Permutation Flowshop Scheduling by Genetic Local Search

Takeshi Yamada[1]*and Colin Reeves[2]

[1]NTT Communication Science Labs., Kyoto, Japan
Email: yamada@cslab.kecl.ntt.co.jp
[2]Coventry University, UK
Email: CRReeves@coventry.ac.uk

Abstract

In this paper, the landscape for the permutation flowshop scheduling problem (PFSP) with stochastic local search and a critical block-based neighbourhood structure has been investigated. Numerical experiments using small benchmark problems show that there are good correlations between the makespans of local optima, the average distances to other local optima and the distances to the known global optima. These correlations suggest the existence of a 'big valley' structure, where local optima occur in clusters over the landscape. An approximation method for PFSP that would make use of this big valley structure is proposed by using a critical block-based neighbourhood structure, and a genetic local search method called MSXF-GA, previously developed for the job shop scheduling problem. Computational experiments using more challenging benchmark problems demonstrate the effectiveness of the proposed method.

1 Introduction

It is well known that Genetic Algorithms (GAs) can be enhanced by incorporating constructive heuristics or point-based local search methods. The incorporation is often referred as Genetic Local Search (GLS) or population-based local search. Basically, GLS can be viewed as a variant of Adaptive Multi-Start (AMS) methods in which new starting points are generated adaptively based on previously found local optima. Yamada and Nakano have extended the idea of GLS and proposed Multi-Step Crossover Fusion (MSXF) [12]. Instead of generating a new starting point from parents by a recombination operator, MSXF uses one of the parents itself as a new starting point and carries out a 'navigated' local search where the search direction is biased toward the other parent.

It has been shown that the success of AMS as well as GAs strongly depends on some global structure in the cost surface or, in GA terminology, fitness landscape. For example, Boese et al. [2] have suggested that, in the cases of the travelling salesman problem (TSP) and graph bisection for which AMS worked well, local optima tend to be relatively close to each other (in terms of a plausible metric) and to the known global optimum. This structure, where local optima occur in clusters, has been called a 'big valley' structure. Jones and Forrest [4] have shown that many GA-easy problems have strong correlation between fitness and distance to the global optimum. More recently, Reeves [7] has re-formulated the landscape concept in the context of an associated neighbourhood structure and confirmed that the landscape of the (makespan minimizing) permutation flowshop sequencing problem (PFSP) induced by rather simple neighbourhood operators also exhibits a big valley structure under a position-based metric.

In this paper, the PFSP landscape induced by a more sophisticated neighbourhood operator is investigated. This neighbourhood focuses on critical blocks and uses a stochastic local search based on the Metropolis criterion. Section 2 gives the problem definitions and explains a critical block-based neighbourhood structure and the distance measures. Section 3 explains the GLS approach based on the stochastic local search and MSXF. Section 4 investigates the existence of a 'big valley structure' for the PFSP with stochastic local search and the representative neighbourhood.

Assuming that a 'big valley' structure holds for a wide range of PFSP landscapes induced by this neighbourhood operator, it is expected that the use of an adaptive multi-start method in which new local search is concentrated on a region between previously found local optima should be effective in finding near-optimal solutions even for more difficult problems. Section 5 shows an implementation of MSXF-GA for the PFSP. In Section 6 MSXF-GA is applied to more challenging benchmark problems. Experimental results demonstrate the effectiveness of the proposed method.

*This work was undertaken when the first author was staying at Coventry University.

2 The permutation flowshop scheduling problem

The permutation flowshop scheduling problem (PFSP) is often designated by the symbols $n/m/P/C_{max}$, where n jobs have to be processed on m machines in the same order. The processing of each job on each machine is an operation which requires the exclusive use of the machine for an uninterrupted duration called the processing time. P indicates that only permutation schedules are considered, where the order in which each machine processes the jobs is identical for all machines. Hence a schedule is uniquely represented by a permutation of jobs. The objective is to find a schedule that minimizes the makespan C_{max}, the time at which the last job is completed on the last machine.

In general, a neighbourhood $N(x)$ of a point x in a search space can be defined as a set of new points that can be reached from x by exactly one transition or move (a single perturbation of x). Several transition operators have been proposed for PFSP. The simplest one is the adjacent pairwise exchange operator which exchanges the positions of two adjacent jobs. The shift operator which takes a job from its current position and re-inserts it to another position is shown to be the most efficient among simple operators[6]. More sophisticated operators are obtained by limiting the size of the neighbourhood by using the notion of critical blocks. Nowicki and Smutnicki have proposed the representative neighbourhood method[5], where a reduced neighbourhood is generated from the original critical block-based neighbourhood by clustering its members and picking up the best move (representative) from each cluster. A new neighbourhood is the set of all representative moves. They proposed a sophisticated tabu search algorithm using this neighbourhood structure and an intensification mechanism called 'back jump tracking'.

In this paper, a simplified form of their representative neighbourhood is adopted. For each job j in a critical block, let S_j^a be a set of moves that shift the job j to some position in the next block; similarly S_j^b shifts j to the previous block. Evaluate schedules obtained from each move in S_j^a and denote the best one by s_j^a. Similarly s_j^b is obtained from S_j^b. Then the representative neighbourhood is defined as a set of all schedules obtained by representative moves $\{s_j^a, s_j^b\}$ for all jobs j in all critical blocks (see Figure 1). This corresponds to the case of $\epsilon = 1$ in the notation of [5].

The difference between two permutation schedules S and T can be measured by an appropriately defined distance. For example, an adjacency-based distance is most commonly used for TSP, where relative ordering is more important than absolute position in the sequence. On the other hand, a distance which respects absolute position more than relative ordering is more suitable for PFSP. In this paper two well-known distances are considered as follows:

precedence-based distance: This distance counts the number of job pairs $\{i, j\}$ in which j is preceded by i in S but not in T.

position-based distance: This distance sums up the positional differences for each job in S and T.

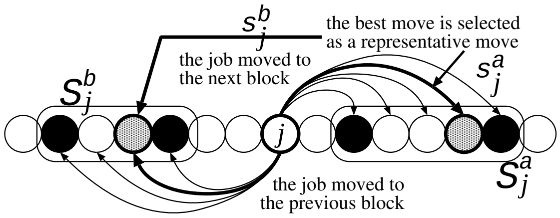

Figure 1: The best move to the next/previous block is selected as a representative.

As shown in the later section, these two distances are strongly correlated with each other, and also with an approximation to the minimal number of steps of the neighbourhood operator to move from S to T. In this paper the precedence-based distance is used.

3 Genetic local search

It is well known that GAs are not well suited for fine-tuning structures that are very close to optimal solutions, and that it is essential to incorporate local search methods, such as neighbourhood search, into GAs. The result of such incorporation is often called *Genetic Local Search* (GLS) [10]. This approach can be viewed as a variant of Adaptive Multi-Start (AMS) methods in which local search is applied repeatedly, each time a new starting point being generated adaptively based on previously found local optima [2]. The Multi-Step Crossover Fusion (MSXF) GA proposed by Yamada and Nakano [12] is one such GLS method, and it has been applied successfully to job-shop scheduling problems. This section briefly reviews neighbourhood search and the MSXF.

3.1 Neighbourhood search

An outline of a neighbourhood search (NS) for minimizing $V(x)$ is described in Algorithm 3.1, where x denotes a point in the search space, $V(x)$ denotes its objective function value and $N(x)$ its neighbourhood. The termination condition can be given, for example, as a fixed number of iterations L.

Step 1 in Algorithm 3.1 defines the NS operator: the main part of NS. This operator is categorized by the way a point is selected from $N(x)$, which is called the *choice criterion*. For example, a descent method selects a point $y \in N(x)$ such that $V(y) < V(x)$. A stochastic method probabilistically selects a point according to the Metropolis

Algorithm 3.1 Neighbourhood search

- Select a starting point: $x_{best} = x = x_0$.

do
1. Select a point $y \in N(x)$ according to the given criterion based on the value $V(y)$. Set $x = y$.
2. If $V(x) < V(x_{best})$ then set $x_{best} = x$.

until some termination condition is satisfied.

---

criterion, i.e. $y \in N(x)$ is selected with probability 1 if $V(y) < V(x)$; otherwise, with probability:

$$P_T(y) = exp(-\Delta V/T), \text{ where } \Delta V = V(y) - V(x) \ . \tag{1}$$

Here P_T is called the *acceptance probability*. Simulated Annealing (SA) is a method in which the parameter T (called the *temperature*) decreases to zero following an annealing schedule as the number of iterations increases.

Although SA is a well-known stochastic method and has been successfully applied to many problems including scheduling problems, it would be unrealistic to apply a full SA search within a GA because it would consume too much time. Therefore a restricted search with a fixed temperature parameter $T = c$ is used in MSXF.

3.2 Multi-step crossover fusion

The genetic crossover operator has two functions, which we denote by F1 and F2. Firstly (F1) it focuses attention on a region between the parents in the search space; secondly (F2), it picks up possibly good solutions from that region. Unlike traditional crossover operators, MSXF is more search oriented: it is designed as an extension of local search algorithm described in Algorithm 3.1, but has the functions F1 and F2, and it is still called 'crossover'.

MSXF carries out a short term 'navigated' local search starting from one of the parent solutions to find new good solutions (F2), where the other parent is used as a reference point so that the search direction is biased toward it and therefore the search is limited between the parents (F1). A stochastic local search algorithm is used for its base algorithm. A similar idea is described under the title 'path relinking' [3] in the context of tabu search. MSXF is defined in a problem-independent manner using a neighbourhood structure and a distance measure, both of which are very common for most combinatorial optimization problems.

Let the parent solutions be p_0 and p_1, and let the distance between any two individuals x and y in any representation be $d(x, y)$. A short term local search is carried out starting from p_0 and using p_1 as a reference point as follows. First x is set to p_0. All members in $N(x)$ are sorted so that $y_i \in N(x)$ with a smaller index i has a smaller distance $d(y_i, p_1)$. Here $d(y_i, p_1)$ can be estimated easily if $d(x, p_1)$ and the direction of the transition from x to y_i are

Algorithm 3.2 Multi-Step Crossover Fusion (MSXF)

- Let p_0, p_1 be parent solutions.
- Set $x = q = p_0$.

do
- For each member $y_i \in N(x)$, calculate $d(y_i, p_1)$.
- Sort $y_i \in N(x)$ in ascending order of $d(y_i, p_1)$.

  do
  1. Select y_i from $N(x)$ probabilistically according to a probability inversely proportional to the index i.
  2. Calculate $V(y_i)$ if y_i has not yet been visited.
  3. Accept y_i with probability one if $V(y_i) \leq V(x)$, and with $P_c(y_i)$ otherwise.
  4. Change the index of y_i from i to n, and the indices of y_k $(k \in \{i+1, i+2, \ldots, n\})$ from k to $k - 1$.

  until y_i is accepted.
- Set $x = y_i$.
- If $V(x) < V(q)$ then set $q = x$.

until some termination condition is satisfied.

- q is used for the next generation.

---

known; it is not necessary to generate and evaluate y_i. One of the members $y_i \in N(x)$ is selected with a probability inversely proportional to the index i. Then y_i is accepted according to the Metropolis criterion using $T = c$ in Equation (1). MSXF is described in outline in Algorithm 3.2. As previously suggested, the termination condition can be given by, for example, a fixed number of iterations L in the outer loop. The best solution q is used for the next generation.

MSXF is not applicable if the distance between p_0 and p_1 is too small compared to the number of iterations. If this happens very often, it means that the population has already converged to a specific region of the search space. In such a case, a mutation operator called *Multi-Step Mutation Fusion* (MSMF) is applied to help diversify the search again. MSMF can be defined in the same manner as MSXF except that $N(x)$ members are sorted in descending order of $d(y_i, p_1)$ in Algorithm 3.2, and the most distant solution is stored and used in the next generation instead of q, if q does not improve the parent solutions.

4 Landscape analysis

According to Höhn and Reeves [7], a landscape is defined by a triple of a search space, an objective function and a distance measure. The link between landscape and search algorithm is given by the NS operators used in the algo-

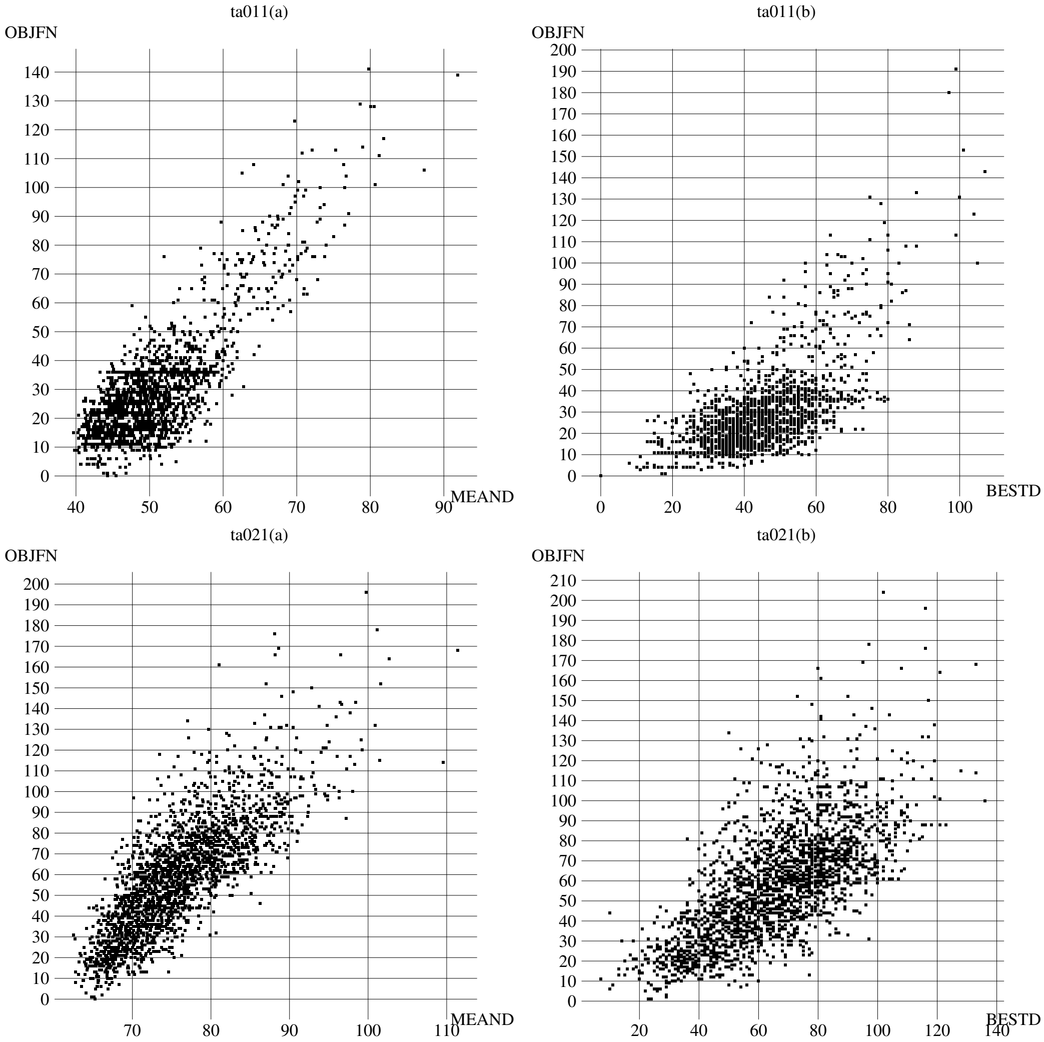

Figure 2: 1841 distinct local optima obtained from 2500 short term local search for the ta011 (20×10) problem and 2313 distinct local optima for the ta021 (20×20) problem are plotted in terms of (a) average distance from other local optima and (b) distance from global optima (x-axis), against their relative objective function values (y-axis).

rithm. Because these operators generate new points in the search space relative to a given point, they define a distance $d_{\mathcal{N}}(s,t)$ on the search space given by the minimum number of applications of operator $\mathcal{N}$ that will convert element t into element s.

One can understand the degree of difficulty of the given combinatorial optimization problem by looking at its landscape: if the landscape is simple and has only one peak, it is very easy to find the global optimum by using simple best ascent search. Unfortunately most NP-hard combinatorial optimization problems, including PFSP, have very 'rugged' landscapes with many false peaks under any NS operator.

Recently, Boese et al. [2] have shown that an appropriate choice of NS operator introduces some neat structure into the landscape. In this 'big valley' structure, local optima occur in clusters – good candidate solutions are usually to be found 'fairly close' to other good solutions. If a landscape has this structure, it would support the idea of generating new starting points for search from a previous local optimum rather than from a random point in the search space.

Before we apply our GLS method to PFSP, we investigate whether there is a big valley structure for the PFSP and the NS operator using the representative neighbourhood and a stochastic search based on Equation (1) at $T = c$ (constant temperature). For the same PFSP but with simpler NS operators, similar experiments reported in Reeves [6] found such a landscape did occur.

As discussed in [4, 2, 6], the existence of a big valley structure can be examined by first generating a set of random local optima and then observing the correlation between their objective function values and their distances to the nearest global optimum, and/or their average distances to other local optima. The distance used here should be $d_{\mathcal{N}}$ for an operator $\mathcal{N}$. However this distance is difficult to compute, and precedence-based distance is used here as an approximation.

Figure 2 shows a scatter plot of random local optima for problems ta011 and ta021, being respectively the first of Taillard's 20×10 and 20×20 groups of problems [9]. Each local optimum is generated by running the neighbourhood search described in Algorithm 3.1 with $L = 5000$ based on the stochastic method with acceptance probability P_c, $c = 5$. Extensive preliminary experiments found only two distinct global optima for the ta011 problem, very close to each other in terms of the precedence-based distance (the distance is two) and only one global optimum for ta021 problem; however one cannot rule out the possibility of finding other different global optima by continuing the search. However, more than 2500 global optima were found for the smaller ta001 (20×5) problem by spending the same amount of CPU time.

The x-axis in Figure 2 represents (a) the average precedence-based distance from other local optima (MEAND), and (b) the precedence-based distance from one of the nearer global optima (BESTD). The y-axis represents their objective function values relative to the global optimum. These plots clearly show that there are good correlations between the distances and objective function values. The calculated correlation coefficients for each plot are: ta011(a): 0.74, ta011(b): 0.50, ta021(a): 0.62 and ta021(b): 0.44. These values are statistically significant at the 0.1% level, on the basis of 1000 replications in a randomization test [6]. These high correlations suggest that the local optima are radially distributed in the problem space with the global optima as the centre, and the more distant are the local optima from the centre, the worse are their objective function values. Hence, by tracing local optima step by step, moving from one optimum to nearby slightly better one, without being trapped, one can eventually reach a near global optimal solution.

In the analysis above, the precedence-based distance is used as a surrogate for $d_{\mathcal{N}}$, because the minimum number of steps for the neighbourhood operator to reach the global optimum is difficult to compute. Although the precedence-based distance seems to be a good alternative, the approximation still need to be justified. For this purpose, the approximate number of steps to reach the global optimum from each local optimum was calculated by choosing the closest move to the global optimum each time from the neighbourhood. While this does not necessarily give the best distance between two points, it seems likely to give a fairly close upper bound.

Figure 3 (a) shows the correlation between the precedence-based distance and the approximate number of steps for the local optima shown in Figure 2 ta011(a) (correlation coefficient is 0.66). Figure 3 (b) shows that

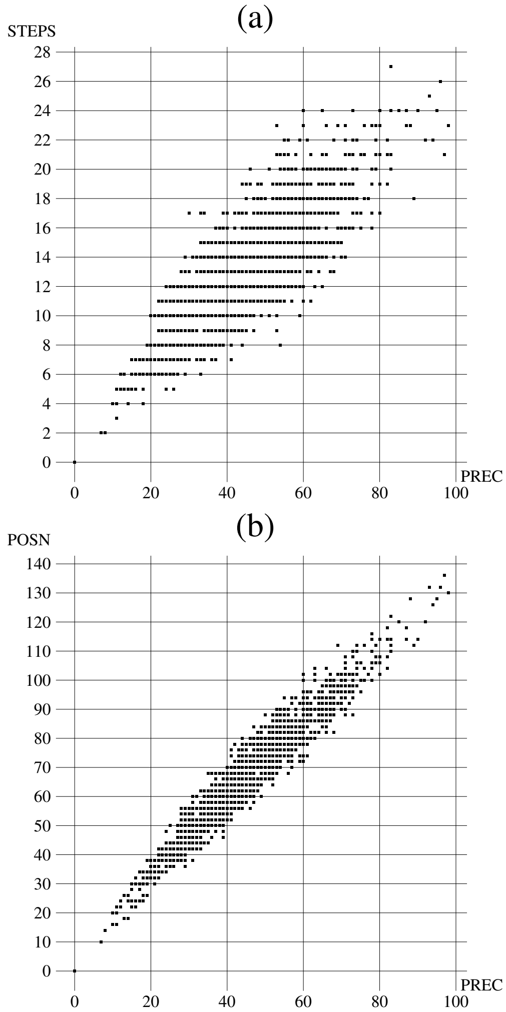

Figure 3: analysis of 1000 random local optima for 20×10 flowshop problem. The x-axis shows precedence-based distance from the global minimum and the y-axis shows the makespan.

there is a strong correlation between the precedence-based distance and the position-based distance for the same local optima (correlation coefficient is 0.91). Thus it does not matter which distance is used. The same kind of experiments were carried out for all Taillard's 20×10 and 20×20 benchmarks, and similar results were obtained in every case. Therefore, the use of the easily-computed precedence-based distance appears to be justified, and the 'big valley' structure can be assumed to hold for this neighbourhood.

5 MSXF-GA for PFSP

As described in Section 3.2, the MSXF operator is designed to find a new local optimum based on previous ones. MSXF-GA provides a framework for traversing local optima without being trapped, by concentrating its attention on the area between the parent solutions and thus eventually finding a very good solution under the assumption of a 'big valley'. MSXF-GA was applied to PFSP using the representative neighbourhood described in Section 2 and the precedence-based distance.

Algorithm 5.1 describes the outline of the MSXF-GA routine for the PFSP using the steady state model proposed in [8, 11]. In this model, the population is ranked according to the makespan values, and the parents are selected from the population with a probability inversely proportional to their ranks. The newly generated solution q is inserted into the population only if its makespan is better than the worst in the current population. To avoid premature convergence even under a small-population condition, if an individual with the same makespan already exists in the population, then q is not inserted into the population in Step 3.

Algorithm 5.1 MSXF-GA for the PFSP

- Initialize population: randomly generate a set of permutation schedules. Sort the population members in descending order of their makespan values.

do

1. Select two schedules p_0, p_1 probabilistically from the population with a probability inversely proportional to their ranks.
2. Do step (2a) with probability P_x, or otherwise do Step (2b).
   (a) **If** the precedence-based distance between p_1, p_2 is less than d_{min}, apply MSMF to p_1 and generate q.
   Otherwise, apply MSXF to p_1, p_2 using the representative neighbourhood $N(p_1)$ and the precedence-based distance and generate a new schedule q.
   (b) Apply Algorithm 3.1 with acceptance probability P_c and the representative neighbourhood.
3. If q's makespan is less than the worst in the population, and no member of the current population has the same makespan as q, replace the worst individual with q.

until some termination condition is satisfied.

- Output the best schedule in the population.

6 Experimental results

In Section 4, the existence of a big valley structure became clear for the relatively small-size PFSP instances. An adaptive multi-start method (AMS) in which new local search

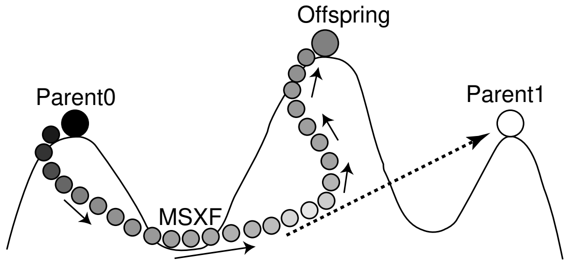

Figure 4: Navigated local search by MSXF-GA: A new search is started from one of the parents and while no other good solutions are found, the search 'navigates' towards the other parent. In the middle of the search, good solutions would be eventually found somewhere between the parents. That direction is then pursued to the top of a hill (or a bottom of the valley, if it is a minimization problem)—a new local optimum.

is concentrated in a region between previously found local optima should be effective at least for these problems. MSXF-GA for PFSP is especially designed as one of the AMS approaches for PFSP. Preliminary experiments show that MSXF-GA is very effective for the problem instances discussed in Section 4, and the global optima are found very quickly. In this section we will extend our investigations to larger-size problems and apply MSXF-GA to a subset of Taillard's benchmark problems.

Table 1 summarizes the performance statistics of MSXF-GA for a subset of Taillard's benchmark problems together with the results found by Nowicki and Smutnicki using their tabu search implementation[5] and the lower and upper bounds, taken from the OR-library [1]. (Upper bounds are the currently best-known makespans, most of them found by a branch and bound technique with computational time unknown). In all, 30 runs were completed for each problem under the same conditions but with different random number seeds. For each MSXF-GA run, population size $= 15$, constant temperature $c = 3$, number of iterations for each MSXF $= 1000$, $d_{min} = n/2$ and $P_x = 0.5$ are used. Each run is terminated after 700 iterations, which takes about 12, 21 and 47 minutes of CPU time respectively for each 50×20, 100×20 and 200×20 problems on a DEC Alpha 600 5/226.

It can be seen that the results for 50×20 problems are remarkable: the solution qualities of our best results are improved over those found in [5] for most of the problems, and some results (marked in bold letters) are even better than the existing best results reported in the OR-library. The results for larger problems are not as impressive as those of 50×20 problems, but still good enough to support our hypothesis. The degradation is probably due to the increasing complexity of the neighbourhood calculation. In fact for problems where the ratio $n/m > 3$, Nowicki and Smutnicki abandoned their representative neighbourhood and used a simple one instead: just moving a job to the beginning or the end of its critical block. They also implemented an efficient way of evaluating all the members in the neighbourhood in a specific order, which is useful for the tabu search but not directly applicable to our stochastic search. (Their paper [5] provides more details.)

7 Conclusions

The landscape for the Permutation Flowshop Scheduling Problem with stochastic local search and the representative neighbourhood structure has been investigated. The experimental analysis using 20×10 and 20×20 Taillard benchmark problems shows the existence of a 'big valley' structure for PFSP. This suggests a well-designed AMS method, such as MSXF-GA in which new local search is concentrated in a region between previously found local optima should be effective in finding near-optimal solutions. MSXF-GA for the PFSP is implemented using the neighbourhood operator and applied to more challenging benchmark problems. Experimental results demonstrates the effectiveness of the proposed method.

References

[1] Beasley, J.E. Or-library: Distributing Test Problems by Electronic Mail. *E. J. of Oper. Res.*, 41:1069–1072, 1990.

[2] Boese, K.D., Kahng, A.B. and Muddu, S. A New Adaptive Multi-Start Technique for Combinatorial Global Optimization. *Operations Research Letters*, pages 101–113, 1994.

[3] Glover, F. and Laguna, M. Tabu Search. *Chapter 3 in Modern Heuristic Techniques for Combinatorial Problems (C.R.Reeves Ed.).* Blackwell Scientific Publications, Oxford, (Recently re-issued (1995) by MacGraw-Hill), London, 1993.

[4] Jones, T. and Forrest, S. Fitness Distance Correlation as a Measure of Problem Difficulty for GAs. In *6th ICGA*, pages 184–192, 1995.

[5] Nowicki, E. and Smutnicki, C. A Fast Tabu Search Algorithm for the Permutation Flow-Shop Problem. *E. J. of Oper. Res.*, 91:160–175, 1996.

[6] Reeves, C.R. Landscapes, Operators and Heuristic Search. *Annals of Operations Research (to appear)*, pages ?–?, 1997.

[7] Reeves, C.R. and Höhn, C. Are Long Path Problems Hard for Genetic Algorithms? In *4th PPSN*, pages 134–153, 1996.

[8] Syswerda, G. Uniform Crossover in Genetic Algorithms. In *3rd ICGA*, pages 2–9, 1989.

[9] Taillard, E. Benchmarks for Basic Scheduling Problems. *E. J. of Oper. Res.*, pages 278–285, 1993.

[10] Ulder, N.L.J., Pesch, E., van Laarhoven, P.J.M., Bandelt, J.H, and Aarts, E.H.L. Genetic Local Search Algorithm for the Traveling Salesman Problem. In *1st PPSN*, pages 109–116, 1994.

[11] Whitley, D. The Genitor Algorithm and Selection Pressure: why rank-based allocation of reproductive trials is best. In *3rd ICGA*, pages 116–121, 1989.

[12] Yamada, T. and Nakano, R. Scheduling by Genetic Local Search with Multi-Step Crossover. In *4th PPSN*, pages 960–969, 1996.

Table 1: Results of the Taillard benchmark problems

	50 × 20				100 × 20				200 × 20			
No.	best	avg.	nowi	lb – ub	best	avg.	nowi	lb – ub	best	avg.	nowi	lb – ub
1	**3861**	3880	3875	3771–3875	6242	6259	6286	6106–6228	11272	11316	11294	11152–11195
2	**3709**	3716	3715	3661–3715	6217	6234	6241	6183–6210	11299	11346	11420	11143–11223
3	**3651**	3668	3668	3591–3668	6299	6312	6329	6252–6271	11410	11458	11446	11281–11337
4	**3726**	3744	3752	3631–3752	6288	6303	6306	6254–6269	11347	11400	11347	11275–11299
5	**3614**	3636	3635	3551–3635	6329	6354	6377	6262–6319	11290	11320	11311	11259–11260
6	3690	3701	3698	3667–3687	**6380**	6417	6437	6302–6403	11250	11288	11282	11176–11189
7	3711	3723	3716	3672–3706	6302	6319	6346	6184–6292	11438	11455	11456	11337–11386
8	**3699**	3721	3709	3627–3700	6433	6466	6481	6315–6423	11395	11426	11415	11301–11334
9	3760	3769	3765	3645–3755	6297	6323	6358	6204–6275	11263	11306	11343	11145–11192
10	3767	3772	3777	3696–3767	6448	6471	6465	6404–6434	11335	11409	11422	11284–11313

best, avg.: our best and average makespan values

nowi: results of Nowicki and Smutnicki

lb, ub: theoretical lower bounds and best known makespans taken from OR-library