# Symbiosis, Synergy and Modularity: Introducing the Reciprocal Synergy Symbiosis Algorithm

Rob Mills and Richard A. Watson

Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, UK
{rmm05r, raw}@ecs.soton.ac.uk

**Abstract.** Symbiosis, the collaboration of multiple organisms from different species, is common in nature. A related phenomenon, symbiogenesis, the creation of new species through the genetic integration of symbionts, is a powerful alternative to crossover as a variation operator in evolutionary algorithms. It has inspired several previous models that use the repeated composition of pre-adapted entities. In this paper we introduce a new algorithm utilizing this concept of symbiosis which is simpler and has a more natural interpretation when compared with previous algorithms. In addition it achieves success on a broader class of modular problems than some prior methods.

## 1 Introduction

Collaboration between organisms of different types is called symbiosis [13], and such relationships are common in nature, particularly amongst prokaryotes [10]. When symbiotic associations persist over evolutionary timescales, the entities involved may become reproductively inseparable and Konstantin S. Merezhkovsky named this symbiogenesis [8]. Symbiogenesis is thought to have been responsible for several major evolutionary transitions [12], including ‘perhaps the most important and dramatic event in the history of life […] – the origin of eukaryotes’ [13, p51] when an archaebacterium and a eubacterium in a symbiotic relationship were genetically joined [11].

The genetic algorithm (GA) is one model of evolution and is useful as it allows us to investigate and understand the process of natural selection. It has distilled some key elements of evolution, such as the ever-repeating process of inherited variation followed by differential selection. However, one key aspect that is not captured by a model like the GA is macroevolution: processes that occur above the level of the species. Symbiosis and symbiogenesis are mechanisms that require a macroevolutionary model to explore successfully. The variation offered by symbiogenesis is qualitatively different from sexual crossover: it offers the union of two sets of pre-adapted genetic material. If this process is repeated, the complexity of the resultant entity can be far greater than that of the original entities involved. We call this process compositional evolution [22], and models that investigate this type of process may shed light on how modular encapsulation facilitates macroevolutionary trends such as the major evolutionary transitions.

Compositional evolution is of interest algorithmically as well as biologically, and this phenomenon has given rise to several algorithms that employ a symbiotic mechanism. Some have been aimed both at a computational biology audience (e.g. [21]) and problem solving in evolutionary computation (e.g. [5][4]). In this paper we consider a new model and investigate its utility upon optimization problems.

A symbiotic approach is particularly suited to hierarchically modular problems: the solution space is searched with entities of increasing complexity and stability, initially identifying optimal configurations for small subsets of variables, but progressively increasing the size of subsets by combining the optimal configurations for different variable subsets. However, we believe that employing symbiotic variation has the ability to exploit any modularity, hierarchically organized or not. We observe modular structures in a wide range of biological [7], physical [18] and engineered systems [1]. In such systems individual parts may be solved in partial or full isolation from other portions of the system [1][18]. In problems exhibiting this type of decomposability, algorithms that exploit this knowledge should outperform algorithms that do not exploit modular structures. The canonical example of an algorithm which does not exploit modularity is a simple hill climbing process which can only advance in the search space with gradual steps of improvement. There are a number of algorithms that attempt to exploit modular structures, including the genetic algorithm (GA) [20][14], cooperative coevolution [17], and estimation of distribution algorithms such as the Bayesian optimization algorithm [16]. Here we address the type of algorithm that is explicitly compositional as mentioned above. There are also a number of test problems in the literature that exhibit modular properties, including concatenated functions of unitation [3], and royal roads [15]. Here we utilize functions with hierarchical modularity [19], and parameterizable structural modularity [14].

In the remainder of this section we review the Symbiogenic Evolutionary Adaptation Model (SEAM) [21], identify limitations and propose possible methods to address certain limitations of this approach with a new computational algorithm.

SEAM is an early evolutionary algorithm inspired by symbiogenesis [21]. It differs from typical evolutionary algorithms (EA) in several aspects. These include the symbiogenic variation operator, an ecosystem of entities which are partially specified, and coevolved ecosystem templates that are used to facilitate evaluation. The main pool of genetic material represents an ecosystem of many different entities, each specifying only part of an overall solution. There is no source of new genetic material (i.e. no mutation), and the only variation operator is the symbiotic join of two entities. The main loop of the algorithm operates as follows: two entities are picked at random, and evaluated in a number of contexts to determine if the pair should make a permanent symbiogenic alliance. A version of Pareto-dominance (see [21]) is used to make these decisions. If this is the case, the two symbionts are removed from the ecosystem and replaced with the chimera. As this process is repeated, the average size of entities will increase until fully specified solutions are discovered.

SEAM has several limitations: the Pareto dominance mechanism employed to determine whether a join should be made is unnecessarily complicated. Assembling contexts from other entities within the ecosystem, although arguably natural, is (we will show) an unnecessary assumption. Finally, SEAM's performance in other problem domains is severely brittle: the utility of SEAM has only been demonstrated on a stylized hierarchical function, Hierarchical If-and-only-If (HIFF) [19].

Following [21] we develop a new symbiosis-inspired algorithm where individuals repeatedly test and make pairwise joins to compose larger and larger entities, resulting in problem solutions. As in prior work, we are also interested in understanding what symbiosis can offer us algorithmically, when compared with mutation and sexual recombination. Symbiosis potentially offers compositional evolution which can, where the problem domain allows, provide a fundamentally different form of adaptation from the gradualist framework by exploiting divide and conquer problem decomposition [22]. The shuffled-HIFF test problem, for example, can be solved by the composition provided by SEAM in time polynomial in the problem size, but neither a hill climbing process nor a GA using sexual crossover can be guaranteed to reach the optimum in less than exponential time [21][22]. Our new compositional algorithm can also reliably solve shuffled-HIFF, but also extends the scope of applicability to include a single-layered modular problem that is somewhat less contrived than shuffled-HIFF. The single-layered modular problem is, however, also beyond the competence of hill climbing and sexual crossover in the simple GA [14]. In addition to being more general, the new algorithm has a much simpler biological interpretation than SEAM, removing several markedly complicated components of the algorithm, as we shall discuss. Thus we provide an illustration of compositional evolution that is shown to be capable of exploiting the divide and conquer advantage of composition by solving a problem with modular interdependency [2]. The mechanism has relatively straightforward biological analogues in the formation of symbiotic associations, yet its adaptive capacity remains algorithmically distinct from that of the simple GA. The modularity exploited by this mechanism, as shown in the problem class we illustrate, is also a straightforward kind of modularity with a simple intuitive structure. Together this indicates that the breadth and potential impact of symbiosis and compositional mechanisms in natural systems could be much broader than the results supported by prior work, and brings us a step closer to developing biologically relevant algorithms of hierarchical adaptive processes and the major transitions in evolution. The new algorithm uses an abstraction of symbiogenesis as its variation operator, and decisions on when to invoke this mechanism to create a permanent association between entities are based on selecting symbiotic relationships with maximal synergy (synergy is essentially a measure of how much greater the symbiont is over the sum of the individual effects).

## 2 The Reciprocal Synergy Symbiosis Algorithm

This section describes the new algorithm presented and investigated in this paper. The Reciprocal Synergy Symbiosis Algorithm (RSSA) uses a generational loop consisting of evaluation, selection and variation, in common with a regular EA. However, significant differences exist in each of these stages as well as the representation used for a candidate. Psuedocode for the algorithm is given in Fig 1.

A population of candidates is used but they are intended to represent distinct entities in an ecosystem, rather than all competing for the same niche. As such a candidate entity only specifies alleles for a subset of the problem loci. The ecosystem is initialized with entities that each cover one allele for one locus such that collectively they cover all values for all problem variables.

The algorithm uses no mutation or other method of genetic variation; the only operator that changes the make-up of the ecosystem is the symbiogenic join. This join produces a new entity, taking on the alleles from each of its symbionts where specified, (usually) resulting in a larger, but still partially specified entity.

Partial evaluation of solutions is not available in general, and is not permitted in this algorithm. Instead a partially specified entity is evaluated in context. A context is simply a set of randomly generated values for each problem variable that is not specified by the entity under evaluation (see templates in [6] and contexts in [21][5]). Whilst individual entities are small with respect to the ecosystem, the ecosystem has niches for many small entities, and we would expect each of these to be occupied at any one time. As the result of the evaluations is to end up with some entities in permanent symbiogenic association, it is not sufficient to simply measure the fitness of each entity in the ecosystem. Symbiotic relationships are evaluated on a pairwise basis, where the synergy afforded by a particular join is assessed. Synergy is measured for all pairs of entities in the ecosystem, and for a meaningful measurement this is performed over various contexts.

We define a fitness delta, δ, as the difference in fitness between a context, *c*, alone and an entity, *A*, in that context (Eqn. 1). Synergy, *s*, is defined as the difference in fitness deltas of the symbiont, *A*+*B*, and the sum of the individual entities (Eqn. 2).

$$\delta(A,c) = f(A+c) - f(c) \tag{1}$$

$$s(A,B,c) = \delta(A+B,c) - \delta(A,c) - \delta(B,c) \tag{2}$$

This measure aims to quantify the additional benefit obtained by coexisting, over individual existence. This is closely related to what might be called a measure of epistasis between *A* and *B* when in the context of *c*. The selection phase is a competition between potential joins, and the most viable are those with high synergy scores. The symbiotic relationships are modeled such that whilst high synergy is rewarded, both symbionts must ‘want’ to make the join. This *reciprocal synergy maximization* is key to assigning the correct symbiotic joins, as we describe later. In order to identify joins with both maximal synergy *and* reciprocal desire to unite from *A* and *B*, we can consider the product of the synergy values for *A*+*B* and *B*+*A*. (Contrast this with an alternative rule that is simply ‘maximize the overall synergy for a pair’, which just sums *s*(*A*+*B*) and *s*(*B*+*A*). This would allow joins where one entity benefits significantly more than the other – a scenario we found to be unsuccessful). In order for the product to be consistent all synergy values must be unsigned. For simplicity we also choose to normalize the range, such that all values will lie in the range [0,1]. Once joins have been made, the generational loop returns to the evaluation stage, until all of the remaining entities are fully specified.

A number of points regarding the new algorithm are worth highlighting. We make abstractions of several elements that are often present in evolutionary algorithms. As in SEAM, no population dynamics are modeled: we assume that all values for each variable are already present in the ecosystem at initialization. Unlike in SEAM, the interactions between entities are also idealized. We assume that enough interactions occur in the timescale of a generation that we can perform a fixed number of interactions between every pair of entities in the ecosystem rather than stochastically sampling the interactions. The maximum number of joins that can be made in one generation is equal to half the current ecosystem size. In the current implementation there is no method for undoing a join.

```
Initialize ecosystem with all atomic units available in substrate
For g=1:MAX_GENERATIONS
  For c=1:CONTEXTS
    Generate context as random bit-string
    Evaluate context
    For each pair of entities remaining in ecosystem A,B
      Evaluate A in context
      Evaluate B in context
      Superimpose A and B, evaluate symbiont in context
      Calculate synergy for A+B
  For each pair A,B calculate mean synergy values across all
  contexts
  Process synergy matrix:
    Rescale synergy averages values to lie in range [0,1]
    Calculate reciprocal synergy for each pair A,B, by performing
    the product of s(A,B)*s(B,A)
  Find J highest reciprocal synergy values and make these joins
  If all entities remaining have length N
    Break
Report final ecosystem
```

**Fig. 1.** Psuedo-code for the Reciprocal Synergy Symbiosis Algorithm

# 3 Simulation Experiments

## 3.1 Performance on Shuffled-HIFF

In this section we verify that the RSSA can solve the HIFF problem [19] that has previously been used to exemplify the abilities of the SEAM algorithm [21]. Hierarchical If-and-only-If is a test problem that is built from a single subfunction of If-and-only-If (IFF), the Boolean function of equality. Each pair of variables is grouped into pairs using this same subfunction recursively, which results in an overall function that is dependent on all of its variables. The IFF subfunction incorporates epistasis between functional groups starting from the bottom hierarchical layer upwards and this causes the optima at each hierarchical level to be maximally distant in Hamming space, creating order-*N* dependencies at the highest level. This means that HIFF is pathologically difficult for a mutation-only hill climbing process to solve [19]. However it is an ideal candidate for solving with a compositional approach since these dependencies have a clear modular structure that composition can exploit. Note that a GA using suitable diversity maintenance and one-point crossover is successful on HIFF but only when the linkage map is tight. Randomising the linkage map such that genetic linkage does not correspond to the epistatic structure of the problem (shuffling) defeats the compositional mechanism of crossover in a GA. Simulations verified that the RSSA solves a 256-bit shuffled-HIFF in 30/30 runs demonstrating its ability to exploit modular interdependency successfully in this problem.[1]

### 3.2 Performance on VSM Problem

In this section, we compare RSSA against SEAM and two variants of GA on a modular test problem, the variable structural modularity (VSM) problem [14]. The specific instances used have clean modularity, for both unshuffled and shuffled cases.

The VSM problem is a test problem that parameterizes the amount of structural modularity to be present, and its modularity exhibits modular interdependency [2]. It is the simplest parameterized model to distinguish the performance of the GA and a hill climber with respect to modularity exploitation. As discussed in [14] a GA using crossover can only exploit the modularity to solve the problem correctly if that crossover mechanism preserves linkage. Therefore, by modifying the physical linkage such that it did not correspond to the epistatic linkage (but the modular structures remain), we expect to reveal a distinction between the GA and RSSA. Although the VSM permits a reduced amount of structural modularity to be present in problems, we do not investigate this dimension of freedom in this paper. The reduction in structural modularity makes the problem easier (see [14]) but since both the GA and RSSA can solve the problem in the hardest case, little is to be learned about the distinction between these two algorithms from easier, less modular problems. Thus we only investigate VSM instances with clean modularity and both tight and random linkage.

In the VSM, the fitness of a genotype is defined as the sum of weighted pairwise dependencies between the problem variables. An $N$-by-$N$ matrix is used to define the strength of each interaction, and these can be organized in such a fashion that the problem exhibits structural modularity. Each weight is only included in the summation if the variable values in question satisfy IFF (i.e. when both variables agree) and the problem instance we choose has just two classes of weights: strong weights for interdependencies which are internal to modules, and weak weights for interdependencies between modules. We choose to group the strong weights together near to the leading diagonal of the weight matrix, which introduces a number of modules, the variables of which have tight linkage on the genome. The overall fitness of a candidate $\boldsymbol{x}$ is given by Eqn. 3. Eqn. 4 describes specifically the location of the weights used to structure the modularity. Note that although the problem is built entirely from pairwise interactions, dependencies can be structured to 'act in concert' to create local optima with significant Hamming distances between them [14].

[1] An early version of the RSSA found that joining all pairs of variables with synergy>0 works very well on HIFF. However, this joining rule is sensitive to the assumption that fitness contributions combine linearly by default, and if all fitnesses are increased by a slight geometric scaling this method will erroneously join any pair of variables. To increase robustness to fitness scaling, the RSSA joins the most synergistic pairs of variables, and this method works reliably on HIFF$^{0.5}$ and HIFF$^{2}$ as well as the default fitness scaling of HIFF.

$$F\left(x_0, x_1, ..., x_{N-1}\right) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_{ij}\left(x_i \leftrightarrow x_j\right) \quad \textbf{(3)}$$

$$w_{ij} = \begin{cases} wI, \; if \lfloor i/k \rfloor = \lfloor j/k \rfloor \\ wE, \quad otherwise \; . \end{cases} \quad \textbf{(4)}$$

The problem parameters used are: *N*=400, *Z*=*k*=20, *wI*=400, *wE*=1. RSSA uses 160 contexts and assigns a maximum proportion of 0.5 joins per generation. This uses approximately 412 million evaluations. For the GAs we use deterministic crowding as a selection method, a mutation rate of 4/*N*, a population size of 4000, crossover rate of 0.05 (this is a low value but see [14] for discussion). SEAM uses 200 contexts to make its joining decisions.

**Table 1.** Successful runs out of 30 for the VSM problem, parameters as in text above

| | **VSM, 400 bit** | **S-VSM, 400 bit** |
|---|---|---|
| GA-DC 1point crossover | 30/30 | 0/30 |
| GA-DC uniform crossover | 0/30 | 0/30 |
| SEAM | 0/30 | 0/30 |
| RSSA | 30/30 | 30/30 |

The results shown in Table 1 illustrate the ability of RSSA to solve a second type of modular problem. The joins made in RSSA explicitly represent epistatic dependencies and shuffling the genetic map does not deter RSSA from reliably solving either problem. The GA using one point crossover solves the unshuffled problem as it preserves linkage. However when the genetic map is randomized, preserving (physical) linkage is no longer sufficient to exploit the modularity in the problem, and it should be no surprise that the GA is defeated on the shuffled problem. Note that SEAM is unable to solve either of these problem configurations.

Fig. 2 shows the synergy matrices for examples of HIFF and VSM problems created by RSSA, before any joins have been made. The instances shown are smaller than those tested in order to show relevant detail, but the properties are qualitatively alike. Note that the problem has *N* variables but these matrices are $(2N)^2$ since each variable can take two values, and the relationships are assumed to be independent. Firstly considering Fig. 2(a), some of the structure of the problem is revealed, but not all. The number of contexts used is sufficient to uncover the next two or sometimes three hierarchical layers, but an exponential increase in the number of contexts is required to see further layers. However, the information required to make the correct joins for a single layer at a time is sufficient when the problem is solved in a compositional manner. For the VSM (Fig. 2(b)), which has only a single layer of hierarchy, the entire structure is revealed: the 1-entities produce high synergy with other 1-entities within modules, and likewise for 0-entities. The bright line which runs through all the dark blocks in the top-right and bottom-left quadrants represents high synergy for two entities coding different values for the same locus. None of these joins should be made, but RSSA requires a large number of evaluations to reveal that these joins are incorrect. Taking into account these synergy matrices and the fact that

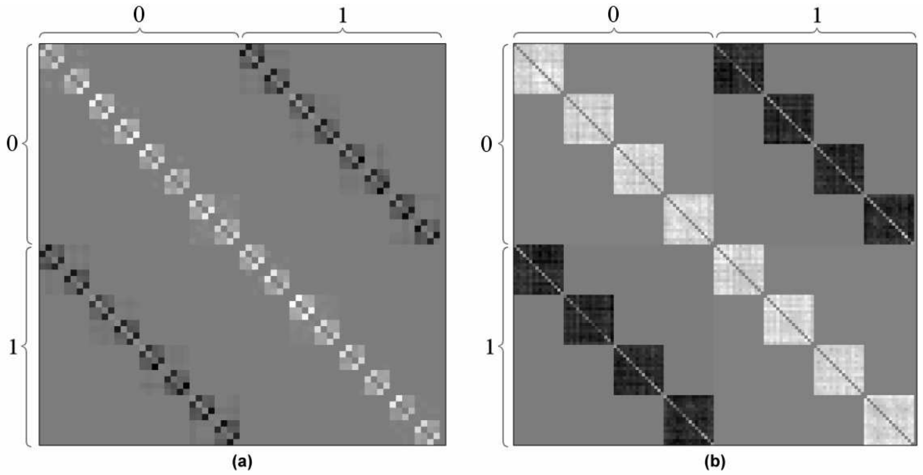

**Fig. 2.** Synergy matrices from (a) 32-bit HIFF and (b) 64-bit VSM, *Z*=4 problems (unshuffled genetic maps) before any joins have been made. Each entry (*i*,*j*) represents the synergy between the *i*th and *j*th problem variables. Strong synergy values are closer to white, weak synergy values are closer to black. Note how the lowest level modules are clear in both cases, but higher level interactions in HIFF are not visible until the search space has been reduced

modular problems are solved by RSSA, we see that our algorithm is capable of correctly identifying modular structures and revealing that structure in the synergy values it assesses.

## 4 Discussion

The RSSA introduced in this paper is both a simplification and generalization of SEAM and thereby expands the relevance of symbiosis-inspired algorithms. In section 1 we identified a number of shortcomings with SEAM. Here we revisit these limitations, noting how they are addressed by RSSA. The use of a Pareto dominance mechanism to make joining decisions is weak biologically: it is difficult to imagine two entities not making a join until they have made absolutely sure there is no other alternative symbiotic partner they would rather be with. We replace this mechanism with one that feels more likely in a biological system; that is, the associations that have the strongest synergy for both partners result in symbiotic joins most readily. This also has the algorithmic advantage that it more directly assesses and exploits the epistasis in the problem. Where SEAM required coevolved contexts, the RSSA shows that randomly generated templates are sufficient to facilitate the evaluation of partially specified entities. Although comprehensively describing the algorithmic niche of the RSSA remains an open question, we are confident that it is broader than SEAM: the only problem SEAM is shown to be competent on is shuffled-HIFF and the RSSA is here demonstrated to solve both shuffled-HIFF and another, and less contrived, form of modular problem that SEAM cannot solve.

There are several areas of future research and development, two of which concern the interactions which are used to determine symbiotic joins. Currently RSSA only makes joins between two entities (although each entity may contain many variable values), but there is scope for uniting larger groups. This would not necessarily require the evaluation of symbiotic groups of more than two entities at a time, but could occur when high synergy was identified in all pairs within the group. The interactions could also be focused selectively. Currently all $(2N)^2$ pairs of entities are evaluated in the same number of contexts. But we might imagine that when one pair of entities consistently discovers low synergy scores that their frequency of interaction (i.e. the number of additional contexts they are tested in) would decline, but pairs which show promise of high reciprocal synergy would be allocated further contexts to confirm the value of a join. Although not a key factor here since we have been examining the generality of the algorithm rather than its efficiency, the computational expense of the algorithm is high. Both of these two modifications would reduce the average number of evaluations per join. Finally, we aim to ascertain the ability of RSSA to solve a wider set of modular problems, both hierarchical and flat in structure. It would also be of interest to examine its robustness to the parameterizable modular structures described in [14].

## 5 Conclusion

In this paper we have presented a new model inspired by symbiosis, with a more straightforward compositional mechanism than that used by SEAM and a simpler biological interpretation than any previous symbiosis-inspired algorithm. We have used the algorithm to explore various possibilities enabled by using symbiosis as a variation operator. As in previous studies, we find it can solve problems that cannot be solved with the more traditional variation operators of crossover and mutation. The experiments performed show the applicability of RSSA is widened when compared with SEAM and it is able to demonstrate an algorithmic advantage over the simple GA on problems with a very simple and intuitive modular structure. In addition, we can observe the structure that is discovered by the algorithm which may provide insight into the characteristics of less neatly formed problems.

Understanding the adaptive significance of the major evolutionary transitions, genetic events of symbiogenesis, and more generally, the formation of adaptive associations among symbionts is critical to evolutionary theory and explaining how evolution works in full [12]. In our work we want to understand not only the conditions under which such events may occur, but also what consequence such events might have for evolutionary adaptation and the evolvability of complex systems. Intuitively, modularity has a significant impact on evolvability but only if there are appropriate mechanisms to exploit it. Crossover in the simple GA cannot exploit modularity unless genetic linkage corresponds with epistatic dependencies, but because the symbiotic joining mechanism modeled here is insensitive to genetic linkage it can exploit modularity that cannot be exploited by the simple GA. It has been shown previously that symbiosis-inspired mechanisms can in principle provide a fundamentally different form of adaptation from the variation of mutation and crossover in the simple GA [21]. However, the exploration of the space in which these types of mechanisms have an advantage is in its infancy. In this paper we make some significant improvements over previous attempts to provide a simple algorithmic model of the formation of symbiotic associations and to describe the kind of modularity it can exploit.

## References

1. Baldwin, C.Y. and Clark, K.B.: *Design Rules, Vol 1: The Power of Modularity*, MIT Press, Cambridge, MA, (2000)
2. Dauscher, P., Polani, D. and Watson R.A.: A simple modularity measure for search spaces based on information theory. In procs *Artificial Life X* (2006)
3. Deb, K. and Goldberg D.E.: Analyzing deception in trap functions. In *FOGA 2*, ed D. Whitley, Morgan Kauffman (1993) 98-108
4. Defaweux, A., Lenaerts, T. and van Hemert J.I.: Evolutionary Transitions as a Metaphor for Evolutionary Optimisation. In procs *ECAL* (2005) 342-352
5. de Jong, E.D., Watson R.A. and Thierens D.: on the complexity of hierarchical problem solving. In procs *GECCO* (2005)
6. Goldberg, D.E., B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems* 3 (1989) 493-530
7. Higgs, P. G.: RNA secondary structure: physical and computational aspects. Q*uarterly Reviews of Biophysics* 8(4) (2000) 373-391
8. Khakhina, L. N.: *Concepts of Symbiogenesis: Historical and Critical Study of the Research of Russian Botanists*, ed. Margulis L. and McMenamin M., trans.: Merkel S. and Coalston, R. Yale University Press, New Haven (1992)
9. Mahfoud, S.: Crowding and Preselection Revisited. *PPSN 2*, Elsevier (1992) 27-36
10. Margulis, L: *The Symbiotic Planet: A new look at evolution*. Phoenix, London (1998)
11. Margulis, L., Dolan, M.F. and Guerrero, R.: The chimeric eukaryote: Origin of the nucleus from the karyomastigont in amitochondriate protists. *PNAS* 97(13) (2000) 6954-6959
12. Maynard Smith, J. and Szathmary, E.: *The major transitions in evolution*. W.H. Freeman, New York (1995)
13. Mayr, E.: *What Evolution is*. Phoenix, London (2001)
14. Mills, R. and Watson, R.A.: Variable Discrimination of Crossover Versus Mutation Using Parameterized Modular Structure. In press *GECCO* (2007)
15. Mitchell, M., Forrest, S. and Holland, J.H.: The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In *ECAL 1*, MIT Press, Cambridge, MA (1992)
16. Pelikan, M. Goldberg, D.E. and Cantú-Paz, E.: BOA: The Bayesian optimization algorithm. In procs *GECCO* (1999)
17. Potter, M.A.: *The Design and Analysis of a Computational Model of Cooperative Coevolution*. Ph.D. dissertation, George Mason University, Fairfax, Virginia. (1997)
18. Simon, H.: *Sciences of the Artificial*. MIT Press, Cambridge MA (1969)
19. Watson, R.A., Hornby, G.S. and Pollack J.B.: Modeling Building Block Interdependency. In procs *PPSN V*, Springer, Berlin (1998) 97-106
20. Watson, R.A. and Jansen, T.: A building block royal road where crossover is provably essential. In press *GECCO* (2007)
21. Watson, R.A. and Pollack J.B.: A Computational Model of Symbiotic Composition in Evolutionary Transitions. *BioSystems* 69(2–3) (2003) 187–209.
22. Watson, R.A. *Compositional Evolution*. MIT Press, Cambridge, MA (2006)