, which – in principle – scales with 2^N. Importantly, however, the total numbers of filaments required are low enough to allow for the solution of problems of interesting size. Specifically, 2^{30} filaments are required to explore every possible path through a device with $N = 30$. Assuming further that the filaments are, on average, 2 µm long, we can use the known filament structure (13, 14) to calculate that each filament contains 364 actin subunits or 1625 tubulin dimers, respectively. Taking into account the molar mass of actin (42kDa) and a tubulin dimer (110 kDa), the 2^{30} actin filaments or microtubules required to be faster than a laptop computer would have a mass of ~60 ng or ~600 ng, respectively. For comparison, a routine tubulin purification that starts with ~10 pig brains yields ~1g of tubulin.

Pre- and post-computation operational time. Similar to the booting of an electronic computer, the time required for operations prior to the actual computing process adds to the overall time required to obtain a solution. This pre-computing time consists of the *loading time*, i.e., the time the filaments take to land on and traverse the loading zone (upper left in Fig. 2 and Fig. S2.1); and the *feeding time*, i.e., the time required by all agents needed for the computation to enter the computational network from the loading zone. Also, to actually get the solution to the problem, the result of the computation needs to be read out, a process that requires a post-computation *read-out time*.

Loading time. For the microtubule-based device the loading time was measured in a separate experiment. This experiment used a lower concentration of microtubules, started the imaging after 30 s and observed several loading zones. Regarding actin filaments-based devices, filling of the loading zones is achieved in approximately 1 min during incubation, see further in Materials and Methods in the main text. In recently shown detailed experimental and theoretical studies (15) loading zones with a nearly triangular shape, capped by a hemisphere, allowed for a rate of emptying with time constants on the order of 1 min after optimization. Inputting these rates, and taking into consideration that the loading areas are orders of magnitude larger than the active areas of the computational network comprising narrow channels, it can be conservatively estimated that the loading time does not contribute importantly to the overall time required to obtain a solution.

Feeding time. Because the number of agents needed for computation increases exponentially with N, also the feeding time will grow exponentially (see *Scaling of required agent mass* above). Fundamentally, the non-polynomial increase of the feeding time with the increase of the size of the problem can be addressed by the multiplication of the agents while they explore the network (see discussion in main text).

Read-out time. The read-out time varies largely with the nature of the problem to be solved. For instance, the *existence* of a solution to the SSP requires only the inspection of the exits, but the actual composition of the

solution would require the more time consuming back-tracking of the agents that exited at the correct exits. However, in the context of solving combinatorial problems, the tracking of a small number of agents, i.e., only those that solved the problem, is a much faster process, e.g., tens of minutes, than the overall computation process, e.g., hours. Importantly, automatic tracking procedures, such as those described recently (12) will further accelerate this process.

Motors-related operational longevity. Presently, the motility assays are operational for several hours. However, recent work (16) showed that a small chemical species protects molecular motors (myosin) against denaturation and provides up to 10 times amplification of its enzymatic activity, thus offering a possibility for a greater expansion of the working time of the molecular motors-based computation.

Scaling of fabrication. The present proof-of-principle device has been fabricated using e-beam lithography, which is the generally accepted patterning technique for prototyping, before embarking on mass production. In the first instance, the slow speed of e-beam lithography is not problematic, as each computation requires only a small batch of chips. Scaling up of our networks for larger problem sizes requires only the addition of more rows of identical split and pass junctions, which is easily achievable and facilitated by the "unit cell"-based design proposed here that makes the translation to step and repeat patterning seamless. Should the mass production of chips be envisaged in the future, the size of the minimum features proposed here, i.e., approx. 200 nm, could be easily fabricated by deep-UV lithography, or by imprint technology (NIL, SCIL) which could be cheaper for smaller batches of computing chips. Using existing, parallel lithography techniques such as optical lithography or nano-imprinting lithography (NIL), large areas (up to 4-inch wafers with NIL and even larger with optical lithography) of a network can be made in one step. Step and repeat patterning can be used to parallelize the computation by the replication of the computing networks several times on the same chip, which is readily achievable with the present fabrication technology. Finally, the suggested improvements of the proposed device, e.g., programmable gates, would lead to universal, programmable devices, which then would enable modest-scale mass production.

Energy requirements. For a detailed analysis of energy needs and comparison to other technologies we refer to section S7. In brief, energy requirements are very favorable and do not pose a scaling limit.

Scaling and propagation of pass junction errors. Each time an agent takes a wrong turn at a pass-junction, it enters an unintended path and thus may ultimately emerge at an exit corresponding to an incorrect solution (exits labeled in magenta in Fig. 1, main text). Therefore, the fraction of useable agents (agents that will find correct solutions) decreases exponentially with the number of pass-junctions passed: $f(x) = (1 - E)^x$, where f is the fraction of useable agents, x is the number of pass-junctions passed, and E is the fractional error (for example, 0.003 for a 0.3 % error). The number of pass-junctions that must be passed is equal for each path through the network and can be calculated from the numbers s_i in the set as $x = J - N$, where J is the total sum of the set of numbers and N is the problem size ($J = \sum_{i=1}^{N} s_i$). All filaments that take at least one wrong turn together form a background of filaments emerging from all exits. From this background, the correct solutions must be distinguished. In simulations we found that $f \geq 0.15$ allows us to still clearly distinguish correct solutions from the background. Based on these considerations we can estimate the size of achievable networks. For the presently realized pass-junction errors (2.1% for actin and 0.3% for microtubules, see section S5 for details), and again considering the first prime numbers as our set, the largest problems that can be solved are $N = 9$ for actin and $N = 20$ for microtubules, respectively. However, for the microtubule-kinesin

device, the errors caused by filaments landing in the channels will likely restrict the device size to 5-7 numbers. In order to solve our benchmark network with N= 30 we would need to reduce the error rates to 0.1% (and –in case of the microtubule device– eliminate the errors caused by filaments landing in channels). This error rate is a factor of three better than what we have shown so far, which – based on simulations of improved junction designs – seems to be clearly achievable. Also, preventing filaments from landing in channels could be achieved by microfluidic focusing. More, importantly, however, we have already demonstrated "filament tunnels" which would avoid pass junction errors altogether (17). With such tunnels or bridges implemented into a network, pass junctions errors can be reduced to zero and will not pose a limitation to scaling.

Scaling analysis summary. Devices that can challenge a modern personal computer in brute-force solving the SSP (prime numbers, $N = 30$) would require networks of less than $1cm^2$, less than a microgram of filaments to solve one problem, and negligible amounts of energy. To realize such a device, the pass junction error would need to be reduced by about a factor of three below the level achieved in our very first designs. Given that we already have demonstrated filament transport through tunnels, allowing for pass junctions with zero error, this appears fully achievable.
We found that none of the following factors pose limits to scaling the device to interesting sizes of $N > 30$: the longevity of the bio-chemical system, the amount of filaments needed, loading time, read-out time, error rates at pass junctions, energy requirements, and fabrication considerations. The most serious limitation to scaling is the "boot" time it takes to feed the required number of filaments into the network, which, in the current design scales exponentially with N. Practical solutions to managing this requirement are discussed in the main text.

S2. Overall layout of the computation device used with actin filaments

Similar to the microtubule device shown in Fig. 2 of the main text, the actin-myosin device features several loading zones where filaments can land and are guided towards the entrance of the network (Fig. S2.1 top left). The size and shape of the loading zones have been optimized to provide a good compromise between a large area (to catch many filaments) and a small time (to guide the filaments to the loading-zone exits).

The network itself has the same layout as the one used for the microtubule device, except that the size of the unit cell was reduced to 4 μm across (see supplementary text S3 for junction design details). For the actin filaments device, exiting agents are not collected but rather recycled through the network via feedback loops at the exit row. To ensure that the filaments do not enter the network from below, "molecular rectifiers" (18, 19) were integrated into the feedback loops that direct the filaments from the computing network back to the loading zone (see Fig. S2.1 bottom right for a detailed description of a rectifier). The efficiency of the rectifiers is 96 % (15), thus the five rectifiers in a row preventing filaments from entering the network from below have a combined efficiency of > 99.99 %.

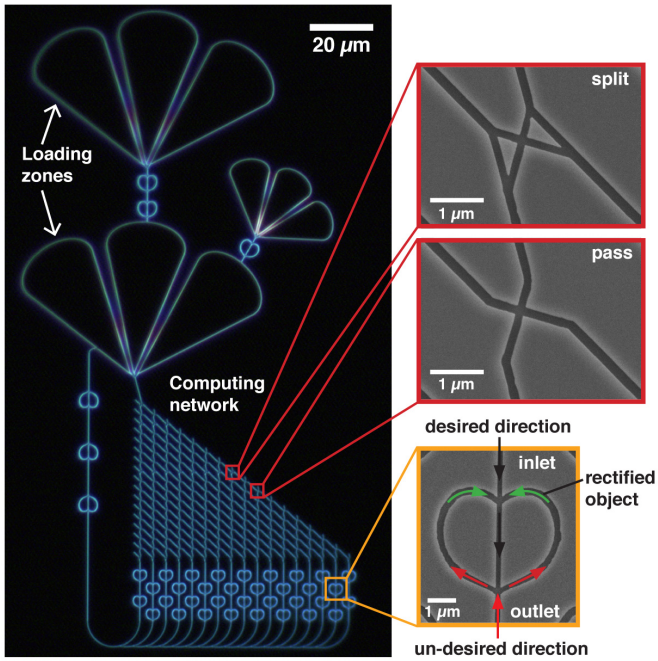


Fig. S2.1. *Design of the {2, 5, 9} subset sum device for testing with actin–myosin. A series of loading zones (large open areas, top left), functionalized with HMM (see Materials and Methods for details of the motility assay), were used to bind actin filaments to the surface and to guide them toward the device entrance along the loading-zone edges (3, 20). Insets: scanning electron micrographs of the split and pass junctions, and of the heart-shaped rectifiers used to maintain unidirectional actin filament motion.*

The rate of turning around (U-turns) for actin filaments and microtubules within the network was also evaluated. The measured rate of U-turns is 0.77 % for 2327 actin filaments, and 2.5 % for 712 microtubules entering the computation devices. This rate represents a very low percentage of total uni-directional guided filaments in the devices on the whole. More importantly, the design of the network layout makes it, to a large extent, error-tolerant. The filaments have to take at least two subsequent U-turns in order to reach an exit, in order for U-turns to adversely affect the calculation. Also, in order to reach an incorrect exit, the filaments have to take at least two turns at split junctions, or one turn at a pass junction in-between subsequent U-turns. Otherwise they will stay on a correct path in the network. Consequently, the estimated error is less than 0.1 % due to U-turns. Therefore, this effect is negligible compared to errors at pass junctions (see S5)

S3. Junction design details

Because of differences in the filament properties, the channel dimensions, and the fabrication processes employed, distinct junction designs have been used for microtubules and the actin filaments (see also supplementary text S4 for design optimization). The junction layouts are shown in Figs. S3.1 and S3.2 and Tabs. S3.1 and S3.2 for the actin-myosin and microtubule-kinesin systems, respectively.

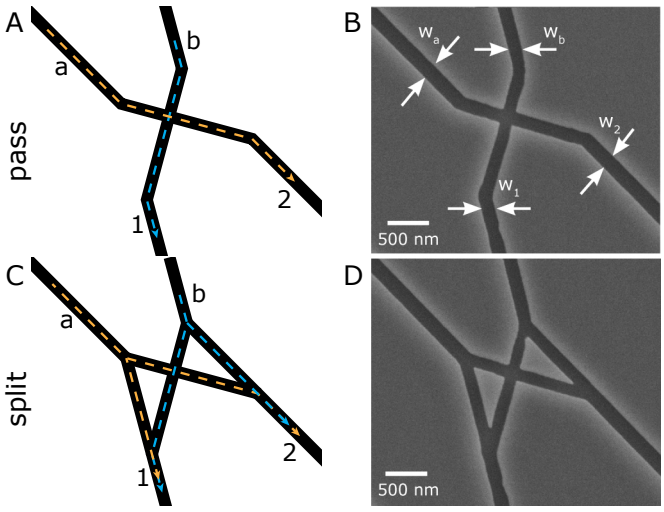


Fig. S3.1. *Schematics and scanning electron micrographs (SEM) of junctions used with the actin–myosin system. (A) Unit cell of a pass-junction with designed paths depicted by colored arrows. (B) Unit cell of a split-junction with colored arrows showing the designed paths. (C) SEM of a pass-junction. (D) SEM of a split-junction. SEM images were obtained after sputtering samples with 5 nm of Pt.*

Tab. S3.1: *Junction dimensions used with the actin–myosin system.*

Junction type	w_a (nm)	w_b (nm)	w_1 (nm)	w_2 (nm)
Split	182 ± 6	165 ± 6	171 ± 10	182 ± 5
Pass	175 ± 4	170 ± 7	176 ± 2	184 ± 4

Note: *Widths (± standard deviations) are measured from top view in scanning electron micrographs of junctions after completed fabrication, demarcated by arrows in Figure S3.1c.*

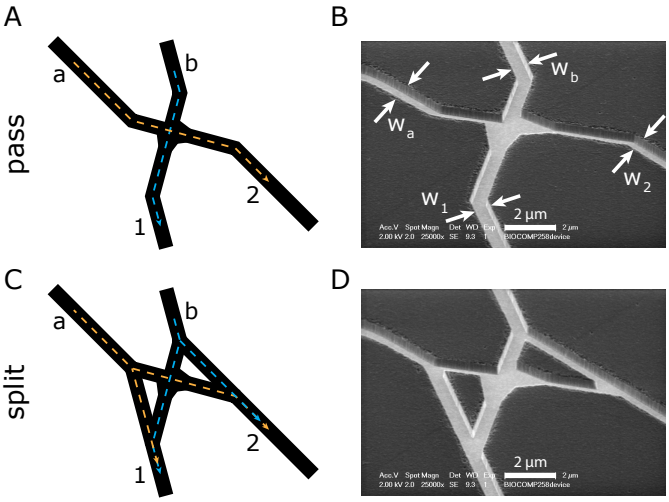


Fig. S3.2. *Schematics and scanning electron micrographs (SEM) of junctions used with the kinesin–microtubule system. (A, C) Schematic unit cells of pass- and split-junctions with designed paths depicted by colored arrows. (B, D) SEM of pass- and split-junctions, respectively, with funnels at the crossing that reduce turning of the filaments (see also supplementary text S4).*

Tab. S3.2: *Junction dimensions used with the microtubule–kinesin system.*

Junction type	w_a (nm)	w_b (nm)	w_1 (nm)	w_2 (nm)
Split	243 ± 13	238 ± 12	256 ± 11	231 ± 10
Pass	273 ± 19	256 ± 13	251 ± 15	250 ± 15

Note: *Widths (± standard deviations) are measured from top view in scanning electron micrographs of junctions after completed fabrication, demarcated by arrows in Figure S3.2b.*

S4. Simulation used to guide junction design

Because the experiments to be used for optimization are time consuming and because relatively extensive knowledge exists (1, 18, 21-23) regarding the motility of cytoskeletal filaments in micro- and nano-fabricated channels, simulation of the paths of motor-propelled actin filaments and microtubules was used to test and optimize the design of split- and pass-junctions. These designs have been then validated by experiments. The Monte Carlo simulations (24) quantitatively took into account filament flexibility, thermal motion, and the confining effect of the channel walls (assumed to be completely smooth), as described in previous work (20). The effect of the thermal fluctuations of filaments sliding at velocity v_f was taken into account by updating the sliding direction at defined short time intervals (Δt) with an angular change, which was obtained from a Gaussian distribution with zero mean value and standard deviation (SD):

$$SD = (v_f \Delta t / L_P)^{0.5} \quad (1)$$

Where L_p is the persistence length of the filament. Matlab (Mathworks, Natick, MA) was used as the random number generator for normal distributions. The simulation of the filament behavior at edges was performed as described in detail elsewhere (20).

The simulation focused on pass-junctions where error rates are critical to device performance (see supplementary text S5). The simulation results are briefly summarized as follows:
The model predicted only minor effects of channel width both for actin filaments and for microtubules (Fig. S4.1, B and C, respectively) when the input and output channels at a junction were of similar width.
An increased persistence length (for example, microtubules, $L_p \approx 100$ µm, as compared with actin, $L_p = 10$ µm) reduced the pass-junction error rate for a given channel width (compare Fig. S4.1, B and C).
Simulations were also used as a basis for optimization of the design details of the pass-junctions. Most importantly, the results in Fig. S4.1, B and C, indicate the use of wider output than input channels to reduce error rate. Surprisingly, the measured error rates for the actin-myosin system were well below the predicted error rates for the used geometry. An explanation for this effect could be that HMM forms a dense layer with a thickness >30 nm (25) on the channel wall (inset of Fig. S4.1B). This layer forms a barrier to actin filaments longitudinally during gliding along the input channel, but can be penetrated from the side, after the leading end of the filament crossed an intersecting channel (right part of filament in the inset of Fig. S4.1B). This

feature is inherent to the thick and dense HMM layer. Therefore, no design improvements along these lines were tested for the actin–myosin device. However, a modified design with funnel-shaped output channels (Fig. S3.2) appreciably reduced error rates for the microtubule-kinesin device, where negligible amounts of protein are believed to adhere to the channel walls.

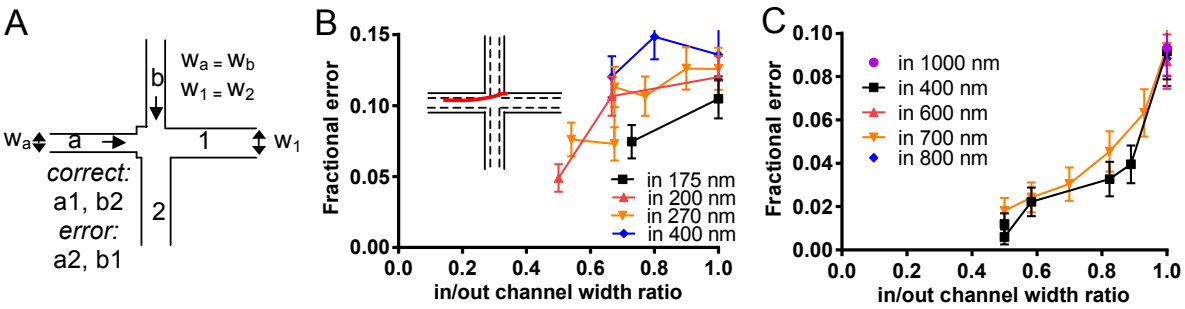


Fig. S4.1. *(A) Geometry used for the Monte-Carlo simulation of pass-junctions (see Fig. S3.1). Filaments enter from the left* ***a*** *and from above* ***b****. Correct pass is from* ***a*** *to* ***1****,* ***a1*** *and* ***b*** *to* ***2****,* ***b2****. Filament paths from* ***a*** *to* ***2****,* ***a2*** *and* ***b*** *to* ***1****,* ***b1*** *are regarded as errors. (B) Fractional error for various ratios of channel widths,* $w_a/w_1 = w_b/w_2$ *for various absolute values of* $w_a = w_b$*. Simulations for conditions corresponding to actin filament behavior (*v_f *= 10 µm/s,* L_p *= 10 µm). Inset: Pass-junction width* $w_a = w_b = w_1 = w_2$ *indicating HMM layer (dashed lines) of 35 nm thickness (25) on channel walls. Possible path of actin filament (red) mimicking behavior in channel with* $w_a = w_b < w_1 = w_2$*. (C) Simulations as in (B) but for conditions corresponding to behavior of kinesin-propelled microtubules (*v_f *= 0.5 µm/s,* L_p *= 100 µm).*

S5. Detailed results of junction-performance tests

The performances of the pass-junctions and split-junctions were analyzed by tracking individual filaments through the devices. This information was used to gauge the performance of the junctions and as an input to Monte Carlo modeling of overall device performance (see supplementary text S6).

S5.1 Results for the actin–myosin system

The behavior of actin filaments moving through pass- and split-junctions is summarized in Tables S5.1.1 and S5.1.2, respectively.

Table S5.1.1: *Pass-junction performance in the {2, 5, 9} subset sum device for the actin–myosin system.*

	a1	**a2**	**b1**	**b2**	Tot **a**	Tot **b**	Tot
# Filaments	55	1885	1706	23	1940	1729	3669
Total Fraction	2.8%	97.2%	98.7%	1.3%			

Note*:* ***a1****,* ***a2****,* ***b1****,* ***b2*** *denote the paths taken through the junction. For example: "****b1****" indicates a filament that entered the device through channel* ***b*** *and left the device through channel* ***1*** *(see Fig. S3.1 for an explanation of channel annotation). Paths for which the pass-junction was designed are shown in black. Those paths not intended in the design are shown in red. The total fraction was obtained by taking the ratio of the number of filaments following a given path from one side divided by the total number of filaments entering the junction at that side.*

Table S5.1.2: *Split-junction performance in the {2, 5, 9} subset sum device used with the actin–myosin system, and expected statistical distribution (EStD).*

	a1	**a2**	**b1**	**b2**	Tot **a**	Tot **b**	Tot
# Filaments	429	437	262	364	866	626	1492
Total Fraction	50%	50%	42%	58%			
EStD	50 ± 5%	50 ± 5%	50 ± 6%	50 ± 6%			

***Note: a1**, **a2**, **b1**, **b2** denote the paths taken through the junction. For example: "**b1**" indicates a filament that entered the device through channel **b** and left the device through channel **1** (see Fig. S3.2 for an explanation of channel annotation). The total fraction was obtained by taking the ratio of the number of filaments following a given path from one side divided by the total number of filaments entering the junction at that side. The EStD is given in terms of equation (2) to within three standard deviations based on the total number of filaments entering side **a** and **b**, respectively.*

Note that, even for a perfectly symmetrical split-junction with precisely 50/50 probabilities for left/right splitting, there is a standard maximal error, which can be described by equation (2):

$$E = Z/2n^{1/2} \qquad (2)$$

Within a given standard deviation the error (E) can be assigned to estimate the minimal number of filaments (n) required to obtain an even distribution. Here, however, the input of the number of filaments was counted from side **a** and **b** in the split-junction, and the error was found to within three standard deviations or $Z = 3$ for a standard normal distribution.

Table S5.1.2 shows that the expected distribution of filaments entering a given split-junction from side a corresponds to a perfect 50 % to 50 % split. In contrast, filaments entering from side b are split 42 % to 58 %. This is more uneven than the expected statistical distribution (44 % to 56 % for a total of 626 filaments), indicating a systematic error in the junction geometry. This, however, does not limit the functionality of the junctions (see also supplementary text S6).

S5.2 Results for the kinesin–microtubule system

The behavior of microtubules moving through pass- and split-junctions is summarized in Tables S5.2.1 and S5.2.2, respectively.

Table S5.2.1: *Pass-junction performance in the {2, 5, 9} subset sum device for microtubule-kinesin system.*

	a1	**a2**	**b1**	**b2**	Tot **a**	Tot **b**	Tot
# Filaments	2	1007	794	3	1009	797	1801
Total Fraction	0.2%	99.8%	99.6%	0.4%			

***Note: a1**, **a2**, **b1**, **b2** denote the paths taken through the junction. For example: "**b1**" indicates a filament that entered the device through channel **b** and left the device through channel **1** (see Fig. S3.2 for an explanation of channel annotation). Paths for which the pass-junction was designed are shown in black. Those paths not intended in the design are shown in red. The total fraction was obtained by taking the ratio of the number of filaments following a given path from one side divided by the total number of filaments entering the junction at that side.*

Table S5.2.2: *Split-junction performance in the {2, 5, 9} subset sum device used with the microtubule-kinesin system.*

	a1	**a2**	**b1**	**b2**	Tot **a**	Tot **b**	Tot
# Filaments	74	76	283	272	150	555	705
Total Fraction	49%	51%	51%	49%			

***Note: a1**, **a2**, **b1**, **b2** denote the paths taken through the junction. For example: "**b1**" indicates a filament that entered the device through channel **b** and left the device through channel **1** (see Fig. S3.2 for an explanation of channel annotation). For the split-junction, the intended performance is for filaments entering at either side to be split evenly to **1** or **2**. The total fraction was obtained by dividing the number of filaments following a given path from one side (for example, **a1** and **a2**) divided by the total number of filaments entering the junction at that side (for example, Tot **a**).*

Tables S5.2.1 and S5.2.2 show a detailed summary of the paths taken by microtubules in the pass-junctions and split-junctions of the {2, 5, 9} device, respectively. The same data are shown in less detail in Fig. 3C in the main text. It is important to note that the design of the split and pass junctions ensured an error rate which is considerably lower than that reported previously (21) in similar type of junctions. Following the same statistical considerations as discussed in S5.1, Table S5.2.2 shows that the results for the kinesin–microtubule device are in good agreement with an even distribution between the exit channels at the split-junctions.

S6 Monte Carlo simulations of device performance

Monte Carlo simulations were used to assess device performance. The algorithm simulates a given number of agents travelling through the computation network. At each junction, an evenly distributed random number between 0 and 1 is assigned to each agent. If this number is above a turning threshold defined for that particular junction, the respective agent changes its direction (for example, from moving diagonally down to moving straight down). All other agents pass the junction and continue on their path. The turning threshold corresponds to the splitting ratio or the error rate of the split-junctions or the pass junctions, respectively.

When the experimentally measured error rates of pass-junctions and split-junctions (see S5) were used, the overall shape of the experimental data was very well reproduced in the case of the actin-myosin device (see left column of Fig. 4b and c, in the main text). Because the experimental data were averaged over all junctions, this shows that each individual junction performed approximately the same as the average, confirming that the device performed as expected. Furthermore, simulations with different parameters (not shown) supported the intuition that the error rate at pass-junctions, leading to computation of illegal sums, is critical for device performance, but that split-junction errors have a smaller effect (see also supplementary text S5).

In the case of the microtubule-kinesin device, the simulations could not reproduce the experimentally observed data as well as in the case of the actin-myosin data. Careful analysis of the microtubule data showed that one reason for this was landing of filaments from solution into the channels (see caption of Fig. 3). To account for this additional source of error, we assigned a low probability to have a filament added to each junction in case of the microtubule-kinesin simulations. Another error observed in the microtubule-kinesin device was that many filaments detached from the channel leading towards exit 11, resulting in an unusually low number of filaments reaching said exit. Because this happened at the same position on two independent devices, we attribute this to a systematic fabrication error in our very early prototype devices. Unfortunately, we were not able to fabricate a new generation of devices without this error, because Philips closed down its electron beam lithography facility that produced these devices.

S7. Energy efficiency of the parallel computation system

The following discussion refers to the energy-requirements per operation for different computation technologies, demonstrating that the molecular-motors based approach presented here has extremely reduced energy needs compared to other practical approaches.

Thermodynamic limit. The lowest possible benchmark is the fundamental, thermodynamic limit for the energy cost of a reversible computational step, namely $kT\ln 2 \approx 2.9\ 10^{-22}$ J/operation (26).

Electronic computers. The energy requirements and the associated heat dissipation problems have become severe limitations for electronic computers (27). Assuming an Ivy Bridge Intel chip, built on the 22-nm manufacturing process (Core i7-3770K), the energy required is 6.3×10^{-10} J/operation (28). Also at the supercomputing end, the energy requirements per operation are still very high, i.e., between 2.3 to $5.3\ 10^{-10}$ J/operation, calculated for the highest energy efficient and highest computing speed, respectively, from the latest data (29). Moreover, taking into consideration the overall power consumption for high-end computation, it has been reported (30) that the next generation supercomputing system will require 100–200 MW, which is close to the power generated by a small power plant. Clearly, aside from the difficulties of solving combinatorial problems with sequential computing machines, it appears that the energy consumption is an additional severe limitation for electronic computers.

Microfluidics-based computers. Microfluidics devices can be used, either to host DNA computation processing steps (31, 32), or to encode a mathematical problem in designed networks, which are explored by

inert agents, e.g., beads (33). Separate from the computational advantages and disadvantages of the microfluidics-based computers, the problems related to the power consumption will be challenging for the scale-up of these devices, because very high pumping pressures would be required to drive fluids, with beads, through long and narrow microfluidics channels in complex, highly miniaturized devices. The power required for pumping fluids through small sectors of the microfluidics device is small, but the combinatorial nature of the network will inherently lead to the exponential amplification of the energy requirements even for devices solving small problems. For instance, for a hypothetical chip proposed (33) comprising a network of channels with a diameter of 200 nm and a length of the computing unit of 1000 nm, with a fluid flow of 1 x 10^{-8} μL/min (which translates in a fluid velocity of 5 μm/sec, similar to the velocity of myosin-propelled actin filaments), a competitive 1.29 10^{-12} J/operation was obtained, roughly two orders of magnitude better than electronic computers. However, for the above chip, which would have 20 vertices, the pressure required to pump the fluid would be around 440 atm, which is clearly unachievable from an engineering standpoint.

DNA computing. The initial paper (34) on DNA computing reports an energy consumption of 5 x 10^{-20} J/operation, which is remarkably low and only two orders of magnitude from the ultimate benchmark calculated above. Contrary to the computational systems discussed above, the distributed nature of the availability of energy in the fluids where DNA synthesis occurs, allows for an estimated very low power consumption for the overall DNA computation.

Molecular motors-based computation. One mathematical operation (addition) in the motor driven device reported here corresponds to molecular motor driven transportation of a cytoskeletal filament between two subsequent split-junctions. We first consider the actomyosin system. For the SSP of the 30 first prime numbers (N=30; average numerical value of the primes: 53) this distance is 53 x 5 = 265 μm for a 5 μm unit cell. If trimethylsilane functionalized surfaces are used for adsorption of heavy meromyosin to give saturating surface density in the channels then, on average, <5 myosin heads are attached simultaneously to a 1 μm long actin filament at each given point in time (35) and each head stays attached for sliding of at least 10 nm (36). On these assumptions, one operation requires less than 5 x 265000/10 actomyosin interaction cycles, each with consumption of 1 ATP (corresponding to 25 kT, i.e. 100 10^{-21}J of chemical energy), i.e. a total energy consumption of less than 2 10^{-14}J/operation (or 5 10^{6} kT) A similar value is obtained for kinesin propelled microtubules if the step size is 8 nm and less than 10 kinesin motor domains are interacting with a 1 μm long microtubule at each given point in time. Clearly, the energy consumption per operation is several orders of magnitude lower for a molecular motor powered device than for an electronic computer.

The very low energy consumption by a motor driven device of the present type may also be illustrated by considering the device with N = 30 presented in Fig. S1.1. Let us assume that each filament in this device moves 9000 μm, i.e. the distance traversed in 0.5 h by a filament moving at 5 μm/s and corresponding to the solution that takes the longest to calculate (thus overestimating energy consumption). If each of 1 billion filaments moves this distance, then less than 5 x 10^{6} x 9 x 10^{9}/10 = 4.5 x 10^{15} ATP molecules will be consumed, corresponding to 7.5 nmole of ATP. This should be compared to the amount of ATP using a standard ATP concentration of 1 mM and a flow cell volume (also standard) of around 10 μl, which would contain 10 nmole ATP. Naturally, for long channels as considered above, the total amount of available ATP would be considerably higher as larger flow cells and volumes of assay solution are used. Importantly, the above calculations show that the ATP consumption, and any issues related to heat dissipation are largely irrelevant when the scalability of a molecular motor powered biocomputation device is considered.

A synthesized view of the data regarding the actual or estimated energy efficiency of various computing systems is presented in Table S7.1.

Table S7.1*. Energy efficiency for various computing systems*

Computation system	**Mode of calculation**	**Energy required J/operation**	**Notes, and source**
Thermodynamic limit	Theoretical	2.90×10^{-22}	Thermodynamics considerations (26)
Electronic	Actual	6.33×10^{-10}	Intel Sandy Bridge (28)
	Actual	5.26×10^{-10}	Tianhe-2 (MilkyWay-2), top speed, 2013 (29)
	Actual	2.34×10^{-10}	TSUBAME-KFC, top energy efficiency, 2013 (29)
Microfluidics	Estimated	1.29×10^{-12}	Chip with d=200nm x L= 1000nm units; fluidics design (33)
DNA	Estimated	5.00×10^{-20}	Thermodynamics considerations (34)
Molecular motors	Estimated	4.95×10^{-14}	Kinesin/microtubule system, Thermodynamics + design
	Estimated	2.00×10^{-14}	Myosin/actin filament system

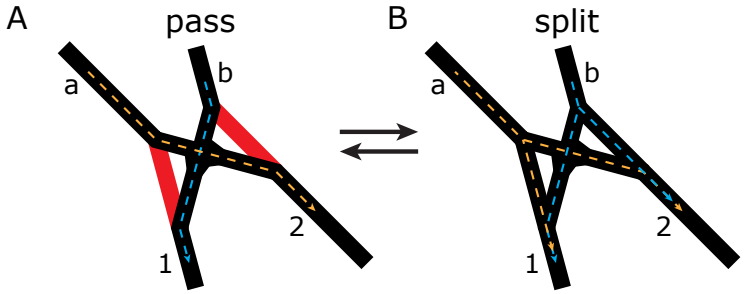


Fig. S7.1. *Schematics of a junction that can be switched between split- and pass functionality. If the channels marked in* ***(A)*** *are blocked or bypassed, the junction would function as a pass junction, if the respective channels are open, the junction would function as a split junction* ***(B)****.*

Energy use of gate operation. In the discussion of the main text, we have proposed to use gated junctions to build a programmable device that can flexibly encode different problems. Of course, the added flexibility would require additional power for the gating, which we have estimated based on gated junctions that have been demonstrated for microtubules by thermal (37) or electrostatic (38) switching. Our existing junction design offers a straightforward approach to gating, because in each junction two channels make the difference between a pass- and a split junction (marked in red in Fig. S7.1A). For thermal switching only the pass

junctions need to be heated in order to open the channels that convert a pass to a split junction (because the thermoresponsive polymers collapse and enable motility when heated). This means that 30 rows of 1 to 1563 junctions or 46890 channels (two per junction) in total would need to be heated to solve the SSP for $N = 30$ with thermal switching. Based on the power density of 150 PW/m^3 published in (37) the whole thermally gated device would consume 6.5 W/mm^2 (about ten times higher than a core i7-3770K CPU) or 2 x 10^{-5} J/operation. For electrical switching, our junction design would require a voltage across two channels in each pass junction in order to convert split to pass junctions. This means that 1533 rows of 2 to 1563 junctions or 2396080 channels in total would need to be electrically switched to solve the SSP for the first 30 primes. Based on the power required for electrical switching published in (38), the device would consume 0.25 mW/mm^2 (about two thousand times lower than a core i7-3770K CPU) or 8 x 10^{-10} J/operation. We caution that these power densities are calculated from switchable junctions that were designed for different purposes and are likely subject to substantial improvement upon optimization to our junction design. For example, our design could work with much lower heat gradients than used by (37) and thus much lower power densities. Crucially, as opposed to the very low power required for filament propulsion, the power required for gating does not scale exponentially with problem size but scales linearly with the network size and thus ~N^2 with the problem size (see S1). This means that for very large problems, the energy required for gating will become small compared to the energy required for filament propulsion and thus the energy efficiency of the device will converge towards the energy efficiency estimated above for the non-gated molecular motors-based computation.

References

1. Sundberg M*, et al.* (2006) Actin filament guidance on a chip: toward high-throughput assays and lab-on-a-chip applications. *Langmuir* 22(17):7286-7295.
2. Sundberg M*, et al.* (2003) Silanized surfaces for in vitro studies of actomyosin function and nanotechnology applications. *Anal. Biochem.* 323(1):127-138.
3. Bunk R*, et al.* (2005) Guiding motor-propelled molecules with nanoscale precision through silanized bi-channel structures. *Nanotechnology* 16(6):710-717.
4. Margossian SS & Lowey S (1982) Preparation of myosin and its subfragments from rabbit skeletal muscle. *Methods Enzymol.* 85(Pt B):55-71.
5. Kron SJ, Toyoshima YY, Uyeda TQ, & Spudich JA (1991) Assays for actin sliding movement over myosin-coated surfaces. *Methods Enzymol.* 196:399-416.
6. Pardee JD & Spudich JA (1982) Purification of muscle actin. *Methods Cell Biol.* 24:271-289.
7. Sundberg M*, et al.* (2006) Selective spatial localization of actomyosin motor function by chemical surface patterning. *Langmuir* 22(17):7302-7312.
8. Coy DL, Wagenbach M, & Howard J (1999) Kinesin takes one 8-nm step for each ATP that It hydrolyzes. *J. Biol. Chem.* 274(6):3667-3671.
9. Hyman A*, et al.* (1991) Preparation of modified tubulins. *Methods in Enzymology*, ed Richard BV (Academic Press), Vol 196, pp 478-485.
10. Nitzsche B*, et al.* (2010) Studying kinesin motors by optical 3D-nanometry in sliding motility assays. *Methods Cell Biol.* 95:247-271.
11. van den Heuvel MGL, Butcher CT, Smeets RMM, Diez S, & Dekker C (2005) High rectifying efficiencies of microtubule motility on kinesin-coated gold nanostructures. *Nano Lett* 5(6):1117-1122.
12. Ruhnow F, Zwicker D, & Diez S (2011) Tracking single particles and elongated filaments with nanometer precision. *Biophys. J.* 100(11):2820-2828.
13. Hild G, Bugyi B, & Nyitrai M (2010) Conformational dynamics of actin: effectors and implications for biological function. *Cytoskeleton* 67(10):609-629.

14. Wade RH & Hyman AA (1997) Microtubule structure and dynamics. *Curr Opin Cell Biol* 9(1):12-17.
15. Lard M*, et al.* (2013) Ultrafast molecular motor driven nanoseparation and biosensing. *Biosens. Bioelectron.* 48(0):145-152.
16. Radke MB*, et al.* (2014) Small molecule-mediated refolding and activation of myosin motor function. *eLife* 3:e01603.
17. Lard M, ten Siethoff, L., Generosi, J., Månsson, A., Linke, H. (2014) Molecular motor transport through hollow nanowires. *Nano Lett* 14(6):3041-3046.
18. Hiratsuka Y, Tada T, Oiwa K, Kanayama T, & Uyeda TQ (2001) Controlling the direction of kinesin-driven microtubule movements along microlithographic tracks. *Biophys. J.* 81(3):1555-1561.
19. Vikhorev PG*, et al.* (2008) Diffusion dynamics of motor-driven transport: Gradient production and self-organization of surfaces. *Langmuir* 24(23):13509-13517.
20. Månsson A, Bunk, R., Sundberg, M., Montelius, L. (2012) Self-organization of motor-propelled cytoskeletal filaments at topographically defined borders. *J. Biomed. Biotechnol.* 2012(647265).
21. Clemmens J*, et al.* (2004) Motor-protein "roundabouts": Microtubules moving on kinesin-coated tracks through engineered networks. *Lab Chip* 4(2):83-86.
22. Clemmens J, Hess H, Howard J, & Vogel V (2003) Analysis of microtubule guidance in open microfabricated channels coated with the motor protein kinesin. *Langmuir* 19(5):1738-1744.
23. Hess H*, et al.* (2003) Molecular Shuttles Operating Undercover: A New Photolithographic Approach for the Fabrication of Structured Surfaces Supporting Directed Motility. *Nano Lett* 3(12):1651-1655.
24. Nitta T, Tanahashi A, Hirano M, & Hess H (2006) Simulating molecular shuttle movements: Towards computer-aided design of nanoscale transport systems. *Lab Chip* 6(7):881-885.
25. Persson M*, et al.* (2010) Heavy meromyosin molecules extending more than 50 nm above adsorbing electronegative surfaces. *Langmuir* 26(12):9927-9936.
26. Landauer R (1961) Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.* 5(3):183-191.
27. Madden PH (2013) Dispelling the myths of parallel computing. *IEEE Design and Test* 30(1):58-64.
28. Gepner P, Fraser DL, & Gamayunov V (2012) Evaluation of the 3rd generation Intel Core Processor focusing on HPC applications. *Technology* 7:i7–2600.
29. Meuer HS, E.; Dongarra, J.; Simon, H. (2013) Top 500 Supercomputers.
30. Mukhanov OA (2011) Energy-Efficient single flux quantum technology. *IEEE Transactions on Applied Superconductivity* 21(3 PART 1):760-769.
31. Ying N, Xuncai Z, & Guangzhao C (2011) Solving maximum clique problems with microfluidic DNA computer. *Computational Intelligence and Design (ISCID), 2011 Fourth International Symposium on* 1:224-227.
32. Ying N, Xuncai Z, & Guangzhao C (2012) Solving graph vertex coloring problem with microfluidic DNA computer. *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*:5061-5065.
33. Chiu DT, Pezzoli E, Wu H, Stroock AD, & Whitesides GM (2001) Using three-dimensional microfluidic networks for solving computationally hard problems. *Proc. Natl. Acad. Sci. U. S. A.* 98(6):2961-2966.
34. Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266(11):1021-1024.
35. Persson M, Bengtsson E, ten Siethoff L, & Mansson A (2013) Nonlinear cross-bridge elasticity and post-power-stroke events in fast skeletal muscle actomyosin. *Biophys. J.* 105(8):1871-1881.
36. Walcott S, Warshaw DM, & Debold EP (2012) Mechanical coupling between myosin molecules causes differences between ensemble and single-molecule measurements. *Biophys. J.* 103:501-510.
37. Schroeder V, Korten T, Linke H, Diez S, & Maximov I (2013) Dynamic guiding of motor-driven microtubules on electrically heated, smart polymer tracks. *Nano Lett* 13(7):3434-3438.
38. van den Heuvel MG, de Graaff MP, & Dekker C (2006) Molecular sorting by electrical steering of microtubules in kinesin-coated channels. *Science* 312(5775):910-914.