

*Electronic version of an article published as [Computers & Operations Research, 2006, vol. 33, No. 4, p. 928-939] [DOI: <http://dx.doi.org/10.1016/j.cor.2004.08.004>]
© [copyright Elsevier]*

Solving the minmax product rate variation problem (PRVP) as a bottleneck assignment problem^{*}

Natalia Moreno^a and Albert Corominas^b

^aDepartment of Business, Faculty of Business Administration, Universitat Politècnica de Catalunya, MUN, St. John's, Spain

^bInstitute of Control and Industrial Engineering, Universitat Politècnica de Catalunya, Spain

Abstract

In this paper, we consider the *minmax* product rate variation problem (PRVP), which consists in sequencing copies of different products on an assembly line in such a way that the maximum value of a discrepancy function between actual and ideal productions is minimum. One means of solving this problem lies in its reduction to a bottleneck assignment problem with a matrix of a special structure. To solve it, three different approaches have been adopted. These approaches exploit specific *minmax* PRVP matrix properties. This paper presents a computational experiment with symmetric and asymmetric objective functions and offers conclusions about the most efficient way to find optimal solutions.

Keywords: Product rate variation problem; Bottleneck assignment problem

^{*}This research has been supported by the Natural Sciences and Engineering Research Council of Canada Grant OPG0105675. The support of NSERC is gratefully acknowledged.

1. Introduction

The product rate variation problem (PRVP) is an important production problem that arises on mixed-model, just-in-time, assembly lines (a particular case of the problem formulated by Monden [1]). This problem consists in sequencing copies of different products in such a way that the actual productions are as close as possible to their ideal productions (Kubiak [2]).

For the ease of explanation, next we will introduce the notation used in this paper. Usually, the problem is considered under the following assumptions: there are V products to be produced on an assembly line that needs negligible time to switch production from one product to another. Each copy, regardless of its corresponding product, requires the same production time (the cycle time of the line, which can be assumed as the unit of time without loss of generality). There are U units to be produced, of which u_i correspond to product i ($i=1, \dots, V$) with $\sum_{i=1}^V u_i = U$. The time horizon is

of U time units, where one copy of product i , $i=1, \dots, V$ will be produced in each time period. The ideal production (i.e. the production that would correspond to a continuous regular production over the time horizon) of each product per time unit can be calculated as $r_i h$ ($h=1, \dots, U; i=1, \dots, V$), where $r_i = u_i / U$ ($i=1, \dots, V$). The sequence can be described by means of x_{ih} variables (the total production of product i in time periods from 1 to h inclusive, $h=1, \dots, U$). In order to evaluate the deviation between actual x_{ih} and ideal productions, discrepancy functions $f_i(x_{ih} - r_i h)$ (a particular case of $f_i(x_{ih}, h)$ functions) can be used.

Miltenburg [3] has formulated the PRVP as a nonlinear integer-programming problem with the goal of minimizing the total deviation in the production rates on a mixed-model assembly line. The objective function has been defined as a sum of discrepancy functions (such a problem has been called the *minsum* PRVP). Miltenburg has also proposed different discrepancy functions for this problem.

Kubiak [2] has introduced the term PRVP. Kubiak and Sethi [4] and [5] have explored the *minsum* PRVP and demonstrated that this problem can be reduced to an assignment problem (AP) when the functions that compose the objective function are nonnegative and convex.

Steiner and Yeomans [6] have dealt with the PRVP, introducing a new non-convex objective function ($\max_{1 \leq i \leq V, 1 \leq h \leq U} |x_{ih} - r_i h|$) to be minimized, which is the maximum deviation between actual production and ideal production (such a problem has been named the *minmax* PRVP). The main objective to be achieved solving the *minmax* PRVP is to obtain a schedule that enables us to avoid too large undesirable deviations at every time period of a total production time horizon. Steiner and Yeomans have shown that a schedule always exists where the deviation of actual production from the ideal for all products is never greater than one copy. These authors have developed an optimization procedure for the *minmax* PRVP with symmetric discrepancy functions, which is a modified version of Glover's [7] algorithm for a maximum matching in a convex bipartite graph.

One means of solution lies in the reduction of the *minmax* PRVP to a bottleneck assignment problem (BAP). Such an approach is independent of discrepancy functions and allows us to use either symmetric or asymmetric discrepancy functions. The idea of reducing the *minmax* PRVP to a BAP was first mentioned by Kubiak [2], although it has not been further developed since its formulation. Bautista et al. [8] were first to demonstrate such a reduction.

Still [9] has shown that at least one sequence for the apportionment problem always exists in which $|x_{ih} - r_i h| < 1 \quad \forall i, h$, that is $T_{\text{opt}} < 1$. This bound is a valid upper bound for the *minmax* PRVP, which has been shown by Bautista et al. [10], who have established a connection between the PRVP and the apportionment problem (Balinski and Young [11]). Recently, Brauner and Crama [12] have again taken up the *minmax* PRVP establishing some new bounds for the optimal target value. One of these bounds is the upper bound $T_{\text{opt}} \leq 1 - U^{-1}$. Tijdeman [13] has demonstrated another upper bound for T , namely, $T_{\text{opt}} \leq 1 - (2(V-1))^{-1}$, which results in a lower value for $V < (U+2)/2$. All of these bounds can significantly reduce the size of a BAP.

Until now, there was no computational experiment reported in the literature on the *minmax* PRVP solution as a BAP. The only published computational results of the *minmax* PRVP solution by means of solving a matching problem in a bipartite convex graph were offered by Kovalyov et al. [14], who have restricted their computational experiment to $U = 200$ units. In their paper, the authors have formulated, as a conjecture, that “the value of the optimal *minmax* PRVP objective will always be at the problem's theoretical lower bound, $1 - r_{\text{max}}$ ”. These authors have refuted this conjecture with a counter-example. Our computational experiments have also shown that this conjecture is not true.

In industrial practice, it is very usual to determine schedules that could correspond to thousands of copies of different products to be sequenced. For example, in the automobile industry the total production for 2 weeks may achieve 10,000 units. Furthermore, the possible number of different products may be extended to hundreds or even thousands. Therefore, the problem with both a large number of total copies to be produced (up to 10,000) and a large number of products (up to 5000) is of special interest. The purpose of our work was to design and computationally test different approaches in order to obtain an efficient one, which allows solving large *minmax* PRVP instances rapidly. To our knowledge, there was not any computational experiment with large instances reported in the literature.

The present paper deals with the *minmax* PRVP regarded as a BAP. Specific procedures have been designed to make good use of the properties of the problem considered as a BAP. Firstly, we have taken into account all already known and available properties of the *minmax* PRVP. Secondly, we have identified new techniques to solve the *minmax* PRVP. Thirdly, we have carried out computational experiments for large instances and symmetric and asymmetric discrepancy functions.

Section 2 presents the mathematical model; Section 3 introduces three approaches to solve the *minmax* PRVP as a BAP with specific properties; Section 4 provides computational results; and in Section 5, some conclusions are drawn.

2. Mathematical model

In this section, the mathematical model of the *minmax* PRVP is formalized. Let $f_i(x_{ih}, h) = f_i(x_{ih} - r_i \cdot h)$ ($i=1, \dots, V$) be unimodal, convex and nonnegative discrepancy functions with minimum in $f_i(0) = 0$ ($i=1, \dots, V$). Then, the *minmax* PRVP problem can be formulated as follows:

$$\begin{aligned}
\text{Minimize } T &= \max_{1 \leq i \leq V, 1 \leq h \leq U} f_i(x_{ih}, h), \\
\text{s.t. } \sum_{i=1}^V x_{ih} &= h \quad h = 1, \dots, U, \\
x_{i,h-1} &\leq x_{ih} \quad i = 1, \dots, V; h = 1, \dots, U, \\
x_{i0} &= 0, \quad x_{iU} = u_i \quad i = 1, \dots, V, \\
x_{ih} &\geq 0 \text{ and integer } i = 1, \dots, V; h = 1, \dots, U.
\end{aligned}$$

Bautista et al. [8] have demonstrated that the *minmax* PRVP is equivalent to a BAP with matrix element values that can be computed in the following way:

$\hat{\phi}_{ik}(t) = \max[f_i(k-1, t-1), f_i(k, t)]$ ($t = 1, \dots, U; i = 1, \dots, V; k = 1, \dots, u_i$), where $\hat{\phi}_{ik}(t)$ corresponds to the production of a copy k of product i at instant t .

For solving this problem with symmetric discrepancy functions, it suffices to use the functions $f_i(k, t) = |k - r_i \cdot t| = |k \cdot U - u_i \cdot t| / U$, since for any other symmetric, non-decreasing function of the absolute discrepancy the order of the matrix element values will remain the same. Therefore, despite the original matrix element values being rational, we can work with integer values, $U \cdot f_i(k, t)$. This enables us to achieve more rapid performance of algorithms. The numerical example with the assignment matrix for one *minmax* PRVP instance ($U=10, V=3, u_1=2, u_2=3, u_3=5$) is presented in Fig. 1.

t	1	2	3	4	5	6	7	8	9	10
$\hat{\phi}_{1,1}(t)$	8	6	4	6	8	10	12	14	16	18
$\hat{\phi}_{1,2}(t)$	18	16	14	12	10	8	6	4	6	8
$\hat{\phi}_{2,1}(t)$	7	4	6	9	12	15	18	21	24	27
$\hat{\phi}_{2,2}(t)$	17	14	11	8	5	5	8	11	14	17
$\hat{\phi}_{2,3}(t)$	27	24	21	18	15	12	9	6	4	7
$\hat{\phi}_{3,1}(t)$	5	5	10	15	20	25	30	35	40	45
$\hat{\phi}_{3,2}(t)$	15	10	5	5	10	15	20	25	30	35
$\hat{\phi}_{3,3}(t)$	25	20	15	10	5	5	10	15	20	25
$\hat{\phi}_{3,4}(t)$	35	30	25	20	15	10	5	5	10	15
$\hat{\phi}_{3,5}(t)$	45	40	35	30	25	20	15	10	5	5

Fig. 1. Example of the assignment matrix for the *minmax* PRVP.

In order to solve the problem with asymmetric discrepancy functions, we have used the functions $f_i(k, t) = \alpha_1 \cdot \max(0, k - r_i \cdot t) + \alpha_2 \cdot \max(0, r_i \cdot t - k)$ with $\alpha_1=1$ and $\alpha_2=3$, in our experiment, to penalize overproduction and underproduction, respectively.

3. Algorithms for solving the minmax PRVP as a BAP

In this section we briefly review our main approaches for solving the *minmax* PRVP as a BAP. The *minmax* PRVP considered as a BAP has some specific properties presented below:

1. There are good linear time heuristic algorithms available for the *minmax* PRVP.
2. The BAP matrix values $\hat{\varphi}_{ik}(t)$ grow to the right and to the left from the ideal positions, which for symmetric discrepancy functions can be computed in the following way: $t_{ik} = \lceil (2 \cdot k - 1) / 2 \cdot r_i \rceil$, $i=1, \dots, V; k=1, \dots, u_i$ (Kubiak [2]).
3. The values $\hat{\varphi}_{ik}(t)$ are easy to calculate, and it is possible to avoid computing a complete matrix by only computing elements for which $|x_{ih} - r_i \cdot h| < 1$ (quota band) holds since there are always optimal solutions within this quota band for symmetric absolute value discrepancy functions.

In this paper, three main approaches have been adapted for the *minmax* PRVP solution as a BAP. They are:

Approach 1: Solving the problem by means of specific BAP algorithms. Two codes for solving a BAP have been used in our study, BASS (Carpaneto and Toth [15]) and BOTJV (Jonker and Volgenant [16]).

Approach 2: Solving the problem as a sequence of AP. We have used the method of Woolsey and Swanson [17], which consists in the solution of a sequence of assignment problems (we have used codes NAUC, APC, LAPJV and LAPJVsp, described, for example, in the study on AP by Dell'Amico and Toth [18]). We have introduced two specific modifications to this approach. They are: (a) the use of a binary matrix instead of the original bottleneck assignment matrix (for a detailed description see Moreno [19]), and (b) an application of bisection search to find the optimal bottleneck value.

Approach 3: Solving the problem as a sequence of matching problems (MP). This approach has been designed using the idea proposed by Steiner and Yeomans [6], who considered a matching problem on a bipartite convex graph. Our approach uses a bipartite maximum cardinality matching (BMCM) algorithm. We have introduced in this approach a new element that consists of starting with a heuristic PRVP solution of good quality. If the heuristic solution is close to the optimal solution the search interval for T can be reduced significantly. This allows us to obtain the optimal solution by solving a sequence of matching problems in a less number of iterations. In our approach, we have used Webster's method as a heuristic algorithm for the *minmax* PRVP. Initially, Webster's method had been developed for solving the apportionment problem (Balinski and Young [11]). A connection between the PRVP and the apportionment problem established by Bautista et al. [10] allows us to use some specific apportionment procedures as heuristic algorithms for the PRVP. We have implemented the third approach with a bisection search (proposed by Steiner and Yeomans [6]) and without application of a bisection search. The main steps of the latter are as follows: (1) obtain a heuristic solution of the *minmax* PRVP; (2) build a bipartite graph whose edges correspond to quota band; (3) examine heuristic solution; if the maximum value of all the edges that enter into the solution is equal to lower bound $1 - r_{\max}$, end the algorithm; if there are edges with associated values greater than upper bound $= \min\{1 - U^{-1}; 1 - 2 \cdot (V - 1)^{-1}\}$ remove them from the solution and the bipartite graph and then apply the BMCM algorithm with a reduced initial

matching to the reduced bipartite graph, otherwise go to (4) next; (4) find the largest value of the given feasible solution; remove in the bipartite graph all edges whose correspondent values are greater or equal to it; apply the BMCM algorithm to the reduced graph with a reduced initial matching; continue this process until the matching of cardinality U cannot be obtained, which means the optimal solution has been reached in the previous iteration.

We have named all the above approaches and their modifications for the ease of making reference to them. Fig. 2 maps all of our approaches.

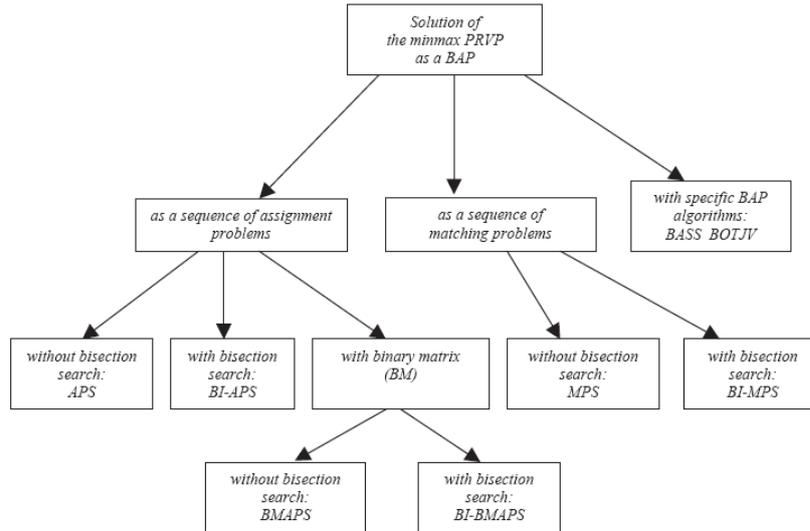


Fig. 2. Approaches for solving the *minmax* PRVP as a BAP.

4. Computational experiment

This section presents comparative computational results of the application of designed approaches to some instances of the PRVP that are generated by fixing the total number of units to be produced and the product number and randomly selecting the number of copies of each product, which is uniformly distributed. All available codes from the literature and codes that we have additionally implemented for the first computational experiment are in FORTRAN. The first computational experiment has been carried out with algorithms already available and newly designed in this research in order to draw conclusions about their efficiency. The algorithms have been executed on a SUN 450 Ultra SPARC2 with a processor of 250 Mhz and 512 Mb of RAM. Later, in order to confirm the results on the most efficient approaches selected after the first computational experiment, a new one has been carried out. The algorithms corresponding to these approaches have been coded in C++. All of the computations have been performed with a processor of 750 Mhz and 512 Mb of RAM.

The first computational results have shown that the approach that requires the use of a binary matrix is non-competitive with respect to the same method with the original BAP matrix. Therefore, we have omitted its comparison with other approaches. More than 30 instances with U up to 200 have been involved in this part of the experiment. The comparison of APS and BMAPS approaches for one instance with input data $U=200$, $V=17$, $u=(9,11,13,12,12,13,5,14,4,19,16,11,13,12,14,11,11)$ is illustrated in Table 1.

Table 1.

Comparison of computing times of APS and BMAPS approaches

Applied approach	Total CPU time (s) (with matrix)	Total CPU time (s) (with matrix)
	LAPJV (for dense matrix)	LAPJVsp (for sparse matrix)
BMAPS	0.90 (20 ^a)	0.10 (20 ^a)
APS	0.19 (6 ^a)	0.08 (6 ^a)

^a Number of times the assignment problem has been solved.

Also, based on the results of the initial experiment, we do not present the APS method with AP algorithms for dense matrices (NAUC, APC, LAPJV) since this method is more time consuming, than, for example, the APS with AP algorithm for sparse matrices (LAPJVsp) and specific BAP algorithms (BOTJV and BASS), even for small instances, as can be seen in Table 2 for the instance from Table 1.

Table 2.

APS approach with AP algorithms and specific BAP algorithms

Applied approach	Total CPU time (s) (with matrix)	Number of times the assignment problem has been solved
LAPJV	0.19	
APC	0.46	6
NAUC	1.81	
LAPJVsp	0.08	

Applied approach	Total CPU time (s) (with matrix)	Number of times the assignment problem has been solved
BASS (specific for BAP)	0.02	1
BOTJV (specific for BAP)		

After this last elimination step, we have selected four approaches (APS, BI-APS, both with LAPJVsp, MPS and BI-MPS) designed for the *minmax* PRVP solution and two specific BAP algorithms. In this part of the experiment about 100 instances with different (V,U) values have been explored. It is to be noted that the time required for any matrix computation has not been taken into account, as it is a constant for a fixed problem dimension.

To demonstrate how these methods perform, CPU times (s) (without including matrix computing time) for one instance with input data $U=5000$, $V=20$, $u=(218,68,999,146,123,26,11,167,333,388,327,91,111,437,14,274,555,301,124,287)$ are presented in Table 3.

Table 3.

Comparison of computing times

APS approach	BI-APS approach	MPS approach	BI-MPS approach	BOTJV	BASS	Bottleneck value T:
47.72 (23 ^a)	12.40 (5 ^a)	8.42 (150 ^a)	3.42 (10 ^a)	35.33	18.79	0.8002

^a Number of times assignment (APS, BI-APS) or matching (MPS, BI-MPS) problem have been solved.

After the comparison of the selected approaches for different instances, the APS approach has been excluded from further computations since it has shown the worst computational time results.

Next, we have run BI-APS, MPS, BI-MPS, BASS and BOTJV algorithms with other 100 instances, randomly generated for the fixed product number $V=50$ and the total copy number $U=2000$. Table 4 contains maximum, minimum and average CPU times for each tried approach.

Table 4.

Comparison of maximum, minimum and average CPU times

CPU time (s)	BI-APS approach	MPS approach	BI-MPS approach	BOTJV	BASS
Maximum	17.93	15.85	14.84	12.11	719.11
Minimum	3.43	0.38	0.38	1.96	2.81
Average	8.83	3.08	2.55	7.16	84.64

Computing time results of the five chosen approaches for all (100) instances are presented in Fig. 3. With regard to the BASS algorithm we have not included the computational time results for the instances 21–25, 84–86, 89–93, 96–100 because it was too high in comparison with results provided by other approaches.

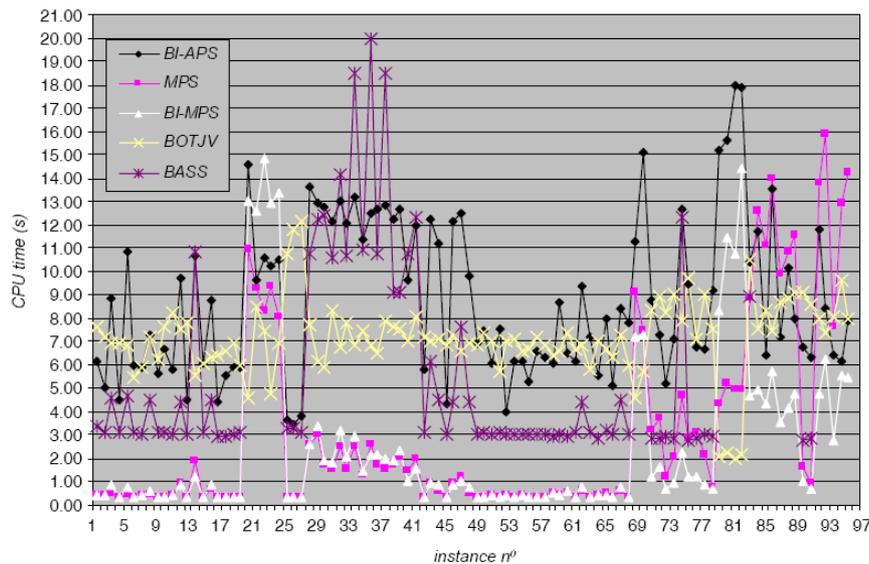


Fig. 3. Time comparison for the set of 100 instances with $U=2000$ and $V=50$.

The method with the specific assignment problem algorithm LAPJVsp and bisection search technique (BI-APS) with a low number of products (as in the example with $U=2000$ and $V=50$) has comparable computing times with those of MPS, BI-MPS, BOTJV and BASS approaches. However, BI-APS computing time increases with the increase of product number, since the number of elements in sparse matrix is larger for the same fixed number U . We have also observed that BASS, an algorithm that is specific for BAP, always performs more quickly than BOTJV, except for cases where input data has some specific structure (where the major part of a total demand belongs to one particular product type i). BOTJV runs much faster for such instances than for

others, while performance time of the BASS algorithm has grown significantly. According to our computational experiments, the most rapid approach is one that uses the BMCM algorithm either with bisection search or without it (BI-MPS or MPS).

Finally, all methods have been tried with asymmetric discrepancy functions. The lower bound, $T_{opt} \geq \alpha_1 \cdot (1 - r_{max})$, on the bottleneck value has been derived following Steiner and Yeomans [6]. The upper bound has been obtained taking the greatest matrix element value from the quota band $[beg_k, end_k]$, $k=1, \dots, U$ (columns in which the band begins and ends for each row k). Thus, $T_{opt} \leq \max_{1 \leq k \leq U} \hat{\varphi}(k, beg_k), \alpha_1 > \alpha_2$ or $T_{opt} \leq \max_{1 \leq k \leq U} \hat{\varphi}(k, end_k), \alpha_1 < \alpha_2$. When underproduction is penalized more than overproduction the maximum value will be situated on the right end of the quota band, otherwise on the left end. Then, in the case of asymmetric functions, we have introduced an extension of the band that does not have a symmetric property as the quota band. This extension is achieved by adding in the examination all matrix element values less or equal to the above-mentioned upper bound. With regard to the computing time, it seems that matrix values have an influence on the performance of some methods and, in general, a growth of computing time (in comparison with symmetric discrepancy functions) has been noted. For such methods as APS or BI-APS approaches with LAPJVsp, the increase could be explained by the extension of the searching interval for T . This suggests that more feasible T values must be explored. However, for the MPS approach it appears that the computational time result has even been improved, while the addition of the bisection search (BI-MPS), leads to a worst result in the case of tried instances. Table 5 presents an example of the application of all methods to the particular instance with symmetric and asymmetric discrepancy functions ($\alpha_1=1$ and $\alpha_2=3$).

Table 5.

Example with symmetric and asymmetric discrepancy functions

Instance:	APS approach	BI-APS approach	MPS approach	BI-MPS approach	BOTJV	BASS	Bottleneck value T:
V=40							
U=1000	CPU time (s) (without matrix computational time)						
Symmetric	6.59 (23)	1.93 (6)	0.18 (7)	0.17 (5)	1.34	0.86	0.922
Asymmetric	93.72 (64)	13.73 (9)	0.16 (4)	0.54 (12)	2.67	3.21	2.766

As a last step of our study, we have extended the initial computational experience for larger dimension instances (up to $V=9000$ and $U=10,000$) applying the best performing approaches MPS and BI-MPS and using symmetric discrepancy functions. Their respective FORTRAN codes were translated in C++. In this part of the computational experiment we report and compare clock times. The bisection technique allows for significant reduction of the search interval, although in each step we remove more edges, thus having more exposed nodes in initial matching which increases performance time of the BMCM algorithm. The number of times the BMCM is run as the result of this reduction can decrease significantly (see example in Table 3), nevertheless it does not happen in all cases. The latest computational experiment has shown that MPS and BI-MPS are very similar in relation to the total clock time for all instances except one particular type, when the major part of total units to be produced is distributed between few products. It is then that the MPS approach performs faster than BI-MPS. An example of such a specific type of instance is presented next:

$$U=2000, V=150, u_1=555, u_2=913, u_3=291, u_4=95, u_5=\dots=u_{150}=1.$$

For the above-mentioned instance the total MPS computing time is 0.77 s with 25 iterations of the BMCM algorithm, while the total computing time of BI-MPS is 5 s with 10 iterations. The explanation is in the excessive reduction of the search interval when applying the bisection technique to the initial interval, which leads to the increase of the number of exposed nodes and consequently of the BMCM computing time. As well, in such cases we have observed the increase of the number of iterations after which the matching was not completed, having to move to the upper half of the search interval.

Further to the comparison of two approaches, Table 6 contains maximum and average computing times (s) for both approaches, respectively (computing time of heuristic algorithm is not included) for different large dimension instances. In total, more than 22,000 instances with different (V, U) values have been explored. It has been observed that high computing times (in relation to times of other instances) occur in cases of types of particular instances described before. Among such instances the maximum time value corresponds to those whose ratio is $V/U \approx 0.4-0.7$. This type of instances has influenced the average computing time, especially of the BI-MPS approach.

Table 6.

Computing time results for the large dimension instances

U	Number	MPS approach				BI-MPS approach			
		Max	Max	Average	Average	Max	Max	Average	Average
	of tried								
	instances	_____				_____			
		number of	clock	number of	clock	number of	clock	number of	clock

<i>U</i>	Number	MPS approach				BI-MPS approach			
	of tried								
		iterations	time (s)	iterations ^a	time (s)	iterations	time (s)	iterations ^a	time (s)
500	3842	88	0.33	9	0.05	11	0.49	3	0.06
1000	2385	103	1.43	10	0.34	12	3.73	3	0.48
1500	2488	253	3.95	11	1.12	13	12.30	4	1.47
2000	1080	165	9.18	12	2.65	13	27.4	4	3.28
2500	1140	191	19.93	12	5.12	13	56.24	4	6.24
3000	4250	360	36.63	11	9.05	14	225.58	4	12.45
3500	1520	272	47.57	13	13.75	14	142.7	4	16.87
4000	1760	245	89.53	13	20.26	14	200.97	4	25.09
5000	1200	391	139.62	13	43.81	14	415.18	4	47.34
6000	714	375	361.51	17	72.93	14	3155.85	4	173.53
7000	446	725	378.93	17	96.27	14	507.56	4	101.26
8000	575	668	746.76	18	160.57	14	6215.44	4	291.88
9000	699	1274	807.07	17	165.48	14	2425.22	4	239.05
10,000	732	740	1097.95	17	233.10	14	3216.78	4	322.49
Total	22,831	^a Average of number of iterations is rounded to integer							

Fig. 4 presents graphic comparison of average computing times of MPS and BI-MPS approaches for the set of instances with $U=3000$, divided in 889 subsets of 12 instances each, corresponding to different V values, where $V_{\min}=10, V_{\max}=2990, V_{k+1}=V_k+40$.

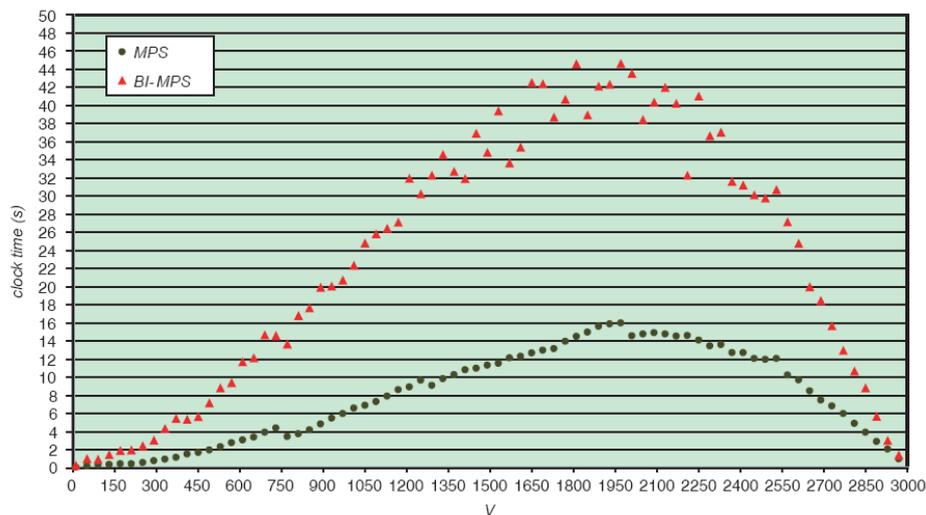


Fig. 4. Comparison of computing times for the set of instances with $U=3000$.

In order to compare the efficiency of MPS and BI-MPS approaches we have additionally implemented the Steiner and Yeomans approach [6], designed for the *minmax* PRVP solution. Coding this approach we have used more recent upper bounds on the bottleneck value, mentioned earlier in this work. A new set of 4620 instances has been generated. Table 7 presents average computing times for the three chosen approaches. Average times of the Steiner and Yeomans approach are higher in comparison to MPS and BI-MPS times. One possible explanation may be that the computational time of the matching algorithm is increased when starting each iteration with empty matching.

Table 7.

Comparison of average computing times

U	Number of tried instances	MPS approach	BI-MPS approach	S&Y approach
		Average clock time (s)		
500	1188	0.05	0.10	0.73

<i>U</i>	Number of tried instances	MPS approach	BI-MPS approach	S&Y approach
		_____	_____	_____
		Average clock time (s)		
1000	1782	0.34	0.84	5.75
1500	1650	0.88	2.34	17.67

5. Conclusions

In this research, the *minmax* PRVP of large dimensions regarded as a BAP has been solved successfully. Several efficient methods have been adapted. Also, the computational experiment with small dimensions and, for the first time, with large dimension instances of the *minmax* PRVP has been carried out. According to our computational experiment, an approach using a sequence of matching problems to solve the *minmax* PRVP proposed by Steiner and Yeomans [6], which has been modified by introducing a new initialization, outperforms computational results of other approaches for the majority of the instances. The improvement of MPS computational result has been attempted, so the bisection search technique has been implemented (BI-MPS).

To sum up, in cases where the *minmax* PRVP instance has to be solved, we recommend the application of the MPS or BI-MPS approaches of those included in the present research, since they seem to be the most efficient for the solution of the *minmax* PRVP.

References

- [1] Y. Monden, Toyota production system, Institute of Industrial Engineers Press, Norcross, GA (1983).
- [2] W. Kubiak, Minimizing variations of productions rates in just-in-time systems: a survey, *European Journal of Operational Research* 66 (1993), pp. 259–271.
- [3] J.G. Miltenburg, Level schedules for mixed-model assembly lines in just-in-time production systems, *Management Science* 35 (1989) (2), pp. 192–207.
- [4] W. Kubiak and S. Sethi, A note on level schedules for mixed-model assembly lines in just-in-time production systems, *Management Science* 37 (1991) (1), pp. 121–122.
- [5] W. Kubiak and S. Sethi, Optimal just-in-time schedules for flexible transfer lines, *International Journal of Flexible Manufacturing Systems* 6 (1994), pp. 137–154.
- [6] G. Steiner and S. Yeomans, Level schedules for mixed-model, just-in-time processes, *Management Science* 39 (1993) (6), pp. 728–735.

- [7] F. Glover, Maximum matchings in a convex bipartite graph, *Naval Research Logistics Quarterly* 4 (1967), p. 3.
- [8] J. Bautista, R. Companys and A. Corominas, Modelling and solving the production rate variation problem, *TOP* 5 (1997) (2), pp. 221–239.
- [9] J.W. Still, *A class of new methods for congressional apportionment*. *SIAM Journal of Applied Mathematics* 37 (1979) (2), pp. 401–418.
- [10] J. Bautista, R. Companys and A. Corominas, A note on the relation between the product rate variation (PRV) problem and the apportionment problem, *Journal of the Operational Research Society* 47 (1996) (11), pp. 1410–1414 (ISSN 0160-5682).
- [11] M.L. Balinski and H.P. Young, Fair representation, Yale University Press, New Haven, CT (1982).
- [12] Brauner N, Crama Y. Facts and questions about the maximum deviation just-in-time scheduling problem. Working Paper GEMME 0013, Université de Liège, 2001.
- [13] R. Tijdeman, On a distribution problem in finite and countable sets, *Journal of Combinatorial Theory* 15 (1973), pp. 129–137.
- [14] M.Y. Kovalyov, W. Kubiak and J.S. Yeomans, A computational analysis of balanced JIT optimization algorithms, *INFOR* 39 (2001) (3), pp. 299–316.
- [15] G. Carpaneto and P. Toth, Algorithm for the solution of the bottleneck assignment problem, *Computing* 27 (1981), pp. 179–187.
- [16] R. Jonker and A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing* 38 (1987), pp. 325–340.
- [17] R.E.D. Woolsey and H.S. Swanson, Operations research for immediate application. A quick & dirty manual, Harper & Row (1995).
- [18] Dell'Amico M, Toth P. Algorithms and codes for dense assignment problems: the state of the art. DEP, Università di Modena, DEIS, Università di Bologna, 1998.
- [19] Moreno N. Solving the product rate variation problem (PRVP) of large dimensions as an assignment problem, Doctoral Thesis, DOE, ETSEIB-UPC, 2002.