Path relinking in Pareto Multi-objective Genetic Algorithms

Matthieu Basseur, Franck Seynhaeve and El-Ghazali Talbi

Laboratoire d'Informatique Fondamentale de Lille (LIFL), UMR CNRS 8022,
University of Lille, 59655 Villeneuve d'Ascq Cedex, France.
{basseur,seynhaev,talbi}@lifl.fr

Abstract. Path relinking algorithms have proved their efficiency in single objective optimization. Here we propose to adapt this concept to Pareto optimization. We combine this original approach to a genetic algorithm. By applying this hybrid approach to a bi-objective permutation flow-shop problem, we show the interest of this approach.
In this paper, we present first an Adaptive Genetic Algorithm dedicated to obtain a first well diversified approximation of the Pareto set. Then, we present an original hybridization with Path Relinking algorithm, in order to intensify the search between solutions obtained by the first approach. Results obtained are promising and show that cooperation between these optimization methods could be efficient for Pareto optimization.

1 Introduction

In solving Multi-objective Optimization Problems (MOPs), many methods scalarize the objective vector into a single objective. However, since several years, interest concerning MOPs using Pareto approaches always grows. Many of these studies use Evolutionary Algorithms (EAs) to solve MOPs [1–3].

The evolutionary approach called scatter search, and its generalized form called Path Relinking (PR), contrast with other evolutionary procedures, such as genetic algorithms, by providing unifying principles for joining solutions based on generalized path constructions. Joining solutions can be realized in both decisional and the objective spaces. Path relinking algorithms have recently been investigated in a number of studies for single-objective optimization, and especially in [4], where the Flow-shop problem is solved, in its single objective form.

In this paper, we propose a multi-objective approach to integrate Path relinking algorithms into EAs. We have to take into account several classical questions to implement a PR algorithm, and we propose some solutions for Pareto optimization. We have to define which distance operator has to be to used to join solutions. We propose a distance measure to compute distance in respect to an efficient neighborhood operator, the *Shift* operator. Then we define techniques to have an initial population (with EA), neighborhood generation to approach goal solutions from initial solutions, and path selection between solutions. Then, we propose to integrate path relinking into Pareto evolutionary algorithms to solve

MOPs. We combine an Adaptive Genetic Algorithm (AGA) with Path relinking technique. In order to evaluate the effectiveness of this hybridization, we apply it to solve a Bi-Objective Flow-shop Scheduling Problem (BOFSP).

This paper is organized as follows. In section 2, we present the BOFSP. In section 3, we present a Pareto EA (AGA) developed to find an initial Pareto population. In section 4, we present cooperation between AGA and multi-objective Path relinking. Section 5 presents results on a large class of instances, which are non-exactly solved with exact approaches. In the last section, we discuss the effectiveness of this approach and perspectives of this work.

2 A Bi-Objective Flow Shop Problem (BOFSP)

The Flow-shop Scheduling Problem (FSP) is one of the numerous scheduling problems. The FSP has been widely studied in the literature. Proposed methods for its resolution vary between exact methods such as the branch & bound algorithm [5], specific heuristics [6] and meta-heuristics [7]. However, the majority of works on flow-shop problem studies the problem in its single criterion form and aims mainly to minimize makespan, which is the total completion time. Some 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 the makespan and the sum of completion times [5]. Sivrikaya-Serifoglu et al. proposed a comparison of branch & bound approaches for minimizing the makespan and a weighted combination of the average flowtime, applied to the two-machine flow-shop problem [8]. Rajendran proposed a specific heuristic to minimize the makespan and the total flowtime [6]. Nagar et al. proposed a survey of the existing multi-criteria approaches of scheduling problems [7].

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 its 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.

In this work, 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 follows:

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

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

In the Graham et al. notation [9], this problem is denoted: F/perm, $d_i/(C_{max}, T)$.

In [10] C_{max} minimization has been proved to be NP-hard for more than two machines. The total tardiness objective T has been studied only a few times

for M machines [11], but total tardiness minimization for one machine has been proved to be NP-hard [12]. The evaluation of the performances of our algorithm has been realized on some Taillard benchmarks for the FSP [13], extended to the bi-objective case [14][1].

3 An Adaptive Genetic Algorithm (AGA)

In this part, we present AGA which is applied to MOFSP. AGA's objective is to explore a large and diversified part of the landscape, to offer good solutions to the hybrid part of the algorithm (i.e. PR in our case). AGA proposes an adaptive selection between different mutation operators. In a first time, let us present classical operators of AGA:

- Initialization: Individuals are generated randomly.
- Selection: Elitist selection with NSGA ranking [15].
- Crossover Operator: 2-point crossover (See Fig. 1).
- Mutation Operators: Adaptive selection between 4 mutation operators: insertion, reciprocal exchange, random and inversion operator (See Fig. 2,3,4,5). Adaptive selection is described below.
- Diversification: Combined Sharing (sharing is realized in the decision space and in the objective space). Niche sizes are defined adaptively after each GA generation [16].
- Replacement: Generational (the population is automatically replaced by the new individuals).
- Stopping criterion: Fixed time.

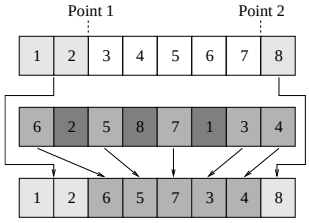

Fig. 1. 2-point crossover operator.

An Adaptive Mutation Selection

Whatever the problem, we can choose between many mutation and crossover operators. Many evaluations must be done on each operator in order to know its effectiveness. Moreover, the efficiency of an operator may change during the evolution process. An operator may offer a better convergence at the beginning of the GA, but this convergence may stop earlier than with another operator. The success of an operator may also depends on the instance of the problem.

[1] bi-objective benchmarks and results obtained are available on the web at `http://www.lifl.fr/~basseur`

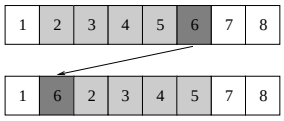

Fig. 2. Shift operator.

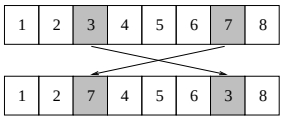

Fig. 3. Swap operator.

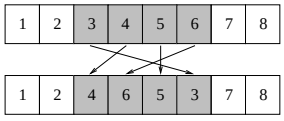

Fig. 4. Random operator.

Fig. 5. Inversion operator.

Therefore, we have proposed an adaptive Pareto GA, in which the choice of the operator is done dynamically during the search. The purpose is to change the probability selection of each operator according to its efficiency.

This adaptive mechanism has the interest to diversify the Pareto population by the use of several operators. Moreover, it allows us to define the best operator to use for the PR algorithm presented in the next section.

To adapt mutation rate during the GA, each mutation operator 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). This progress value allows to refine mutation rates after each GA generation. This approach has been initially proposed by Hong *et al.* [17] for the single-objective case. To have a good evaluation of each operator in the multi-objective case, we use the elitist and ranking approach . As a consequence, the progress value is expressed as follows:

$$\Pi(I_{M_i}) = C_{I_{M_i}} * \left(\frac{R_{I_{M_i}}}{R_I}\right)^k \tag{1}$$

with $C_{I_{M_i}} = \frac{1}{R_{I_{M_i}}}$ ($C_{I_{M_i}}$ is an elitist coefficient), 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 performed by a mutation operator. For our application, we set k to 2.

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

$$Progress(M_i) = \frac{\sum \Pi(I_{M_i})}{\sum C_{I_{M_i}}} \tag{2}$$

The new selection probabilities are computed proportionally to these values, with a minimum selection probability of δ:

$$P_{M_i} = \frac{Progress(M_i)}{\sum_{j=1}^{n} Progress(M_j)} * (1 - n * \delta) + \delta \tag{3}$$

In figure 6, we show an example of mutation rate on a problem with 100 jobs and 5 machines. Global results (detailed in [18]) show that *shift* operator is the most efficient neighborhood operator for this instance, and for the other instances.

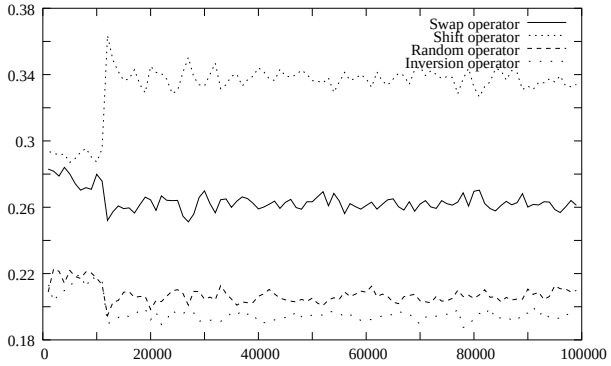

Fig. 6. Mutation rate evolution during GA run: ta_100_5_01 instance

Computational results presented in the last section show that AGA obtain a good initial approximation of the Pareto front. In order to improve these results by intensify the search within the set of solutions founded by AGA, we propose a cooperative scheme between AGA and Path relinking.

4 A Multi-objective Path Relinking Algorithm

Path Relinking (PR) was originally proposed by Glover [19] as an intensification strategy exploring trajectories connecting elite solutions obtained by tabu search or scatter search. [20] offers a good overview of this technique.

In this section we propose a cooperation scheme between PR and AGA in the multi-objective case. The goal is to show the interest of this type of approach, to improve solutions obtained by AGA. In our knowledge, a PR algorithm has already been proposed to solve Flow-shop problem, but only in its single objective optimization form [4]. The concept of PR algorithm is schematized in figure 7. Two solutions are chosen in an initial population (if possible, good solutions). Then by iterative applications of neighborhood operator, join the first solution to the second. These two solutions are called *initiating* and *guiding* solutions. During the algorithm, solutions are evaluated, and in order to improve efficiency of PR algorithms, some of them are selected to be improved by a local search algorithm.

Path relinking is naturally an extension of scatter search algorithms. Only a few studies propose scatter search algorithms to solve MOPs [21]. A study by Gandibleux et al. [22] propose to incorporate PR principles in a multiobjective heuristic using genetic heritage. In a general way, many questions have to be

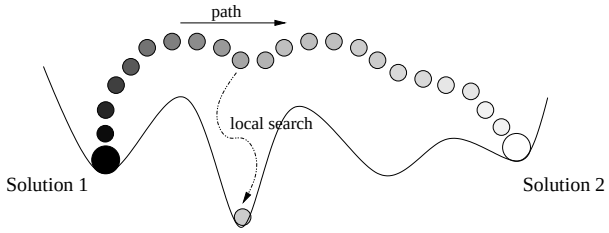

Fig. 7. Path Relinking: the concept

considered before designing a PR algorithm. Several answers exist for single objective optimization, but they have to be reconsidered for the multi-objective case. We have to establish these following mechanisms:

- Neighborhood structure.
- Distance measure (correlated, or not, with the neighborhood).
- Selection criteria, to choose solutions to link.
- Path selection, to establish which path(s) has to be generated.
- Improvement of solutions. A local search algorithm may be applied on solutions belonging to the path. A mechanism of solutions selection to apply the local search algorithm has to be proposed.

Here we propose a first initiative to answer these difficulties, but there are still some open questions.

4.1 Neighborhood operator

In order to link solutions, we need to select a neighborhood operator which allows us to generate the intermediate solutions. In [4], Reeves and Yamada propose a distance measure for the *swap* operator, but not for the *shift* operator. Then they propose other distance measures which are not correlated with these operators. Here, we choose the most powerful operator of AGA algorithm applied on BOFSP: the *shift* operator. We propose next a distance measure in respect to this operator.

4.2 Distance in decision space

Genotypic distance between two solutions S_1 and S_2, in respect to the *shift* operator, is: $d_{perm} = N - s_{max}$, where N is the number of jobs, and s_{max} is the size of the greatest shared substring between S_1 and S_2. In the appendix, we prove that $d_{perm}(S_1, S_2)$ is a distance, and that it corresponds to the minimum number of permutations required to join S_1 with S_2.

Computing a greatest shared substring between two solutions is a well-known problem in the genomic community. Computing a greatest shared substring between two solutions is a classical application of dynamic programming [23]. We implement this algorithm to compute distance between solutions in the decision space, its complexity is in $\mathcal{O}(N^2)$.

An example of distance between two solutions x and y obtained by computing a greatest shared substring is shown in figure 8. In this example, three inversions are necessary to link x from y - the jobs 2, 4 and 6 have to be moved to link these solutions. They correspond to the set of jobs which are not in the greatest shared substring between the two solutions (let remark us that the greatest shared substring is not necessary unique: the substrings '13478' and '12478' can be viewed as largest shared substring too). As in many Path relinking implementations, we choose to generate only minimum path between solutions.

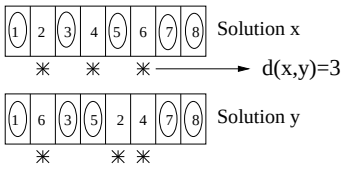

Fig. 8. Distance between two solutions x and y

4.3 Initial solutions for path relinking

In many single objective optimization path relinking algorithms, good local optima solutions are chosen to be linked. We can favor distant solutions to favor the exploration of the search space, or adjacent solutions to favor intensification of the search around good solutions. Then, in the multi-objective case, it may be interesting to link solutions which are closed to each other in the objective space in order to intensify the search around solutions with similar quality on the different objectives.

For this study, we randomly choose among Pareto solutions obtained by the GA. After linking the two first solutions, we compute the new Pareto set. The selection of the two next solutions to link is realized on this set.

4.4 Neighborhood generation

We use the insertion neighborhood operator to generate the path from a solution x to a solution y. The goal is to explore only solutions which reduce distance between the generated solution and y.

So, after defining the set of "well placed" jobs, by computing the largest shared substring between x and y, we have to move misplaced jobs in order to increase the size of this largest shared substring.

Then, we compute the available positions for misplaced jobs, as shown in figure 9. On this example, a greatest shared substring is first computed and represented in this figure by encircled jobs (i. e. the string 13578). Possible moves to apply to x to approach y are also computed for jobs 2, 4 and 6. These available positions are chosen to increase the size of the largest shared substring, and so to decrease the distance between the current solution and the guiding solution.

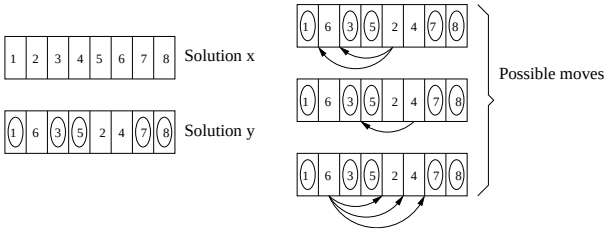

Fig. 9. Neighbors explored (greatest shared substring represented by encircled solutions)

4.5 Generation of the Path

We can generate from x a set of neighbors which reduce the distance between this solution x and the goal solution y. Now, we have to choose which path we will generate to iterate this mechanism and explore the largest landscape possible.

A first possible approach is to generate all the possible paths. But with this approach, the number of explored solutions grows exponentially with the distance between the two considered solutions. So, experimentally, we have to explore only a subset of the possible paths. Many PR algorithms which select a subset of the possible paths propose to explore only the best solutions generated. In the multi-objective case, the best solutions are those which are non-dominated by all the neighborhood. This set is represented in figure 10 (eligible solutions). In order to reduce the size of the exploration, we apply a random aggregation of the objectives to select only one solution in the set of eligible solutions (fig. 10). In future works, it is conceivable to select the totality of the set of eligible solutions, with a mechanism which limits the size of the exploration.

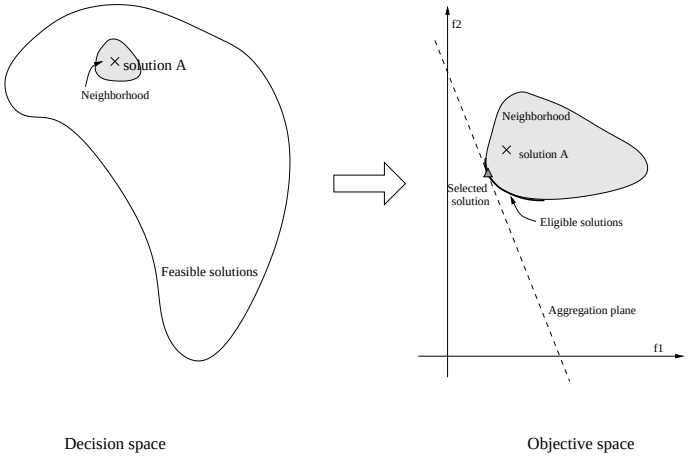

Fig. 10. Path Relinking algorithm: neighborhood exploration.

4.6 A Pareto Local Search

To affine Pareto solutions found by the Path Relinking algorithm, we implement a population-based local search or Pareto Local Search (PLS). Local searches are realized after each PR generation. The local search technique is described in algorithm 1. The neighborhood structure used for this algorithm is the insertion operator.

Algorithm 1 PLS algorithm

Generate an initial Pareto set PO (in our case, with PR algorithm)
do
$S' \leftarrow PO$.
Generate the neighborhood PN_x for each solution x of S'.
Let PO be the set of non-dominated solutions of $S' \cup_x PN_x$.
Until PO=S' (the population has reached the local optima).
Let PO^* be the Pareto optimal set of $PO^* \cup S'$.
Return the Pareto set PO^*

To introduce PLS in PR algorithm, we select the set of non-dominated solutions discovered during the PR algorithm. Then we apply the PLS algorithm with this set of solutions as initial population. This mechanism is shown in figure 11. Let remark us that PLS algorithm may comsume larger computation time if the problem size or the number of objective increase. In this case, PLS algorithm may be implemented with other mecanisms as clustering, neighborhood restriction, etc. to be effective.

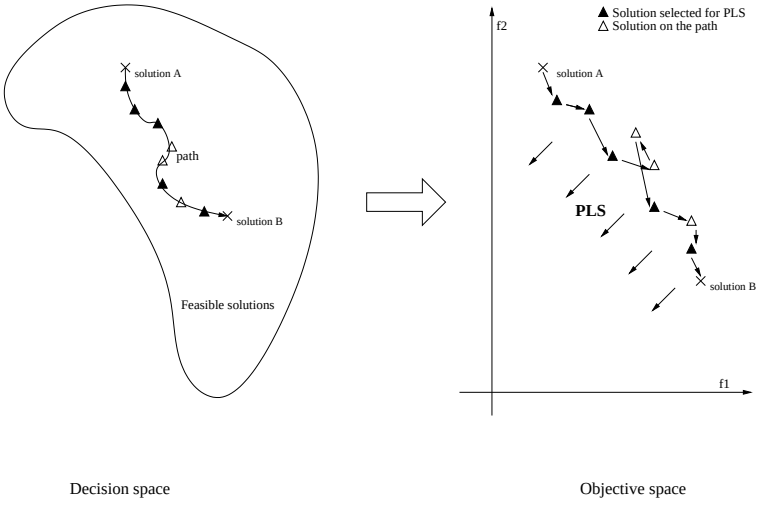

Fig. 11. Path relinking algorithm with local search.

5 Experimental results

5.1 Quality assessment of Pareto set approximation

Solutions' quality can be assessed in different ways. Here, we use the contribution metric [24] to evaluate the proportion of Pareto solutions given by each front, and the S metric [25] which evaluate the dominated area with a reference point. This metric is advised in [26].

Contribution metric: 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^*, where PO^* is 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): $N_i = PO_i \setminus (PO \cup W_i \cup L_i)$.

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

***S* metric:** A definition of the S metric is given in [25]. Let PO be a non-dominated set of solutions. S metric calculates the hyper-volume of the multi-dimensional region enclosed by PO and a reference point Z_{ref}.

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 the evaluation, the reference point is the one with the worst value on each objective among all the Pareto solutions found over the runs.

5.2 Computational results

Evaluations were realized for 10 runs per instance, on a 1.6Ghz machine. We test the different algorithms of several instances proposed in [14]. The smallest instances are exactly solve in a bi-objective exact approach proposed by Lemesre et al. [27]. We test the different approaches on instances with 50 and 100 jobs.

We compare performance of AGA with PR against results obtained only with AGA. In the experiments, runs are realized with a specified time limit (1000 minutes in our experiments). For all tests, 10 runs have been done on each benchmark. For the cooperative approach ($AGA + PR$), we choose to run AGA for 10% of the total run time. Results of tested instances are listed in tables 1, 2 and 3, with:

- $\mathbf{S_{Min}}$, respectively $\mathbf{C_{Min}}$, is the minimum value of S, respectively of the contribution.
- $\mathbf{S_{Max}}$, respectively $\mathbf{C_{Max}}$, is the maximum value of S, respectively of the contribution.

- **Average** is the average value.
- **Std dev** is the average deviation between the different measures.
- **Avg Imp** is the average improvement realized by AGA+PR against AGA in terms of dominance area.

Table 1. Quality assessment (S metric): S(AGA).

Benchmark	S(AGA)			
	S_{Min}	**S_{Max}**	**Average**	**Std dev**
ta_50_10_01	516521	828610	709023.6	93279.2
ta_50_20_01	1841592	2476210	2170418.6	205943.7
ta_100_5_01	431760	446264	439799.2	5526.3
ta_100_10_01	5414203	6956336	6218845.8	464814.8
ta_100_20_01	17167424	20139584	19117894.4	866011.0

Table 2. Quality assessment (S metric): S(AGA+PR)

Benchmark	S(AGA+PR)				
	S_{Min}	**S_{Max}**	**Average**	**Std dev**	**Avg Imp**
ta_50_10_01	870448	1065523	988079.6	67443.2	39.4%
ta_50_20_01	2853166	3113992	3020664.0	93279.6	39.2%
ta_100_5_01	445695	446878	446323.0	286.2	1.5%
ta_100_10_01	7813940	8148625	7978420.0	93335.8	28.3%
ta_100_20_01	26209816	28745238	27752835.2	690416.0	45.2%

Table 3. Quality assessment (Contribution metric): Cont(AGA/AGA+PR) - 1000 minutes runs

Benchmark	Cont(AGA+PR/AGA)			
	C_{Min}	**C_{Max}**	**Average**	**Std dev**
ta_50_10_01	1.00	1.00	1.000	0.000
ta_50_20_01	1.00	1.00	1.000	0.000
ta_100_5_01	0.71	1.00	0.836	0.085
ta_100_10_01	1.00	1.00	1.000	0.000
ta_100_20_01	1.00	1.00	1.000	0.000

The instance *ta_N_M_n*, correspond to the bi-criteria version of the number n problem with N jobs and M machines in Taillard's benchmarks. Results show a great improvement for all the instances excepting *ta_100_5_01*. For this problem, we can expect that we are really near the optimal Pareto front. In fact, the

best value for C_{max} objective given by the hybrid algorithm is the optimal value (found by single objectives algorithms). For the other instances evaluated, with 10 and 20 machines, improvement, in term of dominance area, varies from 27.7% and 45.2%. Table 3 shows that for these problems, all the Pareto set generated by $AGA+PR$ algorithm dominate those generated by AGA ($C(AGA+PR/AGA) = 1.00$). Figure 12 shows an example of Pareto set generated. The first Pareto set is generated by AGA. The second corresponds to the same algorithm in a longer run. The third corresponds to the path relinking algorithm applied on the first Pareto set.

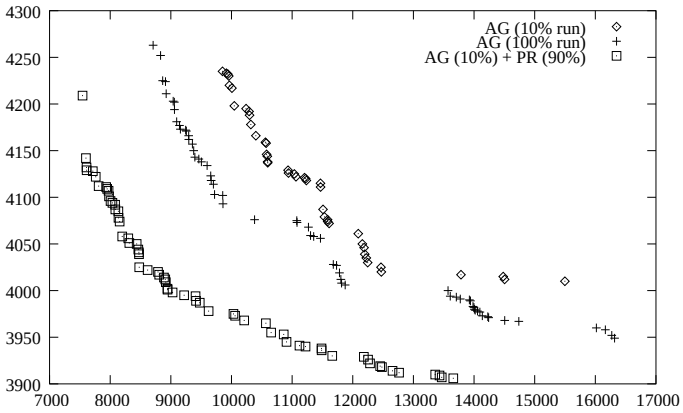

Fig. 12. Result on *ta_50_20_01* instance.

6 Conclusion and perspectives

In this paper, we have first presented an Adaptive Genetic Algorithm to solve MOPs. This approach has been applied on a BOFSP. Then we have proposed a multi-objective path relinking algorithm, with a distance measure corresponding to the best neighborhood operator we select during AGA experiments. We propose several mechanisms to answer some difficulties of multi-objective path relinking implementation. Then we have proposed an original hybridization between AGA and path relinking to improve results. This approach has been tested, and its effectivenesses has been shown in comparison with Pareto fronts obtained by AGA algorithm. These results show the interest of cooperation with path relinking, which can be applied after many meta-heuristics, and especially evolutionary algorithms.

These results could be improved by adding other mechanisms of Path Relinking algorithms, such as replacing a genetic operator of AGA with the PR algorithm, or using a more evolved path selection. Another interesting approach could be to change the transition rules between AGA and PR. We can adapt the transition in respect to the evolution rate of the Pareto set.

Moreover, the hybrid approach presented in this article can be easily parallelized, by running several path generations and local searches simultaneously. This work will be integrated in the framework ParaDisEO (Parallel and Distributed Evolutionary Objects) [28].

References

1. Coello, C.A.C., Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary algorithms for solving Multi-Objective Problems. Kluwer, New York (2002)
2. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester, UK (2001)
3. Zitzler, E., Deb, K., Thiele, L., Coello, C.A.C., Corne, D., eds.: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001). Volume 1993 of Lecture Notes in Computer Science. (2001)
4. Reeves, C., Yamada, T.: Genetic algorithms, path relinking and the flowshop sequencing problem. Evolutionary Computation **6** (1998) 230–234
5. Sayin, S., Karabati, S.: A bicriteria approach to the two-machine flow shop scheduling problem. European journal of operational research (1999) 435–449
6. Rajendran, C.: Heuristics for scheduling in flowshop with multiple objectives. European journal of operational research (1995) 540–555
7. Nagar, A., Haddock, J., Heragu, S.: Multiple and bicriteria scheduling: A litterature survey. European journal of operational research (1995) 88–104
8. Sivrikaya, F., Ulusoy, G.: A bicriteria two-machine permutation flowshop problem. European journal of operational research (1998) 414–430
9. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: Annals of Discrete Mathematics. Volume 5. (1979) 287–326
10. Lenstra, J.K., Kan, A.H.G.R., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics **1** (1977) 343–362
11. Kim, Y.D.: Minimizing total tardiness in permutation flowshops. European Journal of Operational Research **33** (1995) 541–551
12. Du, J., Leung, J.Y.T.: Minimizing total tardiness on one machine is NP-hard. Mathematics of operations research **15** (1990) 483–495
13. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operations Research **64** (1993) 278–285
14. Talbi, E.G., Rahoual, M., Mabed, M.H., Dhaenens, C.: A hybrid evolutionary approach for multicriteria optimization problems : Application to the flow shop. In: Evolutionary Multi-Criterion Optimization (EMO'01). Volume 1993 of Lecture Notes in Computer Science. (2001) 416–428
15. Srinivas, N., Deb, K.: Multiobjective optimisation using non-dominated sorting in genetics algorithms. In: Evolutionary Computation. Volume 2. (1994) 221
16. Basseur, M., Seynhaeve, F., Talbi, E.G.: Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In: Congress on Evolutionary Computation (CEC'02), Honolulu, Hawaii, USA (2002) 1151–1156
17. Hong, T.P., Wang, H.S., Chen, W.C.: Simultaneous applying multiple mutation operators in genetic algorithm. Journal of Heuristics **6** (2000) 439–455
18. Basseur, M., Seynhaeve, F., Talbi, E.G.: Adaptive mechanisms for multi-objective evolutionary algorithms. In: Congress on Engineering in System Application (CESA'03), Lille, France (2003) 72–86

19. Glover, F.: Tabu search and adaptive memory programing advances, applications and challenges. In: Interfaces in Computer Science and Operations Research, Kluwer Academic Publishers, Boston (96) 1–75
20. Glover, F., Laguna, M.: Fundamentals of scatter search and path relinking. Control and Cybernetics **29** (1999) 653–684
21. Beausoleil, R.P.: Multiple Criteria Scatter Search. In: 4th Metaheuritics International Congress, Porto, Portugal (2001) 539–543
22. Gandibleux, X., Morita, H., Katoh, N.: Impact of clusters, path-relinking and mutation operators on the heuristic using a genetic heritage for solving assignment problems with two objectives. In: Metaheuristics International Conference (MIC'03), Kyoto, Japan (2003) 23(1–6)
23. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: 15. In: Introduction to algorithms. The MIT Press, Cambridge, Massachuchetts (1990) 350–355
24. Meunier, H., Talbi, E.G., Reininger, P.: A multiobjective genetic algorithm for radio network optimisation. In: CEC. Volume 1., Piscataway, New Jersey, IEEE Service Center (2000) 317–324
25. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Master's thesis, Swiss federal Institute of technology (ETH), Zurich, Switzerland (1999)
26. Knowles, J.D., Corne, D.W.: On metrics for comparing non-dominated sets. In Center, I.S., ed.: Congress on Evolutionary Computation (CEC'2002). Volume 1., Piscataway, New Jersey (2002) 711–716
27. Lemesre, J., Dhaenens, C., Talbi, E.: A parallel exact method for a bicriteria permutation flow-shop problem. In: Project Management and Scheduling (PMS'04), Nancy, France (2004) 359–362
28. Cahon, S., Melab, N., Talbi, E.G.: Paradiseo: a framework for the flexible design of parallel and distributed hybrid metaheuristics. Journal of Heuristics **10** (2004) 357–380
29. Basseur, M.: Cooperative models for multi-objective optimization. PhD thesis, University of Lille ((to appear))

Appendix

We want to prove that d_{perm} is a distance measure and that only minimum paths are generated. Let us introduce several notations:

- $Shift(i, j, X)$ corresponds to the permutation generated by moving the i^{th} job of a sequence to a new position j.
- $pos_X(i)$ corresponds to the position of the job i in a permutation X.
- $GSS(X, Y)$ corresponds to the greatest shared substring between two permutations X and Y.

First, let us introduce a trivial lemma (proved in [29]):

Lemma 1 *For all the permutations X and Y, $d_{perm}(X,Y)-1 \leq d(X, Shift(i, j, Y)) \leq d(X, Y) + 1$*

In fact, the largest shared substring between X and Y can not vary more than 1 by deleting and adding a 'letter' (job).

Now, let introduce another lemma:

Lemma 2 *For all the permutations X and Y of a size N, with $d_{perm}(X,Y) \neq 0$, $\exists\, i,j \in [0..N] \mid d_{perm}(Shift(i,j,X),Y) = d_{perm}(X,Y) - 1$.*

Proof: Let X and Y be two permutations. If $X \neq Y$, then $\exists x \in Y$ with $x \notin GSS(X,Y)$. Let y (resp. z) be the nearest predecessor (resp. successor) of x in X with $y \in GSS(X,Y)$ (resp. z).

Given $Y' = Shift(x, pos_Y(y)+1, Y)$ [2]. Set $\mathcal{U} = GSS(X,Y)$. $\mathcal{U}$ can be described as: $\mathcal{U}_1 y z \mathcal{U}_2$. Let $\mathcal{U}' = GSS(X,Y')$. So, $\mathcal{U}' = \mathcal{U}_1 y x z \mathcal{U}_2$ (x is now between y and z, either in X than in Y). So, $|\mathcal{U}'| = |\mathcal{U}| + 1$.

As a consequence, for all X, Y with $X \neq Y$, we can apply a *Shift* operator to increase the size of the greatest shared substring of the two permutations. So, lemma 2 is proved.

With the two lemma presented, let us prove that d_{perm} is a distance. We have to establish 3 properties:

- d_{perm} **is symmetrical:** $(d_{perm}(X,Y) = d_{perm}(Y,X))$. This property is trivially verified with the definition of the largest shared substring between two permutations.
- **Space separation by** d_{perm}: $(d_{perm}(X,Y) = 0 \Leftrightarrow X = Y)$. Trivial property too. If, $d_{perm}(X,Y) = 0$ then $|GSS(X,Y)| = N$, so $GSS(X,Y) = X = Y$. Then $X = Y$.
- **Triangular inequality:** $(\forall \{X,Y,Z\}, d_{perm}(X,Z) \leq d_{perm}(X,Y) + d_{perm}(Y,Z))$. This property need a recurrence proof, proposed below.

Let X and Z be two permutations. Let us define the proposition P_n: "$\forall\, Y$ with $d_{perm}(X,Y) \leq n$, we have the relation $d_{perm}(X,Z) \leq d_{perm}(X,Y) + d_{perm}(Y,Z)$". With the space separation, we have P_0 verified. Let us suppose $\mathcal{P}_n$ verified.

Set Y with $d_{perm}(X,Y) \leq n+1$.

If $d_{perm}(X,Y) \leq n$, the inequality is verified according to the recurrence hypothesis. If $d_{perm}(X,Y) = n+1$, we show previously that we can find i,j with $Y' = Shift(i,j,Y)$ and $d_{perm}(X,Y') = d_{perm}(X,Y) - 1$. So, according to the recurrence hypothesis, $d_{perm}(X,Z) \leq d_{perm}(X,Y') + d_{perm}(Y',Z)$. As a consequence, with lemma 1 and 2 and the recurrence relation:

$$d_{perm}(X,Y) = d_{perm}(X,Y') + 1$$
$$d_{perm}(X,Z) \leq d_{perm}(X,Y') + d_{perm}(Y',Z)$$
$$d_{perm}(Y,Z) \geq d_{perm}(Y',Z) - 1$$

As a consequence:

$$d_{perm}(X,Y) + d_{perm}(Y,Z) \geq d_{perm}(X,Y') + 1 + d_{perm}(Y',Z) - 1$$
$$d_{perm}(X,Y) + d_{perm}(Y,Z) \geq d_{perm}(X,Z)$$

The recurrence hypothesis is verified at the rank $n+1$. So, the triangular inequality is verified with d_{perm}, which is a distance measure.

[2] we can choose every element in the interval $[pos_Y(y)+1, pos_Y(z)]$.