

2D and 3D Representations of Solution Spaces for CO Problems

Eugeniusz Nowicki and Czesław Smutnicki

Wrocław University of Technology,
Institute of Engineering Cybernetics,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
enowicki@ict.pwr.wroc.pl, czeslaw.smutnicki@pwr.wroc.pl

Abstract. This paper presents various graphical techniques developed by us to support research of some very hard combinatorial optimization (CO) problems. Thanks to increased efficiency of algorithms and high power of modern PC, we are able to make visualization of the solution space and search trajectories, which in order, allows us to analyze and design modern approximate approaches quickly running towards good solution. We provide unique graphic representations of solution spaces and search trajectories of modern approximate algorithms dedicated for the job-shop problem, the one of the hardest problem in CO.

1 Introduction

Among variety of combinatorial optimization (CO) problems, the *scheduling theory* provides many hard cases, among which the *job shop scheduling problem* is known as a particularly hard one. This multidimensional, discrete CO problem follows from Operations Research (OR) practice, has relatively simple formulation, excellent industrial applications, finite but huge number of solutions and unfortunately is strongly NP-hard. That's why it is also considered as a permanent indicator of practical efficiency of advanced scheduling algorithms, manifesting indirectly the power of OR theory and algorithmic skillfulness. During last fifty years, a lot of various methods have been designed to solve the problem in a quick time, see reviews in [1,8].

Modern approximate approaches, as an example tabu search (TS), simulated annealing (SA) and genetic search (GS), dominate in the job-shop research carried out in the last decade. Although made works improve significantly algorithms efficacy (measured by the accuracy as opposing to the running time) observed phenomena of surprisingly good accuracy and capricious convergence depending on instances remain unexplained to the end. One supposes that the answer lies in the chaotic structure of the solution space, characteristic for CO problems. In particular, amazing TS algorithm, very fast and of excellent accuracy below 4%, designed by us few years ago for the job-shop problem [4], inspires us to explore these phenomena using advanced graphical tools.

We used for visualizations selected public benchmarks with the number of operations varying from 225 to 2,000, [7]. Research was supported by Grant T11A 01624 of the State Committee for Scientific Research.

2 Problem, Models, Solutions

There are a set of jobs $N = \{1, \dots, n\}$, a set of machines $M = \{1, \dots, m\}$ and a set of operations $O = \{1, \dots, o\}$. Set O is decomposed into subsets corresponding to the jobs. Job j consists of a sequence of o_j operations indexed by $(l_{j-1} + 1, \dots, l_{j-1} + o_j)$ which should be processed in that order, where $l_j = \sum_{i=1}^j o_i$, $j = 1, \dots, n$ ($l_0 = 0$), and $\sum_{i=1}^n o_i = o$. Operation i must be processed on machine $\mu_i \in M$ during the time $p_i > 0$, $i \in O$. Each machine can process at most one operation at a time. A feasible schedule is defined by start times $S_i \geq 0$, $i \in O$, such that the above constraints are satisfied. We look for a feasible schedule that minimizes the makespan $\max_{i \in O} (S_i + p_i)$.

The choice of the model (MILP, disjunctive graphs) determines the form of the solution and thus character, structure, dimensionality and cardinality of the space. We operate on the following permutation-and-graph model, [4], which has the smallest cardinality with minimal fraction of infeasibility.

Set O can be decomposed into subsets $M_k = \{i \in O : \mu_i = k\}$, $k \in M$. The sequence of operations from M_k is defined by $\pi_k = (\pi_k(1), \dots, \pi_k(|M_k|))$, permutation on M_k , $k \in M$. Let Π_k be the set of all permutations on M_k . The processing order of operations is defined by m -tuple $\pi = (\pi_1, \dots, \pi_m)$, where $\pi \in \Pi = \Pi_1 \times \Pi_2 \times \dots \times \Pi_m$. It is linked with schedule S_i , $i \in O$ by the digraph $G(\pi) = (O, R \cup E(\pi))$ with a set of nodes O and a set of arcs $R \cup E(\pi)$, where $R = \bigcup_{j=1}^n \bigcup_{i=1}^{o_j-1} \{(l_{j-1} + i, l_{j-1} + i + 1)\}$ and $E(\pi) = \bigcup_{k=1}^m \bigcup_{i=1}^{|\pi_k|-1} \{(\pi_k(i), \pi_k(i+1))\}$. Node $i \in O$ in $G(\pi)$ has weight p_i , arcs have weight zero. Processing order π is feasible if $G(\pi)$ does not contain a cycle. Then S_i equals the length of the longest path going to the vertex i (but without p_i) in $G(\pi)$, $i \in O$. Makespan $C_{\max}(\pi)$ for feasible π equals the length of the longest path (critical path) in $G(\pi)$. Thus, Π is the solution space, where a processing order $\pi \in \Pi$ is its element.

3 Solution Space

Space Π is discrete, has dimension o , is finite, but its cardinality is of astronomical size. For the small historical instance FT10 ($n = 10$, $m = 10$, $o = nm = 100$), we have $|\Pi| = \prod_{k=1}^m (|M_k|!) = (n!)^m \approx 4 \cdot 10^{65}$, however only (!) approximately $4 \cdot 10^{48}$ solutions are feasible, see [5]. The projection of feasible solutions of FT10 on 2D plane printed in high resolution 2400 dpi covers area of $4 \cdot 10^{32}$ km², whereas the surface of Jupiter has only $6.4 \cdot 10^{10}$ km². “Typical” searching procedure is able to check at most 10^9 solutions, which corresponds to only 0.1 m² of the area and can be shown as thread of 0.01 mm width and 10 km length.

The space Π can be perceived as hyper-graph, where nodes represent solutions and each arc links two solutions, so that the former can be obtained from the later by using a *move*. For the given solution π , the subset of solutions (nodes) linked by single arc with π is called the *hyper-neighborhood* of π , and can be generated by various types of moves. The move changes sequence of operations processed on a common machine as follows: (A) A-move swaps a pair $v = (a, b)$ of two adjacent operations a, b , (S) S-move swaps a pair $v = (a, b)$ of

two any operations a, b , (I) I-move for a pair $v = (a, b)$ of operations a, b , cuts out operation a and then inserts it immediately after operation b (if a precedes b), or immediately before operation b (otherwise). Let π_v denote the processing order obtained after the move v . The set of all moves of type Z performed from π will be denoted by $V^Z(\pi)$, whereas the obtained hyper-neighborhood by $\mathcal{N}^Z(\pi) = \{\pi_v : v \in V^Z(\pi)\}$, $Z \in \{A, S, I\}$. The sequence of successive solutions $\pi^0, \pi^1, \dots, \pi^r$, where $\pi^{i+1} \in \mathcal{N}^Z(\pi^i)$, $i = 0, 1, \dots, r - 1$ constitutes the *trajectory* (path in the hyper-graph). Hyper-neighborhoods of type A, S, I have strong connectivity property, i.e. for each pair of solutions $\alpha, \beta \in \Pi$ exists trajectory so that $\alpha = \pi^0, \pi^1, \dots, \pi^r = \beta$ and r is finite.

Because of hyper-neighborhood size, its selected subset has been used in local search and called simply *neighborhood*. There are a few attributes for neighborhoods generated by moves: size, feasibility, usefulness and connectivity, [5], which influence on trajectories going through the space. The most popular neighborhoods are the following two based on A-moves. The first is generated by move set $V^{A1}(\pi) \subseteq V^A(\pi)$, where $v = (a, b) \in V^{A1}(\pi)$ if only a and b belongs to a critical path in $G(\pi)$, [3]. All these moves are feasible (leads from feasible to feasible solution), but some of them are useless (leads to essentially worse solutions). Neighborhood has only weak connectivity property (β is not any but one of optimal solutions). The second is generated by the move set $V^{A2}(\pi) \subseteq V^{A1}(\pi)$, obtained from $V^{A1}(\pi)$ by elimination of all useless moves, with the help of block property; all moves are feasible, [4]. Although $\mathcal{N}^{A2}(\pi)$ does not have the connectivity property, in experiments it behaves better than other neighborhoods.

4 Distances in the Space

We define the distance $D^Z(\pi, \sigma)$ between processing orders $\pi, \sigma \in \Pi$ as the minimal number of moves $Z \in \{A, S, I\}$ necessary to go from π to σ (i.e. the shortest path in hyper-graph). Clearly, we have $D^Z(\pi, \sigma) = 1$ for $\sigma \in \mathcal{N}^Z(\pi)$. It has been shown, [5], that because move is being performed inside single machine, then permutation π_k can be mapped onto permutation on set $N^k = \{1, 2, \dots, |M_k|\}$, $k \in M$. Thus, the measure $D^Z(\pi, \sigma) = \sum_{k \in M} D_k^Z(\pi_k, \sigma_k)$ and its properties can be derived from properties for measure $D_k^Z(\pi_k, \sigma_k)$ between permutations π_k and σ_k on the set N^k . Follows results from [2], there have been collected in [5] basic properties of measures for all types of moves (A,N,I): recipes for calculation, mean value, variance and computational complexity. Here only notice that the measure $D_k^A(\pi_k, \sigma_k)$ equals the number of inversions in permutation $\pi_k^{-1} \sigma_k$, where π_k^{-1} is the inverse permutation π_k , $k \in M$. Hereinafter we analyze the use of A-moves only, thus upper index A will be skipped in proper notions.

For the sake of huge size of the space, its character can be analyzed statistically by random sampling. In Fig. 1 (left) one can find, for typical benchmark ta04 ($n = 15, m = 15, o = 225$), the distribution of $DIST(\pi) = 100 * D(\pi, \pi^{REF}) / D_{max}$, where $D_{max} = \max_{\alpha, \beta} D(\alpha, \beta) = mn(n - 1) / 2$, taking into account the random sample of ALL solutions and the random sample of FEASible solutions, where π^{REF} is a fixed reference solution (optimal solution).

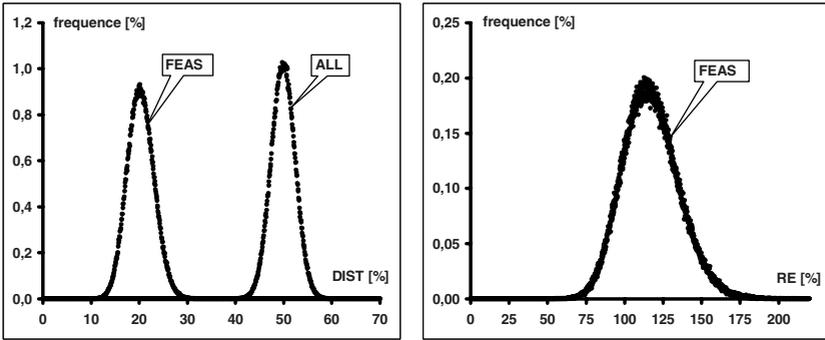


Fig. 1. Instance ta04: Distribution of $DIST(\pi)$ (left) and $RE(\pi)$ (right) for 500,000 random solutions; ALL - for any solutions, FEAS - for feasible solutions

Curve ALL fits ideally to the theoretical distribution of $D(\alpha, \beta)$, with the mean $0.5 * D_{max}$. From curve FEAS one can read the mean value of $DIST$ distribution for feasible solutions equal approximately $0.25 * D_{max}$; this holds for almost all benchmarks ta-. Both distributions do not depend on the choice of π^{REF} .

The distribution of $RE(\pi) = 100 * (C_{max}(\pi) - C_{max}(\pi^{REF}) / C_{max}(\pi^{REF}))$, which is the relative error, for the same sample of FEASible solutions is shown in Fig. 1 (right). The distribution is close to gaussian. Randomly found solution has $RE \approx 120\%$ (standard deviation 15%), whereas we know that the best constructive algorithm called INSA offers $RE \approx 20\%$ and the best improvement algorithm TS has $RE \approx 4\%$. Note also, that finding feasible solution with RE below 50% is statistically infinitesimal. Results remain valid for all ta- benchmarks.

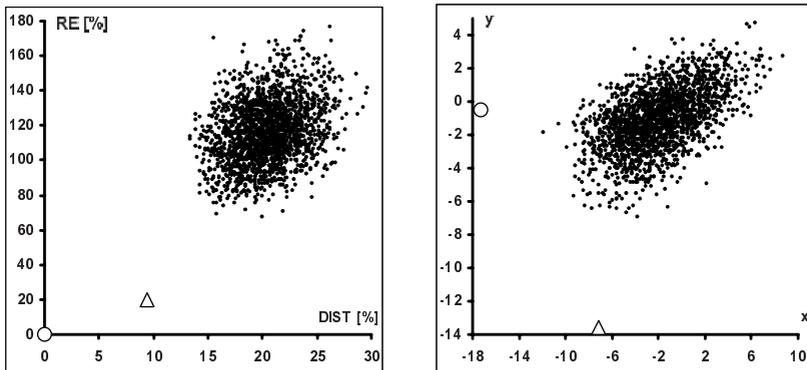


Fig. 2. Instance ta04: 2,000 random feasible solutions in coordinates $DIST/RE$ (left) and X/Y (right); Δ marks starting solution found by INSA, \circ - optimal solution

The main goal of the space analysis is the identification of the *landscape* and its typical element called the *big valley*, which plays role of an *attractor* in the space. Its presence means that distances between good locally optimal solutions and the global one are significantly positively correlated with goal function values of these solutions. It also means that majority of very good local optima concentrates in a relatively small area of the solution space.

In Fig. 2 one can find plot of random feasible solutions showing the correlation 0.3 between DIST and RE (statistically significant on the level 0.1%), both calculated with respect to the optimal solution π^{REF} . This fact confirms the existence of “big valley”, not only for good locally optimal solutions (as it showed in [5]) but also for *all* feasible solutions.

5 2D Transformations

Our aim is to transform solutions $\pi^j \in \Pi, j = 1, \dots, t$ into points on 2D plane “rationally” preserving the distance from Π . Problem of transformation continuous n -dimensional space R^n into space R^2 is known for years, see the review in [6]. However, some troubles appear in the application to CO problems. First, the source space is discrete. Second, metric in Π is not Euclidian, takes only discrete values, and can influence on the final result of the transformation. Third, and the most important in our opinion, the calculation process should be cheap since in practice we need to transform 1,000 . . . 10,000 solutions from the space with dimension 200 . . . 2,000. That’s why we focus chiefly on *reference points* approaches. Among a few proposals, we describe below only one, fast and of satisfactory *mapping accuracy* (accuracy is measured by the correlation between distances in space Π and Euclidian distances of 2D images). Let $\sigma^i, i = 1, \dots, r$ be a sequence of reference solutions. Each π is transformed into point (x, y) on 2D plane according to the following formula

$$\begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^r D(\pi, \sigma^i) \begin{bmatrix} \sin(\frac{(i-1)\pi}{r}) \\ \cos(\frac{(i-1)\pi}{r}) \end{bmatrix} \tag{1}$$

We performed tests on most of 80 ta- benchmarks, with $r = 3$ reference solutions: $\sigma^1 = \pi^{REF}$, σ^2 – solution generated by algorithm INSA used as the starting point for our method TS in [4], and σ^3 – solution generated by SPT rule, one of the most commonly applied in constructive algorithms. In the series of graphical runs for a few known algorithms, we found similar results for all benchmarks. Then, we discuss here only one, typical. In Fig. 3 one can find search trajectories (in various axes) performed by algorithms TS, SA, RANdom; all work on neighborhood \mathcal{N}^{A2} . Compare chaotic track of algorithm RANdom trapped in a local sub-area of the solution space with directional track of TS; in these conditions the behavior of SA is similar to RANdom. (From unpublished here results we found that RANdom and SA behaves even worse if we replace \mathcal{N}^{A2} by greater neighborhood e.g. \mathcal{N}^{A1} or \mathcal{N}^A .) This static figure does not reflect the evolution of the trajectory in time, therefore in Fig. 4 we have provided speed of convergence of tested methods in iterations (in time) – TS is the best.

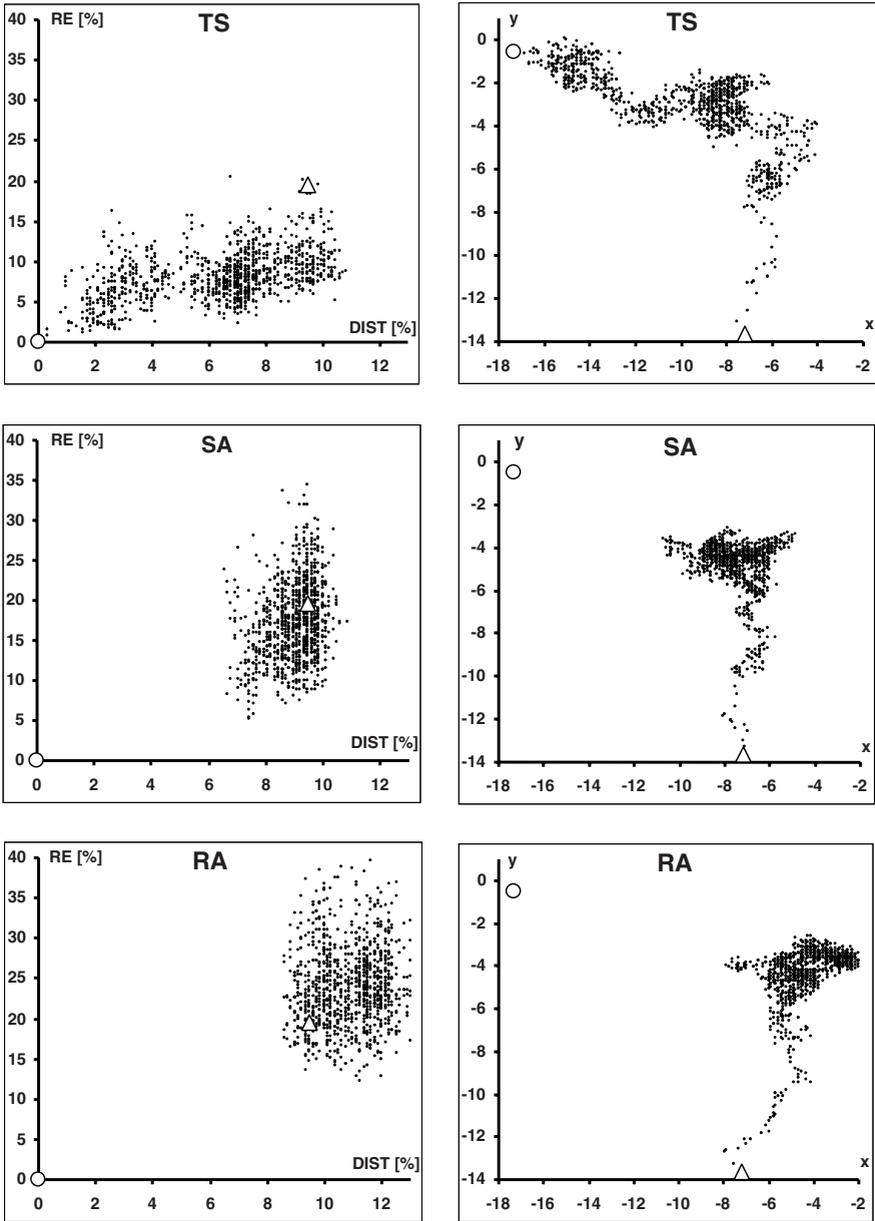


Fig. 3. Instance ta04: 10,000 iterations of algorithms TS, SA, RANDOM (plotted every 10 iterations) in coordinates DIST/RE (left) and X/Y (right); Δ marks starting solution found by INSA, O - optimal solution

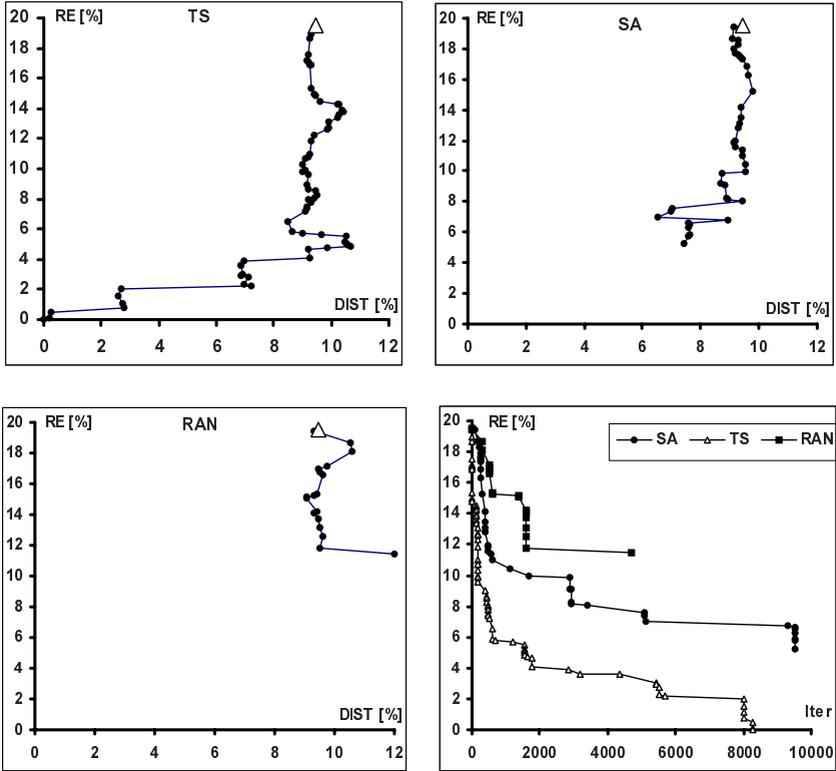


Fig. 4. Speed of convergence for the instance and algorithms from Fig. 3; there are marked only solutions reducing the makespan

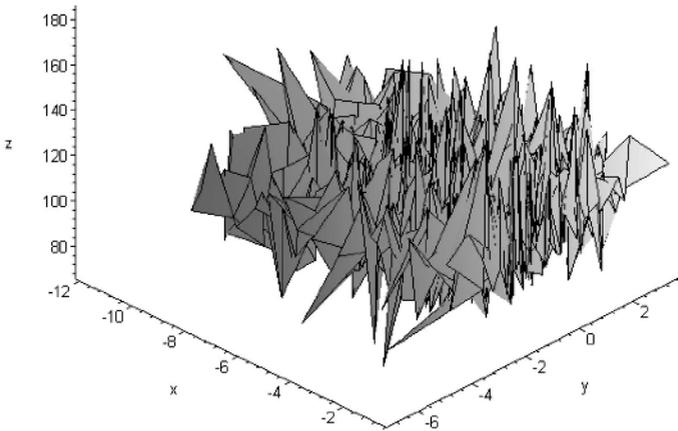


Fig. 5. Instance ta04: Optimization surface spanned on solutions from Fig. 2 right

Clearly, the choice of r and the location of reference solution influences on the image – this subject is currently being studied. Primal tests suggest that reference solutions should be spread widely, whereas the increase of r does not influence on mapping accuracy significantly.

6 Optimization Surfaces

Using certain transformation of solutions into 2D points, we can try to draw in 3D the optimization surface, spanned on points (x_i, y_i, z_i) , $i = 1, \dots, t$, where (x_i, y_i) is the image of solution π^i on 2D, $z_i = RE(\pi^i)$, and π^1, \dots, π^t is the sample of feasible solutions from Π . An example of such surface spanned on a part of points from Fig. 2 is shown in Fig. 5. It destroys our classical imagination about optimization process – the surface is sharp without any regularity, like the brush, fur, high mountain or stone forest. Local extremes are distributed densely, chaotically, and act as deception points. The primal two natural associations are more suitable since through the increase of the sample cardinality one obtains higher density of tops and wells on almost the same area. The figure indicates fundamental troubles in the research of CO problems, formulated as the rhetorical question “How to find the proper path on this surface being a small ant with limited visibility?” This pessimistic attitude is overcome by existing algorithms – TS from [4] finds such path intentionally and consciously, tending to $RE=0\%$ for ta04 ($RE \approx 4\%$ for all ta- instances). Other methods, like SA or GS search space through the sampling points on the surface - properties of such methods are completely predicable. That’s why frequently SA and GS need support by complete/partial neighborhood search procedures in order to reach satisfactory quality.

References

1. Błażewicz, J., Domschke, W., Pesh, E.: The job shop scheduling problem. Conventional and new solution techniques. *European Journal of Operational Research* **93** (1996) 1–33
2. Knuth, D.E.: The art of computer programming. Addison Wesley Longman (1997)
3. Van Laarhoven, P., Aarts, E., Lenstra, J. K.: Job-shop scheduling by simulated annealing. *Operations Research* **40** (1992) 113–125
4. Nowicki, E., Smutnicki C.: A fast tabu search algorithm for the job-shop problem. *Management Science* **42** (1996) 797–813
5. Nowicki, E., Smutnicki, C.: Some new ideas in TS for job-shop scheduling. Technical Report 50/2001. In print in: Rego C., Alidaee B. (eds.): *Adaptive Memory and Evolution: Tabu Search and Scatter Search*. Kluwer Academic Publisher (2004)
6. Siedlecki, W., Siedlecka, K., Sklansky, J.: An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition* **25** (1988) 411–429
7. Home page of E. Taillard: <http://www.eivd.ch/ina/collaborateurs/etd/default.htm>
8. Vaessens, R., Aarts, E., Lenstra, J.K.: Job shop scheduling by local search. *INFORMS Journal of Computing* **8** (1996) 303–317