Adaptive mechanisms for multiobjective evolutionary algorithms

Basseur Matthieu, Seynhaeve Franck, Talbi El-ghazali
Laboratoire d'Informatique Fondamentale de Lille (LIFL), University of Lille, France.
E-mail: {basseur,seynhaev,talbi}@lifl.fr

Abstract— **Many real-world optimization problems are multiobjective. This paper proposes an Adaptive Genetic/Memetic Algorithm (AGMA) with a multiobjective approach applied to a flow shop scheduling Problem (FSP). AGMA is firstly a genetic algorithm (GA) which proposes an adaptive selection between mutation operators. Moreover, AGMA proposes an original hybrid approach, the search alternates adaptively between a Genetic Algorithm and a Memetic Algorithm (MA). We test AGMA on multiobjective FSP (MOFSP). We use different performance indicators to compare with classical algorithms and with our previous work on adaptive mechanism for Genetic Algorithm applied to FSP.**

I. INTRODUCTION

A large part of real-world optimization problems are of multiobjective nature. In trying to solve Multiobjective Optimization Problems (MOPs), many methods scalarize the objective vector into a single objective. Since several years, interest concerning MOPs area with Pareto approach always grows. Many of these studies uses Evolutionary Algorithms to solve MOPs [CVL02] [Deb01] [ZDT+01].

The efficiency of multiobjective genetic algorithms is characterized by several important mechanisms, such as genetic operators (i.e. mutation and crossover), selection, diversification and hybridization.

In our previous works [TRMD01], we compare several classical selection and diversification strategies in order to select the best ones. Then we show the interest of hybridization with local search. This study was applied on a MOFSP.

Then, in [BST02] we give some guidelines to set automatically some of the numerous GA parameters. We propose an adaptive mutation selection method, and a dynamic niche size computation for sharing diversification [Gol89]. We also propose a hybridization by memetic algorithm. These mechanisms were tested successfully on MOFSP.

In this paper, we present AGMA, which is applied to MOFSP. AGMA proposes an new adaptive selection between mutation operators, which try to face drawbacks of our previous approach . Moreover, AGMA proposes an original hybrid approach in which the search alternates adaptively between GA and MA.

We use different performance indicators to compare AGMA with classical algorithms [TRMD01] and our previous work on adaptive mechanisms for GA [BST02].

This paper is organized as follows:

In section II, we define MOP and we present MOFSP. In section III, we present an adaptive mutation selection operator, that performs different mutation operators, the previous results obtained by each operator help in determining the mutation operator to apply. In section IV, we present GA hybridization by a memetic search, and a cooperative method between these two complementary methods. Then we present our results obtained for MOFSP in section V. These results are compared with multiobjective performance indicators. In the last section, we will discuss on AGMA effectiveness and perspectives of this work.

II. MULTI-OBJECTIVE FSP

A. Multiobjective Optimization Problem

Firstly, we have to describe and define MOPs in a general case. We assume that a solution to such a problem can be described by a decision vector $(x_1, x_2, ..., x_n)$ in the decision space X. A cost function $f : X \rightarrow Y$ evaluates the quality of each solution by assigning it in an objective vector $(y_1, y_2, ..., y_p)$ in the objective space Y (Fig. 1). So, multiobjective optimization consists in founding the solutions in the decision space minimizing or maximizing p objectives.

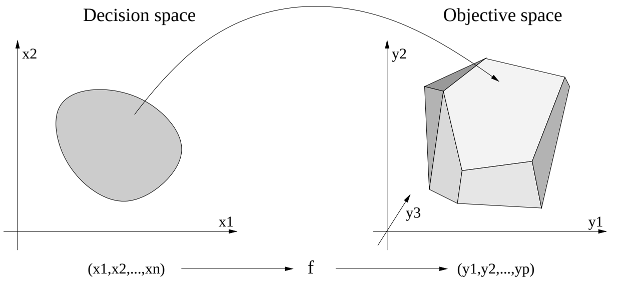

Fig. 1. Example of MOP

For the following definitions, we consider the minimization of p objectives. For maximization problems, definitions are similar. In the case of a single objective optimization, comparison between two solutions x^1 and x^2 is immediate. If $y^1 < y^2$ then x^1 is better than x^2. For multiobjective optimization, comparing two solutions x^1 and x^2 is more complex. Here, we only have a partial order relation, known as Pareto dominance concept:

Definition 1: A solution x^i dominate a solution x^j if and only if:

$$\forall k \in [1..p], f_k(x^i) \leq f_k(x^j)$$

$$\exists k \in [1..p], f_k(x^i) < f_k(x^j)$$

In MOP, we are looking for non-dominated Pareto solutions:

Definition 2: A solution is Pareto optimal if it is not dominated by any other solution of the feasible set.

The set of optimal solutions in the decision space X is denoted as Pareto set, and its image in the objective space is the Pareto front. In fig. 2, the circle solutions represent a non-dominated front for two objectives.

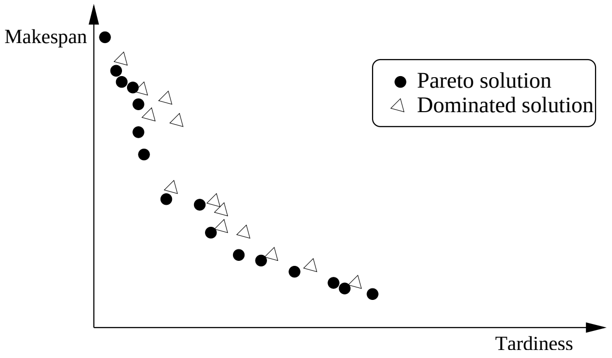

Fig. 2. Example of non-dominated solutions

B. Flow-shop Scheduling Problem

The FSP is one of the numerous scheduling problems. Flow-shop problem, has been widely studied in the literature. The proposed methods for its resolution vary between exact methods, as the branch & bound algorithm, specific heuristics and meta-heuristics. However, the majority of works on flow-shop problem studies the problem in its single criterion form and aims mainly to minimize makespan i.e. the total completion time, the date when the last job is terminated on the last machine. Several bi-objective approaches exist in the literature. Sayin et al. proposed a branch and bound strategy to solve the two-machine flow-shop scheduling problem, minimizing makespan and sum of completion time [SK99]. Sivrikaya-Serifoglu et al. proposed a comparison of branch & bound approaches for minimizing makespan and weighted combination of the average flowtime, applied to the two-machine flow-shop problem [SU98]. Rajendran proposed a specific heuristic to minimize makespan and total flowtime [Raj95]. Nagar et al. proposed a survey of the existing multicriteria approaches of scheduling problems [NHH95].

FSP can be presented as a set of N jobs $J_1, J_2, \ldots, J_N$ to be scheduled on M machines. Machines are critical resources: one machine cannot be assigned to two jobs simultaneously. Each job J_i is composed of M consecutive tasks $t_{i1}, \ldots, t_{iM}$, where t_{ij} represents the j^{th} task of the job J_i requiring the machine m_j. To each task t_{ij} is associated a processing time p_{ij}. Each job J_i must be achieved before the due date d_i. In our study, we are interested in permutation FSP where jobs must be scheduled in the same order on all the machines (Fig. 3).

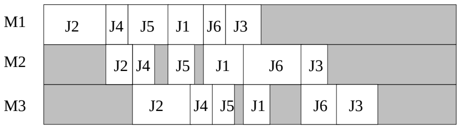

Fig. 3. Example of permutation Flow-Shop scheduling

In our study, we minimize two objectives: C_{max}, the makespan (Total completion time), and T, the total tardiness. Each task t_{ij} being scheduled at the time s_{ij}, the two objectives can be computed as follow:

$$C_{max} = Max\{s_{iM} + p_{iM} | i \in [1 \ldots N]\}$$

$$T = \sum_{i=1}^{N} [max(0, s_{iM} + p_{iM} - d_i)]$$

In the Graham & al. [GLLK79] notation, this problem can be defined as: F/perm, $d_i/(C_{max}, T)$

C_{Max} minimization has been proved to be NP-hard in [LKB77]. The total tardiness objective T, has been studied only a few times for M machines [Kim95], but total tardiness minimization for one machine has been proved NP-hard [DL90].

The evaluation of the performance of our algorithm has been realized on some Taillard benchmarks for the FSP [Tai93], extended to the bi-objective case [TRMD01] (the bi-objective benchmarks and the results obtained are available on the web at **http://www.lifl.fr/~basseur**).

To find a good Pareto set for multiobjective problems, evolutionary algorithms seem to be well suited. The next section presents different mechanisms to improve the effectiveness of GA search.

III. AN ADAPTIVE MUTATION SELECTION

A. Previous work

In [BST02], an adaptive mechanism of mutation operators was presented. The purpose is to change the probability selection of each operator according to its efficiency. To do that, each mutation M_i applied to the individual I was associated with a progress value (I_{M_i} is the individual I modified by the mutation M_i):

$$\Pi(I_{M_i}) = \begin{cases} 1 & \text{if } I \text{ is dominated by } I_{M_i} \\ 0 & \text{if } I \text{ dominates } I_{M_i} \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

At the end of each GA generation, a *Progress* value is assigned to each operator:

$$Progress = \frac{\sum \Pi(I_{M_i})}{\#M_i}$$

where $\#M_i$ is the number of applications of the mutation M_i on the population. The new selection probabilities are computed proportionally to these values.

B. A ranking approach

The previous approach of progress computation compares two solutions with their dominance relation. However, a comparison only between I and I_{M_i} is not sufficient. Firstly, the fact that I and I_{M_i} do not dominate each other is not enough to evaluate the quality of the mutation (Fig. 4). Secondly, if a generated solution dominates the initial solution, we can not measure with precision the progress realized.

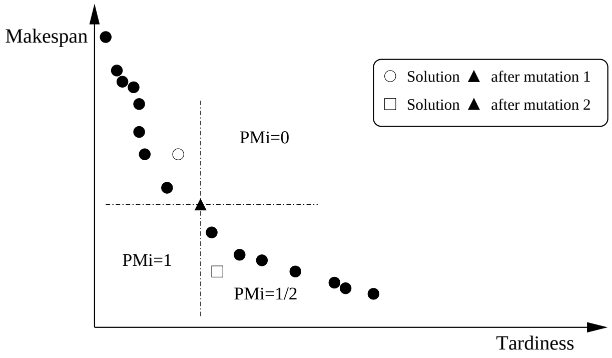

Fig. 4. Progress evaluation based on dominance relation between solutions

These problems can be tackled in the case of evolutionary algorithms using selection by ranking (in our case, we use NSGA ranking [SD94])). The progress value can be replaced by:

$$\Pi(I_{M_i}) = \left(\frac{R_I}{R_{I_{M_i}}}\right)^k$$

where $R_{I_{M_i}}$ is the rank of the solution after mutation, R_I is the rank of the solution before mutation, and k is how much we encourage the progress made by mutation operators. For our application, we set k to 2.

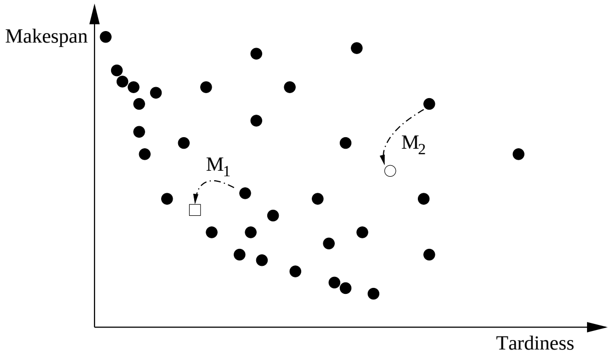

Fig. 5. Example of mutation applications

The interest of the improvement realized by mutation on different solutions is not the same for each solution in the population. The progress realized on good solutions are most important for the front progression (Fig. 5). So we introduce an elitist factor in our last progress indicator:

$$\Pi(I_{M_i}) = C_{I_{M_i}} * \left(\frac{R_{I_{M_i}}}{R_I}\right)^k$$

with $C_{I_{M_i}} = \frac{1}{R_{I_{M_i}}}$

Then, the global progress of a mutation M_i is defined as follows:

$$Progress(M_i) = \frac{\sum \Pi(I_{M_i})}{\sum C_{I_{M_i}}}$$

C. Application to the Flow-Shop Problem

In our application on FSP, we consider four mutation operators. The first operator is an exchange between two jobs (fig. 6). The second one, the insertion operator, consist in a circular permutation between two random points (fig. 7). The third operator randomly re-arranges a sequence between two points (fig. 8). The last operator, 2-opt operator, reverses the sequence of jobs between two random points (fig. 9).

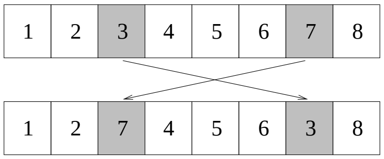

Fig. 6. exchange mutation operator

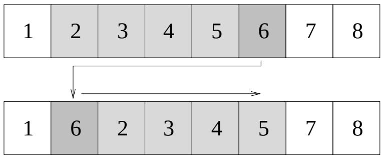

Fig. 7. insertion mutation operator

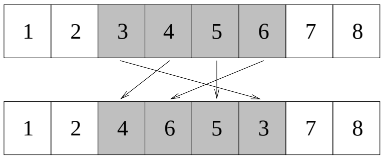

Fig. 8. random mutation operator

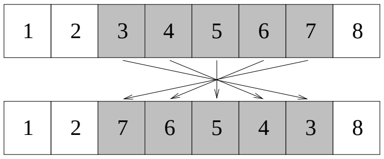

Fig. 9. 2-opt mutation operator

According to experiments, the random operator seems effective for short problems (because of their small complexity), and the exchange and permutation operators perform better on great problems. The 2-opt operator performs worsly than the other operators. Fig. 10 and 11 show the evolution of the selection probabilities of these operators in function of the number of GA generations done. These evolutions are dependent to the problem treated.

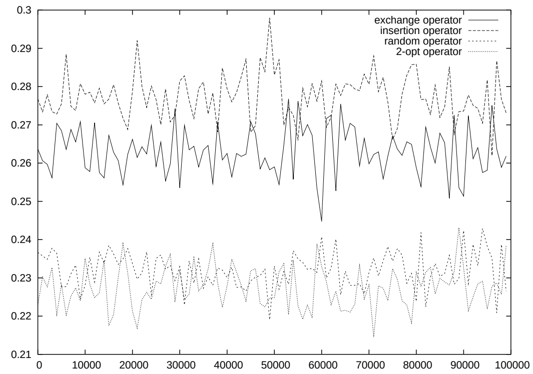

Fig. 10. Mutation selection probabilities: example on instance ta_31bi (50 jobs*5 machines)

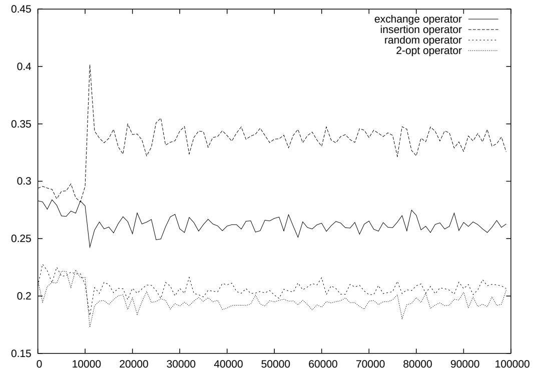

Fig. 11. Mutation selection probabilities: example on instance ta_61bi (100 jobs*5 machines)

IV. ADAPTIVE MEMETIC SEARCH

In single objective optimization, it is well known that GA must be hybridized by local search algorithm to give good results. In fact, GA convergence is too slow to be really effective without any hybrid approach. In [TRMD01], we show that for FSP, local search improves the results obtained by the GA. In [BST02], we test hybridization by memetic algorithm. We run several generations of crossover+local search on the non-dominated set of solutions obtained by the GA. Algorithm 1 describes the Genetic Algorithm hybridized by Memetic Algorithm (GA+MA):

Crossover, selection and diversification operators are described in [BST02]. This hybridization method gives good results. To improve this method we propose to define dynamically transitions between GA and MA. Our idea is to assign a progress value P during the meta-heuristic run. Let P_{PO*} be the value of the modification rate done on the Pareto front $PO*$. If this value goes bellow a threshold α, the MA is launched on the current GA population. When the MA is over, the Pareto front is updated, and the GA is re-run with the previous population (Algorithm 2):

Algorithm 1 GA+MA algorithm

```
Create an initial population
while GA run time not reached do
  Make a GA generation with adaptive mutation
end while
- Here we have a Pareto set $PO$ -
while MA run time not reached do
  Apply crossover on randomly selected solutions of $PO$ to create a set of new solutions.
  Compute the non-dominated set $PO'$ on these solutions
  while New solutions found do
    Create the neighborhood $N$ of each solution of $PO'$
    Let PO' be the non-dominated set of $N \bigcup PO'$
  end while
end while
```

Algorithm 2 AGMA algorithm

```
Create an initial population
while run time not reached do
  Make a GA generation with adaptive mutation
  Update $PO*$ an $P_{PO*}$
  if $P < \alpha$ then
    Make a generation of MA on the population (Algorithm 1)
    Update $PO*$ an $P_{PO*}$
  end if
  Update selection probability of each mutation operator
end while
```

In AGMA algorithm, we can apply the memetic search on the Pareto solutions or on the current population of the GA. Experiments show that the search on the population gives best results. In fact, it allows us to have a better exploration of the search space. Fig. 12 shows the evolution of the ratio Memetic search/Genetic search during one AGMA run.

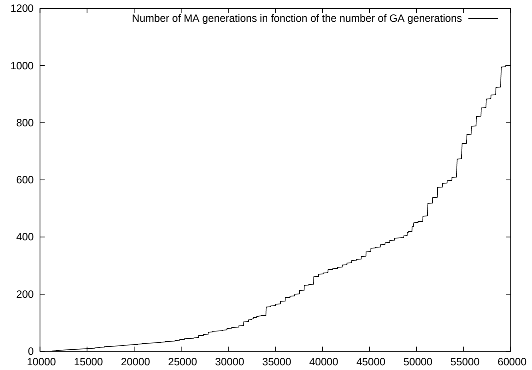

Fig. 12. Example of AGMA evolution - ta_51bi instance (50 jobs*20 machines)

V. COMPUTATIONAL RESULTS

For each problem, we run AGMA for 200 MA hybridization. The time needed by AGMA varies between 1 minute (problems with 20 jobs and 5 machines) and 3 hours (problems with 50 jobs and 20 machines), and is quite similar to the running time needed by AG+MA. Run time are measured on a 1,6 GHz machine.

Table I describes the best results obtained for each objective:

- *Problem* is the instance treated.
- *UB* is the upper bound (best solutions ever found) obtained for the Makespan in single-objective studies.
- C_1 and T_1 are the best C_{Max} and T obtained in [BST02].
- C_2 and T_2 are the best C_{Max} and T obtained with AGMA.
- *Dev* is the deviation between AGMA and the best value obtained in single objective optimization regarding *UB*.

TABLE I

Performance evaluation (best results in bold)

Problem	*UB*	C_1	C_2	*Dev*	T_1	T_2
ta_01bi	1278	**1278**	**1278**	**0%**	**452**	**452**
ta_02bi	1359	**1359**	**1359**	**0%**	491	**469**
ta_11bi	1582	1586	**1582**	**0%**	**1224**	**1224**
ta_12bi	1659	1672	**1659**	**0%**	**1275**	**1275**
ta_21bi	2297	2308	**2297**	**0%**	1097	**1031**
ta_31bi	2724	2729	**2724**	**0%**	3364	**3231**
ta_41bi	2991	3063	**3025**	**1.14%**	**4636**	4706
ta_51bi	3855	3933	**3904**	**1.27%**	7667	**7214**

We obtain the optimal value for C_{Max} objective on all instances except *ta*_41 and *ta*_51. In [BST02], only two optimums has been reached. These results give us an idea on the progress made, but only on the extremities of the front. To compare two fronts, we have to look at the totality of them. In the two-dimensional case we can make this comparison graphically (Fig. 13 and 14). We can observe the progress made for the population diversity, and the progress made in the quality of the solutions, especially for large problems (Fig. 14).

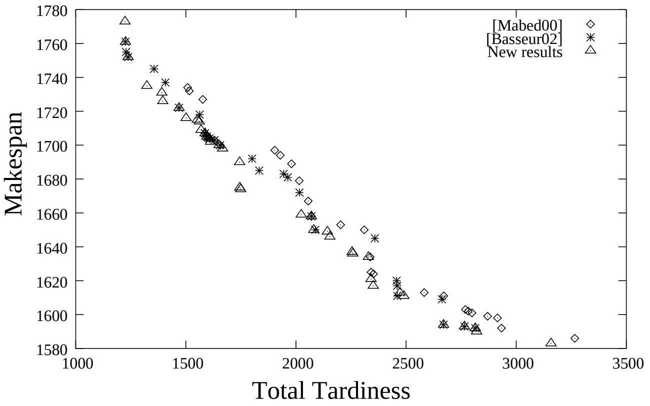

Fig. 13. Example of progress brought by the new mechanisms: ta_11bi instance (20 jobs*10 machines)

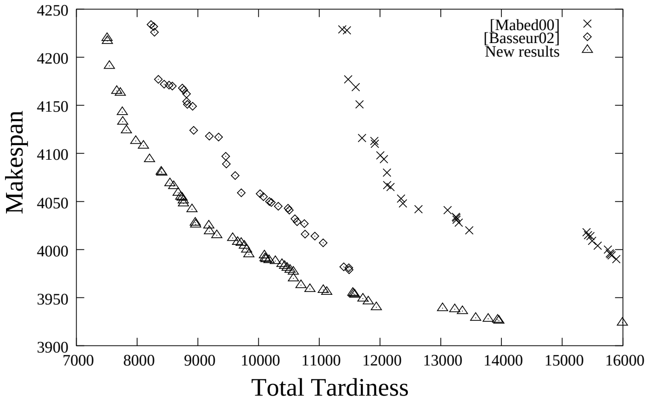

Fig. 14. Example of progress brought by the new mechanisms: ta_51bi instance (50 jobs*20 machines)

These observations show there is a progress made, but we can not quantify this progress. Proper comparison of two multi-objective optimization algorithms is a complex issue and need the use some performance indicators to quantify these progress for two set of runs.

A. *Quality assessment of Pareto set approximation*

Several different solutions have been proposed in recent years. Solutions quality can be assessed in different ways. Some approaches compare the obtained front with the optimal Pareto front [VL00]. Others approaches evaluate a front with a reference point [Jas00]. Some performance measures do not use any reference point or front to evaluate an algorithm [ZT99] [KCO00].

Here, we use contribution metric [MTR00] to evaluate the proportion of Pareto solutions given by each front, and S metric, as suggested in [KC02], to evaluate the dominated area.

A.1 Contribution

The contribution of a set of solutions PO_1 relatively to a set of solutions PO_2 is the ratio of non-dominated solutions produced by PO_1 in PO^*, the set of Pareto solutions of $PO_1 \cup PO_2$.

- Let PO be the set of solutions in $PO_1 \cap PO_2$.
- Let W_1 (resp. W_2) be the set of solutions in PO_1 (resp. PO_2) that dominate some solutions of PO_2 (resp. PO_1).
- Let L_1 (resp. L_2) be the set of solutions in PO_1 (resp. PO_2) that are dominated by some solutions of PO_2 (resp. PO_1).
- Let N_1 (resp. N_2) be the other solutions of PO_1 (resp. PO_2): $N_i = PO_i \setminus (PO \cup W_i \cup L_i)$.

$$Cont(PO_1/PO_2) = \frac{\frac{\|PO\|}{2} + \|W_1\| + \|N_1\|}{\|PO^*\|}$$

Let us remark that $\|PO^*\| = \|PO\| + \|W_1\| + \|N_1\| + \|W_2\| + \|N_2\|$ and $Cont(PO_1/PO_2) + Cont(PO_2/PO_1) = 1$.

For example, we evaluate the contribution of the two sets of solutions PO_1 and PO_2 on Fig. 15: solutions of PO_1 (resp. PO_2) are represented by circles (resp. crosses). We obtain $Cont(PO_1, PO_2) = 0.7$ and $Cont(PO_2, PO_1) = 0.3$.

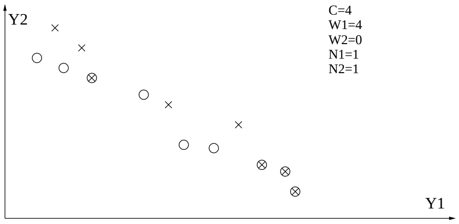

Fig. 15. Example of contribution

Previous study shows that [BST02] outperforms [TRMD01]. So, to compute AGMA assessment, we compare AGMA to [BST02]. Table II presents the average value of the contribution (Cont(AGMA/AG+MA)) obtained on 10 runs.

TABLE II

Quality assessment (contribution metric)

problem	Cont(AGMA/AG+MA)
ta_01bi	0.657
ta_02bi	0.739
ta_11bi	0.751
ta_12bi	0.732
ta_21bi	0.754
ta_31bi	0.690
ta_41bi	0.920
ta_51bi	0.984

Results obtained for contribution metric show that AGMA outperforms AG+MA. Average contributions obtained on each instance is always up to 0.5 (i.e. average contribution of AGMA is better than average contribution of AG+MA on each instance). Contributions obtained for the two last instances show that AG+MA is almost totally dominated by AGMA.

A.2 S metric

A definition of the S metric is given in [Zit99]. Let A be a non-dominated set of solutions. S metric calculates the hyper-volume of the multi-dimensional region enclosed by A and a reference point Z_{ref} (Fig. 16).

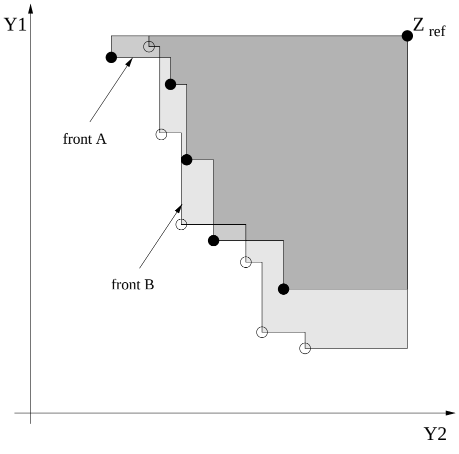

Fig. 16. Example of S value for two sets of solutions PO_1 and PO_2

Let PO_1 and PO_2 be two sets of solutions. To evaluate quality of PO_1 against PO_2, we compute the ratio $(S(PO_1) - S(PO_2))/S(PO_2)$. For our evaluation, the reference point is the worst value on each objective among all the Pareto solutions found by the runs.

Table III presents the average value of the S metric obtained on 10 runs and the average improvement realized by AGMA. This table confirms that AGMA strongly outperforms AG+MA on ta_41bi and ta_51bi. Results are better on all instances especially on ta_02bi, ta_21bi, ta_41bi and ta_51bi instances.

TABLE III

Quality assessment (S metric)

problem	S(AGMA)	S(AG+MA)	Improvement
ta_01bi	4778.1	4707.8	1.49%
ta_02bi	6437.9	5873.5	9.61%
ta_11bi	322121.5	304436.7	5.81%
ta_12bi	180378.2	172749.7	4.42%
ta_21bi	506460.0	442050.9	14.57%
ta_31bi	146462.6	141386.0	3.59%
ta_41bi	1249924.3	1041016.6	20.07%
ta_51bi	2954461.3	2451656.3	20.51%

B. A parallel implementation

Previous tests showed that AGMA performs well on FSP. To improve our results, we implement a parallel version of AGMA. The parallel model is presented in Fig. 17.

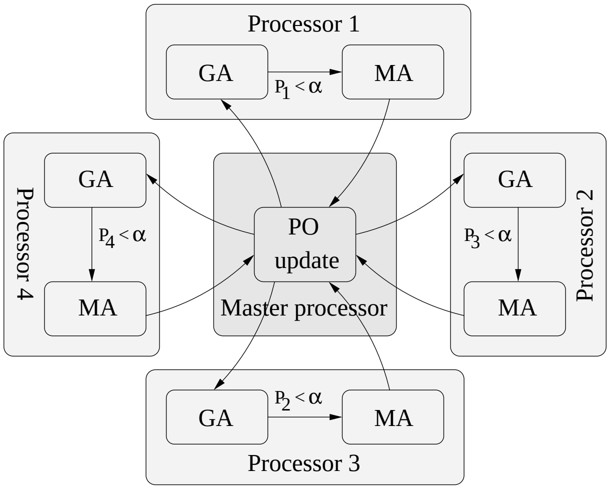

Fig. 17. Parallel model: example with 5 machines

We test this parallel version on 8 1.1GHz/Power4 machines. We launch AGMA on each machine until 1000 hybridization are realized. Time needed by runs never exceed 10 hours.

The set of Pareto solutions obtained for each instance with the parallel version of AGMA are given in annexe. These table represent the best non-dominated solutions obtained during parallel AGMA run. Let us remark that except on ta_41bi and ta_51bi instances, fronts obtained on each parallel run are the same. In respect to these results, we can expect that the front obtained on the six smaller instances are optimal.

VI. Conclusion and Perspectives

In this paper, we have proposed an Adaptive Genetic/Memetic Algorithm with:

- A new mutation performance indicator to use different genetic operators simultaneously in an adaptive manner,
- Adaptive hybridization combining Pareto GA with a MA.

This approach has been tested and evaluated successfully on a FSP.

Perspective of this work is a more general study of landscape (convexity, continuity,...) of MOP for adaptive design of efficient hybrid evolutionary algorithms. Moreover, we can use specific hybrid techniques, to replace the local search realized during the MA. The goal is to speed up the search for larger problems, with up to 500 machines.

References

[BST02] M. Basseur, F. Seynhaeve, and E-G. Talbi. Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Congress on Evolutionary Computation CEC'02*, pages 1151–1156, Honolulu, USA, 2002.

[CVL02] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary algorithms for solving Multi-Objective Problems.* Kluwer, New York, 2002.

[Deb01] K. Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley, Chichester, UK, 2001.

[DL90] J. Du and J. Y.-T. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research*, 15:483–495, 1990.

[GLLK79] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of Discrete Mathematics*, volume 5, pages 287–326. 1979.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[Jas00] A. Jaszkiewicz. On the performance of multiple objective genetic local search on the 0/1 knapsack problem. a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznan University of Technology, Poznan, Poland, jul 2000.

[KC02] J. D. Knowles and D. W. Corne. On metrics for comparing non-dominated sets. In IEEE Service Center, editor, *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002.

[KCO00] J. D. Knowles, D. W. Corne, and M. J. Oates. On the assessment of multiobjective approaches to the adaptive distributed database management problem. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 869–878, september 2000.

[Kim95] Y-D. Kim. Minimizing total tardiness in permutation flowshops. *European Journal of Operational Research*, 33:541–551, 1995.

[LKB77] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.

[MTR00] H. Meunier, E. G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimisation. In *CEC*, volume 1, pages 317–324, Piscataway, New Jersey, jul 2000. IEEE Service Center.

[NHH95] A. Nagar, J. Haddock, and S. Heragu. Multiple and bicriteria scheduling: A litterature survey. *European journal of operational research*, (81):88–104, 1995.

[Raj95] C. Rajendran. Heuristics for scheduling in flowshop with multiple objectives. *European journal of operational research*, (82):540–555, 1995.

[SD94] N. Srinivas and K. Deb. *Evolutionary Computation*, volume 2(3), chapter Multiobjective optimisation using nondominated sorting in genetics algorithms, page 221. 1994.

[SK99] S. Sayin and S. Karabati. A bicriteria approach to the two-machine flow shop scheduling problem. *European journal of operational research*, (113):435–449, 1999.

[SU98] F. Sivrikaya and G. Ulusoy. A bicriteria two-machine permutation flowshop problem. *European journal of operational research*, (107):414–430, 1998.

[Tai93] E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operations Research*, 64:278–285, 1993.

[TRMD01] E. G. Talbi, M. Rahoual, M. H. Mabed, and C. Dhaenens. A hybrid evolutionary approach for multicriteria optimization problems : Application to the flow shop. In E. Zitzler et al., editors, *Evolutionary Multi-Criterion Optimization*, volume 1993 of *LNCS*, pages 416–428. Springer-Verlag, 2001.

[VL00] D. A. Van Veldhuizen and G. B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *In 2000 Congress on Evolutionary Computation, Piscataway, New Jersey*, volume 1, pages 204–211, Jul 2000.

[ZDT+01] E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, Berlin, 2001. Springer-Verlag.

[Zit99] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. Master's thesis, Swiss federal Institute of technology (ETH), Zurich, Switzerland, november 1999.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, nov 1999.

Annexe

Ta01_bi (20 jobs*5 machines)

T	C_Max	T	C_Max
554	1278	453	1297
515	1296	452	1339

Ta02_bi (20 jobs*5 machines)

T	C_Max	T	C_Max	T	C_Max
896	1359	754	1363	603	1367
892	1361	631	1366	469	1368

Ta11_bi (20 jobs*10 machines)

T	C_Max	T	C_Max	T	C_Max
3613	1582	2255	1637	1612	1702
3065	1583	2213	1645	1607	1704
3064	1589	2147	1646	1595	1705
2820	1590	2143	1649	1546	1706
2814	1592	2081	1650	1509	1715
2765	1593	2068	1654	1501	1716
2670	1594	2024	1659	1415	1722
2626	1606	1903	1667	1395	1726
2614	1610	1749	1674	1391	1731
2460	1611	1745	1675	1323	1735
2351	1617	1744	1690	1238	1752
2341	1621	1683	1693	1229	1755
2327	1630	1667	1698	1226	1761
2258	1636	1651	1700	1224	1773

Ta12_bi (20 jobs*10 machines)

T	C_Max	T	C_Max	T	C_Max
3438	1659	2016	1694	1564	1743
2699	1660	1850	1696	1557	1744
2694	1668	1842	1700	1528	1745
2500	1669	1771	1708	1488	1750
2477	1671	1765	1710	1453	1756
2348	1672	1717	1719	1413	1760
2200	1677	1656	1721	1399	1763
2174	1683	1635	1723	1360	1776
2160	1684	1613	1724	1341	1778
2072	1685	1597	1725	1275	1795
2037	1687				

Ta21_bi (20 jobs*20 machines)

T	C_Max	T	C_Max	T	C_Max
3762	2297	2202	2336	1563	2408
3598	2298	2172	2355	1548	2411
3561	2299	2136	2358	1507	2418
3089	2300	2132	2359	1495	2419
3063	2307	2071	2362	1441	2420
3019	2308	2052	2363	1438	2425
2924	2310	2007	2364	1346	2427
2918	2311	1922	2367	1273	2450
2807	2312	1829	2377	1243	2454
2618	2315	1824	2381	1158	2468
2449	2316	1682	2383	1062	2476
2422	2330	1618	2393	1057	2497
2415	2334	1607	2407	1031	2509
2349	2335				

Ta31_bi (50 jobs*5 machines)

T	C_Max	T	C_Max	T	C_Max
4461	2724	3496	2745	3275	2752
3801	2729	3455	2746	3238	2807
3783	2743	3438	2751	3231	2840
3779	2744				

Ta41_bi (50 jobs*10 machines)

T	C_Max	T	C_Max	T	C_Max
7999	3025	5929	3065	5019	3126
7792	3028	5903	3066	5004	3127
7752	3030	5800	3067	4988	3130
7677	3032	5788	3070	4975	3137
7301	3033	5776	3072	4934	3138
7119	3036	5757	3073	4818	3140
7019	3040	5737	3074	4779	3145
6808	3042	5604	3075	4757	3150
6569	3045	5475	3079	4736	3152
6503	3046	5459	3084	4687	3153
6450	3047	5436	3086	4662	3159
6446	3049	5362	3088	4625	3163
6276	3050	5301	3097	4533	3164
6132	3051	5282	3098	4524	3167
6126	3055	5256	3099	4504	3168
6112	3057	5190	3100	4475	3176
6036	3061	5164	3101	4422	3191
6014	3062	5050	3116	4420	3209
5961	3063	5035	3124	4392	3210
5956	3064	5023	3125	4359	3245

Ta51_bi (50 jobs*20 machines)

T	C_Max	T	C_Max	T	C_Max
13720	3899	10349	3966	8028	4072
13514	3900	10209	3968	7915	4084
13455	3901	10122	3971	7853	4089
13142	3902	10103	3972	7852	4093
12550	3903	10059	3973	7846	4096
12339	3906	10008	3975	7777	4097
12250	3909	9792	3976	7737	4102
12021	3911	9624	3978	7687	4105
11906	3913	9410	3984	7604	4109
11794	3915	9370	3994	7575	4113
11780	3920	9220	3995	7530	4116
11748	3921	9029	3998	7506	4123
11733	3922	8947	4001	7482	4127
11641	3926	8944	4002	7470	4136
11562	3928	8917	4009	7461	4137
11438	3930	8904	4012	7436	4142
11054	3932	8886	4014	7410	4145
11010	3937	8802	4017	7402	4149
10995	3940	8788	4020	7390	4165
10947	3941	8617	4022	7389	4166
10902	3947	8478	4025	7378	4169
10663	3948	8393	4038	7334	4173
10658	3953	8355	4048	7307	4182
10632	3954	8330	4050	7275	4186
10569	3955	8316	4051	7265	4199
10565	3957	8235	4054	7244	4200
10481	3958	8198	4058	7213	4213
10431	3962	8160	4068	7211	4245
10356	3963	8125	4070	7210	4248
10353	3964	8117	4071	7204	4268