GENETIC ALGORITHM WITH SELECTIVE LOCAL SEARCH FOR MULTI-OBJECTIVE PERMUTATION FLOW SHOP SCHEDULING PROBLEM

Dominik ŻELAZNY

Abstract: In this paper the flow shop scheduling problem with minimizing two criteria simultaneously is consider. Selected criteria are: makespan and the sum of tardiness of jobs. For each separate criteria the problem is strongly NP-hard, which makes it NP-hard as well. There is a number of heuristic algorithms to solve the flow shop problem with various single objectives, but heuristics for multi-criteria flow shop problems are not so common. In this paper a new idea of the use of local search method for multi-criteria problems is presented.

Key words: flow shop, multi-criteria, genetic algorithms, selective local search.

1. Introduction

Companies wanting to maintain competitive position in fast changing market causes to develop and use new methods of optimization. Due to that competitiveness, developing effective and efficient advanced methods is extremely important. The so-called flow shop scheduling problem (FSSP) represents a class of widely studied cases based on ideas derived from production engineering, which modelled a lot of manufacturing systems, assembly lines, information service facilities [16], and has earned a reputation of being NP-hard to solve [5]. Most of the currently used single objective problems are easily adaptable to real world applications, but modern production scheduling problems need more advanced models.

Since its first formulation, it has received considerable theoretical, computational, and empirical research work. Thus, permutation flow shop scheduling problem is often studied case in the scheduling theory, commonly considered as a practical scheduling problem with still relatively simple mathematical model. Due to its complexity, branch and bound techniques and classical mathematical programming [9] providing exact solutions, are applicable to only small-scale instances. Hence, a lot of various approximate solution methods were proposed, including constructive heuristics, improvement meta-heuristics, and hybrid algorithms. Multi-objective PFSSP is the result of natural evolution of models and solution methods, oriented on practice, since scheduling decisions usually have to take into account several economic indexes simultaneously. Over the last decades, a number of multi-objective evolutionary/genetic algorithms have been suggested. Primarily because of their ability to find multiple Pareto-optimal solutions (an approximation of the Pareto front) in single run. Since, in most cases, it is not possible to have a single solution simultaneously optimizing all objectives, algorithms that give solutions lying on or near the Pareto efficient frontier are of great practical value to different enterprises.

2. Literature on multi-criteria optimization

General multi-criteria optimization received considerable interest in the last twenty years, albeit the multi-criteria discrete (particularly PFSSP) has not been so often studied. Especially in regard to the number of works on PFSSP with single criterion objective function. Most of the multi-criteria PFSSP papers are either based on branch and bound methods or evolutionary algorithms.

2.1. General evolutionary multi-objective algorithms

In [4] Deb and others suggested an Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II), based on the Non-dominated Sorting Genetic Algorithm (NSGA), which was criticised for high computational complexity of non-dominated sorting, lack of elitism and need for specifying the sharing parameter. New version modified its approach to alleviate all of those difficulties, by using fast non-dominated sorting, density estimation and crowded comparison operator allowed it to lessen the computational complexity and guide the selection process of the algorithm towards a uniformly spread out Pareto-optimal front.

Zitzler and Thiele [21] suggested maintaining an external population at every generation storing all discovered non-dominated solutions in theirs Strength Pareto Evolutionary Algorithm (SPEA). It participates in genetic operations. All non-dominated solutions are assigned a fitness based on the number of solutions they dominate, while dominated solutions are assigned a fitness worse than the worst fitness of any non-dominated solution, so that the search is directed towards the non-dominated solutions. Moreover, a clustering technique is used to ensure diversity among non-dominated solutions.

Knowles and Corne proposed Pareto-Archived Evolutionary Strategy (PAES) [7], in which the child is compared with respect to the parent. If the child dominates the parent solution, then the parent is discarded and the child takes its place as the next parent. If the child is dominated by the parent, then the child is discarded and new child solution is generated. On the other hand, if the child and the parent do not dominate each other, the child is compared with the archive to check if it dominates any member of the archive of non-dominated solutions. If so, the child is accepted as the new parent and the dominated solutions are eliminated from the archive. Otherwise, both parent and child are checked for their nearness with the solutions of the archive and the one residing in a least crowded region in the parameter space is accepted as the parent and included in the archive.

In his work, Rudolph [18] suggested, a simple elitist multi-objective evolutionary algorithm based on a systematic comparison of individuals from parent and offspring populations. The non-dominated solutions of both offspring and parent populations are compared, to form new set of non-dominated solutions, which becomes the parent population in the next iteration. If the size of this set is lower than the desired population size, then other solutions from the offspring population are included. Unfortunately this algorithm lacks in the task of maintaining diversity of Pareto-optimal solutions.

2.2. Multi-criteria optimization in flow shop problems

Most of multi-criteria algorithms developed for FSSP use Pareto efficiency evaluation, which is considered as one of the best approaches to the appraisal of solutions. Although there were some attempts at non-Pareto algorithms as well.

2.2.1. Pareto efficiency

Theory of Pareto efficiency states that solution to a multi-objective problem is the set of non-dominated solutions called the Pareto front, where dominance is defined as follows: solution $y = (y_1, y_2, \ldots, y_n)$ dominates a solution $z = (z_1, z_2, \ldots, z_n)$ if and only if for each $i \in \{1 \ldots n\}$, where $y_i \leq z_i$ and exists $i \in \{1 \ldots n\}$, where $y_i < z_i$.

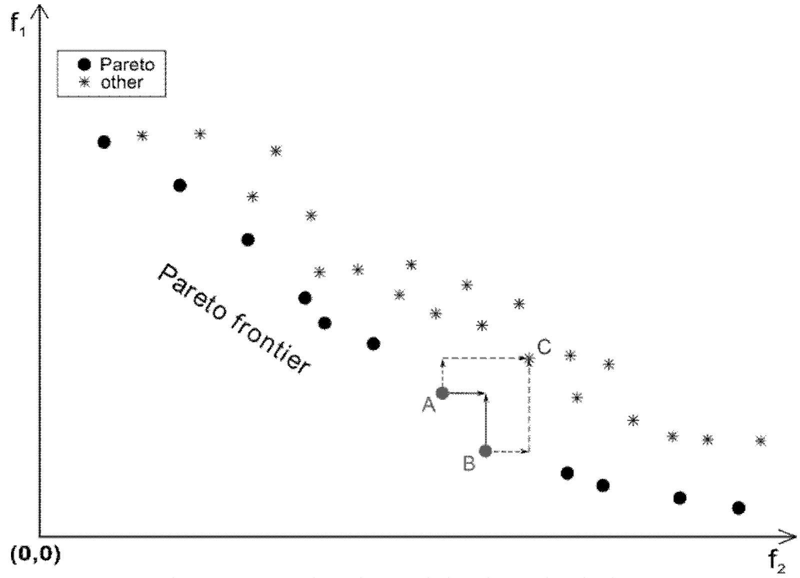

Fig. 1. Pareto frontier and dominated solutions

Fig. 1 shows an example of dominance of points A and B over point C and lack of such between aforementioned points A and B. Both of which are non-dominated and as such included in the approximation of the Pareto front.

2.2.2. Literature on multi-objective PFSP algorithms

The Multi-Objective Genetic Algorithm (MOGA) of Murata et al. [11], being part of evolutionary algorithms, was developed to solve multi-objective flow shop problem. Other than a modified selection operator, this algorithm was a simple genetic approach to scheduling. Selection is interrelated with a set of weights assigned to the objectives, which allowed to distribute the search towards different criteria directions. Elitist preservation method was also incorporated, so that solutions from the actual Pareto frontier were copied to the next generation. The MOGA was furthermore enhanced by Murata et al. [12], by changing the way of weight distribution between objectives. Using a cellular structure permitted a better weight selection, which in turn led to finding a finer approximation of Pareto front. New algorithm was called CMOGA.

Armentano and Arroyo [1] described a tabu search approach for makespan and maximum tardiness objectives. The proposed method was compared with the genetic algorithm of Ishibuchi and Murata for the m-machines case. The proposed meta-heuristic was shown to outperform existing methods.

Khan and Govindan [6] presented a multi-objective simulated annealing algorithm for the m-machine permutation flow shop scheduling problems with the objectives of minimizing makespan and maximum tardiness of jobs. To prove the efficiency of the proposed algorithm, different test problems were solved and compared to other genetic algorithms and to the evolutionary algorithm. According to computational experiments, the proposed algorithm was more effective than others.

There are some implementations of the particle swarm optimization algorithm for multi-criteria FSSP. Chandrasekaran et al. [2] presented a multi-objective particle swarm optimization algorithm with aim to minimize makespan, total flowtime and completion time variance. The authors did not compare this algorithm against any others. They only identified a Pareto solution set for the flow shop instances in the literature. The proposed particle swarm optimization was shown to yield more non-dominated solutions.

Local search method was proposed in the work of Chakravarthy and Rajendran [3]. Their goal was to minimize the weighted sum of two objectives using a simple SA algorithm. Initial solution is selected from the following methods: a) Earliest Due Date (EDD), Least Static Slack (LSS) and NEH heuristic [13], while generating a neighbourhood was performed by the adjacent interchange scheme (AIS). Since it uses weighted objectives, this algorithm does not belong to the set of Pareto approach algorithms.

Inspired by the PAES algorithm, Suresh and Mohanasundaram proposed a Pareto Archived Simulated Annealing (PASA) [19], in which new perturbation was suggested. Mechanism called Segment Random Insertion was used to generate the neighbourhood of a given sequence. In order to retain non-dominated solutions, an external archive is used. Initial solution is randomly generated, while new current solution is selected by a scaled weighted sum of the objective values.

A hybrid multi-objective algorithm based on a shuffled frog-leaping algorithm and a variable neighbourhood search was studied Rahimi-Vahed et al. [17] It was set with the objectives of minimizing weighted mean earliness and weighted mean tardiness. The computational results showed that the proposed algorithm outperformed two multi-objective genetic algorithms and was able to improve the quality of the obtained solutions, especially for large-sized problems.

Variation of genetic algorithm, with an initialization procedure which inserts four good solutions into initial random population, was proposed by Pasupathy et al. in [14]. It used an external population, for non-dominated solutions. Evolution strategy is similar to the one used in NSGA-II, while crowding distance procedure is used as a secondary population selector. Improving the quality of Pareto frontier is based on two different local search procedures, applied to the external population afterwards.

3. Problem description

Let us consider a manufacturing system with a structure consisting of m machine given by the set $M=\{1,\ldots,m$. Consider a set $J=\{1,\ldots,n$ of jobs to be processed on machines. Each job has to be processed on a machine $1, 2,...,m$ in that order. Job j, $j \in J$, consists of a sequence of m operations $O_{j,1}$, $O_{j,2}$,..., $O_{j,m}$. Operation $O_{j,k}$ corresponds to the processing of job j on a machine k during an uninterrupted processing time $p_{j,k} \geq 0$. In the permutation flow shop problem the job sequence must be the same on all machines. Each machine can execute at most one job at a time and each job can be processed on at most one machine at a time. Each job $j \in J$ should be delivered before its due date $d_j \geq 0$.

The schedule of jobs (solution of the problem) can be described by starting $S_{j,k}$ and completion times $C_{j,k}$ of operations $j \in J$, $k \in M$, satisfying the above mentioned constraints. The operation $O_{j,k}$ starts in $S_{j,k}$ and completes in $C_{j,k}$. The job j, $j \in J$, is delivered by the production system in the time moment $C_{j,m}$. Two objective functions are considered: the total completion time C_{max} and the total tardiness T_{tot}. For a given schedule described by $C_{j,k}$ the objective values can be calculated with the following expressions:

$$C_{\max} = \max_{j \in J} C_{j,m} \tag{1}$$

and

$$T_{tot} = \sum_{j \in J} T_j, \tag{2}$$

where $T_j = \max(0, C_{j,m} - d_j)$ is a tardiness of job $j \in J$.

In the paper there is a reference to another (equivalent) characterization of the solution which uses loading sequence instead of the schedule. Let a permutation π of n jobs determine the processing order on all machines. The completion times can be calculated with following recursive expression:

$$C_{\pi(j),k} = \max(C_{\pi(j-1),k}, C_{\pi(j),k-1}) + p_{\pi(j),k}, \tag{3}$$

where $\pi(0)=0$, $C_{j,0}=0$ for $j=1,\ldots,n$ and $C_{0,k}=0$ for $k=1,\ldots,m$.

The completion times obtained from (3) are as small as possible. Finally, due to regularity of both objectives, for a given processing order described by π the minimal value of objectives can be calculated with the following expressions:

$$C_{\max}(\pi) = \max_{1 \le s \le n} \{C_{\pi(s)m}\} \tag{4}$$

and

$$T_{tot}(\pi) = \sum_{s=1}^{n} T_{\pi(s)}, \tag{5}$$

where $T_{\pi(s)} = \max(0, C_{\pi(s)} - d_{\pi(s)})$ is a tardiness of job $\pi_{(s)} \in J$.

4. Genetic algorithm with selective local search method

Research and general studies show that good solutions can be found near other good solution for the flow shop scheduling problem. Those features were used to construct Genetic Algorithm with Selective Local Search (GASLS), being a development of previously proposed Genetic Algorithm with Local Search [22].

In previous work in every iteration of the algorithm, for each offspring, given number of iteration of stochastic Local Search (LS) method is performed to enhance the offspring. There is searched neighbourhood generated by adjacent interchange moves. New solution is generated by random swapping two adjacent jobs in current permutation. If it dominates the old solution, then it replaces the parent solution in next iteration. In other case it gets discarded. Although the computational complexity increased. In GASLS, after each iteration of the GA algorithm, solutions are divided into two groups: a) dominated and b) non-dominated (Pareto efficient). Subsequently, unique solutions from the second group are selected to be subjected to LS method. It allowed to decrease computational time of LS method and also, without increasing computational time of whole algorithm, allowed more thorough search the neighbourhoods of unique non-dominated solutions.

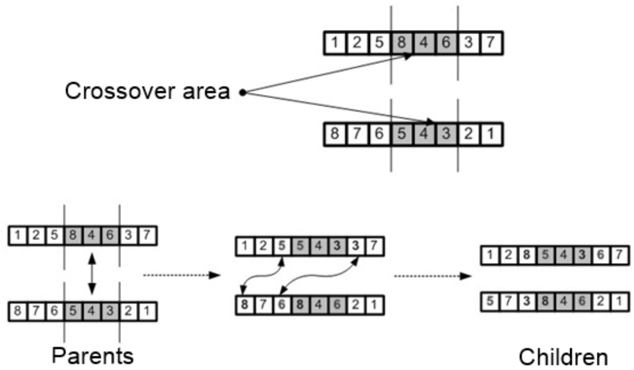

Fig. 2. Sample of PMX Crossover

In GASLS implementation the individuals in population are represented by jobs permutation and values of their criteria functions. The GASLS algorithm uses partially matched crossover (PMX) crossover (see Fig. 2) and tournament selection, while fitness value is based on non-domination level and distance from direct neighbours. Initial population includes solutions obtained from NEH algorithm adapted to solve PFSSP problem with each of criteria from objective function. Such initialization allows to faster designate approximation of Pareto front. Moreover, non-dominated solutions are archived in external population.

5. Computational evaluation

The aim of the experiment was to compare the effectiveness of a proposed algorithm with the benchmarks taken from literature. The algorithms were tested on 100 benchmark instances provided by Taillard [20] for the flow-shop problem and modified by Minella [10] for total tardiness criterion. The benchmark set contains 10 groups of 'hard' instances of different sizes.

The implementations were compiled and executed on the Dell Inspiron 7720 SE with Intel i7-3610QM 2.30 GHz processor. The computation times of GASLS were comparable to those of former GA with LS method.

Comparison of multi-criteria algorithms is not as easy as comparing algorithms with single criterion objective function, where lower/higher (minimization/maximization) value of solution translates directly to a better solution. Considering different methods (scalar, Pareto and others) of solution evaluation, there is no single way to evaluate which set of non-dominated solutions is clearly better than the other. Selected approaches are based on a dominance relation, so a returned result for each instance is a set of non-dominated solutions, called approximation of Pareto front.

5.1. Quality indicators and reference data

In this paper three ways of comparing results were used. Both algorithms were tested with the same computation time limit, and furthermore compared using following methods.

5.1.1. Number of Pareto efficient solutions

In earlier work [15] certain method of comparison was devised, which used a percentage of non-dominated solutions in the aggregation of compared sets as an indicator. Solutions from all the algorithms were flagged and aggregated into a single set, which was then purged of dominated solutions. A number of solutions in this global Pareto-efficient set was computed for each algorithm and those numbers were compared to evaluate solution sets.

5.1.2. Hyper-volume Indicator

Knowles et al. [8] provided a few necessary tools for a better evaluation and comparison of multi-objective algorithms. They proposed, among others, a hyper-volume indicator I_H to measure quality of the Pareto frontier approximations. Hyper-volume indicator measures the area covered by the approximated Pareto fronts for each of the algorithms. In order to bound this area, a reference point is used. A greater value of I_H indicates both a better convergence to as well as a good coverage of the optimal Pareto front, see Fig. 3.

In this case, reference points were calculated as follows. For each of the criteria used worst value from both (LS and SLS) non-dominated sets were taken, multiplied it by 1.2 and assigned as reference points value of that criteria.

5.1.3. Subspace coverage

Using I_H values, coverage of GA with LS of the GASLS method was computed. The lower the value of such, the better results obtained by GASLS algorithm in comparison to GA with LS algorithm.

$$Coverage = \frac{I_H(GAwLS)}{I_H(GASLS)} \tag{6}$$

6. Experimental evaluation

Solution sets obtained from both algorithms were compared using previously mentioned indicators and presented in the summarized groups of instances.

As can be seen in Tab. 1, proposed GASLS algorithm outperforms classic GA with stochastic LS method in number of found Pareto solutions. Moreover, in above mentioned table only solutions non-dominated in both sets were counted. Proposed GASLS, with similar computational time, found over three times more solutions.

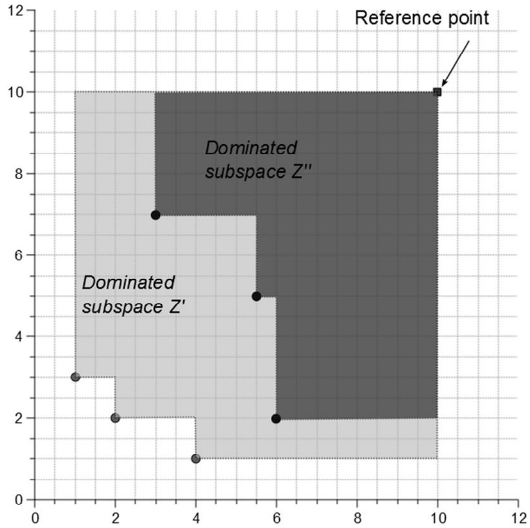

Fig. 3. Visualization of Hyper-volume indicator

Tab. 1. Number of Pareto solutions

Instance Size	GA with LS	GASLS
20x5	58	236
20x10	128	395
20x20	114	328
50x5	80	181
50z10	100	415
50x20	135	752
100x5	68	228
100x10	145	372
100x20	145	514
Instance Size	GA with LS	GASLS
200x10	137	392
Sum	1110	3813

Previously proposed algorithm covered, in average, only around 72% of newly found non-dominated subspace. Newly proposed GASLS mostly outperformed previous algorithm in cases with 10 and more machines, while with only 5 machines they obtained more similar results. It could be caused by lower number of criteria function computations in case with fewer machines.

Tab. 2. Coverage of GA with LS over GASLS

Instance Size	GA with LS	GASLS	Coverage
20x5	0,08	0,12	68,1%
20x10	0,20	0,24	82,2%
20x20	0,19	0,22	84,3%
50x5	0,05	0,06	82,5%
50z10	0,05	0,09	59,7%
50x20	0,09	0,16	59,2%
100x5	0,04	0,05	85,2%
100x10	0,04	0,06	70,0%
100x20	0,04	0,07	58,9%
200x10	0,03	0,05	67,3%
Average	0,08	0,11	71,7%

In all cases, proposed GASLS outperforms previous GA with LS considerably and shows it potential in instance cases of higher computational time of criteria objectives. Furthermore, it allows for more accurate solution space search.

7. Conclusions and future work

Proposed GASLS allows better approximation of Pareto frontier in approximate computational time. It is the result of lowering number of solutions undergoing local search and increasing the spread of said search, which confirms earlier assumption that good (or non-dominated) solutions are placed in the neighbourhood of previously found good solutions. Future work with multi-objective algorithms, including genetic algorithms, assumes the use of Multi-Criteria Decision Making, e.g. Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) or Light Beam Search (LBS).

Reference

1. Armentano V.A., Arroyo J.C.: An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem. Journal of Heuristics, 10, 2004, 463-481.
2. Chandrasekaran S, Ponnambalam SG, Suresh RK, Vijayakumar N.: Multiobjective particle swarm optimization algorithm for scheduling inflowshops to minimize makespan, total flowtime and completion time variance. IEEE Congress on Evolutionary Computation (CEC 2007), 4012-4018.
3. Charavarthy K., Rajendran, C.: A heuristic for scheduling in a flowshop with the bicriteria of makespan and maximum tardiness minimization. Production Planning and Control, 10 (7), 1999, 707-714.
4. Deb K., Pratap A., Agarwal S., Meyarivan T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation, 6 (2), 2002, 182-197.
5. Garey M., Johnson D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

6. Khan BSH, Govindan K.: A multi-objective simulated annealing algorithm for permutation flow shop scheduling problem. International Journal of Advanced Operations Management, 3 (1), 2011, 88-100.
7. Knowles J., Corne D.: The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation. 1999, 98-105.
8. Knowles J., Thiele L., Zitzler E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Tech. rep., ETH Zurich, 2006.
9. Lageweg B., Lenstra J., Kan A.R.: A general bounding scheme for the permutation flowshop problem. Operations Research, 26 (1), 1978, 53-67.
10. Minella G., Ruiz R., Ciavotta M.: A review and evaluation of multi-objective algorithms for the flow shop scheduling problem. Journal on Computing, 20 (3), 2008, 451-471.
11. Murata T., Ishibuchi H., Tanaka H.: Multi-objective genetic algorithm and its applications to flowshop scheduling. Computers and Industrial Engineering, 30 (4), 1996, 957-968.
12. Murata T., Ishibuchi H., Gen M.: Specification of genetic search directions in cellular multiobjective genetic algorithms. Springer, 2001, 82-95.
13. Nawaz M., Enscore E., Ham I.: A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. The International Journal of Management Science, 11 (1), 1983, 91-95.
14. Pasupathy T., Rajendran C., Suresh R.: A multi-objective genetic algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. The International Journal of Advanced Manufacturing Technology, 27 (7-8), 2006, 804-815.
15. Pempera J., Smutnicki C., Żelazny D.: Multi-objective optimization of production schedules. Proc. of the Production Engineering, 2011.
16. Pinedo M.: Scheduling: Theory, Algorithms and Systems. 2nd ed, Prentice-Hall, 2002.
17. Rahimi-Vahed AR, Dangchi M, Rafiei H, Salimi E.: A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem. International Journal of Advanced Manufacturing Technology, 41, 2009, 1227-1239.
18. Rudolph G.: Evolutionary search under partially ordered fitness sets. Tech. rep., 2001.
19. Suresh R., Mohanasundaram K.: Pareto archived simulated annealing for permutation flowshop scheduling with multiple objectives. 2004, 712-717.
20. Taillard E.: Benchmarks for basic scheduling problems. European Journal of Operational Research, 64, 1993, 278-285.
21. Zitzler E., Thiele L.: Multiobjective optimization using evolutionary algorithms - a comparative case study and the strength pareto approach. Evolutionary Computation, IEEE Transactions, 3 (4),1999, 257-271.
22. Żelazny D.: Optymalizacja wielokryterialna w problemie marszrutyzacji (VRP). [w:] Automatyzacja procesów dyskretnych: teoria i zastosowania. T.1, pod red. Andrzej Świerniak i Jolanta Krystek, Gliwice, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 157-163.

Mgr inż. Dominik ŻELAZNY
Instytut Informatyki Automatyki i Robotyki, Politechnika Wrocławska
50-370 Wrocław, ul. Janiszewskiego 11/17
tel.: (71) 320 27 45 / 321 26 77
e-mail: dominik.zelazny@pwr.wroc.pl