Big Valley in scheduling problems landscape – metaheuristics with reduced searching area

Wojciech Bożejko
Department of Automatics,
Mechatronics and Control Systems,
Faculty of Electronics,
Wrocław University of Science and Technology,
Wyb. Wyspiaskiego 27, 50-370 Wrocław, Poland
Email: wojciech.bozejko@pwr.edu.pl

Czesław Smutnicki
Department of Computer Engineering,
Faculty of Electronics,
Wrocław University of Science and Technology,
Wyb. Wyspiaskiego 27, 50-370 Wrocław, Poland
Email: czeslaw.smutnicki@pwr.edu.pl

Mariusz Uchroński
Department of Automatics,
Mechatronics and Control Systems,
Faculty of Electronics,
Wrocław University of Science and Technology,
Wyb. Wyspiaskiego 27, 50-370 Wrocław, Poland
Email: mariusz.uchronski@pwr.edu.pl

Mieczysław Wodecki
Telecommunications and Teleinformatics Department
Faculty of Electronics,
Wrocław University of Science and Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
Email: mwd@ii.uni.wroc.pl

***Abstract*—Created in the 90s of the past century methods of constructing algorithms (metaheuristic), inspired by the *no free lunch* theorem of Wolpert and Macready, using specific properties of problems, do not meet present expectations of practitioners. Commonly used artificial intelligence algorithms in recent years have also proved to be ineffective in solving a large group of extremely difficult instances of various problems. In the work we present some empirical methods of exploration of solution space in optimization problems whose solutions are represented by permutations. While sampling the set of permissible solutions we designate the histogram of the frequency of occurrence of local minima and on this basis we verify the statistical hypothesis concerning the (normal) distribution of occurrence of these minima. Due to this process we can flexibly change the "radius" of the searched area. Computational experiments performed on examples of the job shop problem are promising and inspire to conduct further research in this direction.**

I. Introduction

Discrete optimization problems, important from a practical and interesting from a theoretical point of view, are mostly placed in the class of NP-hard problems. This fact greatly limits the scope of use of accurate algorithms but inspires the researchers to carry out the study on construction of approximate algorithms that would not only be efficient in terms of time but also could designate acceptable suboptimal solutions. For several years there have been observed an intensive development of methods inspired by nature, classified as the so-called computational intelligence, in which different probabilistic strategies of searching solution space are used. In spite of passing time, the most effective ones are still classic metaheuristic created in the last decade of XX century such as: tabu search, simulated annealing, genetic algorithms and their hybrids based on strategies improving solutions through iterative generation of elements of the solution space (neighborhoods). It follows from the practice of computation that the quality of solutions determined by the approximate algorithms largely depends on the number of tested space elements that are strictly limited by the time of calculations (in practice it is currently around 10^9 solutions). Therefore, it is necessary to generate neighborhoods including most promising solutions (preferably if they are local minima or solutions leading directly to such minima). For this purpose, there are properties being proven enabling intermediate space view of elements, which means fast elimination of not promising solutions from the neighborhood in practice.

The results of many computational experiments conducted at the beginning of the 90s of the last century showed that, despite the dynamic development of computing equipment, the barrier that cannot be exceeded is 100 as the size of the data. In addition, at that time Wolpert and Macready published the so-called no free lunch theorem [19]. It follows from the theorem that without the use of specific properties of the problem being solved, in the construction of the algorithm, it is not possible to obtain an algorithm generally better than others. Therefore, there appeared the need for intensive research on problems concerning properties that could be applied in the construction of accurate and approximate algorithms. As a result of the conducted study many properties have been proven which, through the intermediate view, enable the elimination of large areas of solution space. In addition, as shown by the calculations, they direct the trajectory of calculations in promising areas accelerating the convergence of the algorithm. In spite of passing years, the developed algorithms are still classified as the best ones. Among them there are algorithms solving classical, from the present standpoint of discrete optimization, scheduling problems:

1) *job shop* $(J||C_{\max})$ - Nowicki and Smutnicki [16],
2) *flow shop* $F(||C_{\max})$ - Nowicki and Smutnicki [15], Grabowski and Wodecki [13]

978-1-5386-2402-9/17/$31.00 ©2017 IEEE

3) single-machine with cost-effective criteria $(1||\sum w_iT_i)$ - Wodecki [20], $(1||\sum(v_iU_i + w_iT_i))$ - Bożejko, Grabowski and Wodecki [5].

and many other less known problems. The effectiveness of these algorithms is evidenced by the fact that, known in the literature, example of job shop problem ft10 (Fisher, Thompson [12]), which was finally solved only 25 years after its publication, can be solved today on PC equipment in just a few seconds. Despite such huge progress, it turns out that there is still a large group of examples that are difficult to be solved. Currently used in the design of algorithms properties of problems are not sufficiently effective to designate, in a reasonable time, solutions accepted by practitioners. It seems that general properties of the problem are ineffective in such cases and specific description of individual instances should be taken into account. First of all, the history of the computations (e.g. during the entire run of tabu search algorithm) is tens of terabytes per hour. Analysis of the solution space has been subject to some research, including the traveling salesman problem [3], the graph coloring [2] and some problems of tasks scheduling [17], [14]. Although the solutions to separate instances vary greatly, they have some common features, as they show Big Valley property. There is a significant positive correlation between the value of the objective function of the solution and its distance from the optimal solution (or the best known solution). Frequency of occurrence of local minima there is much bigger than in the other areas of solution space. Previous results are the inspiration for further research of this type.

In this paper we present a new method for the construction of algorithms for solving discrete optimization problems based on landscape analysis of a solved problem instance. In the course of calculations, the search area is dynamically modified. The basis for such course of action is the statistical analysis generated by the local minima algorithm. It enables potential location of Big Valley and leads to intensification of calculations. However, if it turns out that we are stuck at some minimum, then we can use the mechanism enlarging the search area, namely - diversification of calculations. In summary, we present the method of algorithms construction as an iteration consisting of two cycles: external and internal one. In the external cycle, basing on history of calculations, the parameters of the search areas are set, whereas in the inner cycle, by searching out the areas defined in the external cycle, local minima are designated. The set of these minima constitutes the data for analysis in the external cycle. Computational experiments were performed on the basis of certain instances of the job shop scheduling problem.

II. Local search method

For many strongly *NP*-hard combinatorial optimization problems natural representation of solutions are permutations. This is a large group of problems, important for practical reasons, related to the ordering of elements of a finite set. In practical applications to solve such problems there are successfully used algorithms based on the local search method. Well-designated neighborhood is one of the essential elements affecting the effectiveness of this method. The number of neighborhood elements, the method of their designation and reviewing has a decisive effect on an algorithms efficiency (calculation time and the value of the objective function of the determined solution). In the literature there are presented many procedures of generating the neighborhoods for different combinatorial optimization problems. Classical neighborhoods, used for several decades, contain a polynomial number of elements (usually $O(n^2)$) generated by transformations commonly called *moves*, i.e. "slight" changes in some elements of permutations. They are also increasingly frequently used neighborhoods with an exponential number of elements, generated, for instance, by composition of single, traditional moves ([7]). For some of them there exist search algorithms with the polynomial computational complexity (dynasearch method, Congram et al. [9]), whereas for others designating the optimal element is the *NP*-hard problem. Such neighborhoods were used in the first place in TSP problem solving algorithms (their review can be found in the works by Ahuja et al. [1] or Deineko and Woeginger [10]).

III. Solution spaces

In good optimization (exact and approximate) algorithms there are usually many specific, for a given problem, properties of the objective function and space permissible solutions used. Optimization combinatorial problems have no classical analytical properties, such as: convexity, consistency, variability and they mostly have a lot of extremes. In addition, the power of the space (relative to data size) is growing very fast ("exponential explosion"), what excludes the use of conversion methods (even application of multiprocessor ones). Therefore, there is an urgent need for intensive study of *fitness landscape*, with triple $(\mathcal{S}, F, d)$, where $\mathcal{S}$ is the solution space, F the goal function and d the measure of the distance between the solutions, see Burke(red.)[8]. (*fitness landscape*), thus the relationship between the values of the objective function and the distances between solutions in space. Knowledge of such dependencies can be particularly useful in certain advanced heuristic methods such as the *path relinking*), where various measures are used to evaluate the distance between solutions.

Let Φ (the set of all *n*-element permutations) be the solution space for some optimization problem. For analyzing the distance between pairs of any arbitrary permutations $\pi, \sigma \in \Phi$) the following measures are most commonly used (Diaconis [11]):

1) Kendall's tau:
   $I(\pi,\sigma)$ = minimum number of swapped adjoining pairs while conducting a permutation π^{-1} in σ^{-1}.
2) Cayley's distance:
   $T(\pi,\sigma)$ = minimum number of transpositions (swaps) that should be done while carrying out a permutation π in a permutation σ.
3) Ulam's distance:
   $L(\pi,\sigma) = n -$ (the length of the longest rising substrate in a permutation product $\sigma \cdot \pi^{-1}$).
4) Footrule:
   $D(\pi,\sigma) = \sum |\pi(i) - \sigma(i)|$.
5) Spearman's rank correlation:
   $S^2(\pi,\sigma) = \sum(\pi(i) - \sigma(i))^2$.
6) Hamming distance:
   $H(\pi,\sigma) = n-$(the number of i such that $\pi(i) \neq \sigma(i)$).

The first three of these measures are directly related to typical moves defined in the permutations and currently used in the process of generating neighborhoods.

Kendal's tau is associated with performing moves of swap type only on adjacent elements in the permutation. The neighborhood generated in such a way has $n-1$ elements, whereas the algorithm calculating the value for this measure has computational complexity $O(n^2)$. *Cayley measure* corresponds to the number of swap type moves (transpositions) transforming any space element into any other. The neighborhood generated by these movements has $n(n-1)/2$ elements and the computational complexity of this measure is $O(n)$. In turn, *Ulama*measure is closely related to insert type moves. The neighborhood generated by these moves has $n(n-1)$ elements. It is possible to show ([4]) that for any permutations $\alpha, \beta \in \Phi_n$ the value of Ulam's measure $L(\alpha, \beta) = Ins(\alpha^{-1}, \beta^{-1})$, where $Ins(\pi, \delta)$ is the minimum number of insert type moves needed to perform a permutation π in δ. The algorithm for determining the value of Ulam's measure has a computational complexity $O(n \ln n)$.

The presented measures characterize the relationships (distances) between permutations generated by single moves. However, the problem becomes much more difficult to consider if the compositions of moves are taken into consideration, what is increasingly being used in the search of solution space for very difficult combinatorial optimization problems. In such cases a useful procedure may be an introduction and then a study of the complex move composed as a permutation of special properties, e.g. involution.

IV. Fitness landscape analysis

We are considering an optimization problem:

$$\min\{F(\pi) : \ \pi \in \Phi\},$$

where Φ is the set of permutations of elements of the set $j = \{1, 2, \ldots, n\}$, and the objective function (optimization criterion) $F : \Phi \to \mathcal{R}^+$. For a fixed set of moves (neighborhood generator) let $G = (\Phi, A)$ be a neighborhood graph, and τ a measure (distance) specified on pairs of elements of Φ (vertices of the graph G). A is a set of edges – moves between elements of Φ generated by a considered set of moves. Let D be a set of local minima of the considered problem. We construct a graph $H = (D, U)$, where the edge set

$$U = \{\{u, v\} : \ u, v \in D, u \neq v \text{ and}$$
$$\{u, v\} \in U \iff \{u, v\} \in A\}.$$

Notation:

- $d(u, v)$ - the length of the shortest path from the vertex u to v in graph H,
- $d_u = \max\{d(u, v) : \ v \in U\}$.

The size of $d^* = \min\{d_u : \ u \in U\}$ is called the *radius*, whereas vertex $c \in D$ such that $d_c = d^*$ *central vertex* of the graph H. Obviously the radius value depends on the choice of measurement of distance. Tests of empirical distribution of measurement of distance in relation to local extremes of the central vertex gave results similar to the normal distribution (example `ta11` of Taillards test instances for the job shop problem is given in Figures 2–3).

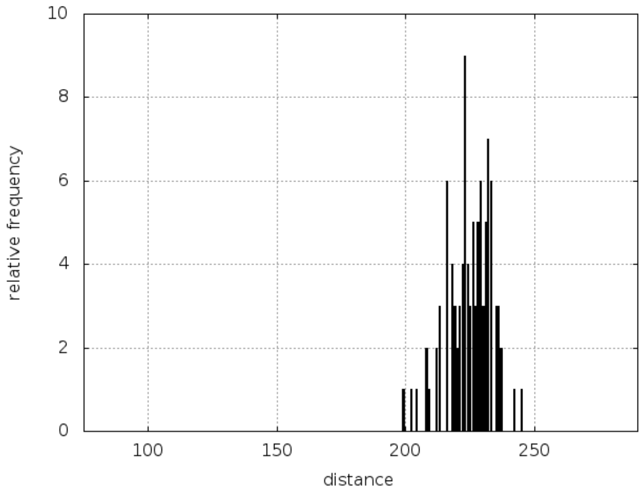

Fig. 1. Distribution of locally optimal solutions distances to the center solution for the `ta11` instance.

In the studies we propose the design of local or population based search algorithms, which in the process of searching the solution space will be limited to the area of the prevalence of local extremes. If it turned out that this is a normal distribution $\mathcal{N}(m, \sigma)$, it would be a range $(m - 3\sigma, m + 3\sigma)$, so it would be also possible to limit the empirically established range $(min\_distance, max\_distance)$, designated when creating a histogram of frequency, where $min\_distance$ and $max\_distance$ are respectively the smallest and greatest distance from the central vertex to one of the local extremes determined by sampling the solution space.

We will take a central vertex as a reference point for the searching area determination for a metaheuristic (see Figure 1). Radius of the area (related to solutions with maximal distance to the center c^*) will be dependent to standard deviation σ. We will also consider optimal (or the best known) solution as a reference point (Figure 2), but such a method can be only done after finding this solution, so it is not usable during optimal solution searching – only for post-analysis. For the comparison, distribution of distances are also presented for random feasible solutions in Figure 3. Although this is only an example (instance `ta11` for the Taillard's job shop benchmarks) this situation is typical also for other job shop instances – the distribution curve, which looks like Gauss curve, is shifted to the left for local optima, which is visible in Figures 1 and 2. This observation indirectly testifies to the existence of a Big Valley.

V. Big Valley researching

The algorithm of the job shop problem solving, in which the searched area is limited by the number of moves (distance) related to a local minimum (center of the graph D) can be presented as follows.

Step 1: *Solution space sampling*

We determine a random solution set

$$\mathcal{P} = \{\pi_1, \pi_2, \ldots, \pi_n\}.$$

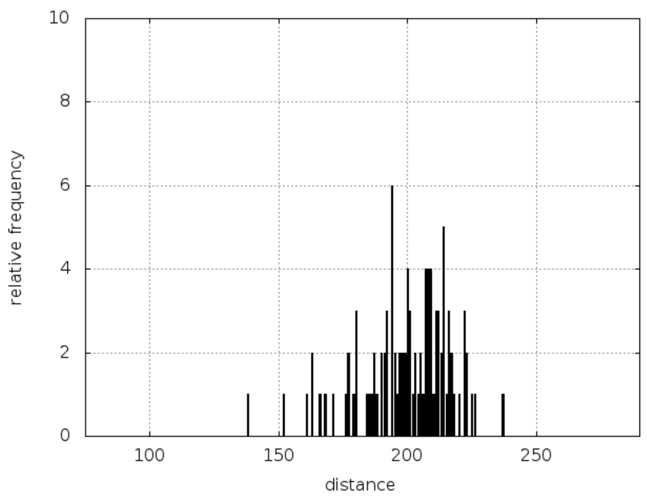

Fig. 2. Distribution of locally optimal solutions distances to the best known solution for the `ta11` instance.

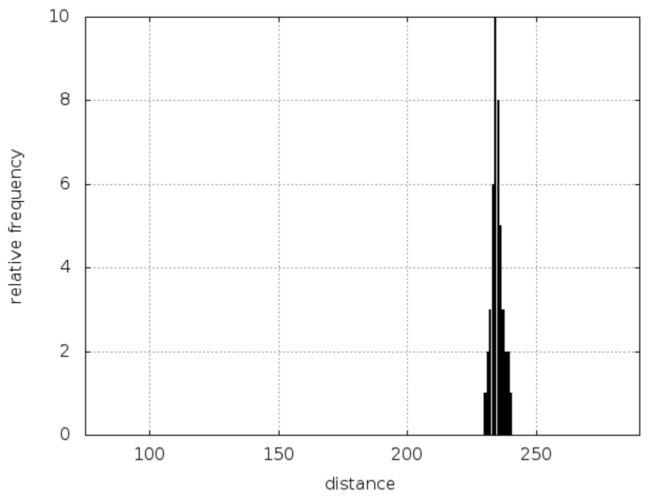

Fig. 3. Distribution of random feasible solutions distances to the best known solution for the `ta11` instance.

Step 2: *Local minima determination*

Applying local improvement algorithm (e.g. tabu search), starting from solutions of the set $\mathcal{P}$, we determine (e.g. in parallel) the set of local minima

$$\mathcal{P}^* = \{\pi_1^*, \pi_2^*, \dots, \pi_n^*\}.$$

Remark: the set $\mathcal{P}^*$ is a random sample from the set of (all) local minima.

Step 3: *Empirical local minima distribution – hypothesis verification*

Hypothesis H_0: elements of the set $\mathcal{P}^*$, due to the chosen distance measure, has normal distribution $N(\mu, \sigma)$.

Step 4: Determine the central vertex π_{cen}^* in the set $\mathcal{P}^*$.

Comment: we assume then in the distance (radius) not greater then 3σ there are 99.97% of local minima (including global minimum). Therefore we limit the search process to this area, using 3σ as a "leash" (we will omit solutions which lie further than 3σ from the center).

Step 5: *Leash algorithm*

(a) The length θ of the leash is the algorithm parameter, $\theta = \delta \cdot \sigma$, where $1 \leq \delta \leq 3$.

(b) We apply a TS algorithm in space searching.

Algorithm 1. Leash algorithm

the center π_{cen}^* is the starting point π;

determine the leash length θ;

repeat

Generate $\mathcal{N}(\pi)$ – neighborhood of π of elements with
thedistance from π not greater than the
leash length θ;

Determine the best element in $\mathcal{N}(\pi)$ and take it as π;

Change the leash length (increase, decrease,
keep the same);

until stop criterion.

The stop criterion can be: maximal iterations number, computations time, determination of a solution with a fixed value, etc. In the searching step other metaheuristic can be applied, e.g. simulated annealing, scatter search or other metaheuristic.

VI. Computational experiments

Numerical experiments were conducted to check the proposed hypothesis that it is profitable to search with limited radius of the searched area. We have used tabu search algorithm NTS taken from the literature [6] and added to it searching area limitation constraints described abowe. Proposed algorithm (lNTS) is based on NTS algorithm and it was implemented in C++ language under CentOS 6.8 operating system. The lNTS algorithm was executed in two phases. In the first phase during running NTS algorithm $maxe$ local minima were gathered. Next, based on local minima set mean values of distances from best/central solutions were calculated. Also values of σ were computed. In the second phase of lNTS algorithm – neighbourhood generation – candidate solution was picked if the distance to center of the graph (or the best solution) was in the interval $[m - \delta\sigma, m + \delta\sigma]$.

Computational experiments were performed on BEM cluster located in Wrocław Centre for Networking and Supercomputing. The calculations were performed on Taillard [18] benchmarks for the job shop scheduling problem.

At the first stage of computational experiments lNTS algorithm was tested for $maxe = 250, 300, 350$ and for central and best solution. Tables I-III contain computational best results for lNTS algorithm for different values of δ and $maxe$ (number of local minima used for determining δ). The results of computational experiment show that best values of average Percentage Relative Deviation (PRD) to the best known solutions was obtained for $maxe = 300$, $\delta = 2$ and central solution as a fixed point for keeping the centrum of the searching area

TABLE I. PRD TO THE BEST KNOWN SOLUTIONS UPPER BOUND FOR $maxe = 350$ AND CENTRAL SOLUTION.

problem	$n \times m$	NTS	lNTS $\delta = 1$	lNTS $\delta = 2$	lNTS $\delta = 3$
TA01-10	15×15	0.5344	0.8997	0.4871	0.5828
TA11-20	20×15	1.0371	1.2127	1.1346	1.1121
TA21-30	20×20	1.0101	1.0415	0.7710	0.9163
TA31-40	30×15	0.9911	0.9529	1.0031	0.9911
TA41-50	30×20	1.7772	1.5248	1.6721	1.7772
TA51-60	50×15	0.0915	0.0915	0.0915	0.0915
TA61-70	50×20	0.1479	0.1368	0.1479	0.1479
TA71-80	100×20	0.0089	0.0089	0.0089	0.0089
average		0.6998	0.7336	0.6645	0.7035

TABLE II. PRD TO THE BEST KNOWN SOLUTIONS UPPER BOUND FOR $maxe = 300$ AND BEST SOLUTION.

problem	$n \times m$	NTS	lNTS $\delta = 1$	lNTS $\delta = 2$	lNTS $\delta = 3$
TA01-10	15×15	0.5344	1.0923	0.3648	0.4793
TA11-20	20×15	1.0371	1.5050	1.2401	1.2668
TA21-30	20×20	1.0101	1.3573	0.8505	0.9498
TA31-40	30×15	0.9911	1.0145	0.9880	1.0801
TA41-50	30×20	1.7772	1.6895	1.6316	1.6458
TA51-60	50×15	0.0915	0.1443	0.1019	0.0915
TA61-70	50×20	0.1479	0.2295	0.1788	0.1061
TA71-80	100×20	0.0089	0.0089	0.0089	0.0089
average		0.6998	0.8802	0.6706	0.7035

of the solution space. The best known solutions for the Taillard benchmark instances are placed on the OR-Library web page http://people.brunel.ac.uk/~mastjjb/jeb/info.html (online; accessed 10-June-2017).

VII. CONCLUSION

We propose the new methodology of Big Valley area determination for scheduling problems. Computational experiments were conducted on the literature job shop problem instances from Taillard [18] and tabu search metaheuristic. We use two versions of the algorithm with (lNTS) and without (NTS) limitation constraints which keep solution in the fixed maximal distance from the center (which is determined before). Results of numerical experiments show, that the proposed mechanism improves results of the average percentage relative deviation of the obtained solutions to the reference ones.

ACKNOWLEDGMENT

Computational experiments were performed in the Wrocław Centre for Networking and Supercomputing under Computational Grant No. 96.

REFERENCES

[1] Ahuja R.K., Ergun O., Orlin J.B., Punnen A.P., *A survey of very large-scale neighborhood search techniques*, Discrete Applied Mathematics, 123, 2002, 75–102.

[2] Bacci, S., and Praga, N., *Ultrametricity, frustration and graph colouring problem*, Journal of Physics A: Mathematical and General 22, 1986, 3023-3032.

[3] Boese, K. D., Kahng, A. B., and Muddu, S., *A new adaptive multi-start technique for combinatorial global optimizations*, Operations Research Letters 16(2), 1994, 101-113.

[4] Bożejko W., Wodecki M., *Solving the flow shop problem by parallel tabu search*, IEEE Computer Society, 2002, 189-194.

[5] Bożejko W., Grabowski J., Wodecki M., *Block approach – tabu search algorithm for single machine total weighted tardiness problem*, Computers & Industrial Engineering 50, 1/2, 2006, 1–14.

TABLE III. PRD TO THE BEST KNOWN SOLUTIONS UPPER BOUND FOR $maxe = 300$ AND CENTRAL SOLUTION.

problem	$n \times m$	NTS	lNTS $\delta = 1$	lNTS $\delta = 2$	lNTS $\delta = 3$
TA01-10	15×15	0.5344	0.6736	0.3662	0.5355
TA11-20	20×15	1.0371	1.1340	1.1454	1.0657
TA21-30	20×20	1.0101	0.9690	0.8529	0.9973
TA31-40	30×15	0.9911	0.7171	1.0652	0.9911
TA41-50	30×20	1.7772	1.9260	1.7074	1.7772
TA51-60	50×15	0.0915	0.0915	0.0915	0.0915
TA61-70	50×20	0.1479	0.2935	0.0747	0.1235
TA71-80	100×20	0.0089	0.0000	0.0089	0.0089
average		0.6998	0.7256	0.6640	0.6989

[6] Bożejko W., Uchroński M., Wodecki M., *Parallel neuro-tabu search algorithm for the job shop scheduling problem*, Proceedings of ICAISC 2013, Lecture Notes in Artificial Intelligence No. 7895, Springer, 2013, 489–499.

[7] Bożejko W., Wodecki M., *On the theoretical properties of swap multimoves*, Operations Research Letters 35/2, 2007, 227–231.

[8] Burke K.E., Kendall G. (eds.), *Search Methodologies. Introductory Tutorials in Optimization and Decision Support Techniques*, Springer, 2005.

[9] Congram, R.K., Potts C.N., Van de Velde S.L., *An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem*, INFORMS Journal on Computing 14, 1, 2002, 52–67.

[10] Demirkol E., Mehta S., Uzsoy R., *Bechmarks for shop scheduling problems*, European Journal of Operational Research, 109, 1998, 137–141.

[11] Diaconis P., *Group Representations in Probability and Statistics*, Lecture Notes Monograph Series vol. 11, Institute of Mathematical Statistics, Harvard University, 1988.

[12] Fisher H., Thompson G.L., *Probabilistic learning combinations of local job-shop scheduling rules*, in: J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice-Hall, Englewood Cliffs, New York, 1963.

[13] Grabowski J., Wodecki M., *A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion*, Computers & Operations Research 31, 2004, 1891–1909.

[14] Nowicki E. and Smutnicki C., *An advanced tabu search algorithm for the job shop problem*, Journal of Scheduling 8, 2, 2005, 145-159.

[15] Nowicki E., Smutnicki C., *A Fast tabu search algorithm for permutation flow shop problem*, European Journal of Operational Research 91, 1996, 160–175.

[16] Nowicki E., Smutnicki C., *A Fast tabu search algorithm for the job shop problem*, Management Science 42, 1996, 797–813.

[17] Reeves, C. *Landscapes operators and heuristic search*, Annals of Operations Research 86, 1990, 473-490.

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

[19] Wolpert, D.H., Macready, W.G., *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation 1, 1997, 67–82.

[20] Wodecki M., *A block approach to earliness-tardiness scheduling problems*, International Journal on Advanced Manufacturing Technology 40, 2009, 797–807.