

5-2005

Solving the Car Sequencing Problem Using Heuristic Approaches (Définition de la Séquence de Production pour une Linge d'Assemblage d'Automobiles)

Soumia Ichoua
Embry-Riddle Aeronautical University, ichouasoumia@msn.com

Abdelilah Chiguer

Fayez F. Boctor
Universite Laval

Follow this and additional works at: <https://commons.erau.edu/ww-leadership>



Part of the [Automotive Engineering Commons](#)

Scholarly Commons Citation

Ichoua, S., Chiguer, A., & Boctor, F. F. (2005). Solving the Car Sequencing Problem Using Heuristic Approaches (Définition de la Séquence de Production pour une Linge d'Assemblage d'Automobiles). , (). Retrieved from <https://commons.erau.edu/ww-leadership/8>

This Conference Proceeding is brought to you for free and open access by the College of Business at Scholarly Commons. It has been accepted for inclusion in Leadership and Management - Worldwide by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

DÉFINITION DE LA SÉQUENCE DE PRODUCTION POUR UNE LIGNE D'ASSEMBLAGE D'AUTOMOBILES

Le principe de la personnalisation de l'offre amène l'industrie de l'automobile à proposer, pour chaque modèle vendu, un nombre important d'options, telles que le toit ouvrant, l'air climatisé ou les freins ABS. Cette diversité d'options nous oblige à accorder une grande attention au problème de la détermination de la séquence de voitures à assembler car l'équilibrage de la ligne d'assemblage en dépend. Pour définir une bonne séquence pour une journée de travail, nous proposons d'utiliser des heuristiques pour générer une séquence initiale puis d'utiliser des procédures pour en améliorer la qualité.

1- CONTEXTE

Une ligne d'assemblage de voitures est généralement dédiée à la production d'un modèle donné de voiture mais assemble plusieurs variantes de ce modèle. Un modèle se définit par un ensemble de caractéristiques techniques que partagent toutes les variantes du modèle et chaque variante se définit par un ensemble d'options qui lui sont propres. La plupart de lignes d'assemblage de voitures est composée de trois ateliers : l'atelier de tôlerie, l'atelier de peinture et l'atelier de montage. Un contrôle se fait à la fin de chaque atelier et les voitures ayant besoin de retouches sont retirées de la ligne principale pour faire les réparations nécessaires puis réinsérées dans la ligne à nouveau. Le plan de production d'une journée précise la séquence exacte des voitures à assembler et leurs options. Cependant le retrait des voitures pour des retouches peut modifier de façon significative cette séquence car une voiture retouchée est généralement réinsérée dans une position différente de sa position initiale dans la séquence. Pour partiellement remédier à ces changements de séquence, les voitures quittant l'atelier de peinture sont dirigées vers une zone de stockage dite le stock de tri qui a pour objectif de permettre à rétablir autant que possible la séquence initiale.

Une bonne séquence de production permet d'équilibrer la ligne, c'est-à-dire d'égaliser autant que possible la charge de travail des postes de la ligne. Cependant, la charge d'un poste varie selon la variante (l'ensemble d'options) à assembler. On observe donc des variations de la charge d'un même poste au long d'une même journée. Pour bien équilibrer la charge de travail des postes, la séquence de production doit respecter des contraintes appelées les contraintes d'espacement des options. Une telle contrainte s'exprime par deux paramètres, n_o et m_o , qui signifient que dans

n'importe quelle séquence de m_o voitures consécutives au plus n_o voitures doivent avoir l'option o . Par exemple, Une contrainte d'espacement de l'option freins ABS de 1/4 signifie que sur n'importe quelle série de 4 voitures successives au plus une doit avoir cette option.

Le problème qui nous intéresse ici est un problème générique qui se trouve dans la plupart des usines d'assemblage de voiture. Il s'agit du problème de la définition de la séquence ou de l'ordre de passage des voitures dans les différents postes de la ligne d'assemblage qui respecte toutes les contraintes d'espacement de toutes les options.

2- REVUE DES TRAVAUX PRÉCÉDENTS

2.1- Formulations mathématiques

En 1996, Smith, Palaniswami et Krishnamoorthy [1] ont présenté une formulation mathématique de ce problème sous forme d'un programme non linéaire en nombres binaires. Dans cette formulation, les contraintes d'espacement portent sur les différentes variantes de la voiture à assembler et pas sur chaque option séparément. Rappelons qu'une variante se définit par l'ensemble d'options que la voiture doit avoir. Aussi, cette formulation modélise uniquement la contrainte d'espacement qui s'exprime comme suit : au plus une seule voiture de la même variante j toutes les m_j voitures successives. La notation suivante sera utilisée pour présenter cette formulation :

k	indice de la position dans la séquence à assembler; $k = 1, \dots, N$
j	indice de la variante (définie par un ensemble d'options) à assembler; $j = 1, \dots, M$
P_{sj}	la pénalité de placer deux voitures de la même variante j séparées par s position dans la séquence
m_j	nombre minimal de positions séparant deux voitures de la même variante j
N_j	nombre d'unités de la variante j à assembler dans la journée à planifier; $N = \sum_{j=1}^M N_j$
x_{kj}	variable binaire qui prend la valeur 1 si et seulement si la position k dans la séquence est occupée par une voiture de la variante j

En utilisant cette notation le problème peut être formulé comme suit :

$$\text{Minimiser : } \sum_{j=1}^M \sum_{k=2}^N (x_{kj} \sum_{s=1}^{\min\{k-1, m_j\}} P_{sj} x_{k-s, j}); \quad (1)$$

$$\text{Sous les contraintes : } \sum_{k=1}^N x_{kj} = N_j \quad ; j = 1, \dots, M \quad (2)$$

$$\sum_{j=1}^M x_{kj} = 1 \quad ; k = 1, \dots, N \quad (3)$$

$$x_{kj} \in \{0,1\} \quad ; k = 1, \dots, N, \quad j = 1, \dots, M \quad (4)$$

Dans cette formulation, la fonction objectif (1) donne la pénalité totale de violation des contraintes d'espacement et la contrainte (2) assure que la demande de chaque variante du modèle est satisfaite. La contrainte (3) assure qu'une et une seule voiture est assignée à chaque position dans la séquence.

Drexel et Kimms [2] proposent une autre formulation qui considère explicitement l'espacement des options où le problème est formulé en tant qu'un problème de satisfaisabilité (constraints satisfaction problem CSP) et pas en tant qu'un problème d'optimisation. Nous présentons ci-après cette formulation. Notons que Kis [3] a démontré en 2003 que ce problème est NP-complet. Pour présenter cette formulation nous utiliserons la notation supplémentaire suivante :

o indice de l'option requise; $o = 1, \dots, O$
 n_o et m_o les paramètres de la contrainte de l'espacement de l'option o indiquant que dans une séquence de m_o voitures consécutives au plus n_o doivent avoir cette option
 a_{jo} binaire égale à 1 si la variante j requiert l'option o

Le problème est alors formulé comme suit :

$$\sum_{j=1}^M x_{kj} = 1 \quad ; k = 1, \dots, N \quad (5)$$

$$\sum_{k=1}^N x_{kj} = N_j \quad ; j = 1, \dots, M \quad (6)$$

$$\sum_{j=1}^M \sum_{s=k-m_o+1}^k a_{jo} x_{kj} \leq n_o \quad ; o = 1, \dots, O, \quad k = m_o, \dots, N \quad (7)$$

$$x_{kj} \in \{0,1\} \quad ; k = 1, \dots, N, \quad j = 1, \dots, M \quad (8)$$

Dans cette formulation, la contrainte (5) impose qu'exactlyement une voiture (variante du modèle fabriqué) doit être produite dans chaque position de la séquence, la contrainte (6) assure la satisfaction de la demande de chaque variante, et la contrainte (7) est celle de l'espacement des options.

2.2- Heuristiques de résolution

Le problème de séquençage de voitures étant NP-complet, les algorithmes exacts sont incapables d'en résoudre des instances réelles souvent de grandes tailles dans un délai adéquat. Les heuristiques sont tout à fait indiquées pour réaliser un bon compromis entre l'effort de calcul et la qualité de la solution.

Hindi et Ploszajski [6] ont présenté en 1993 une heuristique de construction pour résoudre le problème. Le principe de l'heuristique est qu'on part d'une séquence initiale et on insère de

nouvelles voitures sans violer les contraintes. Si aucune voiture n'est trouvée pour la position courante, la position est laissée vide.

Smith *et al.* [1] ont présenté en 1996 trois approches pour résoudre une version simplifiée du problème. Ils proposent une adaptation de la méthode de descente ou l'heuristique d'amélioration locale, une adaptation de l'algorithme du recuit simulé [4] ainsi que des approches basées sur les réseaux neurones de Hopfield [5]. La méthode de descente part d'une solution initiale et cherche la meilleure solution dans un voisinage prédéfini. Pour ce faire, l'algorithme fait des permutations et retient la permutation qui génère le coût minimal. La solution trouvée dans ce cas est un minimum local. Les auteurs proposent également une adaptation de l'algorithme du recuit simulé. Le principe de base de cette méthode est d'autoriser temporairement des augmentations du coût afin d'échapper au minimum local et atteindre ainsi le minimum global. Les auteurs rapportent que le recuit simulé performe mieux que la méthode de descente et que l'approche basée sur les réseaux neuronaux est aussi bonne que le recuit simulé et fait mieux lorsque la séquence à définir contient plus que 40 voitures.

De leur part, Gottlieb *et al.* [7] ont présenté en 2003 des heuristiques voraces, une approche de recherche locale et un algorithme de fourmis pour résoudre le problème à l'étude. Les auteurs rapportent que les meilleurs résultats sont obtenus par l'algorithme de fourmis.

3- DESCRIPTION DU PROBLÈME ET DE L'APPROCHE PROPOSÉE

Le problème traité ici est celui de la définition de la séquence de production d'un ensemble de voitures à fabriquer le même jour sur une ligne d'assemblage mixte. Chaque voiture a un ensemble d'options et chaque option a une contrainte d'espacement. La séquence à définir doit prendre en considération les voitures non achevées de la veille dont l'assemblage est déjà commencé et la séquence ne peut donc être modifiée.

Dans la plupart des situations pratiques, il est difficile de trouver une séquence qui permet de satisfaire toutes les contraintes d'espacement. Il est donc habituel de diviser ces contraintes en deux types de contraintes d'espacement, à savoir les contraintes prioritaires et les contraintes non prioritaires. Les contraintes prioritaires correspondent à des contraintes «lourdes» de l'atelier de montage alors que les contraintes non prioritaires sont associées à des opérations plus légères.

Notre problème consiste à définir la séquence de production de la journée, en prenant en considération la séquence des voitures restantes de la veille, de façon à minimiser le nombre total des violations de contraintes prioritaires d'espacement. Les contraintes non prioritaires doivent, dans la mesure du possible, être respectées mais jamais au détriment des contraintes prioritaires.

Pour résoudre ce problème nous avons utilisé une approche heuristique hybride qui commence par utiliser une ou plusieurs heuristiques de construction pour construire une solution initiale puis utiliser une procédure d'amélioration pour améliorer cette solution initiale. Les heuristiques de construction utilisées sont présentées dans la section suivante tandis que les heuristiques d'amélioration sont données dans la section 5.

4- APPROCHES DE CONSTRUCTION DE LA SOLUTION INITIALE

Dans cette section nous présentons trois familles d'heuristiques qui serviront à construire une solution initiale. Toutes ces heuristiques sont des heuristiques de construction séquentielle qui sélectionnent la variante de voiture à placer à la première position après les voitures restantes de la veille. Puis elles sélectionnent celle à placer à la position suivante et ainsi de suite.

Certaines de ces heuristiques utilisent un indice de criticité pour sélectionner la variante de voiture à placer dans la position suivante. L'indice de criticité de la variante de voiture j , noté c_j , est la somme des indices de criticité de ses options. L'indice de criticité d'une option o , noté α_o , varie au fur et à mesure que l'on construit la séquence. Il est égal à la différence entre le ratio du nombre de voitures ayant cette option et qui restent à placer dans la séquence par rapport au nombre total des voitures restantes et le ratio n_o/m_o . Si cette différence est négative alors on met $\alpha_o = 0$. Les variantes de voiture dont $c_j > 0$ sont appelées variantes critiques.

D'autres heuristiques utilisent ce que l'on appelle le principe de conservation des proportions. Selon ce principe, la variante à placer dans la position suivante de la séquence est celle qui fait que, pour chacune des options, le pourcentage des voitures placées dans la séquence et ayant l'option considérée est le plus proche possible du même pourcentage pour l'ensemble de toutes les voitures à assembler. Pour mesurer le degré de conservation des proportions pour la variante j , noté d_j , on calcule la somme, sur toutes les options, de la différence entre le pourcentage de l'option dans toutes les voitures à assembler et celui dans l'ensemble composé des voitures déjà placées et la voiture à placer. Une deuxième mesure, notée e_j , est également utilisée. Il s'agit de la somme, sur toutes les options, de la différence entre le pourcentage de l'option dans toutes les voitures à assembler et celui dans l'ensemble des voitures non placées dans la séquence.

4.1- L'heuristique vorace (H1)

La première heuristique est une heuristique vorace simple qui consiste à affecter la position suivante de la séquence à la variante de voiture qui minimise la violation totale des contraintes prioritaires et dont la demande n'est pas encore entièrement satisfaite. Si plusieurs variantes satisfont ces conditions, on applique une règle secondaire (R2) et si l'égalité persiste, une règle tertiaire (R3) et même une quatrième et cinquième règles (R4 et R5). Six versions de cette heuristique ont été testées; voici dans l'ordre les règles utilisées par chacune de ces six versions :

- H1-1** : (R2) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R3) La variante ayant la valeur la plus élevée de l'indice de criticité c_j . (R4) Au hasard.
- H1-2** : (R2) La variante ayant la valeur la plus élevée de l'indice de criticité c_j . (R3) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R4) Au hasard.
- H1-3** : (R2) La variante ayant la valeur la plus élevée de l'indice de criticité c_j . (R3) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R4) La variante ayant la plus faible valeur de l'indice de conservation des proportions d_j . (R5) Au hasard.

- H1-4** : (R2) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R3) La variante ayant la valeur la plus élevée de l'indice de criticité c_j . (R4) La variante ayant la plus faible valeur de l'indice de conservation des proportions d_j . (R5) Au hasard.
- H1-5** : (R2) La variante ayant la plus faible valeur de l'indice de conservation des proportions d_j . (R3) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R4) Au hasard.
- H1-6** : (R2) La variante qui minimise le nombre des violations de contraintes non prioritaires. (R3) La variante ayant la plus faible valeur de l'indice d_j . (R4) Au hasard.

Les heuristiques de cette famille ont un défaut important. Elles aboutissent souvent à placer les modèles critiques à la fin de la séquence ce qui génère beaucoup de violations de contraintes d'espacement de leurs options. L'heuristique suivante a été conçue pour remédier à ce défaut.

4.2- L'heuristique de placement et de réservation (H2)

Cette heuristique a la particularité de réserver certaines positions dans la séquence à définir pour placer les variantes critiques uniquement. Au départ, aucune position n'est réservée. Pour sélectionner la variante de voiture à placer dans la position suivante de la séquence, soit la position k , on utilise les mêmes règles que l'heuristique H1-3 si k n'est pas réservée. La version H1-3 a été choisie car c'est la version qui a permis d'obtenir les meilleurs résultats. Par contre, si la position k est réservée, on choisit parmi les variantes critiques celle qui minimise le nombre de violations de contraintes prioritaires. Aussi, si la position n'est pas réservée et si la variante choisie pour occuper cette position est critique (c.-à.-d. telle que $c_j > 0$), l'heuristique calcule ce que l'on appelle le pas, noté p , en prenant le plus grand entier inférieur ou égal au rapport entre le nombre de positions restant $(N-k)$ et le nombre de voitures de cette variante qui restent à placer, noté N'_j . Puis l'heuristique déclare les positions $k + up; u = 1, \dots, \lfloor (N - k)/p \rfloor$ comme réservées.

4.3- L'heuristique de conservation des proportions (H3)

Deux versions de cette heuristique seront testées. La première version (H3-1) de cette heuristique consiste à affecter à la position suivante de la séquence la variante de voiture j qui minimise d_j . En cas d'égalité on retient celle qui minimise la violation totale des contraintes prioritaires et, si l'égalité persiste, on retiendra celle qui minimise la violation totale des contraintes non prioritaires. La deuxième version (H3-2) est identique excepté qu'elle utilise e_j à la place de d_j .

5- MÉTHODES D'AMÉLIORATION

Nous utilisons deux méthodes pour améliorer la solution initiale. Il s'agit de l'heuristique d'amélioration locale et l'utilisation de la même heuristique dans le cadre d'une approche de perturbation. Dans les deux cas, les solutions voisines sont obtenues en permutant les voitures occupant deux positions choisies au hasard.

5.1- L'heuristique d'amélioration locale

Partant d'une solution initiale, à chaque itération, cette heuristique cherche la meilleure solution voisine de la solution courante. Ce processus est répété jusqu'à ce qu'aucune permutation ne puisse améliorer la solution.

5.2- La méthode de perturbation

La méthode proposée est une méthode itérative où chaque itération consiste à exécuter l'algorithme d'amélioration locale jusqu'à ce qu'aucune amélioration ne puisse être obtenue pour ensuite modifier la solution finale afin d'en créer la solution initiale de l'itération suivante. Pour modifier (perturber) une solution, nous permutons les positions d'un nombre de paires de voitures choisies au hasard. Le nombre de paires est également choisi au hasard entre 10 % et 15 % du nombre total des voitures à assembler. La méthode s'arrête lorsqu'aucune amélioration n'est obtenue au cours de trois itérations successives.

6- ÉVALUATION DES MÉTHODES PROPOSÉES

6.1- Critères d'évaluation et problèmes-tests

Pour évaluer la performance des différentes variantes des heuristiques proposées nous nous basons sur le nombre des violations de contraintes prioritaires et sur le nombre des violations de contraintes non prioritaires.

Aussi pour tester nos méthodes, nous nous sommes servi de 23 instances du problème fournies par le constructeur d'automobiles Renault. Parmi ces instances on a 12 instances faciles et 11 instances difficiles. Une instance est considérée facile si la méthode utilisée présentement par Renault parvient à construire une séquence ne produisant aucune violation sur l'ensemble des contraintes prioritaires d'espacement d'options. À l'opposé, une instance est considérée difficile si cette méthode ne parvient pas à construire une séquence ne produisant aucune violation de l'ensemble des contraintes prioritaires. Le Tableau 1 résume les caractéristiques de ces instances.

6.2- Résultats

Les résultats numériques (voir Tableau 2) démontrent que les meilleurs résultats sont obtenus avec la version 2 de l'heuristique 1. Cette version a réalisé en moyenne, sur les vingt-trois instances, 94 violations de contraintes prioritaires et 428 violations de contraintes non prioritaires. Les versions 1 et 3 de l'heuristique 1 ont donné aussi de bons résultats. La version H1-3 a réalisé en moyenne 103 violations de contraintes prioritaires et 398 violations de contraintes non prioritaires et la variante 1 a réalisé en moyenne 104 violations de contraintes prioritaires et 409 violations de contraintes non prioritaires.

D'autre part, la deuxième version de la troisième heuristique (H3-2) donne de meilleurs résultats que sa première version (H3-1). Finalement, la deuxième heuristique (H2) donne en moyenne de meilleurs résultats que les deux versions de la troisième heuristique ainsi que la sixième variante de la première heuristique (H1-6). Enfin, notons qu'aucune variante ne surclasse toutes les autres pour toutes les instances du problème.

Puisque le temps de calcul de la solution initiale est négligeable (quelques secondes), alors on peut combiner plusieurs versions sans augmenter le temps de calcul de façon significative. Nous proposons alors de construire la solution initiale de chaque instance en utilisant les différentes versions des différentes heuristiques de construction et d'en retenir la meilleure séquence. Cette séquence est ensuite améliorée par les procédures d'amélioration proposées. Le tableau 3 présente les résultats ainsi obtenus pour toutes les instances du problème après l'application de la méthode de perturbation.

Tableau 1
Les instances du problème et leurs caractéristiques

Instances	Voitures restantes de la veille	Voitures à séquencer	Nombre de variantes	Nombre d'options	Difficulté
1	15	525	20	9	Facile
2	20	1110	41	17	Difficile
3	49	1270	66	13	Difficile
4	96	1161	172	16	Facile
5	21	365	91	20	Facile
6	37	65	10	9	Facile
7	43	730	12	7	Difficile
8	5	128	7	4	Difficile
9	24	269	8	5	Difficile
10	32	1231	86	11	Facile
11	119	1000	22	11	Facile
12	311	591	181	25	Facile
13	49	546	112	23	Difficile
14	29	825	46	14	Facile
15	39	412	10	5	Facile
16	14	485	18	9	Facile
17	14	1260	50	13	Difficile
18	14	1315	43	13	Difficile
19	228	1004	223	22	Facile
20	27	954	20	5	Difficile
21	29	875	25	9	Facile
22	5	128	7	4	Difficile
23	24	269	8	5	Difficile

Tableau 2
Résultats moyens obtenus par les heuristiques de construction
suivies par la méthode de perturbation

	Heuristiques								
	H1-1	H1-2	H1-3	H1-4	H1-5	H1-6	H2	H3-1	H3-2
Nombre moyen des violations de contraintes prioritaires	104	94	103	107	137	187	161	189	170
Nombre moyen des violations de contraintes non prioritaires	409	428	398	465	1025	518	450	676	599

Tableau 3
Solutions retenues

Instances	Difficulté	Heuristiques de construction utilisées	Violations contraintes prioritaires	Violations contraintes non prioritaires
1	Facile	H1-2	0	9
2	Difficile	H2	62	35
3	Difficile	H3-1	870	2342
4	Facile	H1-2	0	1478
5	Facile	H1-3	53	136
6	Facile	H1-1	0	68
7	Difficile	H1-3	32	2129
8	Difficile	H1-3	63	83
9	Difficile	H2	339	358
10	Facile	H1-1	0	35
11	Facile	H1-1	0	4
12	Facile	H1-6	0	102
13	Difficile	H1-1	3	6
14	Facile	H1-4	0	15
15	Facile	H1-1	0	69
16	Facile	H1-2	0	0
17	Difficile	H1-2	74	265
18	Difficile	H1-2	95	155
19	Facile	H1-2	0	593
20	Difficile	H1-2	21	0
21	Facile	H1-1	0	724
22	Difficile	H1-4	64	75
23	Difficile	H3-1	336	334
MOYENNE			87	392

7- CONCLUSION

Dans cet article, nous avons proposé une approche heuristique pour résoudre le problème de définition de la séquence de production de voitures en deux étapes. Une étape de construction et une étape d'amélioration. Pour construire une solution initiale, nous avons développé trois familles d'heuristiques. Ensuite, nous avons proposé d'améliorer la solution initiale par deux méthodes : l'heuristique d'amélioration locale et une méthode des perturbations.

La performance de ces différentes méthodes a été évaluée en utilisant des données provenant de l'entreprise Renault. Les résultats nous ont permis d'identifier des solutions prometteuses. Ainsi nous pouvons dire que résoudre le problème par chacune des méthodes de construction, retenir la meilleure des séquences obtenues puis appliquer la méthode de perturbation afin d'améliorer cette séquence, donne des résultats encourageants.

Dans des travaux futurs nous essayerons de traiter de façon coordonnée le problème de sélection des variantes de voitures à assembler dans une journée de travail et celui de la définition de la séquence à utiliser pour les assembler. Ces deux problèmes sont liés et on gagne à les traiter ensemble.

RÉFÉRENCES :

- [1] Smith, K., M. Palaniswami, and M. Krishnamoorthy, "Traditional heuristic versus Hopfield neural network approaches to a car sequencing problem", *European Journal of Operational Research*, 93 (1996), 300-316.
- [2] Drexl, A., and A. Kimms, "Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints", *Management Science*, 47 (2001), 480-491.
- [3] Kis, T., "On the complexity of the car sequencing problem", *Operations Research Letters*, 32 (2004), 331-335.
- [4] Kirkpatrick, S., C. Gelatt, and M. Vecchi, "Optimisation by simulated annealing", *Science*, 220 (1983), 671-680.
- [5] Tank, D. W., and J. J. Hopfield, "Neural' computation of decisions in optimization problems", *Biological Cybernetics*, 52 (1985), 141-152.
- [6] Hindi, K. S., and G. Ploszajski, "Formulation and solution of a selection and sequencing problem in car manufacture", *Computers and industrial Engineering*, 26 (1994), 203-211.
- [7] Gottlieb, J., M. Puchta1, and Ch. Solnon, "A Study of greedy, local search, and ant colony optimization approaches for car sequencing problems", *Euro Workshops 2003*, LNCS 2611, 246-257.