Homogeneous and Heterogeneous Island Models for the Set Cover Problem

Andrea Mambrini[1], Dirk Sudholt[2], and Xin Yao[1]

[1] University of Birmingham, Birmingham, UK
[2] University of Sheffield, Sheffield, UK

Abstract. We propose and analyse two island models that provably find good approximations for the SetCover problem. A homogeneous island model running parallel instances of the SEMO algorithm—following Friedrich *et al.* (Evolutionary Computation 18(4), 2010, 617-633)—leads to significant speedups over a single SEMO instance, but at the expense of large communication costs. A heterogeneous island model, where each island optimises a different single-objective fitness function, provides similar speedups at reduced communication costs. We compare different topologies for the homogeneous model and different migration policies for the heterogeneous one.

Keywords: Parallel evolutionary algorithms, set cover, island model, theory, runtime analysis.

1 Introduction

Due to the current development in computer architecture and the steeply rising number of processors in modern devices, parallelisation is becoming a more and more important issue. Evolutionary algorithms (EAs) can be parallelised by using island models, also called coarse-grained EAs or multi-deme models [1,2]. Several subpopulations are evolved on different processors. Subpopulations coordinate their search by a process called migration, where selected individuals, or copies thereof, are sent to other islands. Migration often happens periodically or probabilistically and islands are typically connected by spatial structures such as rings or torus graphs [3]. Compared to panmictic populations, this decreases the spread of information. A slower spread of information can increase the diversity in the whole system, and by choosing the right topology and the frequency or probability of migration, the communication effort can be tuned.

Despite being applied and researched intensively, the theoretical foundation of parallel EAs is still in its infancy. Even the effect of the most fundamental parameters on performance is not well understood [1] and more research is needed to understand the search dynamics in island models [4]. Present theoretical studies include takeover times and growth curves (see, e. g., [5] or [1, Chapter 4]). Recently the expected running time of parallel EAs has been studied, leading to a constructed example where island models excel over panmictic populations [6,7] and examples where the diversity in island models makes crossover

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part I, LNCS 7491, pp. 11–20, 2012.

a powerful operator [8,9]. Also the speedup in island models has been studied rigorously: how the number of generations can be decreased by running multiple islands instead of one. Studies include pseudo-Boolean optimisation [10,11] and polynomial-time solvable problems from combinatorial optimisation [12].

These works form a solid foundation towards a theory of parallel metaheuristics, but they leave open many important questions. None of these works addresses how island models behave on general instances of NP-hard problems, or how they deal with multiobjective fitness functions. Furthermore, studies have been limited to homogeneous island models, where all islands run the same algorithm. In many settings heterogeneous models make more sense—islands can use different parameters, different operators, and even different fitness functions. This closely relates to the emerging area of hyper-heuristics [13].

In this work we propose and analyse homogeneous and heterogeneous island models for the SetCover problem. Given a set S with m elements and a collection of n subsets of S with associated costs, the SetCover problem asks for a selection of subsets of S that cover the whole set and have minimum cost. This classic NP-hard problem is one of the most fundamental problems in computer science. Friedrich *et al.* [14] studied a SEMO algorithm on a biobjective formulation of the problem and showed that SEMO efficiently computes an H_m-approximation, where $H_m = \sum_{i=1}^{m} \frac{1}{i}$ is the m-th Harmonic number.

We study a parallel version of this algorithm where each island runs an instance of SEMO. Each island use the same bi-objective fitness functions to be minimised: one criterion counting the number of uncovered elements and the other representing the cost of the selection. Each island stores a population of non-dominated solutions. At the end of each generation migration occurs transmitting a copy of the whole population to all neighbouring islands.

We show that this leads to significant speedups, depending on the topology and the migration probability, for probabilistic migration policies. However, this homogeneous island model has large communication costs as whole populations are exchanged between islands. To this end, we propose a heterogeneous island model that has a lower communication cost and islands run simpler algorithms.

The heterogeneous island model consists of $m + 1$ islands using different single-objective fitness functions. Each island stores one individual and runs a (1+1) EA (or RLS that just differs in using local instead of global mutation). The fitness functions are such that on island i only selections covering i elements are feasible. Therefore, each island i keeps the best individual covering i elements of S. The island model can be implemented on fewer than $m + 1$ processors by running multiple islands on each processor. We show that the collection of islands is able to guarantee the same performance and approximation quality as in the homogeneous model, but with lower communication costs and simpler operations. We also study different migration policies for the heterogeneous model and show how the migration policy affects running time and communication costs.

Due to space restrictions, many proofs are omitted or reduced to proof sketches.

2 Preliminaries

Let $S = \{s_1, \cdots, s_m\}$ be a set containing m elements and $C = \{C_1, \cdots, C_n\}$ be a collection of non-empty sets such that $C_i \subset S$ for $1 \le i \le n$ and $\bigcup_{i=1}^{n} C_i = S$. Each set C_i has a cost $c_i > 0$. We call $X = x_1 \cdots x_n$ a *selection* of C and we say that C_i is in the selection X iff $x_i = 1$. The optimal solution to the SetCover problem is an X such that $\bigcup_{i:x_i=1} C_i = S$ and $\sum_{i:x_i=1} c_i$ is minimum.

We define the following measures:

- $c(X) = |\bigcup_{i:x_i=1} C_i|$ is the number of covered elements of the selection.
- $|X|_1 = \sum_{i=1}^{n} x_i$ is the number of selected sets of a selection.
- $\mathrm{cost}(X) = \sum_{i:x_i=1} c_i$ is the cost of a selection.
- $c_{\max} = \max_i c_i$ is the maximum cost of a set.
- $c_e(C_i, X) = \frac{\left|C_i \setminus \bigcup_{j:x_j=1} C_j\right|_1}{c_i}$ is the cost-effectiveness of a set w. r. t. X.

The homogeneous island model consists of an archipelago of μ islands each one running the SEMO algorithm, minimising the fitness function $f(X) = (m - c(X), \mathrm{cost}(X))$. SEMO always maintains a set of non-dominated search points. New solutions are created by selecting uniformly a search point from the current population and mutating it. The offspring is added to the current population and then all dominated search points are removed. SEMO uses local mutations: one bit is chosen uniformly at random and then flipped. A variant called *global SEMO* uses standard bit mutations instead (called *global mutations*), flipping each bit independently with probability $1/n$. In the homogeneous island model based on SEMO or global SEMO (see Algorithm 1), each island maintains such a population. For migration, a copy of this whole set is transmitted to all neighbouring islands. The union of this set with the target island's set is considered and then all dominated solutions are removed. This way, the best solutions among source and target islands are maintained and combined.

Algorithm 1. Homogeneous island model based on (global) SEMO

```
Initialise P^(0) = {P_1^(0), ..., P_μ^(0)}, where P_i^(0) = {0^n} for 1 ≤ i ≤ μ. Let t := 0.
repeat forever
    for each island i do in parallel
        Simulate one generation of (global) SEMO, updating P_i^(t).
        Send a copy of the population P_i^(t) to all neighbouring islands.
        Unify P_i^(t) with all populations received from other islands.
        Remove all dominated search points from P_i^(t).
    Let t := t + 1.
```

The heterogeneous island model consists of a fully connected archipelago of $m+1$ islands indexed $0, \dots, m$. Each island stores just one individual and runs an (1+1) EA (or RLS) using a single-objective function that is different on each island. For island i we define the fitness function (to maximise) as:

$$f_i(X) = \begin{cases} nc_{\max} - \mathrm{cost}(X) & \text{if } c(X) = i \\ -|c(X) - i| & \text{if } c(X) \neq i \end{cases}$$

The idea is that island i stores an individual that represents the so far best selection covering i elements (referred to as *feasible*). If the solution does not cover i elements, the fitness is negative and hints are given towards covering i elements[1]. Each island is thus assigned a different part of the search space to optimise. This is similar to what happens in dynamic programming [15].

The heterogeneous island model is shown in Algorithm 2. Note that the heterogeneous island model can be easily implemented on $\mu \leq m$ processors by running up to $\lceil \frac{m+1}{\mu} \rceil$ islands on each processor.

Both island models are initialised with empty selections. This is a sensible strategy for SetCover and theoretical results [14] as well as preliminary experiments have shown that this only speeds up computation.

Algorithm 2. Heterogeneous island model based on (1+1) EA (or RLS)

```
Initialise the island individuals X_0^(0), ..., X_m^(0) to 0^n. Let t := 0.
repeat forever
    for each island i do in parallel
        Produce a global (or local) mutation X̃_i^(t) of the individual X_i^(t).
        Send a copy of X̃_i^(t) to each other island.
        Choose X_i^(t+1) with maximal f_i-value among X_i^(t), X̃_i^(t) and all immigrants.
    Let t := t + 1.
```

The homogeneous and heterogeneous island models differ fundamentally in their search behaviour. Following Skolicki [16], we distinguish *intra-island evolution* (the evolution within each island) and *inter-island evolution* (evolution among and between islands). The homogeneous model uses intra-island evolution to generate improvements by mutation, and migration helps to propagate these improvements to other islands. The heterogeneous island model strongly relies on inter-island evolution; in fact, beneficial mutations as in the homogeneous model yield solutions that are only feasible on other islands. The two island models also differ in the population size. In the heterogeneous model the population of each island consists of just one individual, while in the homogeneous model the population size of each island is upper bounded by m. This generally means that the time and space required to compute a generation in the homogeneous model is larger than in the heterogeneous one.

We define the *parallel running time* as the number of generations of an island model until it has found a satisfactory solution, in our case an H_m-approximation. We also refer to the *sequential running time* as the product between the parallel running time and the number of islands. This represents the computational effort to simulate the model on a single processor. The *speedup* of an island model with μ islands is defined as the rate between the expected parallel running time of the island model and the expected running time of the same EA using only a single island. This kind of speedup is called *weak orthodox speedup* in Alba's taxonomy [17]. If the speedup is of order $\Theta(\mu)$, we speak of

[1] Our analysis holds for any negative function for the second case of f_i.

a *linear speedup*. Furthermore, we also consider the effort for performing migration. We define the *communication effort* as the total number of individuals sent between islands, throughout a run of an island model. The (expected) communication effort is given by the (expected) parallel time, multiplied by the number of islands and the (expected) number of emigrants sent by one island.

In order to achieve a good balance between the communication effort and the parallel running time, we consider the following migration policies. The first two policies make sense for both island models. The last two policies are tailored towards the heterogeneous model.

complete migration: each island sends migrants to all other islands.
uniform probabilistic: each island sends migrants to every other island independently with a migration probability p.
non-uniform probabilistic: each island i sends migrants to every other island $(i + k) \bmod (m + 1)$ independently with probability $1/k$.
smart migration: Each island i sends migrants to island $c(\tilde{X}_i)$, where $\tilde{X}_i$ is the offspring generated on island i.

3 Analysis of the Homogeneous Island Model

We first consider the homogeneous model with uniform probabilistic migration as this includes complete migration. In their analysis of SEMO, Friedrich *et al.* [14] consider the time until SEMO finds an empty selection, and how long it takes to get a H_m-approximate solution from there. Their results are as follows.

Theorem 1 (Friedrich *et al.* [14]). *For any initialisation and every* SET-COVER *instance, SEMO and global SEMO find an H_m-approximate solution in $O(m^2 n + mn \log(nc_{\max}))$ expected generations. When starting with a population containing only an empty selection, the time bound is $O(m^2 n)$ generations.*

The following lemma is at the heart of their–and our–analysis. It goes back to Chvatal's analysis of the greedy algorithm [18]. Starting with an empty set, the greedy algorithm subsequently adds the most cost-effective set to the current solution. When k elements are covered, for some $0 \leq k \leq m$, the cost of this partial solution is at most $\mathrm{cost}(X) \leq (H_m - H_{m-k})\,\mathrm{OPT}$, where OPT denotes the cost of an optimal solution. For $k = m$ this gives an H_m-approximation.

Lemma 1. *Let* OPT *be the cost of an optimal set cover and X be such that $c(X) = k$ (with $k < m$) and $\mathrm{cost}(X) \leq (H_m - H_{m-k})\,\mathrm{OPT}$. Adding the most cost-effective set to X creates X' with $c(X') = k'$ and $\mathrm{cost}(X') \leq (H_m - H_{m-k'})\,\mathrm{OPT}$.*

Proof. The selection X leaves $m - k$ elements of S uncovered. These elements can be covered at cost OPT since the optimal cover covers the whole set. Then there is a set with cost-effectiveness at least $\frac{m-k}{\mathrm{OPT}}$. Let i be the number of newly covered elements by adding this set, then after adding the set we get a solution covering $k' = k + i$ elements at cost no more than

$$\left(H_m - H_{m-k} + \frac{i}{m-k}\right) \cdot \mathrm{OPT} \leq (H_m - H_{m-k'}) \cdot \mathrm{OPT}\,. \qquad \square$$

This behaviour can be mimicked by SEMO [14] and the homogeneous island model. Friedrich *et al.* [14] define the *potential* of the population of the archipelago as the largest k such that there is an individual in the population that covers k elements and costs at most $(H_m - H_{m-k}) \cdot \mathrm{OPT}$. The potential can never decrease as SEMO always keeps some solution with k covered elements in the population. Starting with empty selections, the initial potential is at least 0.

The probability of increasing the potential is at least $1/((m+1)en)$ for the following reasons. It is sufficient to select the solution defining the potential and to add a set with maximum cost-effectiveness (Lemma 1). The population contains at most $m+1$ individuals, so the probability of selecting the right parent is at least $1/(m+1)$. The probability of a specific 1-bit mutation is at least $1/n \cdot (1-1/n)^{n-1} \geq 1/(en)$ for both local and global SEMO.

This analysis can be transferred to our homogeneous island model using the general method by Lässig and Sudholt [10] based on fitness levels. Assume the search space can be partitioned into fitness-level sets ordered w. r. t. fitness such that an EA never decreases its current level. If we have lower bounds on the probability that the EA will leave a current level towards a better fitness-level set, we get an upper bound on the expected hitting time of the final level. For island models we get upper bounds on the expected parallel running time that depend on the topology at hand and the probability that migration successfully transmits information about the current best fitness level. A rapid spread of information enables more islands to search on the current best fitness level, which gives better performance guarantees than a slow spread of information.

In [10] upper bounds are stated for common topologies: ring graphs, torus or grid graphs, and the complete topology. In our case instead of using fitness levels, we argue with the potential of islands. As seen above, the potential can never decrease. We have $m+1$ potential values, and the probability of increasing the potential on any island is at least $1/((m+1)en)$. Plugging this into the results from [10,19], we get the following bounds on the expected parallel time. The expected communication effort is by a factor of $pd(m+1)\mu$ larger than the expected parallel time, where d is the degree of any node in the topology.

Theorem 2. *For the homogeneous island model based on (global) SEMO on μ islands and migration probability $p > 0$ the expected parallel time until an H_m-approximation for* SetCover *is found is bounded by*

- $O\left(\frac{n^{1/2}m^{3/2}}{p^{1/2}} + \frac{nm^2}{\mu}\right)$ *for any ring topology,*
- $O\left(\frac{n^{1/3}m^{4/3}}{p^{2/3}} + \frac{nm^2}{\mu}\right)$ *for any undirected $\sqrt{\mu} \times \sqrt{\mu}$ grid or torus graph*
- $O\left(\frac{m}{p} + \frac{nm^2}{\mu}\right)$ *for the complete topology K_μ.*

The expected communication effort is $O\big(p^{1/2}\mu n^{1/2}m^{5/2} + pnm^3\big)$ for rings, $O\big(p^{1/3}\mu n^{1/3}m^{7/3} + pnm^3\big)$ for grids and $O\big(\mu^2 m^2 + p\mu nm^3\big)$ for K_μ.

The upper bounds are asymptotically minimised for choosing the number of islands as $\mu = \sqrt{pnm}$, $\mu = (pnm)^{2/3}$, and $\mu = pnm$, respectively. With these choices we get expected parallel times of $O(n^{1/2}m^{3/2}/p^{1/2})$, $O(n^{1/3}m^{4/3}/p^{2/3})$, and $O(m/p)$, respectively (see Table 1 in Section 5). The expected communication effort is $O(pnm^3)$, $O(pnm^3)$, and $O(p^2n^2m^4)$, respectively. Multiplying all parallel times by μ, we see that the expected sequential time is bounded by $O(nm^2)$ in all three cases. This asymptotically matches the upper bound from Theorem 1 for initialisation with empty selections. This means that, apart from constant factors hidden in the asymptotic notation, in these cases parallelization does not increase the (upper bounds on the) total running time, but the (upper bounds on the) parallel time can decrease significantly. In fact, all numbers of islands up to the values mentioned above yield linear speedups—for cases where the $O(nm^2)$-bound for a single (global) SEMO is asymptotically tight.

As remarked in [11], the bound for the complete topology with $p = 1$ also applies to an offspring population-version of SEMO where λ offspring are created and added to the population, before removing dominated solutions.

4 Analysis of the Heterogeneous Island Model

For the heterogeneous model based on (1+1) EA or RLS we first present an analysis for the complete migration policy.

Theorem 3. *The heterogeneous island model with complete migration finds an H_m-approximate solution for* SetCover *in an expected parallel time of $O(n \cdot \min(m, n))$. The expected communication effort is $O(nm^2 \cdot \min(m, n))$.*

Proof. As in Theorem 2 we calculate the expected time to produce a solution that is at least as good as the greedy solution, starting from 0^n and always adding the most cost-effective set. We define again the potential of the population of the archipelago as the largest k such that there is an individual in the population that covers k elements and costs at most $(H_m - H_{m-k}) \cdot \mathrm{OPT}$. At the end of each generation (after migration and selection) the potential can't decrease. In fact the individual X^k on island k can only be replaced by an individual with the same number of covered elements but a lower cost (and that would not affect the potential). Instead the potential can be increased to k' mutating X^k such that the most cost-effective set is added. That would produce an individual $\tilde{X}^k$ such that $c(\tilde{X}^k) = k' > k$ and $\mathrm{cost}(\tilde{X}^k) \leq (H_k - H_{m-k'}) \cdot \mathrm{OPT}$ (Lemma 1).

After migration and selection this individual will replace the individual on the island k' (which had higher cost and therefore lower fitness). This specific 1-bit mutation happens with probability at least $1/n \cdot (1 - 1/n)^{n-1} \geq 1/(en)$ for both local and global mutation. At most n sets can be included in a selection but, if $n > m$, at most m of them will be selected since each most cost-effective set covers at least one new element (otherwise its cost-effectiveness would be 0). So after $O(n \cdot \min(m, n))$ expected generations $k = m$ and then on island m we get an $H_m - H_{m-m} = H_m$-approximate solution. $\square$

Comparing this time with [14] and assuming $n = O(m)$, we get that our parallel time is by a factor of $\Theta(m)$ lower, while we get the same upper bound for the sequential running time.

For uniform probabilistic migration with migration probability $p < 1$, the island model only increases the potential if migration happens on the edge that links the two islands involved (k and k'). The probability estimate for this event decreases by a factor of p, and the waiting time thus increases by $1/p$.

Theorem 4. *The heterogeneous island model with uniform probabilistic migration and migration probability p finds an H_m-approximate solution for* SetCover *in an expected parallel time of $O(n \cdot \min(m, n)/p)$. The expected communication effort is $O(nm^2 \cdot \min(m, n))$.*

We see that our estimate of the communication effort has not improved. This is not surprising as we only rely on inter-island evolution for making progress. A uniform migration probability delays the inter-island evolution and the reduced communication effort in a single generation is nullified by a larger parallel running time.

With non-uniform probabilistic migration, the chance of making the right migration is generally higher than for uniform migration probabilities. Typically only few new elements are covered, when adding a most cost-effective set. A large number of new elements implies that we make large progress. This balances out a small migration probability: if adding the most cost-effective set covers j new elements, the probability of making this move is at least $1/j \cdot 1/(en)$. In expectation, the potential increases by at least $j \cdot 1/j \cdot 1/(en) = 1/(en)$, regardless of j. A straightforward drift analysis gives the following.

Theorem 5. *The heterogeneous island model with non-uniform probabilistic migration finds an H_m-approximate solution for* SetCover *in an expected parallel time of enm. The expected communication effort is at most enm^2H_m.*

Smart migration sends emigrants only to the unique island where they are considered feasible. The proof of Theorem 3 only relies on such migrations. Hence the upper bound also holds for smart migration.

Theorem 6. *The heterogeneous island model with smart migration finds an H_m-approximate solution for* SetCover *in an expected parallel time of $O(n \cdot \min(m, n))$. The expected communication effort is $O(nm \cdot \min(m, n))$.*

In our setting, smart migration outperforms all other migration policies as it leads to the best upper bound for the communication effort.

5 Discussion and Conclusions

We have proposed and analysed two parallel EAs for the SetCover problem that provably find good approximations. Table 1 gives an overview of our results, regarding parallel and sequential expected running times as well as the communication effort. In order to fairly compare heterogeneous and homogeneous models we consider them running on μ processors. For the heterogeneous model this means (for $\mu \leq m$) running up to $\lceil \frac{m+1}{\mu} \rceil$ islands on the same processor and thus increasing the parallel running time by a factor of $\Theta(\frac{m}{\mu})$.

Table 1. Upper bounds on expected parallel times (general bounds and bounds for best μ), expected sequential times and expected communication effort for homogeneous island models with various migration topologies and for heterogeneous island models with various migration policies, until an H_m-approximation is found for any SetCover instance with m elements and n sets. p denotes the migration probability. We simplified $\min(n, m) \leq m$ and we constrained μ to yield linear speedups.

Algorithm	parallel time bounds general b. $\rightsquigarrow$ best bound		seq. time	comm. effort
Non-parallel SEMO	$O(nm^2)$	$\rightsquigarrow O(nm^2)$	$O(nm^2)$	0
Homogeneous island model based on (global) SEMO and topology...				
– ring ($\mu \leq \sqrt{pnm}$)	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O\left(\frac{n^{1/2}m^{3/2}}{p^{1/2}}\right)$	$O(nm^2)$	$O(pnm^3)$
– grid ($\mu \leq (pnm)^{2/3}$)	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O\left(\frac{n^{1/3}m^{4/3}}{p^{2/3}}\right)$	$O(nm^2)$	$O(pnm^3)$
– complete ($\mu \leq pnm$)	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O\left(\frac{m}{p}\right)$	$O(nm^2)$	$O(p^2n^2m^4)$
Heterogeneous island model with $\mu \leq m$ based on (1+1) EA (or RLS) and policy...				
– complete	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O(nm)$	$O(nm^2)$	$O(nm^3)$
– uniform prob.	$O\left(\frac{nm^2}{\mu p}\right)$	$\rightsquigarrow O\left(\frac{nm}{p}\right)$	$O\left(\frac{nm^2}{p}\right)$	$O(nm^3)$
– non-uniform prob.	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O(nm)$	$O(nm^2)$	$O(nm^2 \log m)$
– smart migration	$O\left(\frac{nm^2}{\mu}\right)$	$\rightsquigarrow O(nm)$	$O(nm^2)$	$O(nm^2)$

For the homogeneous model based on (global) SEMO, the topology determines how many islands still give a linear speedup. For dense topologies more islands can be used. The migration probability gives a smooth trade-off between this maximum number of islands and the communication effort. For large migration probabilities the heterogeneous island model based on the (1+1) EA (or RLS) has lower communication costs, when comparing complete topologies or using the right migration policies. It is also easier to implement as unlike for the SEMO-based model it is not necessary for each island to handle large populations and to remove many dominated solutions. Thus the heterogeneous model is also faster when considering the time and space required to compute a generation.

The discussion on migration policies has revealed how adding more knowledge about the problem can decrease the communication effort. The complete migration and uniform migration policies do not require any knowledge about the problem at hand, while non-uniform migration only needs a sensible ordering of islands to work. This ordering should be consistent with the similarity between different islands. We believe that this approach can be fruitful for other heterogeneous island models. Smart migration requires knowledge about the problem at hand since it needs to inspect the genotype to determine the island to send it to. But it leads to the best performance guarantees among all considered policies.

Experiments (not included here) show that on random SetCover instances both island models quickly find better solutions than the greedy algorithm. An experimental study is left for future work. Future work should also investigate whether the approach used in the heterogeneous island model (i. e. assigning a portion of the search space to each island) can solve a broader class of problems.

Acknowledgments. This research was partially supported by EPSRC grants EP/D052785/1 and EP/I010297/1.

References

1. Luque, G., Alba, E.: Parallel Genetic Algorithms–Theory and Real World Applications. Springer (2011)
2. Nedjah, N., de Macedo Mourelle, L., Alba, E.: Parallel Evolutionary Computations. Springer (2006)
3. Tomassini, M.: Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time. Springer (2005)
4. Skolicki, Z., De Jong, K.: The influence of migration sizes and intervals on island models. In: Proc. of GECCO 2005, pp. 1295–1302. ACM (2005)
5. Rudolph, G.: Takeover time in parallel populations with migration. In: Filipic, B., Silc, J. (eds.) Proc. of BIOMA 2006, pp. 63–72 (2006)
6. Lässig, J., Sudholt, D.: The benefit of migration in parallel evolutionary algorithms. In: Proc. of GECCO 2010, pp. 1105–1112. ACM (2010)
7. Lässig, J., Sudholt, D.: Experimental Supplements to the Theoretical Analysis of Migration in the Island Model. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 224–233. Springer, Heidelberg (2010)
8. Watson, R.A., Jansen, T.: A building-block royal road where crossover is provably essential. In: Proc. of GECCO 2007, pp. 1452–1459. ACM (2007)
9. Neumann, F., Oliveto, P.S., Rudolph, G., Sudholt, D.: On the effectiveness of crossover for migration in parallel evolutionary algorithms. In: Proc. of GECCO 2011, pp. 1587–1594. ACM (2011)
10. Lässig, J., Sudholt, D.: General Scheme for Analyzing Running Times of Parallel Evolutionary Algorithms. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 234–243. Springer, Heidelberg (2010)
11. Lässig, J., Sudholt, D.: Adaptive population models for offspring populations and parallel evolutionary algorithms. In: Proc. of FOGA 2011, pp. 181–192. ACM (2011)
12. Lässig, J., Sudholt, D.: Analysis of Speedups in Parallel Evolutionary Algorithms for Combinatorial Optimization. In: Asano, T., Nakano, S.-i., Okamoto, Y., Watanabe, O. (eds.) ISAAC 2011. LNCS, vol. 7074, pp. 405–414. Springer, Heidelberg (2011)
13. Burke, E.K., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-heuristics: An emerging direction in modern search technology, pp. 457–474. Kluwer (2003)
14. Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Approximating covering problems by randomized search heuristics using multi-objective models. Evolutionary Computation 18(4), 617–633 (2010)
15. Doerr, B., Eremeev, A.V., Neumann, F., Theile, M., Thyssen, C.: Evolutionary algorithms and dynamic programming. Theoretical Computer Science 412(43), 6020–6035 (2011)
16. Skolicki, Z.: An Analysis of Island Models in Evolutionary Computation. PhD thesis, George Mason University, Fairfax, VA (2000)
17. Alba, E.: Parallel evolutionary algorithms can achieve super-linear performance. Information Processing Letters 82(1), 7–13 (2002)
18. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research 4(3), 233–235 (1979)
19. Lässig, J., Sudholt, D.: General upper bounds on the running time of parallel evolutionary algorithms. ArXiv e-prints (2012) Extended version of [10], `http://arxiv.org/abs/1206.3522`.