A mathematical model and a hybrid heuristic for sequencing mixed-model assembly lines with disabled workers.

Pâmela Michele C. Cortez, Alysson M. Costa

Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
{pamela,alysson}@icmc.usp.br

August, 2011

Abstract

Assembly lines are flow oriented production systems where parts of a product are assembled in different workstations. Depending on the context, the planning and operation of these lines give rise to different combinatorial optimization problems. In this study, two variants are of particular interest: the mixed-model assembly line sequencing problem, where different versions of a product must be sequenced in the line; and the assembly line worker assignment and balancing problem, where workers in the line present different characteristics. In this article, we study these two variants in an integrated fashion. The problem is defined and a linear mathematical model is introduced. Moreover, a hybrid heuristic approach is developed and tested on small-scale instances. The computational experiments show that the proposed method is both fast and acurate.

Keywords: Assembly lines, disabled workers, multi-models.

1 Introduction

According to the World Health Organization (2010), there are more than 600 million disabled people in the world. This significant part of the population tends to encounter much higher difficulties to enter the work market. As an example, in Brazil, more specifically in the city of São Paulo, a study carried out by Employment Secretary of this city showed that only 10% of disabled people on working age were employed (SERPRO; 2004). This fact highlights the importance of taking actions that guarantee equal work opportunities for disabled people. An example of this kind of actions is the creation of Sheltered Work centers for the Disabled (referred to as SWD henceforth). These centers are not profit-driven and aim to provide a first work-environment for disabled people. Nevertheless, profit and efficiency are welcome in SWDs for they can be used to expand the center and employ a larger number of disabled workers.

Assembly lines are production structures commonly used in SWDs. An assembly line is a flow oriented production system where parts of a product are assembled in different workstations. In the classical situation, workers are considered to be equally effective. The major planning problem is therefore to obtain an appropriate distribution of tasks among the workstations, such that the bottleneck of the line (given by the most loaded station) is minimized.

In the case of assembly lines with disabled workers, the challenge is even more complex. Indeed, since the execution time of a given task may be highly dependable on the worker assigned to its execution (due to each worker particularities), the stations load will depend not only on the tasks assigned to each workstation but also on the distribution of the workers along the line. This gives rise to a double-assignment problem that is known in the literature as the *assembly line worker assignment and balancing problem* (ALWABP).

In this piece of work, we consider the ALWABP with a further extension: different versions of a same product are available. We call this problem the *mixed model assembly line worker assignment and balancing problem* (MALWABP). The goal in this case is still the obtention of an appropriate assignment of tasks and workers to the workstations. Nevertheless, two new important aspects must be considered: 1) different product versions may have different task execution times and 2) besides the task and workers assignment one must also decide in which order the products should be processed. Assembly lines producing different models of a same product are growing in number in practical and research contexts, due to the growing demand for mass-customized products. The ability of assembly lines in SWDs to operate with different product models might open new markets for these centers and contribute to their growth. To the best of the authors knowledge, this is the first study focusing this important case in the literature.

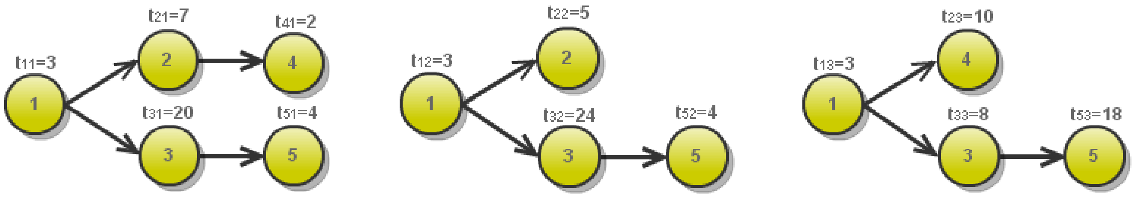
In this exploratory study we propose a model for obtaining an appropriate sequencing of the different models, once a solution has been found for the balancing problem. We also develop a hybrid heuristic that combines simple constructive and local search procedures with the optimal resolution of linear programming models. Computational experiments with both the model and the heuristic procedure are presented and analyzed.

The remainder of this paper is organized as follows. Section 2 reviews related research. In Section 3, the model for the sequencing problems is presented and discussed. Section 4 describes the heuristic methods proposed for the solution of the sequencing problem. Section 5 reports and discusses our computational experiments. Finally, Section 6 ends this paper with some general conclusions and future research directions.

2 Problem definition and literature review

An extension of the classic *simple assembly line balancing problem* (SALBP), where the line is arranged to produce more than one product type or different models of a product was first proposed by Salveson (1955) and named *mixed-model assembly line balancing problem* (MALBP). Since then, many methodologies have been proposed to its resolution. Most of these approaches are based on the idea proposed by Thomopoulos (1970), which consists in solving MALBP instances via SALBP models. This is done simply by computing average task times for all tasks, based on the tasks times for each different model and on the demand of the models. As an example, take the precedence graphs for three different models presented in Figure 1. In these graphs, nodes represent tasks and arcs represent precedence relationships. Task execution times are shown above the nodes and the demands for each model are presented in the captions.

Figure 2 illustrates an unified precedence graph where task times are assumed to be a weighted average over all models, given the expected demands, i.e., $\bar{t}_j = \sum_{m\in M}(d_m/|I|)t_{jm}$, where d_m is the expected demand for model m during the planning period, t_{jm} is the execution time for task j in model m, and $|I| = \sum_{m\in M} d_m$. Figure 2 shows the SALBP instance associated with the mixed-model case represented in Figure 1. A solution to this resulting SALBP instance will be an assignment of tasks and workers that is capable of producing the (equivalent) averaged model in a given cycle time.



model 1 ($d_1 = 2$) model 2 ($d_2 = 1$) model 3 ($d_3 = 1$)

Figure 1: Precedence graph for each model

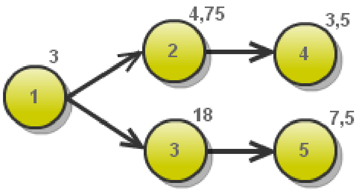


Figure 2: Joint precedence graph, given $d_m = (2, 1, 1)$

Due to the difference among the models, station times may exceed the cycle time for particular models, and therefore, a worker will not be able to finish the current workpiece inside his/her workstation boundaries whenever the workload for the current model is greater than the cycle time. These situations, called work overload, may be compensated by the temporary employment of the so-called utility workers, extra workers that provide help when needed. This support provided by utility workers is assumed to double the processing speed, i.e. to halve the required (residual) processing time, and to start exactly at that point in time that allows the work to be completed at the right-hand border of the station. The employment of utility workers represent an extra cost and should be minimized. This can be done by an appropriate sequencing of the models, giving rise to the *mixed-model sequencing problem* (MSP).

Several papers focus on minimizing the work overload applying different meta-heuristics, especially genetic algorithms (Boysen et al.; 2010; McMullen; 2010; Wang; 2010; Akgündüz and Tunali; 2010; Shao et al.; 2010; Alpay; 2009; Mansouri; 2005; Ponnambalam et al.; 2003; Dong et al.; 2002; Hyun et al.; 1998; Scholl et al.; 1998; Leu et al.; 1996; Kim et al.; 1996; Gouveia and Voß; 1995). The survey of Boysen et al. (2009) provides a good overview of the mixed-model assembly line sequencing literature. Although both balancing and sequencing problems have drawn the attention from the research community, the consideration of these problems in conjuction with ALWABP models has been ignored. In the next section, we propose a mathematical formulation for sequencing problems in the context of SWDs.

3 Mathematical modeling

In this section, we present a model for sequencing mixed-model assembly lines with disabled workers. This model modifies the model proposed by Scholl (1999) to the MSP case in order to consider the fact that workers are now different from each other and from utility workers. Let x_{mi} be binary variables, $x_{mi} = 1$ if a unit of model m is assigned to the i-th position in the production sequence, and $x_{mi} = 0$ otherwise. Also, assume continuous variables s_{ki}, denoting the initial processing point of the i-th unit in station k, and y_{ki}, denoting the operation time employed by an utility worker at station k to finish the i-th item. We consider that:

- a worker cannot start processing the next item before the current item is completed in his/her workstation;
- worker w_k cannot operate across workstation boundaries. As a consequence, whenever worker w_k is not able to finish the current workpiece inside his/her workstation boundaries, the presence of an utility worker is needed at station k, i.e., $y_{ki} > 0$;
- all utility workers are non-disabled workers and the total execution time of tasks N_k for an utility worker is given by $t_{mk}^{min} = \sum_{j \in N_k} min_{w \in W}\{t_{jmw}\}$.

We use the nomenclature shown in Table 1.

Table 1: Notation for the sequencing problem

i	index for sequence positions $(i = 1, \cdots, \|I\|)$,
j	index for tasks $(j = 1, \cdots, \|N\|)$,
k	index for workstations $(k = 1, \cdots, \|K\|)$,
w	index for workers $(w = 1, \cdots, \|W\|)$,
m	index for models $(m = 1, \cdots, \|M\|)$,
N	set of tasks,
N_w^∞	set of tasks that worker w is not able to execute,
K	set of workstations,

W	set of workers,
M	set of models,
I	model sequence,
m_i	model m which is the i-th to be assembled,
d_m	demand for each model m,
d_m^*	residual demand for each model m,
$\lvert I\rvert$	total demand over the planning horizon ($\lvert I\rvert = \sum_{m \in M} d_m$),
w_k	worker assigned for workstation k,
N_k	set of tasks assigned for workstation k,
t_{jmw}	processing time for task j model m when executed by worker w ($t_{jmw} \to \infty$ if $j \in N_w^\infty$),
t_{mk}	time required to worker w_k finished the tasks N_k on an item of model m,
C	cycle time,
l_k	"length" of station k given in time units ($l_k =$ length of station $k/$conveyor velocity $\geq C$).

Let us define the time execution of the i-th item at workstation k as:

$$\rho_{ki} = \sum_{m \in M} t_{mk} \cdot x_{mi}, \; k = 1, \cdots, |K|, \; i = 1, \cdots, |I| \tag{1}$$

Also, let Δt_{mk} be a parameter representing the relative efficiency ratio between worker w_k and an utility worker:

$$\Delta t_{mk} = t_{mk}/t_{mk}^{min}, \; m = 1, \cdots, |M|, \; k = 1, \cdots |K| \tag{2}$$

As a direct consequence of $t_{mk} \geq t_{mk}^{min}$, $\Delta t_{mk} \geq 1$. The larger Δt_{mk}, the lower is the relative efficiency of worker w_k in comparison to an utility worker. In practical terms, this means that each unit (time) of work employed by an utility worker can reduce the assembly of tasks N_k in Δt_{mk} units of time, in average.

With variables x_{mi}, s_{ki} and y_{ki} described above, we can write the sequencing problem as follows:

$$\text{Min} \sum_{i \in I} \sum_{k \in K} y_{ki} \tag{3}$$

subject to

$$\sum_{m \in M} x_{mi} = 1 \qquad \forall i \in I, \tag{4}$$

$$\sum_{i \in I} x_{mi} = d_m \qquad \forall m \in M, \tag{5}$$

$$s_{ki} + \rho_{ki} - C - \sum_{m \in M} \Delta t_{mk} \cdot y_{ki} \leq s_{k,i+1} \qquad \forall k \in K, \forall i \in I, \tag{6}$$

$$s_{ki} + \rho_{ki} - \sum_{m \in M} \Delta t_{mk} \cdot y_{ki} \leq l_k \qquad \forall k \in K, \forall i \in I, \tag{7}$$

$$s_{k1} = 0, s_{k,I+1} = 0 \qquad \forall k \in K, \tag{8}$$

$$s_{ki} \geq 0, y_{ki} \geq 0 \qquad \forall k \in K, \forall i \in I, \tag{9}$$

$$x_{mi} \in \{0, 1\} \qquad \forall m \in M, \forall i \in I. \tag{10}$$

Model (3)–(10) is an adaptation of the model proposed by Scholl (1999) with modifications in constraints (6) and (7). Constraints (6) state that a worker will not start another workpiece before completing the current item, possibly with help of an utility worker. Constraints (7) guarantee that the workstation boundaries (l_k) will not be crossed by w_k. The limit l_k is important because we consider that all workstations are closed, thus the workload of a workstation has no impact on the succeeding station. The difference between these constraints and their counterpart for the case in which every worker is equally skilled is that, on average, the utility worker is Δt_{mk} faster than worker w_k when tasks N_k are considered. Clearly, the utility worker is not executing all tasks in N_k and, therefore, Δt_{mk} is an approximation of the difference in efficiency between worker w_k and the utility worker for the tasks the utility worker actually executes.

The remaining constraints can be understood as follows: constraints (4) and (5) guarantee that, at each cycle time, exactly one model will be started in the assembly line, and that the demand for the planning horizon has to be met. Equations (8) ensure that the line is in the initial state before and after producing all units. Constraints (9) and (10) define the escope of the decision variables.

4 Hybrid heuristic procedure

The model presented in the previous section was tested using the commercial mixed-integer programming solver CPLEX 12.1[1]. Because of the poor scalability of the used branch-and-cut approach, CPLEX was only able to solve instances containing a maximum of 4 models and a total demand of 25 items. Motivated by this, a hybrid solution method is proposed. The rationale of this approach is to use simple constructive procedures and local search movements to determine a good production sequence for the different models and obtain the initial processing positions for each model unity and the appropriate values for the utility workforce via linear programming.

In practical terms, the heuristic first finds an initial sequence for the model unities. This is done via a greedy criterion inspired in the ideas of Scholl et al. (1998). The first step of the heuristic is to choose a model to start the production. A possible idle time of this first model cannot compensate work overload from a previous item, because, obviously, there is not a previous one. Therefore, if the first item of the sequence needs less than the cycle time to be assembled in any workstation k, the worker w_k will finish the item and wait for the next piece to enter the line on the next cycle time. To avoid idle time, the first model to be chosen is the one with lowest idle time over all models. The following models in the sequence are chosen according to the criterion expressed in (11). The rationale behind this criterion is to choose the following model as the one that better compensates the possible work overload of the preceeding item. In case of ties, the model with the lowest idle time over all workstations is selected. At each step, a model m can only be selected if it has a residual demand, i.e., $d^*_m > 0$.

$$g_{im} = \sum_{k \in K} max\{max\{t_{m_{i-1}k} - C, 0\} - max\{C - t_{m_i k}, 0\}, 0\} \qquad (11)$$

The pseudo-code of this first step is shown in Algorithm 1. This procedure defines heuristically the integer variables x_{mi} of model (3)–(10). The remaining (continuous) variables are then obtained by solving the linear problem that remains by fixing integer variables x_{mi} in (3)–(10).

An improvement of this initial solution can be achieved by applying a local search strategy. A simple neighbourhood would be the all-pairs, i.e, to consider all solutions that differ from the current solution in only two sequence positions.

[1] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

Algorithm 1: Greedy heuristic HC

```
for $m = 1;\ m \leq |M|$ do
    $d^*_m = d_m$;
end
$S_1 = m \mid min_{m \in M}\ \sum_{k \in K} max\{C - t_{mk}, 0\}$;
$d^*_{S_1} = d^*_{S_1} - 1$;
for $i = 2;\ i \leq |I|$ do
    bestm = 1;
    bestmvalue = ∞ ;
    bestidle = ∞ ;
    for $m = 1;\ m \leq |M|$ do
        if $d^*_m > 0$ AND $g_{im} < bestvalue$ AND $\sum_{k \in K}\ max\{C - t_{mk}, 0\} < bestidle$ then
            bestm = m;
            bestmvalue = $g_{im}$;
            bestidle = $\sum_{k \in K}\ max\{C - t_{mk}, 0\}$;
        end
    end
    $S_i$ = bestm;
    $d^*_{bestm} = d^*_{bestm} - 1$;
end
```

Note, however, that each modification on the integer variables (model sequence) must be evaluated with a call to the equivalent linear program obtained by fixing the integer variables in (3)–(10). In order to minimize the computational burden of the local search, a more elaborate neighbourhood is proposed and described in the following.

The work overload is caused by the interation between models of subsequente positions: if a worker w_k operates a item i more than C units of time, the item $i+1$ will not be assembled as soon as it enters the workstation k, i.e, $s_{k,i+1} > 0$. But, if the model m_{i+1} needs more than $l_k - s_{k,i+1}$ units of time to be assembled, a work overload in station k and unit $i+1$ will happens. This suggests that, model m_i is not a good precedessor to model m_{i+1}, and therefore, we can try to eliminate the work overload by swapping model in position i with another model.

Let i' be the position in the sequence with highest work overload. We propose a neighbourhood that swaps model $m_{i'-1}$ with all other positions containing different models, i.e, $\forall i^* \mid m_{i^*} \neq m_{i^\star}$ a movement is made: swap(m_{i^*}, $m_{i^\star}$), where we denote $i' - 1$ by $i^\star$. To evaluate the quality of the new solution (S') a linear problem is solved after each swap. The current solution is replaced by the best neighbour solution, i.e., with lowest work overload, if the new solution is better than the incumbent. The algorithm stops when there is no neighbour solution better than the current (incumbent) solution. Algorithm 2 presents the pseudo-code for the local search method.

5 Results

5.1 Instance's generation

Since the MSP with disabled workers has never been considered in the literature, there are no benchmark results that can be used for comparison. Two families of instances, with 80 instances each, commonly used in the ALWABP literature were therefore proposed (further details on these instances can be found in Chaves et al. (2007)). To derive the parameters for the different models, we

Algorithm 2: Local Search

```
$bestS = S = S_{HC}$ (solution from the hybrid constructive heuristic);
$improved = true$;
while $improved$ do
    if $i' > 0$ then
        $i^\star = i' - 1$;
    else
        $i^\star = i' + 1$;
    end
    $improved = false$;
    for $\forall i^* \mid m_{i^*} \neq m_{i^\star}$ do
        $S'$ = new solution generated from $S$ + swap($m_{i^*}$, $m_{i^\star}$);
        if $value(S') < value(S)$ then
            $improved = true$;
            $bestS = S'$;
        end
    end
end
```

increased the task time of the first model (given by the ALWABP instance) by a value randomly obtained in the range [0,4], where only integer values we considered. A task that is not needed for a model is simulated when the multiplier value is zero. In our experiments, we use 4 models and total demand of 25 items.

5.2 Computational results

The experiments have been conducted on a Intel® Core™ 2 Quad 2,66 GHz machine with 4 GB of memory running Ubuntu Linux 9.10. All the algorithms were implemented in C++, compiler g++ 4.4, and CPLEX 12.1. Table 2 reports the optimality gap of the proposed hybrid heuristic followed by Local Search, as well as standard deviation and largest gap in each family of instances, and the average computational time ($\overline{t_h}$) and standard deviation time ($\sigma_{\overline{t_h}}$) for solving the heuristic and the exact approach of CPLEX ($\overline{t_o}$, $\sigma_{\overline{t_o}}$).

Table 2: Gap and time execution for each family of instances

	Heuristic					CPLEX	
	$\overline{gap}$ (%)	$\sigma_{\overline{gap}}$ (%)	largest gap (%)	time $\overline{t_h}$ (s)	$\sigma_{\overline{t_h}}$ (s)	time $\overline{t_o}$ (s)	$\sigma_{\overline{t_o}}$ (s)
Heskia	1.3	1.1	5.7	0.2	0.3	1,634	2,180
Roszieg	1.9	1.7	17.0	0.4	0.3	630	946

The optimality gaps show that the proposed heuristics lead to good quality solutions in a very small fraction of time needed for solve to optimality. Indeed, the average gaps are 1.3% and 1.9% for families Heskia and Roszieg, respectively, with very small standard deviations showing the robustness of the approach.

Moreover, the computational times are never larger than 1.3 s, whereas the large standard deviation time for CPLEX execution, in each family of intances, show that CPLEX was not able to found the optimal solution in a reasonable time on several instances. Indeed, when the optimal solution was trivial (any production sequence had no work overload), CPLEX found that situation quickly, but the time execution reached more than 4,000 s in more complicated instances. For comparison purposes, note that CPLEX spends around 1130s to solve each instance.

The good quality and the satisfactory execution time of the heuristics, for all 160 instances studied, suggest that our method is applicable for the resolution of the proposed problem. To a more complete study, we are testing the heuristics with large instances (more than 25 models and 125 itens to be produced).

6 Conclusions

We proposed a mathematical model for sequencing problems in the context of multi-models being assembled by disabled workers. Although there is a vast literature on the *mixed-model assembly line balancing and sequencing problem* and, more recently, on the *assembly line worker assignment and balancing problem*, these two problems had never been considered in an integrated fashion.

Sequencing problems are combinatorial in nature, suggesting computational intractability. In this paper we proposed a hybrid heuristic that combines simple heuristics based on a greedy criterion, local search and linear programming to tackle this problem. Computational experiments showed that this hybrid approach is capable of obtaining good results both in terms os solution quality and in terms of execution times.

The main contribution of this research is the definition of a new problem, with practical relevance, the modeling of such problem in a linear fashion and the proposal of a simple and efficient hybrid heuristic. Future research will concentrate on testing the method with large-scale instances, eventually reducing the need of solving linear programming models at each neighborhood evaluation.

References

Akgündüz, O. S. and Tunali, S. (2010). An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem, *International Journal of Production Research* **48**: 5157–5179.

Alpay, S. (2009). Grasp with path relinking for a multiple objective sequencing problem for a mixed-model assembly line, *International Journal of Production Research* **47**: 6001–6017.

Boysen, N., Fliedner, M. and Scholl, A. (2009). Sequencing mixed-model assembly lines: Survey, classification and model critique, *European Journal of Operational Research* **192**: 349–373.

Boysen, N., Kiel, M. and Scholl, A. (2010). Sequencing mixed-model assembly lines to minimise the number of work overload situations, *International Journal of Production Research* **online first**. DOI: 10.1080/00207543.2010.507607.

Chaves, A. A., Miralles, C. and Lorena, L. A. N. (2007). Clustering search approach for the assembly line worker assignment and balancing problem, *37th International Conference on Computers and Industrial Engineering*, Alexandria, Egypt, pp. 1469–1478.

Dong, J., Xiao, T., Fan, S. and Qiang, L. (2002). Mixed-model assembly line sequencing with hybrid genetic algorithm and simulation, pp. 541–545.

Gouveia, L. and Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem, *European Journal of Operational Research* **83**: 69–82.

Hyun, C. J., Kim, Y. and Kim, Y. K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines, *Computers & Operations Research* **25**: 675–690.

Kim, Y. K., Hyun, C. J. and Kim, Y. (1996). Sequencing in mixed model assembly lines: A genetic algorithm approach, *Computers & Operations Research* **23**: 1131–1145.

Leu, Y., Matheson, L. A. and Rees, L. P. (1996). Sequencing mixed-model assembly lines with genetic algorithms, *Computers and Industrial Engineering* **30**: 1027–1036.

Mansouri, S. A. (2005). A multi-objective genetic algorithm for mixed-model sequencing on jit assembly lines, *European Journal of Operational Research* **167**: 696–716.

McMullen, P. R. (2010). Jit mixed-model sequencing with batching and setup considerations via search heuristics, *International Journal of Production Research* **48**: 6559–6582.

Ponnambalam, S. G., Aravindan, P. and Subba Rao, M. (2003). Genetic algorithms for sequencing problems in mixed model assembly lines, *Computers and Industrial Engineering* **45**: 669–690.

Salveson, M. E. (1955). The assembly line balancing problem, *Journal of Industrial Engineering* **6**: 18–25.

Scholl, A. (1999). *Balancing and sequencing of assembly lines*, Physica-Verlag.

Scholl, A., Klein, R. and Domschke, W. (1998). Pattern based vocabulary building for effectively sequencing mixed-model assembly lines, *Journal of Heuristics* **4**: 359–381.

SERPRO (2004). Pesquisa aponta taxa de desemprego de 88,7% entre portadores de necessidades especiais. Last access April 15th, 2011.
URL: *http://www.serpro.gov.br/noticias-antigas/noticias-2004/20041221_02*

Shao, X., Wang, B., Rao, Y., Gao, L. and Xu, C. (2010). Metaheuristic approaches to sequencing mixed-model fabrication/assembly systems with two objectives, *International Journal of Advanced Manufacturing Technology* **48**: 1159–1171.

Thomopoulos, N. T. (1970). Mixed model line balancing with smoothed station assignments, *Management Science* **16**: 593–603.

Wang, B. (2010). Sequencing mixed-model production systems by modified multi-objective genetic algorithms, *Chinese Journal of Mechanical Engineering (English Edition)* **23**: 537–546.

World Health Organization, W. H. O. (2010). World report on disability. Last access April 15th, 2011.
URL: *http://www.who.int/disabilities/world_report/concept_note_2010.pdf*