# Supplementary Material: Proofs of results

Todd Wareham and Andrew Vardy

All of our intractability results will be derived using polynomial-time and parameterized reductions[1] from the following problems:

3-Satisfiability (3SAT) [6, Problem LO2]
*Input*: A set $U$ of variables and a set $C$ of disjunctive clauses over $U$ such that each clause $c \in C$ has $|c| = 3$.
*Question*: Is there a satisfying truth assignment for $C$?

Clique [6, Problem GT19]
*Input*: An undirected graph $G = (V, E)$ and a positive integer $k$.
*Question*: Does $G$ contain a clique of size $k$, i.e., is there a subset $V' \subseteq V$, $|V'| = k$, such that for all $u, v \in V'$, $(u, v) \in E$?

Dominating set [6, Problem GT2]
*Input*: An undirected graph $G = (V, E)$ and a positive integer $k$.
*Question*: Does $G$ contain a dominating set of size $k$, i.e., is there a subset $V' \subseteq V$, $|V'| = k$, such that for all $v \in V$, either $v \in V'$ or there is at least one $v' \in V'$ such that $(v, v') \in E$?

Compact Deterministic Turing Machine Computation (CDTMC) ([2]; [6, Problem AL3])
*Input*: A deterministic Turing Machine $M$ with state-set $Q$, tape alphabet $\Sigma$, and transition-function $\delta$, an input $x$, and a positive integer $k$.
*Question*: Does $M$ accept $x$ in a computation that uses at most $k$ tape squares?

The problems 3SAT, Clique, and Dominating Set are $NP$-hard in general and problem CDTMC is $PSPACE$-hard in general; moreover prob-

[1] Given two problems $\Pi$ and $\Pi'$, a polynomial-time reduction from $\Pi$ to $\Pi'$ is essentially a polynomial-time algorithm for transforming instances of $\Pi$ into instances of $\Pi'$ such that any polynomial-time algorithm for $\Pi'$ can be used in conjunction with this instance transformation algorithm to create a polynomial-time algorithm for $\Pi$. Analogously, a parameterized reduction from $\langle K\rangle$-$\Pi$ to $\langle K'\rangle$-$\Pi'$ allows the instance transformation algorithm to run in fp-time relative to parameter-set $K$ and requires that for each parameter $k'$ in parameter-set $K'$, there is a function $g_{k'}()$ such that $k' = g_{k'}(K)$. Such an instance transformation algorithm can be used in conjunction with any fp-algorithm for $\langle K'\rangle$-$\Pi'$ to create an fp-algorithm for $\langle K\rangle$-$\Pi$. The interested reader is referred to [4, 6] for details.

lems Clique and Dominating set and $W[1]$-hard and $W[2]$-hard, respectively, relative to parameter-set $\{k\}$ [4]. For each vertex $v \in V$, let the complete neighbourhood $N_C(v)$ of $v$ be the set composed of $v$ and the set of all vertices in $G$ that are adjacent to $v$ by a single edge, i.e., $v \cup \{u \mid u \in V \text{ and } (\mathrm{u}, \mathrm{v}) \in \mathrm{E}\}$. We assume below for each instance of Clique and Dominating set an arbitrary ordering on the vertices of $V$ such that $V = \{v_1, v_2, \ldots, v_{|V|}\}$; we also assume analogous orderings for the variables and clauses of each instance of 3-Satisfiability such that $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and $C = \{c_1, c_2, \ldots, c_{|C|}\}$.

For technical reasons, all intractability results are proved relative to decision versions of problems, i.e., problems whose solutions are either "yes" or "no". For example, problem ContEnvVer defined in Section 2 is a decision problem. Though problems ContDes and EnvDes are not decision problems, each can be made into a decision problem by asking if that problem's requested output exists; let the decision version for a problem $\mathbf{X}$ be denoted by $\mathbf{X}_D$. The following two easily-proven lemmas will be useful below in transferring results from decision problems to their associated non-decision and parameterized problems; both follow from the observation that any algorithm for non-decision problem $\mathbf{X}$ can be used to solve $\mathbf{X}_D$ and the definition of fp-tractability.

**Lemma 1** *If $\mathbf{X}_D$ is $NP$-hard then $\mathbf{X}$ is not solvable in polynomial time unless $P = NP$.*

**Lemma 2** *Given a parameter-set $K$ for problem $\mathbf{X}$, if $\mathbf{X}_D$ is $NP$-hard when the value of every parameter $k \in K$ is fixed to a constant value, then $\langle K \rangle$-$\mathbf{X}$ $\notin FPT$ unless $P = NP$.*

**Lemma 3** *CDTMC polynomial-time reduces to ContEnvVer such that in the constructed instance of ContEnvVer, $|T| = |f| = |X| = 1$, $r = 0$, and $|E|$ is a function of $|x|$ and $k$ in the given instance of CDTMC.*

**Proof:** Given an instance $\langle M = (Q, \Sigma, \delta), x, k \rangle$ of CDTMC, construct an instance $\langle E, E_T, T, X, p_I, p_X \rangle$ of ContEnvVer as follows: Let $E_T = \Sigma \cup \{e_B, e_{F1}, e_{F2}\}$, $E$ be a $\max(k, |x|) + 2 \times 1$ grid in which the first $|x|$ squares encode $x$, the next $\max(k - |x|, 1)$ squares are of type $e_B$, and the final square is of type $e_{F1}$, $p_I = W_{1,1}$, and $X$ is a single square of type $e_{F2}$ at $p_X = W_{\max(k,|x|)+2,1}$. Let $T$ consist of a single FSR with $r = 0$ such that $Q' = Q \cup \{q_{F1}\}$ and $\delta'$ consists of the FSR analogues of all transitions in $\delta$ (phrased now in terms of eastward and leftward movements in $E$) plus the transitions $\langle q_A, *, *, goEast, q_{F1}\rangle$, $\langle q_{F1}, enval(e_{F1}, (0, 0)), e_{F2}, stay, q_{F1}\rangle$, and $\langle q_{F1}, enval(e_{F2}, (0, 0)), *, stay, q_{F1}\rangle$ where $q_A$ is the accepting state of $M$. The TM $M$ underlying the single FSR in $T$ is deterministic in the classical automata-theoretic sense described in Footnote 1 and thus has at most one transition enabled at any time; hence, the operation of that FSR in $E$ is deterministic in the sense defined in this paper. Note that this instance of ContEnvVer can be constructed in time polynomial in the size of the given instance of CDTMC.

If $M$ accepts $x$ using at most $k$ tape squares, the single FSR in $T$ starting from $p_I$ (as it is based on the same transitions as $M$) will eventually enter state $q_A$ sitting on one of the first $k$ squares of $E$. The extra transitions described above will then ensure that the FSR proceeds to the eastmost square of $E$, i.e., $p_X$, replaces $e_{F1}$ with $e_{F2}$, i.e., $X$, and then stays there. Conversely, if the single FSR in $T$ can proceed from $p_I$ to create $X$ at $p_X$ then at some point it must have entered $q_A$ (as only the extra transitions added to $M$ originating from $q_A$ could have moved eastwards over the second-last square in $E$ to reach $p_X$), which would imply that $M$ accepts $x$.

To complete the proof, note that in the constructed instance of ContEnvVer, $|T| = |f| = |X| = 1$ and $r = 0$. ∎

As CDTMC is $PSPACE$-hard, Lemma 3 establishes that ContEnvVer is also $PSPACE$-hard. This has some unexpected consequences:[2]

- If $P \neq PSPACE$ and $P \neq NP$ then DesCont and DesEnv are polynomial-time intractable (proof by contradiction: Suppose DesCont$_D$ is in $NP$, the class of decision problems whose solutions can be verified in polynomial time. As this membership implies that it can be verified in polynomial time whether or not a given controller and environment can produce a specified target structure, ContEnvVer can be solved in polynomial time. However, as ContEnvVer is $PSPACE$-hard, this implies that $P = PSPACE$, which is a contradiction. Therefore, DesCont$_D$ cannot be in $NP$. As we also assume that $P \neq NP$, DesCont$_D$ and hence DesCont cannot be in $P$ and hence polynomial-time tractable either. An analogous argument establishes this result relative to DesEnv).

- If $P = BPP$, $P \neq PSPACE$, and $P \neq NP$ then ContEnvVer, DesCont and DesEnv are not polynomial-time tractable by probabilistic algorithms which operate correctly with probability $\geq 2/3$ (proof by

[2] We thank the second anonymous reviewer for pointing out these consequences.

contradiction: The existence of an algorithm with the operation correctness bound for a problem by definition means that this problem is in $BPP$. As we assume that $P = BPP$, that problem is also in $P$. However, this contradicts the proofs above that ContEnvVer, DesCont and DesEnv cannot be in $P$ relative to the conjectures $P \neq NP$ and $P \neq PSPACE$. Hence, none of these problems can be in $BPP$).

As the conjectures $P = BPP$, $P \neq NP$, and $P \neq PSPACE$ are widely believed within Computer Science [3, 5, 6, 10], it is reasonable to also assume that the intractabilities stated in these consequences hold. Hence, Lemma 3 answers Questions 1 and 2 posed in Section 1 of the main text relative to ContEnvVer and the unrestricted forms of DesCont and DesEnv.

It seems likely that all three of these problems are very hard indeed. The precise degree of this hardness should be resolved; however, it is not necessary to do this in this paper. This is so because we are interested here (at least initially) in polynomial-time tractability, and once polynomial-time intractability is established, the precise degree of this intractability does not matter — we are then concerned with parameterized tractability. Additional work should thus address the following issues:

1. The consequences above do not hold for DesCont$^{fast}$ and DesEnv$^{fast}$ (as such problems explicitly disallow the nonpolynomial-length task executions possible under ContEnvVer that allow the proofs above to work).

2. The consequences above have nothing to say about the parameterized complexity of ContEnvVer, ContDes$^{fast}$, or EnvDes$^{fast}$.

This will be done using the reductions in the lemmas below.

**Lemma 4** *3SAT polynomial-time reduces to ContEnvVer$_{syn}$ such that in the constructed instance of ContEnvVer$_{syn}$,* $|Q| = |X| = 1$ *and* $|E_T| = 9$

**Proof:** In this reduction, we will simulate an instance of 3SAT with $U$ variables and $C$ clauses with a team composed of $|U| + 4$ synchronously-operating FSRs. In this simulation, $|U| + 3$ of the FSRs are configured to enumerate through all possible truth-assignments to the variables in $U$. As each truth-assignment is enumerated, an "evaluator" FSR determines if that truth-assignment satisfies the conjunction of the clauses in $C$ and, if so, creates the requested target structure. All of these activities are co-ordinated by FSRs sensing the positions of other FSRs and hence communicating indirectly by stigmergy [1, 8].

It is very important to note that the enumeration of all $2^{|U|}$ truth assignments (which requires FSR timesteps exponential in $|U|$) is not actually done in this reduction — we require only that the FSRs in the constructed instance of ContEnvVer$_{syn}$ can in principle enumerate all such truth assignments. We can then (as is done in the proof of Result F) pair the polynomial-time ContEnvVer$_{syn}$-instance construction process specified in the reduction with a hypothetical polynomial-time algorithm for ContEnvVer$_{syn}$ to derive a polynomial-time algorithm for 3SAT and hence by contradiction prove the polynomial-time intractability of ContEnvVer$_{syn}$ relative to the conjecture $P \neq NP$ (see also Footnote 1).

Given an instance $\langle U, C\rangle$ of 3SAT, construct an instance $\langle E, E_T, T, X, p_I, p_X\rangle$ of ContEnvVer$_{syn}$ as follows: Let $E_T = \{e_B, e_E, e_S, e_W, e_N, e_{CD}, e_{CS}, e_{Ev}, e_{SAT}\}$ (all of which are freespaces) and $E$ be a $|U| + 3 \times 8$ grid with the following structure:

- The westmost $|U| + 1$ squares in the southernmost row consist of a square of type $e_N$ followed by $|U|$ squares of type $e_W$.
- $E_{|U|+2,2} = e_{CD}$.
- $E_{|U|+2,3} = W_{|U|+2,4} = W_{|U|+2.5} = e_S$.
- $E_{1,4} = e_E$.
- The westmost $|U| + 2$ squares in the 6th row consist of a square of type $e_B$ followed by $|U|$ squares of type $e_V$ followed by a square of type $e_{CS}$.
- $E_{1,8} = e_{Ev}$.
- All remaining squares have type $e_B$.

Let $T$ be a team of $|U| + 4$ FSRs composed of five groups of FSRs:

1. **Variable**: $|U|$ FSRs which are initially positioned in the $|U|$ squares of the 4th row starting at $E_{2,4}$.
2. **Carry**: A single FSR which is initially positioned at $E_{2,3}$.
3. **CarryDetect**: A single FSR which is initially positioned at $E_{|U|+3,3}$.
4. **CarrySignal**: A single FSR which is initially positioned at $E_{|U|+3,5}$.

5. **Evaluate**: A single FSR which is initially positioned at $E_{2,8}$.

The Variable FSRs in column $i+1$, $1 \leq i \leq |U|$ encodes the truth-value of variable $u_i$ in the instance of 3SAT, with this value being $False$ if the FSR is in row 4 and $True$ if it is in row 5. If $False$ is interpreted as 0 and $True$ is interpreted as 1, the Variable FSRs encode a binary number in the range $0 - 2^{|U|} - 1$ whose ascending digits are written left to right in the columns of $E$ rather than the traditional right to left. The Carry FSR travels in a repeating loop that goes east in row 3 and west in row 2 such that it allows us to simulate the addition of 1 to the number encoded by the Variable FSRs. In this addition process, a carry bit is active in the addition if the Carry FSR is in row 3; this is detected by the CarryDetect FSR. When a carry bit is not active, i.e., the number encoded in the Variable FSRs has had 1 added to it, the CarryDetect FSR moves south. This causes the CarrySignal FSR to move north, which in turn signals the Evaluate FSR that the truth-assignment encoded in the Variable FSRs is stable and can be evaluated. Once the Carry FSR enters $E_{1,3}$, the CarryDetect and CarrySignal FSR return to their original positions and another addition begins. An example environment $E$ and initial placement of a team $T$ of FSRs described above when $|U| = 5$ is shown in Figure 1.

The controller for these FSRs that implements the activities described above has a single state with the following groups of transitions:

- **Variable**: Two transitions such that (1) if there is an FSR immediately south and a square of type $e_V$ one square north then execute action $goNorth$ and (2) if there is an FSR one square south and square of type $e_V$ immediately north then execute action $goSouth$.

- **Carry**: Four transitions such that (1) if $\mathbf{X}$ and there is either no FSR immediately north, a square of type $e_E$ immediately north, or no square of type $e_S$ immediately east then execute action $goEast$, (2) if $\mathbf{X}$ and there is a square of type $e_S$ immediately east then execute action $goSouth$, (3) if $\mathbf{X}$ and there is a square of type $e_W$ immediately south then execute action $goWest$, and (4) if $\mathbf{X}$ and there is a square of type $e_N$ immediately south then execute action $goNorth$, where $\mathbf{X}$ is $True$ if there is a square of type of type $e_N$ or $e_W$ either immediately or one square to the south and $False$ otherwise.

- **CarryDetect**: Two transitions such that (1) if there is a square of type $e_S$ immediately west and there is no FSR within $|U| + 2$ squares to the west then execute action $goSouth$ and (2) if there is a square of

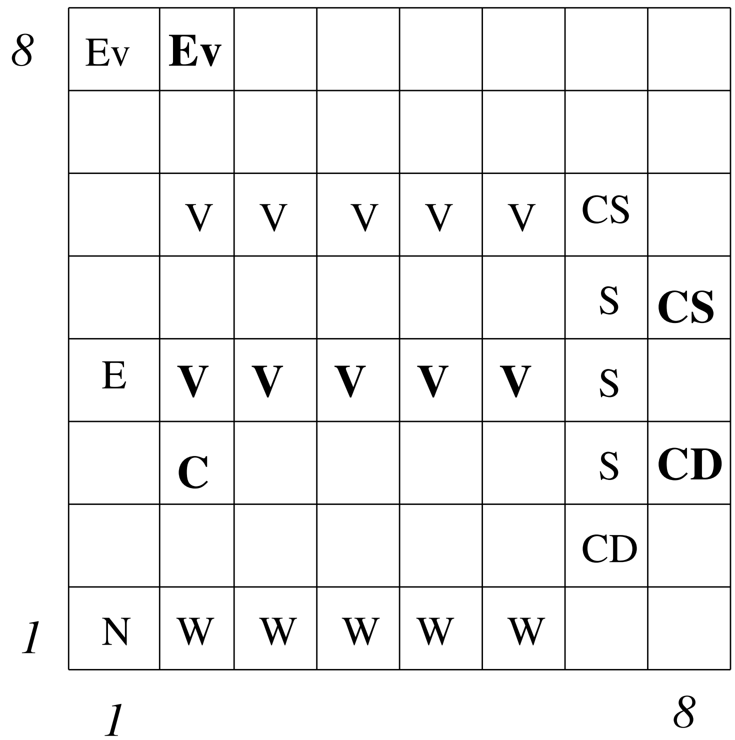


Figure 1: An example environment and initial placement of a FSR team constructed in the reduction in Lemma 4 when $|U| = 5$. Non-$e_B$ square-types are shown in Roman font and the various types of FSRs are shown in boldface (Variable(V), Carry (C), CarryDetect (CD), CarrySignal (CS), Evaluate (Ev)).

type $e_{CD}$ immediately west and there is no FSR within $|U|+2$ squares to the west then execute action $goNorth$.

- **CarrySignal**: Two transitions such that (1) if there is a square of type $e_S$ immediately west and there is an FSR two squares to the south then execute action $goNorth$ and (2) if there is a square of type $e_{CS}$ immediately west and there is an FSR two squares to the south then execute action $goSouth$.

- **Evaluate**: A single transition such that if there is a square of type $e_{Ev}$ immediately west and there is an FSR at $E_{|U|+1,6}$ and the Variable

FSRs encode a truth assignment that satisfies all clauses in $C$ then change the type of the current square to $e_{SAT}$ and execute action $goWest$.

Observe that, starting from the initial positions described above, the synchronous operation of FSRs enabled with this controller will effectively (and repeatedly) enumerate through all possible truth assignments to the variables in $U$. Moreover, courtesy of the initial placement and subsequent evolution of the FSRs in $T$, there is at most one transition that can execute at any point in any FSR in $T$; hence, the operation of the FSRs in $T$ in $E$ is deterministic.

Finally, to complete our construction of the instance of ContEnvVer$_{syn}$, let $p_I$ be as specified above for the FSRs in $T$ and $X$ be a single square of type $e_{SAT}$ at position $p_X = W_{2,8}$.

As for the running time of this reduction, recall that $E$ is an $(|U|+3) \times 8$ grid and there are $|U| + 4$ FSR in this grid, where each FSR is of size at most some polynomial of the summed length of the clauses in $C$ in the given instance of 3SAT. As there are at most $\left( \begin{array}{c} 2|U| \\ 3 \end{array} \right) \leq (2|U|)^3$ clauses in $C$ (namely, all possible choices of 3 variables from the set comprised of the variables in $U$ and the negations of those variables), this means that the constructed instance of ContEnvVer$_{syn}$ is of size polynomial in and moreover can be constructed in time polynomial in the size $|U|$ of the given instance of 3SAT.

Suppose there is a truth-assignment $a$ for the variables in $U$ that satisfies all of the clauses in $C$ in the given instance of 3SAT; let $n$ be the binary number corresponding to $A$. The configuration of the Variable FSRs corresponding to $n$ in the constructed instance of ContEnvVer$_{syn}$ (which is accessible from $p_I$, as all possible truth-assignments are enumerated by the synchronous operation of the FSRs in $T$) will cause the lone transition in the **Evaluate** transition-group to execute in the Evaluate FSR. This in turn will cause (and is indeed the only way to cause) $X$ to be constructed at $p_X$. Similarly, suppose that $X$ can be constructed at $p_X$ by the synchronous operation of the FSRs in the constructed instance of ContEnvVer$_{syn}$. As this can only occur if the lone transition in the **Evaluate** transition-group executes in the Evaluate FSR, this implies that there is a truth-assignment for the variables in $U$ that satisfies all of the clauses in $C$ in the given instance of 3SAT.

To complete the proof, note that in the constructed instance of ContEnvVer$_{syn}$, $|Q| = |X| = 1$ and $|E_T| = 9$. ∎

**Lemma 5** DOMINATING SET *polynomial-time reduces to ContDes$_D^{fast}$ such that in the constructed instance of ContDes$_D^{fast}$, $|T| = |Q| = |f| = |X| = c_1 = c_2 = 1$, $r = 0$, and $d$ is a function of $k$ in the given instance of* DOMINATING SET.

**Proof:** The reduction used here is a modification of that given in the proof of Theorem 3.1 in [9]. That proof establishes that DOMINATING SET (the problem of determining if a given graph $G$ has a dominating set of size $k$) reduces to the problem of designing a team $T$ consisting of a single Gauci-style reactive robot [7] whose sensor points straight down and controller is a nested `if-then-else` block consisting of $k+1$ `if-then` statements whose conditions are $k+1$ elements of $E_T$ followed by a single `else`-statement. This controller enables the robot to navigate from the southernmost westmost corner ($p_I$) of a $(k + |V|) \times (|V|^2 + |V| + 4)$ square-based environment $E$ to the northernmost eastmost corner of $E$. Note that such a computation-free robot is equivalent to a single-state FSR with $r = 0$, $k + 1$ regular transitions corresponding to the $k + 1$ `if-then` statements, and a single default transition corresponding to the `else` statement.

Given an instance $\langle G = (V, E'), k\rangle$ of DOMINATING SET, construct an instance $\langle E'', E'_T, |T'|, p'_I, X, p_X, |Q|, |f|, r, d, c_1, c_2\rangle$ of ContDes$_D^{fast}$ as follows: Let $E''$ be $E$ with the addition of a new northernmost row consisting of $k + |V| - 1$ squares of type $e_N$ followed by a single square of type $e_{F1}$, $E'_T$ be $E_T$ with the addition of square-types $\{e_{F1}, e_{F2}\}$, $p'_I = p_I$, $X$ be a single square of type $e_{F2}$ at $p_X = E''_{k+|V|,|V|^2+|V|+4}$, $|T'| = |Q| = f = 1$, $r = 0$, $d = k+3$, and $c_1 = c_2 = 1$. This instance of ContDes$_D^{fast}$ can be constructed in time polynomial in the size of the given instance of DOMINATING SET. By slight modifications of the proof of correctness of the reduction in Theorem 3.1 in [9], it can be shown that there is a dominating set of size $k$ in graph $G$ in the given instance of DOMINATING SET if and only if there is an FSR with the structure specified in the constructed instance of ContDes$_D^{fast}$ such that (1) $X$ can be constructed at $p_X$ if this FSR starts at $p_I$ and (2) the transitions in this FSR are $\langle q_0, enval(e_{F1}, (0, 0)), e_{F2}, stay, q_0\rangle$, $\langle q_0, enval(e_{F2}, (0, 0)), *, stay, q_0\rangle$, $k$ east-moving transitions whose activation-formula predicates correspond to the vertices in a dominating set of size $k$ in $G$, and default transition $\langle q_0, *, goNorth, q_0\rangle$. As $r = 0$ and $f = 1$, this FSR can sense at most one square and each transition in this FSR has an activation-formula consisting of either $*$ or a single predicate evaluating if that square has a particular square-type. Hence, there can be at most one transition enabled at a time in this FSR and the operation of this FSR in $E''$ is deterministic. As the single robot in $T$ can only move north or east and does one of either in each move, the number of transitions executed in this construction task is the Manhattan distance from $p_I$ to $p_X$ in $E$. This distance is $(k + |V|) + (|V|^2 + |V| + 4) < (k + |V|) \times (|V|^2 + |V| + 4) = c_1|E|^{c_2} = |E|$, which means that the construction task is $(c_1, c_2)$-completable in both the synchronous and asynchronous senses wrt $T$ and $p_I$ when $c_1 = c_2 = 1$. To complete the proof, note that in the constructed instance of ContDes$_D^{fast}$, $|T'| = |Q| = |f| = c_1 = c_2 = 1$, $r = 0$, and $d = k + 3$. ▮

**Lemma 6** *3SAT polynomial-time reduces to EnvDes$_D^{fast}$ such that in the constructed instance of EnvDes$_D^{fast}$, $|T| = |X| = c_1 = 1$, $c_2 = 3$, and $|f| = |E_T| = 5$.*

**Proof:** Given an instance $\langle U, C\rangle$ of 3SAT, construct an instance $\langle G, E_T, T, X, p_I, p_X, c_1, c_2\rangle$ of EnvDes$_D^{fast}$ as follows: Let $G$ be a $2 \times (|U| + 1)$ grid, $E_T = \{e_T, e_F, e_0, e_{F1}, e_{F2}\}$ consist of 5 different types of freespace squares, and $p_I = W_{1,1}$. Team $T$ will consist of a single FSR based on states $Q = \{q_0, q_1, \ldots, q_{|C|}\}$ with the following $|C| + 2$ transitions:

1. For each clause $c_i = v_{i,1} \vee v_{i,2} \vee v_{i,3}$, $1 \leq i \leq |C|$, a transition $\langle q_{i-1}, x_1 \vee x_2 \vee x_3, *, stay, q_i\rangle$, where $x_k = enval(e_T, (0, j))$ if $v_{i,k} = u_j$ and $enval(e_F, (0, j))$ if $v_{ik} = \neg u_j$; and
2. Transitions $\langle q_{|C|}, enval(e_0, (0, 0)), *, goEast, q_{|C|}\rangle$ and $\langle q_{|C|}, enval(e_{F1}, (0, 0)), e_{F2}, stay, q_{|C|}\rangle$.

Finally, let $X$ be a single square of type $e_{F2}$ positioned at $p_X = W_{2,1}$, $c_1 = 1$, and $c_2 = 3$. Note that this instance of EnvDes$_D^{fast}$ can be constructed in time polynomial in the size of the given instance of 3SAT.

If there is an assignment $assign : U \rightarrow \{True, False\}$ to the variables of $U$ that makes each of the clauses in $C$ true, construct an environment $E$ as follows:

1. $E_{1,1}$ has square-type $e_0$;
2. $E_{1,j+1}$, $1 \leq j \leq |U|$, has square-type $e_T$ if $assign(u_j) = True$ and $e_F$ otherwise;
3. $E_{2,1}$ has square-type $e_{F1}$; and
4. All remaining environment-squares have square-type $e_0$.

Observe that $E$ will allow $c$ to progress from $p_I$ to create $X$ at $p_X$; moreover, as at any point in this progress at most one transition in $c$ is enabled, the operation of $c$ in $E$ is deterministic. In order to do this, $c$ must execute all of its $|C|+2$ transitions. As each clause in $C$ is a combination of three negated or unnegated variables from $U$, $|C| < \left(\begin{array}{c} 2|U| \\ 3 \end{array}\right) < (2|U|)^3 = 8|U|^3$. This gives us $|C|+2 < 8|U|^3+2 < 8|U|^3+8 < (2|U|+2)^3 = c_1|E|^{c_2}$ when $c_1 = 1$ and $c_2 = 3$, which means that this construction task is $(c_1, c_2)$-completable in both the synchronous and asynchronous senses wrt $T$ and $p_I$ when $c_1 = 1$ and $c_2 = 3$.

Conversely, suppose there is an environment $E$ such that when FSR $c$ is started at $p_I$, the task of constructing $X$ at $p_X$ is $(c_1, c_2)$ completable. This completability, by the rules of FSR operation given in Section 2, implies that the operation of $c$ in $E$ is deterministic. By the structure of $c$, $c$ can only move right from $E_{1,1}$ to $E_{2,1}$ and subsequently create $X$ there if it can change state from $q_0$ to $q_{|C|}$, $E_{1,1}$ has square-type $e_0$, and $E_{2,1}$ has square-type $f_{F1}$. However, this can only happen if environment-squares $E_{1,2}$ through $E_{1,|U|+1}$ in the westmost column of $E$ have square-types which correspond to a truth-assignment to the variables of $U$ that satisfies each clause in $C$.

To complete the proof, note that in the constructed instance of $\mathrm{EnvDes}_D^{fast}$, $|T| = |X| = c_1 = 1$, $c_2 = 3$, and $|f| = |E_T| = 5$. ∎

Observe that the converse part of the proof of correctness of the preceding reduction can be tightened. Namely, one can show that the operation of $c$ in $E$ is deterministic by noting that courtesy of the structure of $c$, any $E$ that allows $c$ to create $X$ at $p_X$ is such that at most one transition ss enabled in $c$ at any time. This tightening can also be done for the following reduction.

**Lemma 7** *3SAT polynomial-time reduces to $EnvDes_D^{fast}$ such that in the constructed instance of $EnvDes_D^{fast}$, $|T| = |X| = |Q| = c_1 = c_2 = 1$ and $|E_T| = 5$.*

**Proof:** The reduction below is a modification of that given in the proof of Lemma 6. Given an instance $\langle U, C\rangle$ of 3SAT, construct an instance $\langle G, E, T, X,$
$p_I, p_X, c_1, c_2\rangle$ of $\mathrm{EnvDes}_D^{fast}$ as follows: Let $G$, $E_T$, $p_I$, $X$, and $p_X$ be as in the proof of Lemma 6. Team $T$ will consist of a single FSR based on $|Q| = 1$ states with the transitions $\langle q_0, f, *, goEast, q_0\rangle$ and $\langle q_0, enval(e_{F1}, (0,0)), e_{F2}, stay, q_0\rangle$ such that $f$ is the conjunction of the individual clause-transition formulas described in the proof of Lemma 6. Note that this instance of

$\text{EnvDes}_D^{fast}$ can be constructed in time polynomial in the size of the given instance of 3SAT. The proof of correctness of this reduction is the same as that given in the proof of Lemma 6. As $c$ started at $p_I$ must execute 2 transitions to construct $X$ at $p_X$ and $2 < (2|U| + 2) = c_1|E|^{c_2}$ when $c_1 = c_2 = 1$, this construction task is $(c_1, c_2)$-completable in both the synchronous and asynchronous senses wrt $T$ and $p_I$ when $c_1 = c_2 = 1$. To complete the proof, note that in the constructed instance of $\text{EnvDes}_D^{fast}$, $|T| = |X| = |Q| = c_1 = c_2 = 1$ and $|E_T| = 5$. ∎

**Lemma 8** Clique *polynomial-time reduces to* $EnvDes_D^{fast}$ *such that in the constructed instance of* $EnvDes_D^{fast}$*,* $|T| = |X| = c_1 = 1$*,* $c_2 = 2$*,* $|f| = 3$*, and* $|Q|$*,* $r$*, and* $|E|$ *are functions of* $k$ *in the given instance of* Clique*.*

**Proof:** Given an instance $\langle G = (V, E), k\rangle$ of Clique, construct an instance $\langle G', E_T, T, X, p_I, p_X, c_1, c_2\rangle$ of $\text{EnvDes}_D^{fast}$ as follows: Let $G'$ be a $2 \times k+1$ grid, $E_T = \{e_0, e_1, \ldots e_{|V|}, e_{F1}, e_{F2}\}$ consist of $|V|+3$ different types of freespace squares, $p_I = W_{1,1}$, and $X$ be a single square of type $e_{F2}$ positioned at $p_X = W_{2,1}$. Team $T$ will consist of a single FSR $c$ based on states $Q = \{q_0 = q_{U,0}, q_{U.1}, q_{U,2} \ldots q_{U,k}, q_{E,1,1}, q_{E,1,2}, \ldots, q_{E,1,k}, q_{E,2,2}, q_{E,2,3}, \ldots, q_{E,2,k}, \ldots q_{E,k,k}, q_F, q_{Err}\}$, $|Q| = k + k(k-1)/2 + 3$, with the following transitions:

1. For each $i$, $1 \leq i \leq k$, the set of transitions $\{\langle q_{U,i-1}, enval(v_j, (0, i)) \wedge enval(v_j, (0, l)), *, stay, q_{Err}\rangle \mid 1 \leq j \leq |V|$ and $1 \leq \text{l} < \text{i}\}$ and transition $\langle q_{U,i-1}, *, *, stay, q_{U,i}\rangle$;
2. A transition $\langle q_{U,k}, *, *, stay, q_{E,1,1}\rangle$;
3. For each $i$, $1 \leq i < k$, the sets of transitions $\{\langle q_{E,i,j-1}, enval(e_u, (0, i)) \wedge enval(e_v, (0, j)), *, stay, q_{E,i,j}\rangle \mid i < j \leq k$ and (u, v) $\in$ E$\}$ and $\{\langle q_{E,i,j-1}, enval(e_v, (0, i)) \wedge enval(e_u, (0, j)), *, stay, q_{E,i,j}\rangle \mid i < j \leq k$ and (u, v) $\in$ E$\}$ and transition $\langle q_{E,i,k}, *, *, stay, q_{E,i+1,i+1}\rangle$; and
4. Transitions $\langle q_{E,k,k}, enval(e_0, (0, 0)), *, goEast, q_F\rangle$ and $\langle q_F, enval(q_{F1}, (0, 0)), e_{F2}, stay, q_F\rangle$.

The transitions in (1) ensure that no two square-types of the $k$ squares $E_{1,2}, E_{1,3}, \ldots W_{1,k+1}$ are the same and the transitions in (3) ensure that each pair of squares in these $k$ squares has types that correspond to an edge $e \in E$; together, these transitions ensure that theses $k$ squares encode a clique of size $k$ in $G$. Finally, set $c_1 = 1$ and $c_2 = 2$. Note that this instance of $\text{EnvDes}_D^{fast}$ can be constructed in time polynomial in the size of the given instance of Clique.

If there is a $V' \subseteq V$ such that $V'$ is a clique of size $k$ in $G$, construct an environment $E$ as follows:

1. $E_{1,1}$ has square-type $e_0$;

2. $E_{1,j+1}$, $1 \leq j \leq k$, has square-type corresponding to the $j$th vertex in $V'$;

3. $E_{2,1}$ has square-type $e_{F1}$; and

4. All remaining environment-squares have square-type $e_0$.

Observe that $E$ will allow the lone FSR in $T$ to progress from $p_I$ to create $X$ at $p_X$; moreover, as at any point in this progress at most one transition in that FSR is enabled, the operation of that FSR in $E$ is deterministic. In order to do this, $c$ must execute $k + k(k-1)/2 + 2$ transitions. As $k + k(k-1)/2 + 2 < 2k^2 + 2 < 4k^2 + 4 < (2k+2)^2 = c_1|E|^{c_2}$ when $c_1 = 1$ and $c_2 = 2$, this means that this construction task is $(c_1, c_2)$-completable in both the synchronous and asynchronous senses wrt $T$ and $p_I$ when $c_1 = 1$ and $c_2 = 2$.

Conversely, suppose there is an environment $E$ such that when $c$ is started at $p_I$, the task of constructing $X$ and $p_X$ is $(c_1, c_2)$-completable wrt $T$ and $p_I$. This completability, by the rules of FSR operation given in Section 2, implies that the operation of $c$ in $E$ is deterministic. By the structure of this FSR, it can only move right from $E_{1,1}$ to $E_{2,1}$ if it can change state from $q_0$ to $q_F$, $E_{1,1}$ has square-type $e_0$, and $E_{2,1}$ has square-type $e_{F1}$. However, this can only happen if environment-squares $E_{1,2}$ through $E_{1,k+1}$ in the westmost column of $E$ have square-types which correspond to a clique of size $k$ in $G$.

To complete the proof, note that in the constructed instance of $\text{EnvDes}_D^{fast}$, $|T| = |X| = c_1 = 1$, $c_2 = 2$, $|f| = 3$, $|Q| = k + k(k-1)/2 + 3$, $r = k$, and $|E| = 2k + 2$. ∎

Observe that the converse part of the proof of correctness of the preceding reduction can be tightened. Namely, one can show that the operation of $c$ in $E$ is deterministic by noting that courtesy of the structure of $c$, any $E$ that allows $c$ to create $X$ at $p_X$ is such that at most one transition is enabled in $c$ at any time. This tightening can also be done for the following reduction.

**Lemma 9** CLIQUE *polynomial-time reduces to* $\textit{EnvDes}_D^{fast}$ *such that in the constructed instance of* $\textit{EnvDes}_D^{fast}$, $|T| = |X| = |Q| = c_1 = c_2 = 1$ *and* $r$ *and* $|E|$ *are functions of* $k$ *in the given instance of* CLIQUE.

**Proof:** The reduction below is a modification of that given in the proof of Lemma 8. Given an instance $\langle G = (V, E), k\rangle$ of CLIQUE, construct an instance $\langle G', E_T, T, X, p_I, p_X, c_1 = c_2\rangle$ of $\text{EnvDes}_D^{fast}$ as follows: Let $G'$, $E_T$, $X$, $p_I$, and $p_X$ be as in the proof of Lemma 8, and team $T$ consist of a single FSR based on $|Q| = 1$ states with the transitions $\langle q_0, f, *, goEast, q_0\rangle$ and $\langle q_0, enval(e_{F1}, (0, 0)), e_{F2}, stay, q_0\rangle$ such that $f$ is the conjunction of (1) the negations of the non-default transition formulas in the transitions described in item 1 and (2) the non-default transition formulas of the transitions described in item 3 of the transition-list for $c$ in the proof of Lemma 8. Finally, set $c_1 = c_2 = 1$. Note that this instance of $\text{EnvDes}_D^{fast}$ can be constructed in time polynomial in the given instance of CLIQUE. The proof of correctness of this reduction is essentially the same as that given in the proof of Lemma 8. As $c$ started at $p_I$ must execute 2 transitions to construct $X$ at $p_X$ and $2 < (2k + 2) = c_1|E|^{c_2}$ when $c_1 = c_2 = 1$, this construction task is $(c_1, c_2)$-completable in both the synchronous and asynchronous senses wrt $T$ and $p_I$ when $c_1 = c_2 = 1$. To complete the proof, note that in the constructed instance of $\text{EnvDes}_D^{fast}$, $|T| = |X| = |Q| = c_1 = c_2 = 1$, $r = k$, and $|E| = 2k + 2$. ∎

**Result A**: If either $\text{ContEnvVer}_{syn}$ or $\text{ContEnvVer}_{asy}$ is polynomial-time tractable then $P = NP$.

**Proof:** The reduction in Lemma 3 in conjunction with the $PSPACE$-hardness of CDTMC proves that ContEnvVer is $PSPACE$-hard — that is, by the transitivity of polynomial-time reductions, every problem in $PSPACE$ reduces to ContEnvVer. Hence, if ContEnvVer is polynomial-time tractable then every problem in $PSPACE$ is also polynomial-time tractable and $P = PSPACE$. As $NP$ is contained in $PSPACE$, by an analogous argument, if ContEnvVer is polynomial-time tractable then $P = NP$. Given the attention focused on this latter conjecture [5, 6], it is this conjecture that we use to phrase our result. To complete the proof, note that in the reduction in Lemma 3, $|T| = 1$ in the constructed instance ContEnvVer, in which case synchronous and asynchronous team operation are equivalent, ∎

The observation in the proof above that having a team of size one renders synchronous and asynchronous team operation equivalent will be used in the proofs of many of the results given below.

**Result B**: If either $\text{ContDes}_{syn}^{fast}$ or $\text{ContDes}_{asy}^{fast}$ is polynomial-time tractable then $P = NP$.

**Proof:** The $NP$-hardness of $\text{ContDes}^{fast}_{syn,D}$ and $\text{ContDes}^{fast}_{asy,D}$ follows from the $NP$-hardness of DOMINATING SET and the reduction in Lemma 5 in which $|T| = c_1 = c_2 = 1$. The result then follows from Lemma 1. ▮

**Result C**: If either $\text{EnvDes}^{fast}_{syn}$ or $\text{EnvDes}^{fast}_{asy}$ is polynomial-time tractable then $P = NP$.

**Proof:** The $NP$-hardness of $\text{EnvDes}^{fast}_{syn,D}$ and $\text{EnvDes}^{fast}_{asy,D}$ follows from the $NP$-hardness of 3SAT and the reduction in Lemma 7 in which $|T| = c_1 = 1 = c_2 = 1$. The result then follows from Lemma 1. ▮

**Result D**: If $P = BPP$ and either $\text{ContEnvVer}_{syn}$, $\text{ContEnvVer}_{asy}$, $\text{ContDes}^{fast}_{syn}$, $\text{ContDes}^{fast}_{asy}$, $\text{EnvDes}^{fast}_{syn}$, or $\text{EnvDes}^{fast}_{asy}$ is polynomial-time tractable by probabilistic algorithms which operate correctly with probability $\geq 2/3$ then $P = NP$.

**Proof:** It is widely believed that $P = BPP$ [10, Section 5.2] where $BPP$ is considered the most inclusive class of decision problems that can be efficiently solved using probabilistic methods (in particular, methods whose probability of correctness is $\geq 2/3$ and can thus be efficiently boosted to be arbitrarily close to one). Hence, if any of $\text{ContEnvVer}_{syn}$, $\text{ContEnvVer}_{asy}$, $\text{ContDes}^{fast}_{syn}$, $\text{ContDes}^{fast}_{asy}$, $\text{EnvDes}^{fast}_{syn}$, or $\text{EnvDes}^{fast}_{asy}$ has a probabilistic polynomial-time algorithm which operates correctly with probability $\geq 2/3$ then by the observation on which Lemma 1 is based, their corresponding decision versions also have such algorithms and are by definition in $BPP$. However, if $BPP = P$ and we know that all these decision versions are $NP$-hard by the proofs of Results A, B, and C, this would then imply by the definition of $NP$-hardness that $P = NP$, completing the result. ▮

**Result E**: If $\langle |T|, |f|, r, |E|, |X| \rangle$-$\text{ContEnvVer}_{syn}$ and -$\text{ContEnvVer}_{asy}$ is fp-tractable then $FPT = W[1]$.

**Proof:** The $W[SAT]$-hardness of $\langle |T|, |f|, r, |E|, |X| \rangle$-$\text{ContEnvVer}_{syn}$ and -$\text{ContEnvVer}_{asy}$ when $|T| = |f| = |X| = 1$ and $r = 0$ follows from the $W[SAT]$-hardness of $\langle k \rangle$-CDTMC when $|x| = 0$ [2, Theorem 32] and the reduction in Lemma 3. The result then follows from the fact that if a $W[SAT]$-hard problem is fp-tractable then $W[SAT]$ collapses to $FPT$, which (as $W[1] \subseteq W[SAT]$) would mean that $FPT = W[1]$. ▮

**Result F**: If $\langle |Q|, |E_T|, |X| \rangle$-$\text{ContEnvVer}_{syn}$ is fp-tractable then $P = NP$.

**Proof:** The $NP$-hardness of $\text{ContEnvVer}_{syn}$ when $|Q| = |X| = 1$ and

$|E_T| = 9$ follows from the $NP$-hardness of 3SAT and the reduction in Lemma 4. The result then follows from Lemma 2. ∎

**Result G**: If either $\langle |T|, |Q|, d, |f|, r, |X| \rangle$-ContDes$^{fast}_{syn}$ or -ContDes$^{fast}_{asy}$ is fp-tractable then $FPT = W[1]$.

**Proof:** The $W[2]$-hardness of $\langle |T|, |Q|, d, |f|, r, |X| \rangle$-ContDes$^{fast}_{syn}$ and -ContDes$^{fast}_{asy}$ when $|T| = |Q| = |f| = |X| = c_1 = c_2 = 1$ and $r = 0$ follows from the $W[2]$-hardness of $\langle k \rangle$-Dominating set and the reduction in Lemma 5. The result then follows from the fact that if a $W[2]$-hard problem is fp-tractable then $W[2]$ collapses to $FPT$, which (as $W[1] \subseteq W[2]$) would mean that $FPT = W[1]$. ∎

**Result H**: If either $\langle |T|, |f|, |E_T|, |X| \rangle$-EnvDes$^{fast}_{syn}$ or -EnvDes$^{fast}_{asy}$ is fp-tractable then $P = NP$.

**Proof:** The $NP$-hardness of EnvDes$^{fast}_{syn,D}$ and EnvDes$^{fast}_{asy,D}$ when $|T| = |X| = c_1 = 1$, $c_2 = 3$, and $|f| = |E_T| = 5$ follows from the $NP$-hardness of 3SAT and the reduction in Lemma 6. The result then follows from Lemma 2. ∎

**Result I**: If either $\langle |T|, |Q|, |E_T|, |X| \rangle$-EnvDes$^{fast}_{syn}$ or -EnvDes$^{fast}_{asy}$ is fp-tractable then $P = NP$

**Proof:** The $NP$-hardness of EnvDes$^{fast}_{syn,D}$ and EnvDes$^{fast}_{aszy,D}$ when $|T| = |X| = |Q| = c_1 = c_2 = 1$ and $|E_T| = 5$ follows from the $NP$-hardness of 3SAT and the reduction in Lemma 7. The result then follows from Lemma 2. ∎

**Result J**: If either $\langle |T|, |Q|, |f|, r, |E|, |X| \rangle$-EnvDes$^{fast}_{syn}$ or -EnvDes$^{fast}_{asy}$ is fp-tractable then $FPT = W[1]$.

**Proof:** The $W[1]$-hardness of $\langle |T|, |Q|, |f|, r, |E|, |X| \rangle$-EnvDes$^{fast}_{syn}$ and -EnvDes$^{fast}_{asy}$ when $|T| = |f| = |X| = c_1 = 1$, $c_2 = 2$, and $|f| = 3$ follows from the $W[2]$-hardness of $\langle k \rangle$-Dominating set and the reduction in Lemma 8. ∎

**Result K**: If either $\langle |T|, |Q|, r, |E|, |X| \rangle$-EnvDes$^{fast}_{syn}$ or -EnvDes$^{fast}_{asy}$ is fp-tractable then $FPT = W[1]$.

**Proof:** The $W[1]$-hardness of $\langle |T|, |Q|, r, |E|, |X| \rangle$-EnvDes$^{fast}_{syn}$ and -EnvDes$^{fast}_{asy}$ when $|T| = |Q| = |X| = c_1 = c_2 = 1$ follows from the $W[2]$-

hardness of $\langle k \rangle$-Dominating set and the reduction in Lemma 9. ∎

**Result L**: $\langle |E|, |E_T| \rangle$-ContEnvVer$_{syn}$ is fp-tractable.

**Proof:** Let an environment-configuration consist of both the types of all environment-squares as well as the positions of all FSRs in $T$ in the environment. Consider the algorithm that enumerates all possible non-repeating sequences of environment-configurations starting with the members of $T$ at $p_I$ and ending with $X$ appearing at $p_X$ and checks each such sequence to see if it is valid, i.e., if each environment-configuration can legally evolve into its immediately following environment-configuration in the sequence. We need only consider non-repeating sequences because a team of deterministically-operating synchronous FSR can evolve into at most one following environment-configuration from any given environment-configuration. Hence, if an environment-configuration is repeated without encountering a configuration in which $X$ does not appear at the appropriate location then that sequence can never contain such a configuration. We know that $|T| \leq |E|$, as there can be at most one FSR from $T$ in any square in $E$. The total number $l$ of environment-configurations (and hence the longest possible non-repeating configuration sequence) is $\leq |E_T|^{|E|} \left( \begin{array}{c} |E| \\ |T| \end{array} \right) \leq |E_T|^{|E|}|T|^{|E|} \leq |E_T|^{|E|}|E|^{|E|}$. The number of sequences of non-repeating sequences is therefore $\leq \sum_{i=2}^{l} \left( \begin{array}{c} l \\ i \end{array} \right) i! \leq \sum_{i=2}^{l} i^l i^i \leq \sum_{i=2}^{l} l^l l^l \leq l \times l^{2l} = l^{2l+1}$. As each such sequence can be checked in time polynomial in $l$ and size of the rest on the input instance of ContEnvVer, the algorithm above shows that $\langle |E|, |E_T| \rangle$-ContEnvVer is fp-tractable. ∎

**Result M**: $\langle |Q|, |E|, |E_T| \rangle$-ContDes$^{fast}_{syn}$ is fp-tractable.

**Proof:** ContDes is trivially solvable by the algorithm that considers all possible FSR teams with the required specifications and verifies for each team that it can create $X$ at position $p_X$ in at most $c_1|E|^{c_2}$ timesteps. We already know from Result L that the verification of any team relative to a given environment can be done in fpt time relative to parameter-set $\{|E|, |E_T|\}$; as this verification considers all possible sequences of execution, it can easily restricted to ensure that completion occurs in the required number of timesteps. Observe that all remaining aspects of such a team-specification in the input except $|Q|$ are also upper-bounded by functions of $|E|$ and $|E_T|$:

- $|T| \leq |E|$ (as at most one member of $S$ can be present in each square of $E$).

- $d \leq (|E_T|+1)^{|E|}$ (as our FSRs operate deterministically and the maximum number of formulas in transitions originating from any state $q$ in an FSR that can evaluate to $True$ such that only one of those formulas evaluates to $True$ is the number of possible configurations of $E$ that can be observed by that FSR, which is $(|E_T|+1)^{|E|}$).

- $|f| \leq (|E_T|+1)^{|E|}(2|E|+2)$ (as each formula effectively encodes some subset of the configurations of $E$ observable from an FSR, each formula can be recoded as a disjunction of conjunctive formulas describing these environment-configurations and the subset encoded in that formula can have at most $(|E_T|+1)^{|E|}$ environment-configurations).

- $r \leq |E|-1$ (as the furthest an FSR need see is to all squares in $E$, with the maximum occurring in linear environments like those posited in Lemma 3).

- $|X| \leq |E|$ (as a structure can only consist of squares in $E$).

Hence, as the running time of the algorithm described above is upper-bounded by some function of $|E|$, $|E_T|$, and $|Q|$ times some polynomial of the aspects of the input, this algorithm shows that $\langle |E|, |E_T|, |Q| \rangle$-ContDes is fp-tractable. ▮

**Result N**: $\langle |E|, |E_T| \rangle$-EnvDes$^{fast}_{syn}$ is fp-tractable.

**Proof:** Consider the algorithm that generates all possible environments of size $|E|$ relative to $E_T$ and for each such environment, determines whether or not the given team $T$ started at $p_I$ in that environment can create $X$ at $p_X$ in at most $c_1|E|^{c_2}$ timesteps. As the number of such environments is $|E_T|^{|E|}$ and the assessment of task completion in the required number of timesteps can be done in fpt time for parameter-set $\{|E|, |E_T|\}$ (by an argument analogous to that given in the proof of Result M), the algorithm above shows that $\langle |E|, |E_T| \rangle$-EnvDes is fp-tractable. ▮

## References


[1] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: From natural to artificial systems. Oxford University Press (1999)

[2] Cesati, M.: Structural aspects of parameterized complexity. Ph.D. thesis, Universitá degi Studi di Roma 'La Sapienza' (1996)

[3] Clementi, A.E.F., Rolim, J.D.P., Trevisan, L.: The computational complexity column: Recent advances towards proving $P = BPP$. Bulletin of the European Association for Theoretical Computer Science **64**, 96–103 (1998)

[4] Downey, R., Fellows, M.: Fundamentals of Parameterized Complexity. Springer, Berlin (2013)

[5] Fortnow, L.: The Status of the P Versus NP Problem. Communications of the ACM **52**(9), 78–86 (2009)

[6] Garey, M.R., Johnson, D.S.: Computers and Intractability. W.H. Freeman (1979)

[7] Gauci, M., Chen, J., Li, W., Dodd, T.J., Groß, R.: Self-organized aggregation without computation. The International Journal of Robotics Research **33**(8), 1145–1161 (2014)

[8] Theraulaz, G., Gautrais, J., Camazine, S., Deneubourg, J.L.: The formation of spatial patterns in social insects: from simple behaviours to complex structures. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **361**(1807), 1263–1282 (2003)

[9] Wareham, T., Vardy, A.: Viable algorithmic options for designing reactive robot swarms. ACM Transactions on Autonomous Adaptive Systems (to appear)

[10] Wigderson, A.: P, NP and mathematics — A computational complexity perspective. In: Proceedings of ICM 2006: Volume I, pp. 665–712. EMS Publishing House, Zurich (2007)