A Specialized Island Model and Its Application in Multiobjective Optimization

Ningchuan Xiao[1] and Marc P. Armstrong[1,2]

[1] Department of Geography, The University of Iowa, Iowa City, IA 52242, USA
{ningchuan-xiao,marc-armstrong}@uiowa.edu
[2] Program in Applied Mathematical and Computational Sciences
The University of Iowa, Iowa City, IA 52242, USA

Abstract. This paper discusses a new model of parallel evolutionary algorithms (EAs) called the specialized island model (SIM) that can be used to generate a set of diverse non-dominated solutions to multiobjective optimization problems. This model is derived from the island model, in which an EA is divided into several subEAs that exchange individuals among them. In SIM, each subEA is responsible (i.e., specialized) for optimizing a subset of the objective functions in the original problem. The efficacy of SIM is demonstrated using a three-objective optimization problem. Seven scenarios of the model with a different number of subEAs, communication topology, and specialization are tested, and their results are compared. The results suggest that SIM effectively finds non-dominated solutions to multiobjective optimization problems.

1 Introduction

Though the parallel characteristics of evolutionary algorithms (EAs) have been recognized for decades [14], intensive research on parallelism in EAs did not take place until the 1980s [4,12]. Since that time, two major types of parallel models have been designed for EAs [1]: cellular and distributed models. For a cellular model, the population of an EA is subdivided into a large number of subpopulations (each often containing one individual) and communication is only permitted among neighbor subpopulations. For a distributed model (also called an island or coarse grained model), a population is partitioned into a small number of subpopulations and each performs as a complete EA; a sparse exchange of individuals among subpopulations is conducted.

The goal of parallelizing an EA is to: 1) relieve the computational burden mainly imposed by evaluation (computing objective functions) and evolutionary processes (e.g., selection and recombination), and 2) improve EA results by increasing population diversity. While it has been reported that parallelization can be used to help with both goals [13,17], the use of parallel models in multiobjective optimization has mainly focused on the first [16,18,26], though some progress has recently been made on the second goal [15].

The purpose of this paper is to describe a new approach to modeling parallel EAs with a specific emphasis placed on multiobjective optimization. This

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1530–1540, 2003.

approach, called the specialized island model (SIM), is loosely based on the general concept used by the island model, and can be easily implemented on different kinds of parallel computers. In SIM, an EA is divided into a number of subEAs, and each is specialized to solve a modified version of a multiobjective optimization problem using a subset of the original objective functions. The use of SIM is demonstrated using a three-objective optimization problem; seven scenarios of the model with a different number of subEAs, communication topology, and specialization are tested, and their results are compared.

2 Background

The central idea in multiobjective optimization is Pareto optimality, which is derived from Vilfredo Pareto's original work [21]. To illustrate this concept, consider an optimization problem with k objectives:

$$\begin{array}{ll} \text{Minimize} & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})]^T \\ \text{subject to} & g_i(\boldsymbol{x}) \geq 0 \quad i \in [1, ..., m] \\ & h_i(\boldsymbol{x}) = 0 \quad i \in [1, ..., p] \ , \end{array} \tag{1}$$

where $\boldsymbol{x}$ is a vector of decision variables, g_i is an inequality constraint, and h_i is an equality constraint.

A solution $\boldsymbol{x}^*$ is said to dominate solution $\boldsymbol{x}$ (or $\boldsymbol{x}^* \prec \boldsymbol{x}$) if and only if

$$\forall i \ f_i(\boldsymbol{x}^*) \leq f_i(\boldsymbol{x}) \wedge \exists i \ f_i(\boldsymbol{x}^*) < f_i(\boldsymbol{x}), i \in \{1, ..., k\} \ . \tag{2}$$

In practice, especially when objectives conflict, it is often impossible to find a single optimum that dominates all other solutions. In contrast, it is common to encounter many non-dominated solutions. A solution $\boldsymbol{x}$ is said to be non-dominated if no other solution dominates it. Obviously, the ultimate goal of multiobjective optimization is to find all non-dominated solutions to a problem.

Numerous evolutionary methods have been developed to solve multiobjective optimization problems [5,7,8,10,26,27,28]. Most of these approaches use the global population structure of an EA and encourage the evolution of a set of diverse non-dominated solutions using a variety of techniques such as sharing, niching, and elitism. The method proposed in this paper is based on the island model of parallel EAs that allows each subpopulation to evolve in different "local" directions (in terms of the combination of objective functions) such that a diverse population will emerge. More importantly, the solutions found by each subpopulation can also be improved, through exchange of individual solutions among subpopulations.

3 Specialized Island Model

SIM is derived from concepts used in the island model of EAs. There are many variations of island models [2,3,12,19,23,25]. In general, in an island model, there are N subEAs (or subpopulations), each running in parallel as a complete EA.

A migration matrix $\mathbf{M} = \{0,1\}^{N\times N}$ is used to determine the communication topology among subpopulations. After a certain number of iterations (defined as migration frequency, m_f), a subpopulation P_i will send a number of individuals (defined as migration rate, m_r) to subpopulation P_j if and only if $m_{ij} = 1$, where $m_{ij} \in \mathbf{M}$. The total number of individuals that subpopulation i can receive is $m_r \cdot \sum_j^N m_{ij}$. After receiving individuals, an assimilation strategy is applied on each subpopulation to incorporate a portion of the "alien" individuals (defined as assimilation rate, m_a) into its own population. The formal procedure of the island model is described below, where PAR and END PAR blocks indicate the beginning and end of parallel sections.

Island Model

```
t := 0
PAR
    Initialize each subpopulation P_i(t), 1 ≤ i ≤ N
    Evaluate individuals in P_i(t)
END PAR
REPEAT
    PAR {regular EA }
        Evaluate individuals in P_i(t)
        Apply recombination and mutation on P_i(t)
    END PAR
    PAR
        Migration
    END PAR
    PAR
        Assimilation in P_i(t)
        Evaluate individuals in each subpopulation P_i(t)
    END PAR
    t := t+1
UNTIL termination criterion is satisfied
```

The difference between the SIM and island model is that, when SIM is used for multiobjective optimization, a subpopulation is not required to search for non-dominated solutions with respect to all objectives. Instead, some subpopulations are used only to search for solutions with respect to a subset of original objectives. This idea can be regarded as a generalized version of the vector evaluated genetic algorithm or VEGA [22], where individuals in each subpopulation are evaluated and selected using only one objective function. But in VEGA, crossover and mutation operate on the entire population, while in SIM these operations are used on each subpopulation. Figure 1 shows an example of a SIM with seven subpopulations for a problem with three objectives (f_1, f_2, f_3). Each node in the figure represents a subpopulation and the number(s) inside each node indicate the objective(s) for which the subpopulation is specialized. For instance, the node marked as "2, 3" is a subEA that is designed to search for optimal solutions to a problem with two objectives (f_2, f_3). Arrows in Fig. 1 indicate the source and destination of migration.

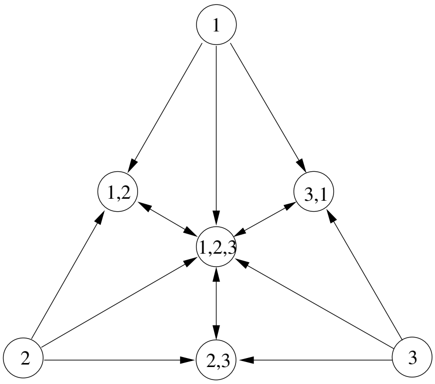


Fig. 1. An example of specialized island model for a 3-objective optimization problem

The specialization of a subEA can be depicted using a binary string $\{0,1\}^k$, where k is the number of objectives in the original problem, and there must be at least one "1" in the string. The i-th bit (starting from the left) is 1 if and only if the i-th objective in the original problem is included in the set of objectives for the subEA. Consequently, in Fig. 1, the specialization for the subEA marked as "2, 3" is "011".

4 Application of SIM

To demonstrate the use of SIM, the following optimization problem was designed:

$$\begin{array}{ll} \text{Minimize} & \boldsymbol{f} = [f_i(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})]^T \\ \text{subject to} & \boldsymbol{x} = (x_1, \ldots, x_m) \\ & 0 \leq x_i \leq 1, i \in \{1, \ldots m\} \\ & m = 30\ . \end{array} \tag{3}$$

The objective functions are defined as:

$$\begin{cases} f_1 = \sqrt{x_1^2 + x_2^2} \\ f_2 = h_2/g_2 \\ f_3 = g_3 \cdot h_3 \\ g_2 = 1 + 10 \cdot (m-1) - \sum_{i=2}^m [\log(x_i^2+1) - 10 \cdot \cos(4\pi x_i)] \\ h_2 = 1 - m \cdot \sqrt{x_1/g_2} \\ g_3 = 1 - \sqrt{x_1/g_3} - (x_1/g_3) \cdot \sin(10\pi x_i) \\ h_3 = (1 - \frac{9}{m-1} \cdot \sum_{i=2}^m x_i)\ , \end{cases} \tag{4}$$

Note that f_2 and f_3 are modified versions of functions originally suggested in [6,27]. More specifically, they are modified from functions f_2's in test problems T_4 and T_3, respectively, in [27]. The reason for the modification is that the original f_2 functions in T_4 and T_3 were designed for problems with two objectives and did not necessarily conflict with each other.

4.1 Settings

Seven scenarios of SIM were designed to test the use of SIM on the problem described in Equations 3 and 4. Each scenario will be called a model in the remainder of this paper. Differences between the models are based on the number of subEAs used, as well as the migration strategy and specialization for each subEA. A description of each model is given in Table 1 and the settings of each model are provided in Table 2.

The parameters used by each subpopulation in the seven models are listed in Table 3. Two selection techniques are used in each subEA: roulette and tournament. In Table 3, a positive tournament size indicates that only tournament selection is used and the tournament size is specified as such; a negative tournament size in Table 3 means that a selection method will be randomly chosen (chance for each method is 50%), and when a tournament approach is used, the tournament size is the absolute value of the negative number. For subEAs specialized for 2 or 3 objectives, a non-dominated sorting approach is used [10,24]; for single-objective subpopulations, a sharing technique is applied [11]. Elitism is used for single-objective subpopulations.

Migrations among the subpopulations are conducted by selecting m_r individuals from subpopulations that are specified as migration sources. This selection process uses a tournament approach with the tournament size equal to two. The selected individuals are sent to a destination according to the migration matrix (Table 2). The destination subEA receives individuals from all source subEAs and puts them into a pool. After a subEA has accepted all alien individuals migrated from other subEAs, m_a individuals are randomly selected from the pool and copied into its own population. When elitism is applicable (for single objective subEAs in this research), the best solution in the population must not be overwritten.

Table 1. Descriptions of the seven models

Model	Description
A	Seven connected subpopulations
A1	An isolated version of A, no connections among subpopulations
B	Four connected subpopulations, one for three objectives, three for two objectives
B1	An isolated version of B, no connections among subpopulations
C	Four connected subpopulations, one for three objectives, three for one objective
C1	An isolated version of C, no connections among subpopulations
D	One subpopulation, specialized for three objectives

Table 2. Settings for the seven models

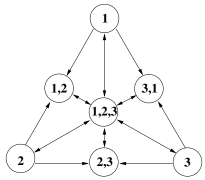
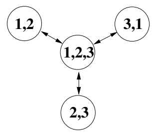
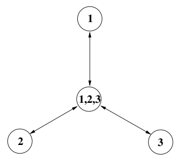
Model	Subpopulation (island)			Illustration
	Specialization	Minimizing	Migration matrix	
A	111	f_1, f_2, f_3	0 1 1 1 1 1 1	
	110	f_1, f_2	1 0 0 0 0 0 0	
	011	f_2, f_3	1 0 0 0 0 0 0	
	101	f_1, f_3	1 0 0 0 0 0 0	
	100	f_1	1 1 0 1 0 0 0	
	010	f_2	1 1 1 0 0 0 0	
	001	f_3	1 0 1 1 0 0 0	
A1	111	f_1, f_2, f_3	0 0 0 0 0 0 0	
	110	f_1, f_2	0 0 0 0 0 0 0	
	011	f_2, f_3	0 0 0 0 0 0 0	
	101	f_1, f_3	0 0 0 0 0 0 0	
	100	f_1	0 0 0 0 0 0 0	
	010	f_2	0 0 0 0 0 0 0	
	001	f_3	0 0 0 0 0 0 0	
B	111	f_1, f_2, f_3	0 1 1 1	
	110	f_1, f_2	1 0 0 0	
	011	f_2, f_3	1 0 0 0	
	101	f_1, f_3	1 0 0 0	
B1	111	f_1, f_2, f_3	0 0 0 0	
	110	f_1, f_2	0 0 0 0	
	011	f_2, f_3	0 0 0 0	
	101	f_1, f_3	0 0 0 0	
C	111	f_1, f_2, f_3	0 1 1 1	
	100	f_1	1 0 0 0	
	010	f_2	1 0 0 0	
	001	f_3	1 0 0 0	
C1	111	f_1, f_2, f_3	0 0 0 0	
	100	f_1	0 0 0 0	
	010	f_2	0 0 0 0	
	001	f_3	0 0 0 0	
D	111	f_1, f_2, f_3		

Table 3. Settings for each subEA

Parameter	Specialization of subEA		
	3-obj	2-obj	1-obj
Size of population	30	30	30
Tournament size	5	3	-3
Crossover probability	0.95	0.95	0.95
Mutation probability	0.15	0.1	0.1
Migration frequency	2	2	2
Migration rate	5	5	5
Assimilation rate	5	5	5
σ_{share} in the sharing function	0.3	0.6	0.6
α in the sharing function	1	1	1

4.2 Performance Metrics

In this research, each model was run 10 times, all solutions generated in the 10 runs were gathered, and the non-dominated ones were selected. Let O_i denote the non-dominated solutions generated by model i ($i \in \{$A, A1, B, B1, C, C1, D$\}$). Assume operation $|\cdot|$ returns the size of a set and $||\cdot||$ is a distance measurement. Two metrics (C and M_2^*) designed in [27] are used to compare the performance of the seven models.

$$C(O_i, O_j) = \frac{|\{b \in O_j \text{ and } \exists a \in O_i : a \prec b \text{ or } a = b\}|}{|O_j|} . \tag{5}$$

Function $C(O_i, O_j)$ measures the fraction of set O_j that is covered by (i.e., dominated by or equal to) solutions in O_i, and $0 \le C(O_i, O_j) \le 1$. $C(O_i, O_j) = 1$ if all solutions in O_j are covered by solutions in O_i; $C(O_i, O_j) = 0$ if no solutions in O_j are covered by those in O_i. Note that $C(O_i, O_j)$ is not necessarily equal to $1 - C(O_j, O_i)$. Generally speaking, if $C(O_i, O_j) > C(O_j, O_i)$, then model i is considered to generate more non-dominated solutions than model j. However, two models will be considered to have similar performance characteristics if $C(O_i, O_j) \approx C(O_j, O_i)$.

$$M_2^*(O_i) = \frac{1}{|O_i| - 1} \sum_{p \in O_i} |\{q \in O_i \text{ and } ||\boldsymbol{f}(p) - \boldsymbol{f}(q)|| > \sigma\}| . \tag{6}$$

Function M_2^* is essentially a measure of the extent of the objective function values in a multi-dimensional space formed by the objectives. A high value of M_2^* indicates the ability of the corresponding model to generate optimal solutions spanning a wide extent.

Two additional metrics were designed to compare the overall non-domination of each model. To calculate these metrics, the non-dominated solutions generated by each model are gathered into a single set, from which the overall non-dominated solutions are picked to form a set $\mathcal{O}$. Let $\mathcal{O}_i$ be the subset of $\mathcal{O}$ that is formed by solutions generated by model i.

$$X_i = \frac{|\mathcal{O}_i|}{\sum_j^N |\mathcal{O}_j|} \; . \tag{7}$$

$$E_i = \frac{|\mathcal{O}_i|}{|O_i|} \; . \tag{8}$$

Here, X_i indicates the overall domination of model i when it is considered with all other models. E_i measures how many non-dominated solutions generated by model i are still non-dominated when the results from all models are considered. E_i can be regarded as the efficiency of model i in generating overall non-dominated solutions. $0 \le E_i \le 1$ because $0 \le |\mathcal{O}_i| \le |O_i|$.

4.3 Results

Figure 2 shows the non-dominated solutions generated by all seven models. Table 4 shows the comparative results using function C, M_2^*, X_i, and E_i. The portion of this table relating to function C is a 7×7 matrix that can be examined along rows and columns. For row $i \in$ {A, A1, B, B1, C, C1, D}, the numbers in that row (denoted as $C(O_i, \cdot)$) indicate the tendency for solutions generated by model i to cover solutions from other models. For column j, the numbers in that column mean (denoted as $C(\cdot, O_j)$) indicate the tendency for solutions generated by model j to be covered by solutions from other models.

It was found that model A gives the best performance since $C(O_A, \cdot) > C(\cdot, O_A)$, and $M_2^*(O_A)$, X_A and E_A are all the highest values for each metric. It can also be noted that model B is relatively competitive with respect to function C. On the other hand, it was found that models C and C1 performed poorly, except that model C has a relatively large range; this is reasonable because model C is specialized to find extreme values for each objective. Model D gives the lowest value in M_2^*, indicating the disadvantage of a single population in finding extreme values for all objective functions. However, model D demonstrates a relatively high value in E_i, which suggests that including a subpopulation specialized in

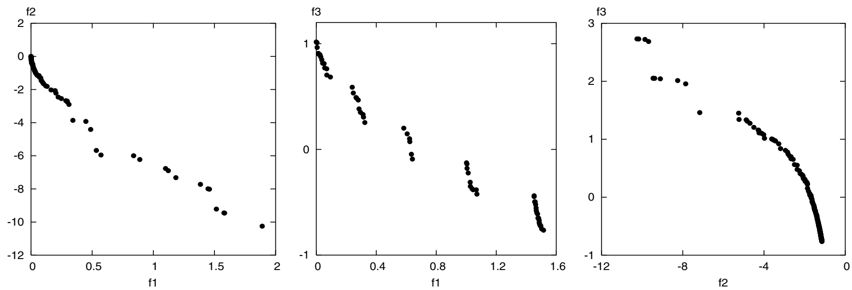


Fig. 2. Scattered plots showing non-dominated solutions for each pair of objective functions

Table 4. Comparison among seven models using metrics C, M_2^*, X_i, and E_i

Model	C							M_2^*	X_i	E_i
	A	A1	B	B1	C	C1	D			
A	-	0.44	0.29	0.39	0.82	0.44	0.26	12.077	0.320	0.63
A1	0.14	-	0.16	0.29	0.50	0.47	0.19	8.802	0.145	0.33
B	0.24	0.45	-	0.44	0.81	0.54	0.30	8.587	0.265	0.55
B1	0.15	0.30	0.15	-	0.52	0.45	0.19	9.059	0.146	0.36
C	0.02	0.12	0.02	0.13	-	0.18	0.09	9.458	0.013	0.07
C1	0.07	0.11	0.07	0.13	0.37	-	0.10	7.862	0.030	0.16
D	0.13	0.23	0.12	0.24	0.47	0.55	-	6.937	0.081	0.55

all objective functions can be very helpful in a SIM model. For models A, A1, B, B1, the connected models always outperform their isolated counterparts. This observation may be used to justify the use of migration among subpopulation in a SIM model.

5 Discussion and Conclusions

SIM is a model of parallel evolutionary algorithms for multiobjective optimization. It is relatively straightforward to implement SIM using tools such as the parallel virtual machine (PVM) [9] or message passing interface (MPI) [20] on different parallel computer architectures. The experimental results in this study suggest that, when connections are properly designed, SIM is effective in helping diversify the population of an EA, and ensures that a large number of non-dominated solutions are found. More generally, SIM can also be regarded as a hybrid approach in which each subEA can be designed to use a set of distinctive settings. Consequently, the entire EA is a hybridization of different evolutionary approaches that can be useful in areas outside multiobjective optimization.

The results generated by the SIM approach appear to be quite promising. However, additional research is needed on the following issues:

- Population size. This study used identical population sizes for all subpopulations, without considering the number of subpopulations. This, however, results in different population sizes among scenarios. Further research may be needed to determine the equivalency of the total population size for the entire EA among different scenarios according to the settings of their subpopulations.
- Niching parameters (α and σ_{share} in this study). EA outcomes are sensitive to the choice of niching parameters. It is necessary to conduct a sensitivity analysis of the results to these parameters so that each subpopulation is tuned to its best performance.
- Migration rate and frequency. These parameters play a critical role in diversifying the receiving subpopulation. A sensitivity analysis is needed to determine the optimal migration strategy for different settings.

– Assimilation rate. A higher assimilation rate can help diversify a population, but can also destroy the original population.

In addition, it is necessary to test SIM using a larger set of benchmark problems that include more than two objectives. Some guidelines for the design of such problems were suggested in [7, pp 346-348]. However, the problems obtained using these guidelines could not guarantee conflicts among objective functions from f_1 to f_{M-1}, where M is the number of objectives. To reflect problems encountered in the real-world, additional test problems with known optimal solutions are needed.

References

1. E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):443–462, 2002.
2. T.C. Belding. The distributed genetic algorithm revisited. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 114–121. 1995.
3. E. Cantú-Paz. *A Survey of Parallel Genetic Algorithms.* University of Illinois at Urbana-Champaign, 1997.
4. E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms.* Kluwer Academic Publishers, Boston, 2000.
5. C.A.C. Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.
6. K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.
7. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Chichester, 2001.
8. C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
9. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing.* MIT Press, Cambridge, MA, 1994.
10. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.
11. D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J.J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.
12. P.B. Grosso. *Computer Simulations of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model.* PhD thesis, University of Michigan, Ann Arbor, MI, 1985.
13. F. Herrera and M. Lozano. Gradual distributed real-coded genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1):43–63, 2000.
14. J.H. Holland. *Adaptations in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.
15. J. Knowles, editor. *The Second Workshop on Multiobjective Problem Solving from Nature (In association with The Seventh International Conference on Parallel Problem Solving from Nature)*, Granada, Spain, September 7-11 2002. http://iridia.ulb.ac.be/~jknowles/MPSN-II.

16. R.A.E. Mäkinen, J.Päriaux, M. Sefrioui, and J. Toivanen. Parallel genetic solution for multiobjective MDO. In A. Schiano, A. Ecer, J. Periaux, and N. Satofuka, editors, *Parallel CFD '96 Conference*, pages 352–359, Capri, 1996. Elsevier.
17. N. Marco and S. Lanteri. A two-level parallelization strategy for genetic algorithms applied to optimum shape design. *Parallel Computing*, 26:377–397, 2000.
18. N. Marco, S. Lanteri, J.-A. Desideri, and J.Périaux. A parallel genetic algorithm for multi-objective optimization in computational fluid dynamics. In K. Miettinen, M.M. Mäkelä, P. Neittaanmäki, and J.Périaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 445–456. Wiley & Sons, Chichester, UK, 1999.
19. W.N. Martin, J. Lienig, and J.P. Cohoon. Island (migration) models: evolutionary algorithms based on punctuated equilibria. In T. Bäck, D.B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C6.3:1–16. Oxford University Press/IOP, New York, 1997.
20. MPI Forum. *MPI-2: Extensions to the Message-Passing Interface.* University of Tennessee, 1997.
21. V. Pareto. *Manual of Political Economy.* Augustus M. Kelley, New York, 1971.
22. J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J.J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 93–100. Lawrence Erlbaum, Hillsdale, NJ, 1985.
23. F. Seredynski. New trends in parallel and distributed evolutionary computing. *Fundamenta Informaticae*, 35:211–230, 1998.
24. N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.
25. R. Tanese. Distributed genetic algorithms. In J.D. Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 434–439. 1989.
26. D.A. Van Veldhuizen and G.B. Lamont. Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.
27. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
28. E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello, and D. Corne, editors. *Evolutionary Multi-Criterion Optimization: Proceedings of the First International Conference*, Berlin, 2001. Springer.