Controlling Tensegrity Robots through Evolution: a review of Iscen *et al.* 2013[1]

Amy Watson

University of Adelaide

Abstract. Iscen *et al.* presented their work on "Controlling Tensegrity Robots through Evolution"[1] as part of the real-world applications track of the 2013 Genetic and Evolutionary Computation Conference (GECCO'13). Structurally, tensegrity robots have many advantageous features, however, these same attributes make them difficult to control. This essay starts with an outline of previous approaches to this problem, which leads to a comparison with Iscen *et al*'s methodology, and concludes with a critical review of their paper. Iscen *et al*'s contribution to the field of tensegrity control is impressive, but there is still a way to go before the super-ball bot is rolling over the surface of Mars.

Keywords: evolutionary computation, robotics, tensegrity

1 Introduction

Tensegrity robots are actuated tensegrity structures (abbreviation of *tensile integrity*); systems of cables connecting axially-loaded rods [2]. This arrangement ensures the components experience either pure tension (cables) or pure compression (rods) and resist external forces by distributing them via multiple axial load paths [2]. The lack of shearing or bending forces is a unique feature of tensegrity structures which allows construction of light weight and robust machines. Obviously these are desirable characteristics for robots and the work presented by Iscen *et al* was supported by both the Idaho Space Grant Consortium and the NASA Innovative Advanced Concepts (NIAC) program with the aim of building landing and mobility systems for planetary exploration [1]. Unfortunately, these same characteristics also make tensegrities difficult to control.

This essay will begin with an outline of previous methods used to develop control systems for tensegrity robots. This will be followed by a review of the work presented by Iscen *et al*, including a discussion of the novelty of their approach and a comparison with related research. Finally, their paper will be critiqued and suggestions for future and alternative approaches to developing control systems for tensegrity robots will be discussed.

2 Background

Robots do the "dirty, dull, and dangerous" tasks that humans can not or will not do [3]. Desirable traits of any machine include being light-weight, robust to failures, and impact tolerant. These are all properties of even static tensegrities as they are composed of multiple, redundant components; the rods do not need to cope with shearing loads and therefore can be relatively light-weight [1]. Additional advantageous features for robots designed for locomotion include being energy efficient, capable of a variety of gaits, and to a certain extent, be able to conform to their changing environment while sustaining and causing minimal damage [4]. All three are inherent qualities of tensegrity robots, which has lead to them being used in the design and deployment of space vehicle components such as extensible masts [6], reflector antennae [5], and satellite reflectors [7].

The shape of a tensegrity structure is a function of the length of its cables and each combination of lengths will result in a stable form due to the tensile and compressive forces reaching equilibrium [8]. Therefore, research into mobile tensegrities has focussed on periodically shortening and lengthening the cables to produce movement by translating and/or rotating the rods in the system [9, 10]; much like animals contract and relax muscles to flex and extend joints. Masic and Skelton proposed the use of an open-loop control system to produce a backward-travelling wave along the length of a *tensegrity worm* which allowed the friction with its surroundings to propel the structure forward [11].

Several studies have been focussed on developing control systems for tensegrities with the minimum number of rods required for locomotion. Paul *et al* used a genetic algorithm (GA) approach to evolve optimal gaits for a 3-strut tensegrity (TR-3); optimal gaits maximise the distance travelled forward in a 10 second simulation [4]. In 2006, the same GA methodology was applied to a 4-strut tensegrity (TR-4) [12], with the addition of tests for gait production with one or two missing actuators and the implementation of the TR-3 design. Rovira and Tur developed a simulator for their TR-3 and reduced the possible states of each cable to 3 for each time step; no change, lengthen or shorten by a predetermined amount [14]. Therefore, they only needed to calculate 3^9 possible combinations to find the best set of cable-states for following a pre-determined path; the author's themselves note that this is only possible for simple tensegrities [14]. Shibat and Imuta used a procedural method to calculate the position and rotation vectors of the rods in a 6-strut tensegrity (TR-6) required to produce forward rolling [15]. The results were used to determine the forces required for simulation (in Matlab) and subsequently to control a pneumatically actuated robot [16, 15].

Iscen *et al* first attempted to evolve a controller for a TR-6 using a genetic algorithm with mutation only [13][1]; fitness of each control system was based on the distance travelled during a 60 second simulation. The authors simplified the problem by dividing the tensegrity cables into 8 groups of 3 based on the symmetry of the robot, and evolved a single length change function (sinusoidal wave function) for each group. The differences between this implementation and the experiments presented in the paper under review were that each cable was controlled individually and, in one case, the inclusion of a simple 1-point crossover

[1] in the paper Iscen *et al* referred to the algorithm as "agent learning"

[1]. The next section compares Iscen *et al*'s latter methodology with Paul *et al*'s approaches for developing control systems for tensegrity robots.

3 Comparison with Previous Approaches

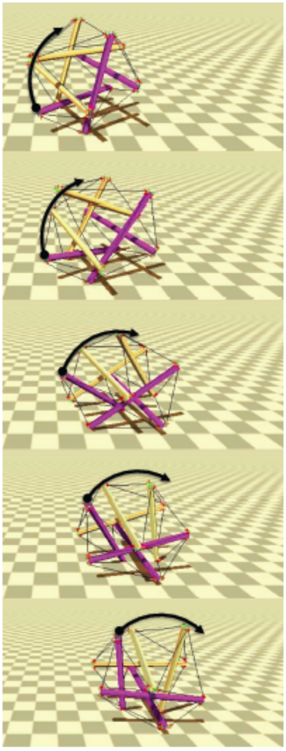

Fig. 1: Tensegrity dynamics. Figure 4 of Iscen *et al*[13]

Figure 1 shows the desired outcome of the evolved control system; a 6-strut tensegrity rolling forward due to changes in cable lengths. Iscen *et al* chose to use a generational genetic algorithm for their evolutionary strategy, as did Paul *et al* for their evolution of TR-3 and TR-4 controllers [4, 12]. The objectives for all three papers were the same, to maximise the distance travelled by the tensegrity. In addition, both used genome strings to represent the candidate solutions, therefore the use of GA for these optimisation problem is appropriate [17]. Although the tensegrities used in each case had different numbers of struts, one of the benefits of using EA over say hand-coding, is because it is scalable. Therefore, a direct comparison can be made between these studies based on genotypes, phenotypes, variation and selection algorithms, fitness, as well as the parameters of the EA itself (number of generations, population size, etc.).

Table 1: Comparison of Evolutionary Algorithms

	Iscen *et al* [1]					
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Paul *et al* [12]
Genotype	string of 96 float parameters $C_1, A_1, \omega_1, \phi_1, \ldots,$ $C_{24}, A_{24}, \omega_{24}, \phi_{24}$		24 agents each with 4 parameters for each agent i, $C_i, A_i, \omega_i, \phi_i$			coefficient p + triplet per cable $[a_i, o_i, d_i]$ all floats $[0, 1]$
	where, C_i = the centre position A_i = amplitude ω_i = angular frequency ϕ_i = phase					where, p = period of gait a_i = amplitude o_i = phase d_i = duration
Phenotype	length of each cable is a function of simulation time t					
	$l_i(t) = C_i + A_i sin(\omega_i t + \phi_i)$					$\lambda = \lambda_{min} + \lfloor p(\lambda_{max} - \lambda_{min} \rfloor$ = period of gait cycle $\alpha_i = \lfloor a_i \alpha_{max} \rfloor$ = amplitude $\phi_i = \lfloor o_i \lambda \rfloor$ = phase $\delta_i = \lfloor d_i \lambda \rfloor$ = duration if t mod $\lambda = \phi_i$, set $t_i^o = t$ $l_i(t) = \begin{cases} l_i^o - \alpha_i l_i^o & t_i^o < t < t_i^o + \delta_i \\ l_i^o & t > t_i^o + \delta_i \end{cases}$ where, l_i^o = original cable length t_i^o = time of onset in each cycle
Fitness	simulated Euclidean distance travelled					simulated distance travelled in y-direction
Agent fitness	as above		gen. average	leniency	hist. average	as above
Generations	unclear, the number of evaluations was reported between 30,000 and 45,000					200
Population	10					200
Parents	5					100
Offspring	5					100
Selection	elitism					tournament
Cross-over	none	1-point, randomly selected	none	none	none	1-point, for 50 randomly selected parents
Mutation	uniform random for each parameter					uniform random with rate $n = \frac{genomelength}{11}$

Table 1 summarises the parameters used in each paper and one of the obvious differences is that Iscen *et al* performed five experiments each using an

EA with slightly different settings. This was done to determine the EA that could consistently find the best control system, that is, the one in which the simulated tensegrity travelled the furthest. In Experiments 1 and 2, Iscen *et al* used what they termed a *centralised* evolutionary algorithm [1]. In these first two experiments the genotype was a string of 96 float values containing the parameters for all 24 cables of the tensegrity. Using this nomenclature, Paul *et al* also used a *centralised* EA strategy, including a global parameter used to determine the period of the gait; between $\lambda_{min} = 200$ and $\lambda_{max} = 500$ milliseconds [12]. The difference in phenotype is discussed below. The advantage of centralised approach is the implicit evolution of a global solution, the disadvantage is the increase in the search space with the addition of each cable.

In experiments 3,4 and 5, Iscen *et al* used *cooperative co-evolution* algorithms (CCAE) in an attempt to exploit the inherent decentralised nature of tensegrity robots to decompose the objective into sub-objectives with smaller search spaces. In CCAE, multiple sub-populations, one for each sub-component, are evolved and sampled to construct a single complete solution[17]. This requires a method to assign global fitness value to each agent and, in order to truly represent symbiosis in nature, an evolutionary framework that promotes co-adaptation. Iscen *et al* trialled three different methods to satisfy the former requirement; generational average, leniency, and historical average (see Iscen *et al* [1] for details). However, they did not implement any method of keeping particularly good combinations of sub-solutions together; referred to as *linkage flags* by Bull and Fogarty [18]. Without such a system, Iscen *et al*'s CCAE experiments would be more appropriately called *multi-agent learning*.

The phenotype used by both Iscen *et al* [1] and Paul *et al* [12] is a cyclic control function to alter the lengths of the tensegrity cables over time. The big difference between them, apart from lack of detail in Iscen *et al*'s paper, is that with Paul *et al*'s version, a cable can be at rest for some period of the gait. It is difficult to compare the effect of this due to the different number of struts in the tensegrity structures used in each paper, but having a method of encoding component inactivity during motion seems logical. Instead of presenting their hand-coded solution for comparison, Iscen *et al*'s paper would have been greatly enhanced by using their best EA with Paul *et al*'s genotype and phenotype.

Unfortunately, it is unclear whether the number of evaluations reported in Iscen *et al*, corresponds to the number of generations or a property of the physics engine. Therefore, the implementation cannot be compared with Paul *et al* using this characteristic. However, the generational properties can be compared; the size of the current, parent and offspring population used by Paul *et al* is 20 times that of Iscen *et al*. The overall effect of smaller population sizes is to reduce the maximum genotypic diversity possible in each generation, which may limit exploration of the search space [17]. In addition, the use of elitism increases the chances of the optimisation converging on local optima, further reducing the region of search space evaluated. In contrast, with tournament selection, as used by Paul *et al*, there is ma chance that a solution ranked low in the population could still be selected if it is included in a tournament with relatively lower ranked individuals; subsequent mutation can then lead to an even better controller evolving [17]. This helps maintain diversity in the population and reduces the chances of being stuck at local optima.

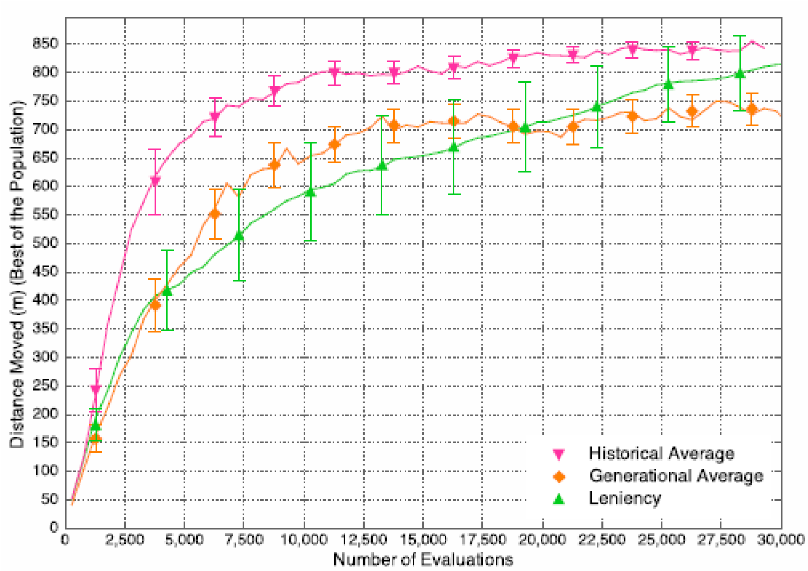

Fig. 2: Cooperative co-evolutionary algorithm evaluation methods. Figure 4 of Iscen *et al* [1]

Iscen *et al* used cross-over only in experiment 2 [1]; single-point cross-over at a random position along the 96 parameter string. The mutation algorithm was used for all experiments to create offspring from the best of the previous generation, which replaced the worst parents. Varying each parameter uniformly at random is basically a random search, which may help offset the potential to converge on local optima by using elitism selection, and maintain population diversity. Paul *et al* use a mutation rate calculated to vary one parameter on average, this reduces the change from the original genotype, however the other parameters of the EA promote diversity, and overall their approach is preferable as the selection pressure can be tuned via the tournament selection parameters [17]. Iscen *et al* presented the standard deviation of fitnesses for 10 runs of each EA experiment as error bars on their results graphs (see Figure 2 for an example). They interpreted small error bars as a sign of the particular EA giving consistent results, which is feasible. However, it would be interesting if they had also presented some measure of population diversity, to determine the extent convergence on local optima.

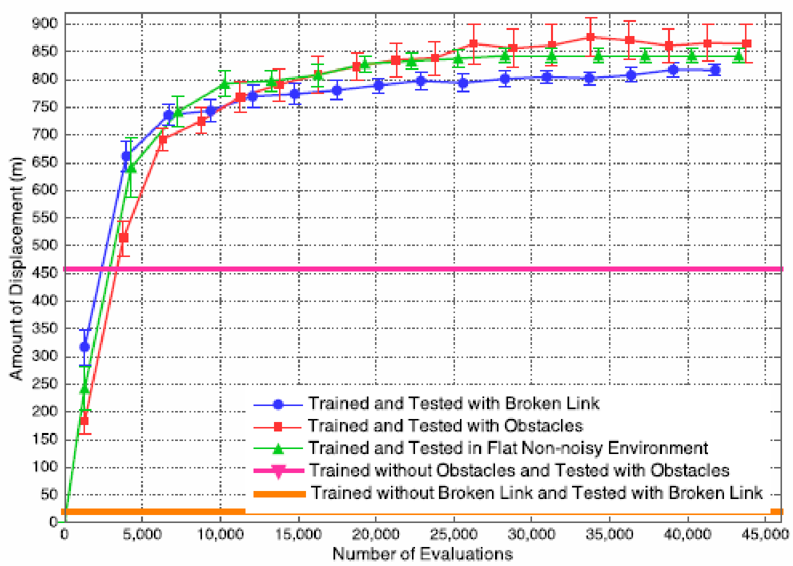

Fig. 3: Robustness tests with broken cable and obstacles. Figure 8 of Iscen *et al* [1]

Iscen *et al*'s paper included additional experiments focussed on tensegrity controller robustness. They tested the effectiveness of pre-adaptation to actuation noise, cable failure and obstacles in the environment. the effectiveness of pre-adaptation for enhancing the robustness of the evolved controller. While Paul *et al* tested cable failure, Iscen *et al*'s tests for the effect of actuation noise and obstacles in the environment appears to be novel. Figure 3 shows the results of the broken cable and obstacles experiments. In these experiments, the authors compared the performance of controllers evolved with and without the disruption, and were able to show the value of pre-adaptation for this type of application; controllers for space vehicles in an unpredictable environment. The only combination they did not include was to train with, but test without the disruptions. Without this result it is difficult to determine if pre-adaptation for robustness, is detrimental to the performance of the tensegrity under normal conditions.

4 Critique

The inclusion of experiments in pre-adaptation to an unpredictable environment by Iscen *et al* was interesting. In nature there are many examples of taxa that are pre-adapted to an environment contributes to their success. For example, true sea snakes (Elapidae: Hydrophiini) evolved from live-bearing terrestrial elapids and account for 90% of marine reptile species, the remainder of which are egglaying [21, 22]. Three situations were included in this paper, and their current work includes tests in undulating and steep environments (Iscen,A, personal communication, 2013). They tested for actuation noise, obstacles and cable failure. Only the latter has been trialled in other TR-6 controller research; Paul *et al* included failure tests for one and two cables [12].

The comparisons of experimental results for different evolution algorithms and environmental conditions were well presented. Contrasting these with the hand-coded solution may be biased as it was produced as a demonstration of the difficulty of hand-coding, not as a benchmark. The results presented by Iscen *et al* in their previous paper[13] showed that even multi-agent and single-agent learning using the same *simplified* model of 8 control groups (all 3 cables in each group are set to the same length) could not produce a control system capable of exceeding 100m displacement under the same simulation conditions. It would have been more informative to see a comparison with a procedurally developed controller (e.g. Shibata and Imuta [15]), or one evolved using an alternative genotype/phenotype representation (e.g. Paul *et al*[4, 12]).

The primary aim of any research article is communication. One of my major criticisms of Icsen *et al*'s paper [1] is the lack of detail regarding the parameters tuned in the optimisation; their experiments could not be repeated using the information presented paper. This lack of detail is in stark contrast to the work presented by Paul *et al* [4, 12], which gives a comprehensive listing and explanation of the use of their parameters. The same is true for the settings used in their simulation environment. The list of missing information includes the material properties of the rods and cables, mass and shape of the rods, spring stiffness and damping coefficients. All of this data would be required to repeat their simulation for validation and/or comparison in future studies and would allow the reader to make an informed judgement about the applicability of their experimental results.

Iscen *et al* chose to use the Bullet Physics Engine [19] for their project. While a comprehensive comparison of physics engines is beyond the scope of this essay, one of the issues the authors encountered was that the rod length was required to be 10m in order for precise calculations [1]. On the other hand, the Open Dynamics Engine(ODE)[20] used by Paul *et al* enabled them to model 1m tensegrity rods [4, 12], a size much closer to Iscen *et al*'s physical robot dimensions. ODE is still freely available under an open source license and it is unclear from the paper, why the authors chose the Bullet Physics Engine over ODE.

Another issue with the simulation is the central payload representing controls etc. This may or may not be realistic if the robot has visual sensors, however for motors and equipment to drive the robot as *landing and mobility* devices this model may be suitable. One of the aims of the simulation was to gather requirements for existing and novel cable actuation methods [1]. The authors deliberately used an abstract model of actuation for this purpose. On the one hand having an abstract model of actuation could produce unrealistic requirements (especially with 10m rods). On the other hand this could allow novel mechanical actuator to be created from the data extracted about the cable length change requirements.

5 Conclusion

One approach to tensegrity control system evolution that has potential is exploitation of the inherent symmetry and modularity of the structure. Iscen *et al* used this property to simplify the structure into 8 control groups [13] and even noted that the fastest gait was forward roll [1] implicitly defining a left and right side, but did not take full advantage of this characteristic. The structure is bio-inspired, the optimisation method is bio-inspired, the missing piece is taking inspiration from the brain-body model [23]. Clune *et al* contrasted direct encoding of system modularity (grouping) with HyperNEAT, an indirect encoding, to evolve artificial neural networks for several problems, including control of a simulated quadruped robot [24]. Clune *et al*'s results and conclusions regarding EC combined with lifetime learning (HybriID)[24] seem to be a path worth pursuing for tensegrity robots, especially for those to be deployed in space.

Iscen *et al* have produced a well written paper about an exciting real-world application and have achieved their stated aims of evolving a control system for a mobile 6-strut tensegrity. Their inclusion of experiments in pre-adaptation for environmental disturbances is intriguing and thought-provoking. However, the lack of detail regarding simulation parameters is disappointing and makes it difficult to compare their results with past (and future) studies in this area. The use of simulation to direct their experimentation with real tensegrity robots gives Iscen *et al* an excellent framework for developing a deployable tensegrity. However, more research is required into alternative strategies of controller evolution for tensegrity robots.

References

1. Iscen, A., Agogino, A., SunSpiral, V., Tumer, K.: Controlling Tensegrity Robots through Evolution. In: Proceedings of the fifteenth annual conference on Genetic and Evolutionary Computation, GECCO'13, 1293–1300. ACM.(2013)
2. Fuller, R.B: Synergetics: explorations in the geometry of thinking, Macmillan, New York (1975)
3. Tilbury, D., Ulsoy, A.G.: A new breed of robots that drive themselves, In: Mechanical Engineering-. 28–30. CIME (2011)
4. Paul, C., Roberts, J.W., Lipson, H., Valero-Cuevas, F.: Gait Production in a Tensegrity Based Robot. In: 12th International Conference on Advanced Robotics, ICAR'05., 216–222. IEEE. (2005)
5. Knight, B.F.: Deployable antenna kinematics using tensegrity structure design. PhD diss., University of Florida, Gainesville, (2000).
6. Furuya, H.: Concept of deployable tensegrity structures in space applications, International Journal of Space Structures, 7, 143–151 (1992).
7. Tibert, A., Pellegrino, S. : Deployable tensegrity reflectors for small satellites, Journal of Spacecraft and Rockets, 39, 701–709, (2002).

8. Skelton, R.E., Adhikari, R., Pinaud, J., Chan, W., Helton, J.: An introduction to the mechanics of tensegrity structures. In: Proceedings of the 40th IEEE Conference on Decision and Control,4254–4259, (2001).
9. Tur, J.M.M., and Juan, S.H.: Tensegrity frameworks: Dynamic analysis review and open problems. Mechanism and Machine Theory, 44, 1–18, (2009).
10. Skelton, R.E., Pinaud, J.P., Mingori, D.L.: Dynamics of the shell-class tensegrity structures. Journal of The Franklin Institute, 338(2), 255–320 , (2001).
11. Masic, M., Skelton, R.E.: Open-loop control of class-2 tensegrity towers. In: Proceedings of SPIE Modeling, Signal Processing, and Control, 5383, 298–308, (2004).
12. Paul, C., Valero-Cuevas, F., Lipson, H.: Design and Control of Tensegrity Robots for Locomotion.In: IEEE Transactions on Robotics, 22(5), 944–957. (2006)
13. Iscen, A., Agogino, A., SunSpiral, V., Tumer, K.: Robust Distributed Control of Rolling Tensegrity Robot. In: The Autonomous Robots and Multirobot Systems (ARMS) Workshop at AAMAS, (2013).
14. Rovira, A.G., Tur, J.M.M.: Control and simulation of a tensegrity-based mobile robot. Robotics and Autonomous Systems, 57, 526–535, (2009).
15. Shibata, M., Imuta, R.: Rolling tensegrity driven by pneumatic soft actuators. In: 2012 IEEE International Conference on Robotics and Biomimetics, 198–204, (2012).
16. Koizumi, Y., Shibata, M., Hirai, S.: Rolling tensegrity driven by pneumatic soft actuators. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), 1988–1993, (2012).
17. Eiben, A.E., Smith, J.E.:, Introduction to evolutionary computing. Springer, Berlin, 2010.
18. Bull, L., Fogarty, T.C.: Horizontal gene transfer in endosymbiosis. In : Langton, C.G. , Shimohara, L. (eds). Proceedings of the 5th International Workshop on Artificial Life : Synthesis and Simulation of Living Systems (ALIFE-96), 6, 77–84, MIT Press, Cambridge, MA, (1997).
19. Bullet Physics Engine. http://www.bulletphysics.org/.
20. Open Dynamics Engine. http://www.ode.org/.
21. de Fraipont, M., Clobert, J., Barbault, R.: The Evolution of Oviparity with Egg Guarding and Viviparity in Lizards and Snakes: A Phylogenetic Analysis, Evolution, 50(1),391–400 (1996)
22. Sanders, K.L., Mumpuni, Lee, M.S.Y.: Uncoupling ecological innovation and speciation in sea snakes (Elapidae, Hydrophiinae, Hydrophiini). Journal of Evolutionary Biology, 23, 2685-2693 (2010)
23. Rieffel, J., Trimmer, B., Lipson, H.: Mechanism as Mind: What tensegrities and caterpillars can teach us about soft robotics. In: ALIFE, 506–512. (2008).
24. Clune, J. Stanley, K.O, Pennock, R.T., Ofrica, C.: On the performance of indirect encoding across the continuum of regularity. In: IEEE Transactions on Evolutionary Computation, 15(3), 346–367 (2011)