

Parallel Strength Pareto Multi-objective Evolutionary Algorithm for Optimization Problems

Shengwu Xiong^{1,2}, Feng Li¹

¹School of Computer Science and Technology,
Wuhan University of Technology, Wuhan 430070, P.R.China
{xiongsw, lif}@mail.whut.edu.cn

²Projet Fractales, INRIA Rocquencourt
Domaine de Voluceau BP 105
78153 Le Chesnay Cedex, France

Abstract- Finding a good convergence and distribution of solutions near the Pareto-optimal front in a small computational time is an important issue in multi-objective evolutionary optimization. Previous studies have either demonstrated a good distribution with a large computational overhead or a not-so-good distribution quickly. Strength Pareto Evolutionary Algorithm (SPEA) produces a better distribution with larger computational effort. In this paper a Parallel Strength Pareto Multi-objective Evolutionary Algorithm (PSPMEA) is proposed. PSPMEA is a parallel computing model designed for solving Pareto-based multi-objective optimization problems by using an evolutionary procedure. In this procedure, both global parallelization and island parallel evolutionary algorithm models are implemented based on Java multi-threaded and distributed computation programtic technology separately. Each sub-population evolves separately with different crossover and mutation probability, but they exchange individuals in the elitist archive. The benchmark problems numerical experiment results demonstrate that the proposed method can rapidly converge to the Pareto optimal front and spread widely along the front.

1 Introduction

Most real-world engineering optimization problems are multi-objective in nature, since they normally have several (usually conflicting) objectives that must be satisfied at the same time. These problems are known as MOP (Multi-objective Optimization Problems) (Coello 1998). The notion of optimum has to be redefined in this context and instead of aiming to find a single solution. A procedure for solving MOP should determine a set of good compromises or trade-off solutions, generally known as Pareto optimal solutions from which the decision maker will select one. These solutions are optimal in the wider sense that no other solution in the search space is superior when all objectives are considered. Pareto optimal solutions form the Pareto front in a k -dimensional objective space, where k is the

number of the objectives in the optimization problem. Evolutionary Algorithms (EAs) have been recognized to be well suited for MOP. Unlike conventional methods that linearly combine multiple attributes to form a composite scalar objective function, evolutionary algorithm for MOP incorporates the concept of Pareto's optimality and modified selection schemes to evolve a set of solutions at multiple points along the trade-off surface simultaneously. Over the past decade, a number of multi-objective evolutionary algorithms (MOEAs) have been suggested (Fonseca 2000). It is clear from the recent studies that there are three distinct goals in the development of an MOEA: (i) The distance of the resulting non-dominated front to the Pareto-optimal front should be minimized. (ii) A good distribution of non-dominated solutions along the front is desirable. (iii) Achieving both the above tasks in a computationally fast manner is also important (Zitzler 1999).

In this paper, we have suggested a parallel MOEA based on the Zitzler's Strength Pareto Evolutionary Algorithm (SPEA). The SPEA produced a much better distribution at the expense of a large computational effort. Parallel computing has been widely applied to EAs (Alba 2002). In the case of MOEAs, parallelism has been successfully applied to the Strength Pareto Evolutionary Algorithm and Parallel Strength Pareto Multi-objective Evolutionary Algorithm (PSPMEA) is proposed in this paper. PSPMEA is a parallel computing model designed for solving Pareto-based multi-objective optimization problems by using an evolutionary procedure. In this procedure, both global parallelization and island parallel evolutionary algorithm models are used. Each sub-population evolves separately with different crossover and mutation probability, but they exchange individuals in the elitist archive. It is observed that PSPMEA is a promising diversity preservation with less computation effort MOEA.

In this paper, section 2 introduces the MOPs, while section 3 introduce the Strength Pareto Evolutionary Algorithm 2. Section 4 presents the models of parallel genetic algorithm used in this paper. In section 5, authors give detail descriptions of PSPMEA. Comparative experimental results and extensive analysis are given in

section 6. Finally, concluding remarks are summarized in section 7.

2 A Standard Multi-objective Optimization Problem

A multi-objective problem consisting of k objectives and m constraints defined as functions of decision variable set X can be represented, without loss of generality, as follows:

$$\text{Maximize } Z(x) = \{Z_l(x) : l = 1, 2, \dots, k\} \quad (1)$$

$$\text{S.T. } g_i(x) \leq 0, \forall i = 1, \dots, m \quad (2)$$

$$x \in X \quad (3)$$

where $x = \{x_j : j = 1, 2, \dots, n\}$ represents the decision vector, x_j is the j_{th} decision variable, D represents the decision space, $g_i(x)$ is the i_{th} constraint, $Z(x)$ is the multi-objective vector, and $Z_l(x)$ is the l_{th} objective function.

More rigorous definitions and related MO terminology are given by Zitzler (Zitzler 1999). Based on the definitions by Zitzler (Zitzler 1999) and notations used in Equations (1)-(3), the following are defined:

Pareto Dominance: A multi-objective vector $u = (u_1, u_2, \dots, u_k)$ is said to dominate $v = (v_1, v_2, \dots, v_k)$ (denoted by $u > v$) if and only if u is partially more than v , $\forall i \in \{1, 2, \dots, k\}, u_i \geq v_i \wedge \exists i \in \{1, 2, \dots, k\} : u_i > v_i$.

Pareto Optimality: A solution $x \in X$ is said to be Pareto optimal with respect to X if and only if there exists no $x' \in X$ for which $v = Z(x')$ dominates $u = Z(x)$.

Pareto Optimal Set: For a given multi-objective problem $Z(x)$, the Pareto optimal set P^* is a set consisting of Pareto optimal solutions. P^* is a subset of all the possible solutions in X . Mathematically, P^* is defined as follows:

$$P^* = \{x \in X \mid \neg \exists x' \in X : Z(x') > Z(x)\} \quad (4)$$

Pareto Optimal Front: The Pareto front, PF^* is the set that contains the evaluated objective vectors of P^* . Mathematically PF^* is defined as:

$$PF^* = \{u = Z(x) \mid x \in P^*\} \quad (5)$$

3 Strength Pareto Evolutionary Algorithm

In (Zitzler 1999), the author presented the SPEA. In (Zitzler 2000)(Zitzler 2001), an improved version, namely SPEA2, was proposed. Zitzler and Thiele suggested two important concepts in evolutionary multi-objective optimization, environmental selection and mating selection. Environmental selection requires an external archive which maintains the elitists in the population every evolutionary generation. Mating selection is based on the Pareto relation, it gives every individual a rank, if

two individuals have the same rank, density metric will be added to determine the final fitness. In principle, both selection schemes are completely independent from each other. However, with many evolutionary methods both concepts are implemented similarly. In SPEA2, these two concepts are implemented simultaneously. SPEA2 uses a fine-grained fitness assignment strategy which incorporates density information. In detail, each individual both dominating and dominated solutions are taken into account. At generation t , each individual in population P_t and external archive \bar{P}_t is assigned a strength value $S(i)$, representing the number of solutions it dominates:

$$S(i) = |\{j \mid j \in P_t + \bar{P}_t \wedge i > j\}|$$

where $>$ corresponds to the Pareto dominances relation. On the basis of the S values, the raw fitness $R(i)$ of an individual i is calculated:

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j > i} S(j)$$

The above raw fitness is determined by the strengths of its dominators in both archive and population. Although the raw fitness assignment provides a sort of niching mechanism based on the concept of Pareto dominances, it may fail when most individuals do not dominate each other. Therefore, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique used in SPEA2 is an adaptation of the k -th nearest neighbor method. Finally, adding density value $D(i)$ to the raw fitness value $R(i)$ yields its fitness $F(i) : F(i) = R(i) + D(i)$.

In environmental selection procedure, copy non-dominated individuals in P_t and P_{t+1} into next generation external archive \bar{P}_{t+1} according to fitness. If the size of \bar{P}_{t+1} exceeds the predefined size then reduce \bar{P}_{t+1} by means of the truncation operator, otherwise if the size of \bar{P}_{t+1} is less than the predefined size then fill \bar{P}_{t+1} with dominated individuals in P_t and \bar{P}_t . In mating selection procedure: perform binary tournament selection from \bar{P}_{t+1} and fill the mating pool P_{t+1} .

4 Parallel Evolutionary Multi-objective Optimization

Parallel Genetic Algorithms (PGAs) are naturally prone to parallelism since the genetic operations on the individuals of the population can be easily undertaken in parallel (evaluate operator and breed operator). A survey of parallelism strategies applied to GAs is provided in (Cantu 1998). In global parallelization, explicit parallelization of the genetic operators and/or evaluations of individuals are performed. The algorithm proceeds in the same way as a sequential GA, but in a faster manner. The rest of the PGA models fit into two classes,

depending on their computation/communication ratio, called coarse grain and fine-grain parallel GAs. Coarse grain PGAs are also known as distributed (dGA) or island GAs, and fine grain PGAs are known as cellular (cGA), diffusion or massively-parallel GAs. A dGA has a bigger subpopulation size than a cGA but fewer of subpopulations and less coupled (Cantu 1999)(Capcar 1999)(Alba 1999). Although much works are reported about PGA models (and implementations on different parallel architectures), involving SOPs, PGA models could also be applied to MOPs, where global parallelization strategy is mostly used. However, in our work, both global parallelization model and coarse grain model are used. In every sub-population, individual's evaluation and breeding operator implementation are performed concurrently. This is the global parallelism model. So better performance of computation is gained. In island model, several sub-populations are maintained and located on different islands. Each subpopulation performs local evolutionary procedure and exchanges individuals in their populations every interval generations. This procedure guarantees the diversity of individuals in the whole population. Along with the continuous evolutionary procedure, the best individuals are found.

5 Parallel Strength Pareto Evolutionary Algorithm

The PSPMEA presented in this paper is based on SPEA2 (Zitzler 2001). Here, we give the description of PSPMEA.

Parameters:

N (population size)

\bar{N} (archive size)

T (maximum number of generations)

G_f (frequency of migration)

G_n (number of migrating individuals)

step1: Generate initial population P_0 , empty the external archive \bar{P}_0 , Set $t = 0$.

step2: Evaluate the individuals in subpopulation and archive. First calculate every individual's $S(i)$, then calculate the raw fitness $R(i)$ according to $S(i)$. Finally the SPEA2 fitness is gained (Zitzler 2001).

step3: Copy nondominated individuals in subpopulation P_t and archive \bar{P}_t to \bar{P}_{t+1} .

step4: if $t = T$, then evolutionary procedure finished, else continue.

step5: Yield the P_{t+1} by using binary tournament selection from elitists archive \bar{P}_{t+1} .

step6: if $t \bmod G_f = 0$, pick G_n individuals from \bar{P}_{t+1} to the sending pool by using binary tournament selection, else go to **step8**.

step7: Send individuals in sending pool to other subpopulation's mailbox.

step8: Apply crossover and mutate operator to the new subpopulation P_{t+1} .

step9: if $t \bmod G_f = 0$, receive individuals from mailbox. Determine which individuals would be replaced by immigrants in mailbox by using binary tournament selection, then use immigrants to replace individuals in subpopulation. Count $t = t + 1$, and go to **step2**.

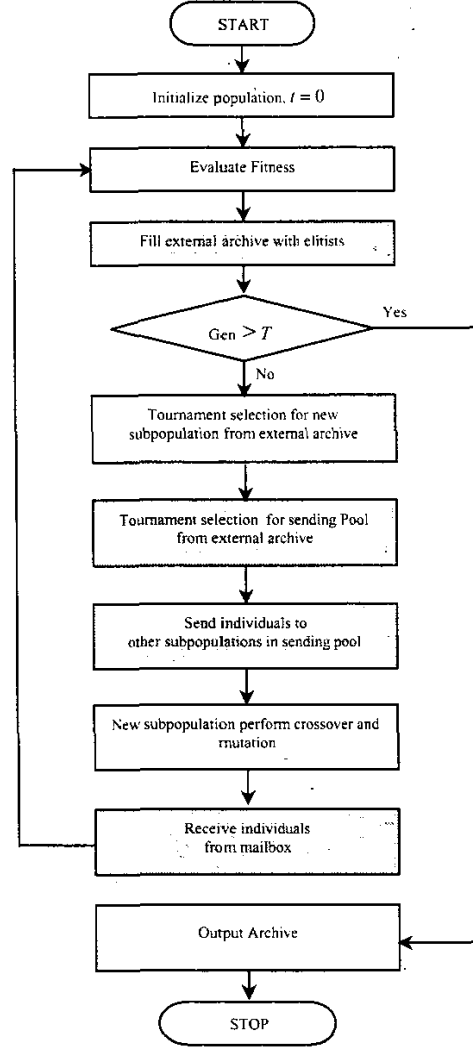


Figure 1: Program flowchart of PSPMEA

In PSPMEA, **step2** and **step8** implement global parallel model. Java language offers multithread mechanism, so we can use java multithread mechanism to implement evaluation and breeding procedure concurrently. According to the number of threads, we divide the subpopulation into chunks and dispatch to every thread, so locally, if there are SMP computers, the evaluation and breeding procedure require less time and gain higher computation speed. **step7** and **step9** implement synchronized island model. Figure 1 shows the

program flowchart of the whole algorithm. Java techniques also offer distributed computation mechanism. In island model, every island maintains a population and an external archive, every population evolves separately with different crossover and mutation probability. After migration interval (predefined generations), every population selects migration size individuals from archive to sending pool, then sends individuals to other population's mailbox. Because we implement synchronize model, so at the same time the population also receives individuals from its own mailbox. For an immigrant we use the following rules: If the immigrant is better than the "to-die" individual, replace the "to-die" individual. Else the "to-die" individual "revive" and the immigrant is abandoned. Populations' individual migration strategy assures the diversity in whole population, while in every sub-population, we also handle individuals in parallel. So the elitists can converge to the Pareto optimal front rapidly.

6 Experiment Design

To evaluate PSPMEA and compare it with SPEA2 algorithms we have used the continuous benchmark functions used by Zitzler (Zitzler 1999) and bi-objective traveling salesman problem (Hansen 2000). The Evolutionary Computation in Java (ECJ), version 9 (Luke 2001), is used in the experiment, where each problem is not available in distribution yet. We design each problem and let it incorporate with the other parts of the system correctly.

6.1 Continuous Test Problems

These functions are selected by taking into account a wide range of features that may cause difficulties for an MOEA such as convexity (function ZDT1), non-convexity (ZDT2), discreteness (ZDT3), multimodality (ZDT4), and non-uniformity (ZDT6). The deceptive function ZDT5 is not used because in PSPMEA the individuals in the population are coded as real numbers. For performance comparison, we used the hyper-volume metric (Zitzler 1999) to evaluate convergence of the obtained solutions to Pareto optimal front, Δ metric (Deb 2002) to evaluate distribution and computation time to evaluate algorithm efficiency. For comparison of two non-dominated solution sets A_i and A_j , convergence to Pareto optimal front, the following measures are computed: $S(A_i)$ is the volume of the space that is dominated by the set A_i . $D(A_i, A_j)$ is the volume of the space that is dominated by the first solution set A_i but non-dominated by the second set A_j . Here, minimization problems are considered. The smaller $S(A_i)$ and $D(A_i, A_j)$, the better A_i related to A_j . The article (Deb 2002) defines a different metric to measure the spread in solutions obtained by an algorithm directly. The metric Δ measures the extent of spread achieved among the obtained solutions. Δ is defined as follow:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}$$

Here, the parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set. The parameter \bar{d} is the average of all distances $d_i, i = 1, 2, \dots, (N-1)$, assuming that there are N solutions on the best non-dominated front. With N solutions, there are $(N-1)$ consecutive distances. Thus the metric Δ takes a higher value with worse distribution of solutions within the extreme solutions.

In all experiments, we use three subpopulations with different P_m and P_c , the first subpopulation $P_m=0.8$, $P_c=0.01$, the second subpopulation $P_m=0.9$, $P_c=0.02$, the third subpopulation $P_m=1.0$, $P_c=0.03$. The first subpopulation assures the steady state evolutionary procedure, the other sub-populations' goal is to search solutions in large scale. Choosing of number of isolated population is based on the complexity of problem. Here, 3 sub-populations are enough to solve these benchmark problems. In PSPMEA all subpopulation and archive size are 100 respectively, the maximum generation is 250, as in (Zitzler 1999). The whole function evaluation is 75,000, for comparison, in SPEA2 we set the population size to 300 and external archive size to 100, the maximum generation is also set to 250. The selection mechanism used in PSPEA and SPEA2 is binary tournament selection. We have used the SBX recombination operator (Deb 1995) and the polynomial mutation operator (Deb 1996) to create an offspring solutions in the two algorithms. For each test problem, each algorithm is run with five different initial populations. Table 2 compares PSPMEA and SPEA2 by using the previously described hyper-volume metric and Δ metric. We also present the computational time needed to run each algorithm through the same function evaluations on the same computers (1.4GHz Pentium IV Processor). In addition, graphical representations of the final non-dominated sets for both PSPMEA and SPEA2 are provided in Figures 2-6 for ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6.

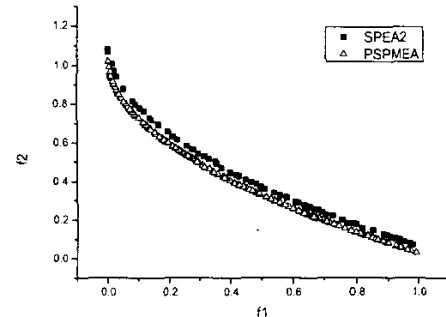


Figure 2: Solutions for ZDT1

Test Problem	MOEA	Hyper-volume		Δ		Time(sec)	
		Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
ZDT1	PSPMEA	0.3651315490	0.000147	0.47868	0.00441	41.109	2.212
	SPEA2	0.4048830111	0.000191	0.59709	0.00645	95.937	3.346
ZDT2	PSPMEA	0.7303524184	0.000103	0.44783	0.00520	42.094	1.355
	SPEA2	0.7763868883	0.000164	0.60341	0.00732	87.187	3.443
ZDT3	PSPMEA	0.2129300549	0.000131	0.28201	0.00217	49.281	1.621
	SPEA2	0.2480549939	0.000223	1.08581	0.00356	77.922	2.782
ZDT4	PSPMEA	0.4509876012	0.000202	0.54710	0.00501	68.063	1.317
	SPEA2	0.9873948323	0.000161	0.62198	0.00624	376.406	7.284
ZDT6	PSPMEA	2.5495801537	0.000144	0.14395	0.01132	78.062	1.399
	SPEA2	2.6949921922	0.000193	0.15234	0.01252	416.047	9.896

Table 1: Performance Comparison of the Two Algorithms for ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6

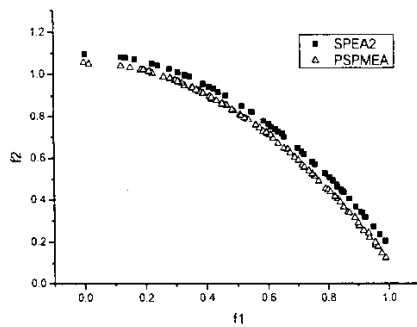


Figure 3: Solutions for ZDT2

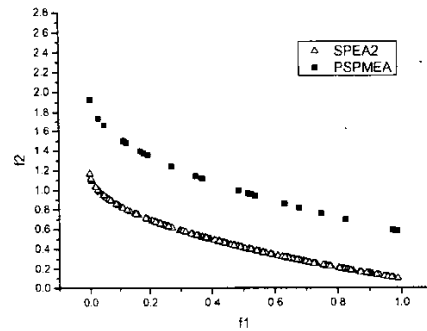


Figure 5: Solutions for ZDT4

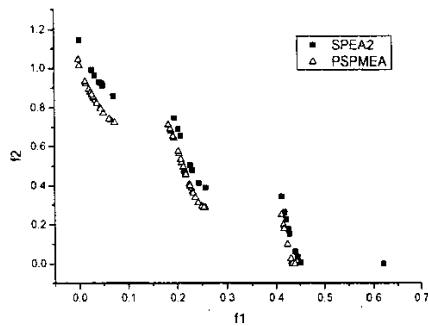


Figure 4: Solutions for ZDT3

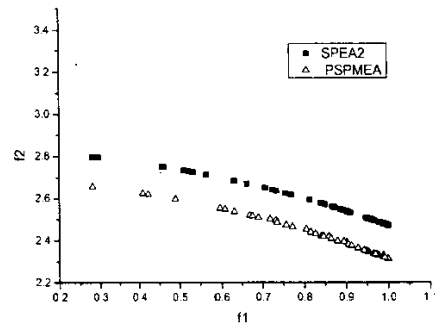


Figure 6: Solutions for ZDT6

The function ZDT1 has a convex Pareto optimal front. From the experiment results we can observe that PSPMEA obtains the better convergence and a good distribution of solutions in a less computational time. The ZDT2 tests a MOEA's ability to find non-convex Pareto optimal solutions. Like in ZDT1, the PSPMEA demonstrates good convergence and distribution in a short computational time. In the case of ZDT3 (discrete), PSPMEA is better than SPEA2 in every discrete area. From the ZDT4 and ZDT6 results comparisons, we find that for multimodal problem, individual's exchanging between elitism archive helps the algorithm converge to the Pareto optimal front and explore widely distributed front. Finally, it can be observed that the biased search space together with the non-uniform represented Pareto-optimal front (ZDT6) does not make it difficulty for the PSPMEA, and PSPMEA outperforms SPEA2. Although the whole population size in both algorithms is same, it is obvious that distributed computation makes good use of computational resource and reduces the computational effort.

6.2 Combinatorial Test Problems

The general traveling salesman problem is defined by a number l of cities and matrix $c = (c_{i,j})$ which gives for each ordered pair (i, j) of cities the non-negative distance $c_{i,j}$ to be covered to get from city i to city j . The optimization goal is to find the shortest route for which each city is entered and left exactly once. By adding an arbitrary number of distance matrices, this SOP can be transformed to an MOP. Formally, given l cities and a set $\{C_1, C_2, \dots, C_k\}$ of $l \times l$ matrices with $C_h = (c_{i,j}^h)$, minimize $f(\pi) = (f_1(\pi), f_2(\pi), \dots, f_k(\pi))$ with $f_i(\pi) = \left(\sum_{j=1}^{l-1} c_{\pi(j), \pi(j+1)}^i \right) + c_{\pi(l), \pi(1)}^i$ and where π is a permutation over the set $\{1, 2, \dots, l\}$ (Zitzler 1999).

test functions except the number of generations which was set to 500. We ran the two algorithms 5 times on the instance independently. Table2 also compares PSPMEA and SPEA2 by using the same metrics as in continuous benchmark problems. Graphical representations of the final non-dominated sets for both PSPMEA and SPEA2 are also provided in Figure 7 for the instance.

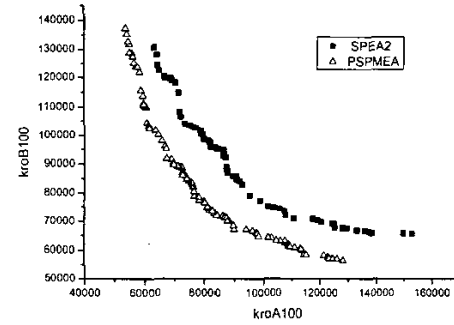


Figure 7: Solutions for(kroA100,kroB100)

It is obvious that PSPMEA outperforms SPEA2 on multi-objective combinatorial problems. After several runs, we found that the fronts produced by PSPMEA almost dominate the fronts produced by SPEA2, and better distribution and spread of Pareto optimal set are gained with less computational effort.

7 Concluding remark

The trade-off between obtaining a well-converged and well-distributed set of Pareto-optimal solutions and obtaining the Pareto-optimal solutions in a small computational time is an important issue in multi-objective evolutionary optimization. In this paper we have

Test Problem	MOEA	Hyper-volume		Δ		Time(sec)	
		Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
(kroA100	PSPMEA	13053347368.1554	23408.9002	0.56130133	0.0081	392.890	5.128
,kroB100)	SPEA2	15509478402.9202	35654.7784	0.59285765	0.0106	492.735	7.293

Table 2: Performance Comparison of the Two Algorithms for (kroA100, kroB100)

As benchmark instances we considered one paired combination of the benchmark 100-city instances kroA100 and kroB100. The benchmark instance was defined in (Hansen 2000) and was also used in (Borges 2000) (Jaszkiewicz 2002). The experiments were carried out using the same parameter settings as the continuous

suggested a Parallel Strength Pareto Evolutionary Algorithm (PSPMEA). PSPMEA is a parallel implementation (using a PGA model) of the Strength Pareto Evolutionary Algorithm 2 (SPEA2), a new elitist MOEA with a fine grained fitness assignment strategy, density estimation technique and an enhanced archive

truncation method. Based on Java multi-threaded technology we implement the global parallel model in PGA: Distributed computation provides us another method to implement parallel island model. The performance of PSPMEA has been analyzed in comparison with SPEA2. The experimental results show that PSPMEA clearly outperforms SPEA2 for the continuous benchmark set and multi-objective combinatorial problems. The multi-subpopulation elitism archive exchanging allows a fast convergence without premature convergence effects. PSPMEA reduces the execution time of SPEA2. Moreover, a better convergence is sometimes observed in PSPMEA for some continuous functions of the benchmark set. The next steps for continuing this research are the study of more complex problems such as three-objective or four-objective optimization problems.

Acknowledgments

We would like to thank Sean Luke ([Luke01]) for the ECJ library and Dr. Zitzler for discussion about SPEA ([Zitzler99]). We would also like to thank four unknown referees for their many useful comments and constructive suggestions. This work was in part supported by the National Science Foundation of China (NSFC) under the grant No.60173046 and "863" project of China under the grant No. 2002AA1Z1490.

Bibliography

- [Alba99] Alba, E. and Troya, J. M. (1999) "A survey of parallel distributed genetic algorithms", *Complexity*, vol.4, no. 4, pp.31-52.
- [Alba02] Alba, E. and Tomassini, M. (2002) "Parallelism and Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5.
- [Borg00] Borges, P.C. and Hansen, P.H. (2000) "A study of global convexity for a multiple objective traveling salesman problem", In C.C. Ribero and P.Hansen, editors, *Essays and Surveys in Metaheuristics*, pp.129-150.
- [Cant98] Cantu-Paz, E. and Goldberg, D. E. (1998) "Predicting speedups of idealized bounding cases of parallel genetic algorithms", in *Proc. 7th Int. Conf. Genetic Algorithms*, T.Back, Ed., pp. 113-120.
- [Cant99] Cantu-Paz, E. and Goldberg, D. E. (1999) "On the scalability of parallel genetic algorithms", *Evolutionary Computation*, vol. 7, no. 3, pp.429-449.
- [Capc99] Capcarrere, M., Tomassini, M. Tettamanzi, A. and Sipper, M. (1999) "A statistical study of a class of cellular evolutionary algorithms", *Evolutionary Computation*, vol. 7, no. 3, pp.255-274.1999.
- [Coel98] Coello, C. A. C. (1998) "An Updated Survey of GA-Based Multi-objective Optimization Techniques", Technical Report Lania-RD-98-08, Laboratorio Nacional de Informática Avanzada (LANIA).
- [Deb95] Deb, K. and Agrawal, R. B. (2000) "Simulated binary crossover for continuous search space", *Complex Systems*, 9(2), pp.115-148.
- [Deb96] Deb, K. and Goyal, M. (1996) "A combined genetic adaptive search (GeneAS) for engineering design", *Computer Science and Informatics*, 26(4), pp.30-45.
- [Deb02] Deb, K., Agrasal, S., Pratap, A. and Meyarivan, T. (2002) "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, 6(2), pp.182-197.
- [Fons00] Fonseca, C. M. and Fleming, P. J., (2000) "Multiobjective optimization", In Th. Back, D. B. Fogel, and Z. Michalewicz, editors, *Evolutionary Computation*, volume 2, pp. 25-37.
- [Hans00] Hansen, M.P. (2000) "Use of substitute scalarizing functions to guide a local search base heuristics: the case of motsp", *Journal of Heuristics*, 6, pp.419-431.
- [Jasz02] Jaszkiewicz, A. (2002) "Genetic local search for multiple objective combinatorial optimization", *European Journal of Operational Research*, 1(137), pp.50-71.
- [Luke01] Luke, S. An Evolutionary Computation and Genetic Programming System. <http://www.cs.umd.edu/projects/plus/ec/ecj>
- [Zitz98] Zitzler, E. and Thiele, L. (1998). "An evolutionary algorithm for multi-objective optimization: The strength Pareto approach", Technical Report No. 43 (May 1998), Zürich: Computer Engineering and Networks Laboratory, Switzerland.
- [Zitz99] Zitzler, E. (1999) "Evolutionary algorithms for multiobjective optimization: Methods and applications", Doctoral thesis ETH NO. 13398, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany: Shaker Verlag.
- [Zitz00] Zitzler, E. and Deb, K. and Thiele, L. (2000) "Comparison of Multi-objective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation Journal*, 8(2), pp.125-148.
- [Zitz01] Zitzler, E., Laumanns, M. and Thiele, L. (2001) "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", *TIK – Report 103*.