VOCAL2022

9th VOCAL Optimization Conference: Advanced Algorithms
Budapest, Hungary, May 25-27, 2022

Short Papers

VOCAL http://vocal.p-graph.org

VOCAL 2022
9th VOCAL Optimization Conference: Advanced Algorithms

Budapest, Hungary, May 25-27, 2022
Short Papers

Published by Hungarian Operations Research Society
Kende utca 13-17., 1111 Budapest, Hungary
E-mail: motelnok@inf.szte.hu
Website: http://www.mot.org.hu/
Contact: Miklós Pintér
Single volume, electronic publication, format: pdf.

ISBN 978-615-01-5987-4

VOCAL 2022. 9th VOCAL Optimization Conference:
Advanced Algorithms. Budapest, Hungary, May 25-27, 2022. Short Papers

Contents

Measuring Digital Development: Ranking Using Data Envelopment Analysis (DEA) and Network Readiness Index (NRI)
Zoltán Bánhidi and Imre Dobos 4

On the Non-Emptiness of the Core of a Cooperative Fuzzy Game
David Bartl 11

Dependence of supplier efficiency on the choice of input-output criteria in Data Envelopment Analysis (DEA)
Imre Dobos and Gyöngyi Vörösmarty 17

Large scale performance analysis of international kidney exchange programmes by the ENCKEP simulator
Kristóf Druzsin, Péter Bíró, Rita Fleiner and Xenia Klimentova 24

Dynamic programming approaches for line balancing problems
András Éles and István Heckl 30

Input design - from convex to non-convex problems and vice versa
László Gerencsér, György Michaletzky and Zsuzsanna Vágó 36

Testing re-optimisation strategies in international kidney exchange programmes by the ENCKEP simulator
Lilla Matyasi and Péter Bíró 42

On Double Roman Domination
Darja Rupnik Poklukar and Janez Žerovnik 48

Parallelizing zero-one linear programs using graph theoretical considerations
Sándor Szabó and Bogdán Zavalnij 54

Solving a dynamic route planning problem in the area of patient transport with mixed integer linear programming
Martin Tóth, Tamás Hajba and Adrián Horváth 60

The effect of problem complexity on the optimal solution of Simple Assembly Line Balancing Problems with learning
Zakaria Zine El Abidine, Imre Dimény and Tamás Koltai 68

Measuring Digital Development: Ranking Using Data Envelopment Analysis (DEA) and Network Readiness Index (NRI)

Zoltán Bánhidi[1], Imre Dobos[1]

[1] Budapest University of Technology and Economics
banhidi.zoltan@gtk.bme.hu

Abstract. The Network Readiness Index (NRI) is a composite index comprising three levels (pillars, sub-pillars and individual indicators) that benchmarks the digital development of countries based on various aspects of "network readiness". In contrast to the International Digital Economy and Social Index (I-DESI) of the European Union (EU), it aims to provide an inclusive, truly global view on digital transformation: NRI's 2021 edition shows the development of 134 countries compared to 45 countries in I-DESI, which measures only the most developed countries. The aim of this presentation is to test the robustness of NRI's scoring model using the Data Envelopment Analysis (DEA) Without Explicit Input (WEI) and Common Weight Analysis (CWA) methods. After establishing the rankings based on the aforementioned DEA models, we compare the digital development of the countries in our data set, and evaluate the impact of the different models on the ranking of countries.

Keywords: Digital transformation measurement, Network Readiness Index, Data Envelopment Analysis (DEA), DEA Without Explicit Input (WEI), Common Weight Analysis (CWA).

1 Introduction

Proper measurement of the digital economy and society is a key factor in understanding the state of digital transformation and formulating sound strategies for the green and digital transition. Based on extant definitions, Vial developed a conceptual definition of digital transformation as "a process that aims to improve an entity by triggering significant changes to its properties through combinations of information, computing, communication, and connectivity technologies" [1]. McKinsey differentiates four types of digital transformation, which are business process, business model, domain and organizational. At the macro level, digital transformation can be understood as a complex process involving transformational changes in the economic and social development of countries, achieved with the help of digital technologies.

In our earlier papers [2], [3], we used a data set comprising the "five principal dimensions" of the European Union's International Digital Economy and Society Index (I-DESI) to establish a ranking of the 28 (pre-Brexit) EU member states and the Russian

Federation with Data Envelopment Analysis (DEA) models. In this paper, we aim to evaluate the digital development of Russia and other CIS (Commonwealth of Independent States) countries with DEA models, using the sub-pillars of the Portulans Institute's Network Readiness Index (NRI).

NRI is a composite index comprising 4 main pillars, 12 sub-pillars and 62 individual indicators [4]. The sub-pillars are presented in Table 1. Digital transformation is a process that is happening at all levels: internationally, nationally and locally. NRI measures digital development at the country level and establishes a ranking of countries based on the digital development of technology, people, governance and impact.

Table 1. Pillars and sub-pillars of the Network Readiness Index.

Pillars	Sub-pillars	Weights of sub-pillars
Technology	Access	1/12
	Content	1/12
	Future Technologies	1/12
People	Individuals	1/12
	Businesses	1/12
	Governments	1/12
Governance	Trust	1/12
	Regulation	1/12
	Inclusion	1/12
Impact	Economy	1/12
	Quality of Life	1/12
	SDG Contribution	1/12

Technology is the main aspect of the network economy and therefore is the first pillar of NRI, which measures the availability and quality of access, content and future technologies. The People pillar measures ICT usage at three levels of analysis: individuals, businesses and governments. The Governance pillar reflects the quality of formal and informal institutions that are crucial to the proper functioning of the digital economy. Finally, the Impact pillar measures the broader economic, social and human impact of "readiness in the network economy".

In the next chapter, we introduce the Data Envelopment Analysis (DEA) Without Explicit Input (WEI) and the DEA Common Weight Analysis (CWA) methods, which are used in the third chapter to evaluate the robustness of NRI's scoring model. The last, fourth chapter presents our conclusions.

2 DEA Without Explicit Inputs and Common Weight Analysis

The first full-fledged DEA model was developed by Charnes, Cooper and Rhodes [5], which was followed by countless model versions and extensions by various authors. For each sub-pillar in the Network Readiness Index (NRI), the ideal value is “as high as possible”, thus all of them should be treated as output criteria in the DEA models. As there are no criteria to be minimized, there are no input variables in our models. In the literature, such models are called DEA Without Explicit Inputs (DEA/WEI) and/or the DEA-type Composite Indicators (DEA/CI) method [6], [7].

The DEA/WEI model was first applied by Fernandez-Castro and Smith to a real-world problem [8] and then by Despotis [9] and Liu et al. [10]. Due to the “shape” of the model, it was also used to create composite indicators, as mentioned above.

The basic model of DEA may be unsuitable for establishing an unambiguous ranking (strict total order) of countries, which is our goal now, because the DEA efficiencies of several decision making units (DMUs, in our case countries) can reach the maximum, i.e. value one, which means that these countries would be tied in the ranking. However, we may also face the problem that each DMU achieves efficiencies that can be calculated with different weights. Therefore, a method must be found to evaluate all possible decision making units with equal weights.

One of the first Common Weight Analysis (CWA) models was proposed by Podinovski and Athanassopoulos [11]. Their model is called the Maximin DEA model, where we first find the decision making unit that gives the lowest efficiency for a given weight vector, and then look for the weight vector for which we maximize this minimum. The method got its name from this procedure.

To solve this problem, a DEA procedure must be found for which all possible DMUs are evaluated with equal weights. This method is called the common weights procedure. The simplest form of the method was proposed by Liu and Peng [12]. The model traces the search for common weights to solve a linear programming problem whose constraint conditions contain linear constraints that include efficiency.

Finally, the third DEA model using common weights determines weights using compromise programming. This procedure was proposed by Kao and Hung [13]. The objective function here is a distance function, which can be Manhattan, Euclidean, or Chebyshev distance functions. To do this, however, an anti-ideal or ideal point must be defined in the space of DMUs’ efficiency. Anti-ideal efficiency can be e.g. zero efficiency, while the ideal may be the previously defined DEA efficiencies, or the efficiency to be achieved for each DMU. Only the two ideal efficiencies are used in this paper. For the ideal point, find the weight at which the ideal point is closest to the set of weights. For anti-ideal points, we look for the furthest efficiency. Vector $\boldsymbol{E^*}$ comprises the DEA efficiencies, which show the best efficiencies for each DMU.

The vector of possible weights of the DEA model can be determined by the system of equations (1) to (2). Inequalities (1) shows the upper limit of DEA efficiency, i.e. one, while inequality (2) defines the non-negativity of weights. The number of decision-making units is p, and vector $\mathbf{y}_j$ comprises the values of the jth decision making unit, in this case country. The $\mathbf{y}_j$ vectors can be summarized in the $\mathbf{Y}$ matrix. Vector $\mathbf{u}$ comprises the DEA weights. The DEA/WEI weights are equal to vector $\mathbf{u}\cdot\mathbf{Y}$.

$$\boldsymbol{u}\cdot \boldsymbol{y}_j \leq 1;\ j = 1,2,\ldots,p. \tag{1}$$

$$\boldsymbol{u} \geq \boldsymbol{0}. \tag{2}$$

The objective functions of possible DEA models are listed in Table 2.

Table 2. The DEA models used to determine the optimal weights.

DEA Models	Objective Functions ($F_i(\mathbf{u})$)	Literature
Maximin model	$F_1(\mathbf{u}) = \min\limits_{1 \leq j \leq p} \mathbf{u} \cdot \boldsymbol{y_j} \to \max$	[11]
Common Weight Analysis	$F_2(\mathbf{u}) = \mathbf{u}\cdot\mathbf{Y}\cdot\mathbf{1} \to \max$	[12]
Compromise Programming with ideal efficiency **1**	$F_3(\mathbf{u}) = d_2(\mathbf{u}\cdot\mathbf{Y};\ \mathbf{1}) \to \min$ $F_4(\mathbf{u}) = d_{+\infty}(\mathbf{u}\cdot\mathbf{Y};\ \mathbf{1}) \to \min$	[13]
Compromise Programming with ideal efficiency ***E****	$F_5(\mathbf{u}) = d_2(\mathbf{u}\cdot\mathbf{Y};\ \boldsymbol{E^*}) \to \min$ $F_6(\mathbf{u}) = d_{+\infty}(\mathbf{u}\cdot\mathbf{Y};\ \boldsymbol{E^*}) \to \min$	[14]

Based on our calculations, we establish six rankings. This is because the result of the Common Weight Analysis for Manhattan distances gives the same result for both ideal vectors, i.e., the DEA efficiency $\boldsymbol{E^*}$ and the summation vector **1** as well [14]. The distance functions of compromise programming are as follows:

Euclidean distance ($k = 2$): $d_2(\boldsymbol{u}\cdot\mathbf{Y};\ \boldsymbol{E}) = \sqrt{\sum_{j=1}^{p}(\boldsymbol{u}\cdot\boldsymbol{y}_j - E_j)^2}$,

Chebyshev distance ($k = +\infty$): $d_{+\infty}(\boldsymbol{u}\cdot\mathbf{Y};\ \boldsymbol{E}) = \max\limits_{1\leq j\leq p}\left|\boldsymbol{u}\cdot\boldsymbol{y}_j - E_j\right|$,

where vector $\boldsymbol{E}$ is a possible ideal efficiency vector, which can be equal to efficiency vectors $\boldsymbol{E^*}$ or **1**.

3 Ranking with the DEA/WEI and CWA methods

Before presenting the results, we show that for the DEA / WEI model, the Manhattan distance gives the same result for all ideal points, so the DEA efficiency and maximum efficiency vectors are the same. Therefore, it is sufficient to determine the efficiency obtained by the common weight analysis method. Since the sum of the given efficiencies does not depend on the weights, it is sufficient to minimize the expression $-\mathbf{u}\cdot\mathbf{Y}\cdot\mathbf{1}$, which means that the minus of the linear function should be maximized. This also means that we got the CWA model back. Therefore, minimizing the Manhattan distance in the DEA / WEI models also leads to the optimization problem of the CWA model.

The mathematical form of the six common weights problems can be described as follows:

$$\boldsymbol{u} \cdot \boldsymbol{y}_j \leq 1;\ j = 1,2,...,p. \tag{1'}$$

$$\boldsymbol{u} \geq \boldsymbol{0}. \tag{2'}$$

$$F_i(\boldsymbol{u}) \rightarrow min/max,\ i = 1,2,...,6. \tag{3'}$$

Problems 1, 2 should be maximized, while problems 3, 4, 5 and 6 should be minimized due to distance. The analytical form of the functions is given in Table 2.

Consider the linear relationship between the efficiencies determined by the common weights methods and the efficiency of DEA. Two types of correlations can be considered.

Table 3 shows the Pearson correlation coefficients. Since these efficiencies are continuous variables, Pearson correlation can be used. The correlation coefficients all have a value close to 1, indicating a strong linear association in all cases. The question can also be asked whether it is worthwhile to calculate efficiencies without defining DEA efficiencies. If there are *p* DMUs (in our case countries), then if DEA efficiency is considered to be an ideal virtual DMU, then these efficiencies must be determined first, which means solving *p* linear programming problems. Determining common weights is a solution to an additional LP. Conversely, if only the maximum available efficiency, i.e. efficiency one, distance from a possible efficiency is determined, then only a single linear programming problem needs to be solved.

Table 3. Pearson correlation between DEA efficiencies and common weights.

Pearson	**Manhattan**	**Euclidean**	**Chebysev**
DEA	0.937*	0.946	0.960
One	0.943	0.932	0.943

*Instead of the Manhattan distance, the weights of the Maximin model are included

According to our results (presented in Table 3), the optimal solution of the Chebyshev distance by minimizing the distance from vector **1** shows a correlation of 0.943 from the DEA distance, which can be considered strong. However, the two different computational efficiencies do not show significantly different results. This may suggest that it is not necessary to determine all *p* DEA efficiencies, which can lead to time and cost savings.

Table 4. Kendall tau-b correlation between DEA efficiencies and common weights.

Pearson	Manhattan	Euclidean	Chebysev
DEA	0.798*	0.806	0.831
One	0.788	0.784	0.807

*Instead of the Manhattan distance, the weights of the Maximin model are included

Kendall's tau-b correlation shows the correlation between rankings. This correlation coefficient indicates a strong stochastic correlation when it is higher than 0.7 and close to 1, which is the case for all values in Table 4, calculated from our NRI data set. The results obtained with the Chebyshev distance provides the strongest linear relationship for both ideal efficiency vectors, with correlation coefficients of 0.831 (DEA) and 0.807 (vector **1**) respectively.

4 Conclusions

In this paper, we demonstrate how the DEA/WEI and DEA/CWA method can be used to provide a viable framework for ranking the CIS countries and the Russian Federation in the absence of explicit input criteria and predetermined weights that are required by the classical DEA and with pre-determined weights for the original NRI scoring model. These methods can eliminate the need for a pre-defined weighting system used by the original composite index, replacing them with an intrinsic one based on the statistical properties of the data set.

According to our rankings, the Baltic countries (Estonia, Lithuania, Latvia) demonstrate respectable results in digital development relative to the Slavic, Caucasian and Central Asian countries of the CIS. Among CIS countries, Russia is ranked fourth and Tajikistan has the lowest ranking. Among Central Asian countries, Kazakhstan has the highest rank when the countries are ranked according to their DEA efficiencies.

In future studies, we plan to examine the sub-pillars of NRI in more detail. A number of statistical methods, such as correlational analysis, principal components analysis and cluster analysis can be used to investigate the statistical properties of the data set.

References

1. Vial, G.: Understanding digital transformation: A review and a research agenda. The Journal of Strategic Information Systems 28(2), 118–144 (2019).
2. Bánhidi, Z., Dobos, I., Nemeslaki, A.: Comparative Analysis of the Development of the Digital Economy in Russia and EU Measured with DEA and Using Dimensions of DESI. Вестник Санкт-Петербургского университета Экономика/St Petersburg University Journal of Economic Studies 35(4), 588–604 (2019).
3. Bánhidi, Z., Dobos. I., Tokmergenova, M.: Russia’s Place Vis-à-Vis the EU28 Countries in Digital Development: A Ranking Using DEA-Type Composite Indicators and the TOPSIS Method. In: Herberger, T. A., Dötsch, J. J. (eds.) Digitalization. Digital Transformation and Sustainability in the Global Economy, pp. 135–146. Springer (2021).

4. Portulans Institute: Network Readiness Index 2020. https://networkreadinessindex.org, last accessed 2022/02/02.
5. Charnes, A., Cooper, W. W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2(6), 429–444 (1978).
6. Cherchye, L., Moesen, W., Rogge, N., Van Puyenbroeck, T., Saisana, M., Saltelli, A., Liska, R., Tarantola, S.: Creating composite indicators with DEA and robustness analysis: the case of the Technology Achievement Index. Journal of the Operational Research Society 59(2), 239–251 (2008).
7. Dobos, I., Vörösmarty, G.: Green supplier selection and evaluation using DEA-type composite indicators. International Journal of Production Economics 157, 273–278 (2014).
8. Fernandez-Castro, A., Smith, P.: Towards a general non-parametric model of corporate performance. Omega 22(3), 237–249 (1994).
9. Despotis, D. K.: A reassessment of the human development index via data envelopment analysis. Journal of the Operational Research Society 56(8), 969–980 (2005).
10. Liu, W. B., Zhang, D. Q., Meng, W., Li, X. X., Xu, F.: A study of DEA models without explicit inputs. Omega 39(5), 472–480 (2011).
11. Podinovski, V. V., Athanassopoulos, A. D.: Assessing the relative efficiency of decision making units using DEA models with weight restrictions. Journal of the Operational Research Society 49(5), 500–508 (1998).
12. Liu, F. H. F., Peng, H. H.: Ranking of units on the DEA frontier with common weights. Computers & Operations Research 35(5), 1624–1637 (2008).
13. Kao, C., Hung, H.: Data envelopment analysis with common weights: the compromise solution approach. Journal of the Operational Research Society 56(10), 1196–1203 (2005).
14. Toloo, M.: Data Envelopment Analysis with Selected Models and Applications. SAEI, vol. 30. Ostrava: VŠB-TU Ostrava (2014).

On the Non-Emptiness of the Core of a Cooperative Fuzzy Game

David Bartl[1]

Department of Informatics and Mathematics, School of Business Administration in Karviná, Silesian University in Opava, Univerzitní náměstí 1934/3, 733 40 Karviná, Czechia
bartl@opf.slu.cz, david.bartl@post.cz

Abstract. We introduce the concept of a fuzzy coalition structure on a finite set of players. Then, we propose a new model of a cooperative fuzzy game with transferable utility: an existing coalition is assumed to endeavour in a branch of industry, and a deviation of a new coalition from the coalition structure is seen as an opportunity of the coalition. Based on these premisses, we introduce the concept of the core of the cooperative fuzzy TU-game with respect to a general fuzzy coalition structure. Finally, we define the concept of balancedness and formulate a generalization of the Bondareva-Shapley Theorem.

Keywords: Cooperative fuzzy TU-game · Core · Balanced game · Bondareva-Shapley theorem.

1 Introduction

Consider a classical cooperative game of n players with transferable utility. The *coalition* is any subset of the set $N = \{1, 2, \dots, n\}$ of the players, and the potency set $\mathcal{P}(N) = \{\, K : K \subseteq N \,\}$ of the set N is the collection of all coalitions $K \subseteq N$ that can potentially emerge. Finally, if a coalition $K \subseteq N$ emerges, then it will achieve its total profit of $v(K)$ units of some transferable utility (e.g. money); it is assumed that $v(\emptyset) = 0$. In other words, the cooperative game is given by its *coalition function*, which is a mapping $v \colon \mathcal{P}(N) \to \mathbb{R}$ such that $v(\emptyset) = 0$.

The *coalition structure* is any partition of the set N of the players; that is, the coalition structure is any collection $\mathcal{S} = \{S_1, S_2, \dots, S_r\}$ of coalitions such that $\bigcup_{p=1}^{r} S_p = N$ and $S_p \cap S_q = \emptyset$ whenever $p \neq q$ for $p, q = 1, 2, \dots, n$, and also $\emptyset \notin \mathcal{S}$.

Assume that a coalition structure $\mathcal{S} = \{S_1, S_2, \dots, S_r\}$ has crystallized. It means that the coalitions $S_1, S_2, \dots, S_r$ have emerged, they exist now, and they will achieve the profits $v(S_1)$, $v(S_2)$, …, $v(S_r)$, respectively. Now, the purpose is that the players within each coalition $S_1, S_2, \dots, S_r$ divide the total profit of their coalition among themselves. The division of the profit among the players is described by the payoff vector.

The *payoff vector* is any vector $\boldsymbol{a} = (a_i)_{i=1}^{n} \in \mathbb{R}^n$, where a_i is the profit apportioned to the i-th player for $i = 1, 2, \dots, n$. It is usual to require that the payoff vector belongs to a certain solution concept of the cooperative game. Informally speaking, the solution concept is a mapping that assigns a certain set of payoff vectors (i.e. a subset of $\mathbb{R}^n$) to the coalition function $v\colon \mathcal{P}(N) \to \mathbb{R}$ and to the coalition structure $\mathcal{S} = \{S_1, S_2, \ldots, S_r\}$. The core [8, 6, 7] is an example of the solution concept.

The *core* of the cooperative game with transferable utility (TU-game) given by the coalition function v with respect to the coalition structure $\mathcal{S}$ is the set

$$\mathcal{C} = \left\{ \boldsymbol{a} \in \mathbb{R}^n : \sum_{i \in S} a_i = v(S) \text{ for } S \in \mathcal{S} \text{ and } \sum_{i \in K} a_i \geq v(K) \text{ for } K \in \mathcal{P}(N) \setminus \mathcal{S} \right\},$$

see [1]. In words, the core is the set of all the payoff vectors $\boldsymbol{a} \in \mathbb{R}^n$ that satisfy the conditions of feasibility ($\sum_{i \in S} a_i \leq v(S)$ for $S \in \mathcal{S}$), efficiency or group rationality ($\sum_{i \in S} a_i \geq v(S)$ for $S \in \mathcal{S}$), and group stability ($\sum_{i \in K} a_i \geq v(K)$ for $K \in \mathcal{P}(N) \setminus \mathcal{S}$). Now, the key question is whether the core is non-empty.

The next classical result provides an answer to the question:

Bondareva-Shapley Theorem [3, 9]**.** *The core $\mathcal{C}$ of the cooperative TU-game given by the coalition function v with respect to the coalition structure $\mathcal{S} = \{N\}$ is non-empty if and only if the game is balanced.*

As we can see, the classical Bondareva-Shapley Theorem provides the answer in the special case when the coalition structure consists of the grand coalition ($\mathcal{S} = \{N\}$) only. We ask whether we can define the concept of balancedness with respect to a general coalition structure $\mathcal{S} = \{S_1, S_2, \ldots, S_r\}$ and prove the respective generalization of the Bondareva-Shapley Theorem. Regarding the generalization in the case of cooperative crisp TU-games, see [2]. Now, our purpose is to extend the results further to the case of cooperative fuzzy TU-games.

2 The core and balancedness of fuzzy TU-games

Consider again a cooperative game of n players with transferable utility. Now, the *fuzzy coalition* is any fuzzy subset $\tilde{K}$ of the set $N = \{1, 2, \ldots, n\}$ of the players; we denote this fact by writing $\tilde{K} \,\tilde{\subseteq}\, N$. Recall that any fuzzy subset $\tilde{K} \,\tilde{\subseteq}\, N$ is given by its *membership vector* $\boldsymbol{\kappa} \in [0,1]^N$, which is here understood as a row vector $\boldsymbol{\kappa} = (\kappa_1 \;\; \kappa_2 \;\; \ldots \;\; \kappa_n)$ with $0 \leq \kappa_i \leq 1$ for $i \in N$. Notice that if the membership vector is restricted so that $\boldsymbol{\kappa} \in \{0,1\}^N$; that is, $\kappa_i \in \{0,1\}$ for $i \in N$, then it corresponds to the crisp coalition $K \subseteq N$, with $i \in K$ if and only if $\kappa_i = 1$ for $i \in N$. The membership vector corresponding to the empty coalition $\emptyset$ and to the grand coalition N is $\boldsymbol{\chi}^{\emptyset}$ and $\boldsymbol{\chi}^N$, with $\chi_i^{\emptyset} = 0$ and $\chi_i^N = 1$, respectively, for $i \in N$.

The collection $\tilde{\mathcal{P}}(N) = \{\, \tilde{K} : \tilde{K} \,\tilde{\subseteq}\, N \,\}$ of all fuzzy subsets of the set N contains all the fuzzy coalitions $\tilde{K} \,\tilde{\subseteq}\, N$ that can potentially emerge. This collection is identified with the aforementioned set $[0,1]^N$ of all the membership vectors $\boldsymbol{\kappa}$.

The *fuzzy coalition structure* is any indexed collection $\tilde{\mathcal{S}} = (\tilde{S}_p)_{p \in \mathcal{R}}$ of fuzzy coalitions $\tilde{S}_p \,\tilde{\subseteq}\, N$ with membership vectors $\boldsymbol{\sigma}_p \in [0,1]^N$ for $p \in \mathcal{R}$, where $\mathcal{R}$ is an index set, such that $\sum_{p \in \mathcal{R}} \boldsymbol{\sigma}_p = \boldsymbol{\chi}^N$ and $\boldsymbol{\sigma}_p \neq \boldsymbol{\chi}^{\emptyset}$ for $p \in \mathcal{R}$. Notice that, even though the set N of the players is finite, the index set $\mathcal{R}$ may be infinite and a fuzzy coalition $\tilde{S} \tilde{\subseteq} N$ may be present several times in the fuzzy coalition structure $\tilde{\mathcal{S}}$; that is, we may have $\tilde{S}_p = \tilde{S}_q$ for distinct $p, q \in \mathcal{R}$. Moreover, if the membership vectors are restricted so that $\boldsymbol{\sigma}_p \in \{0,1\}^N$, then the index set $\mathcal{R}$ is finite, let $\mathcal{R} = \{1, 2, \ldots, r\}$, say, and the fuzzy coalition structure $\tilde{\mathcal{S}}$ reduces to the crisp coalition structure $\mathcal{S} = \{S_1, S_2, \ldots, S_r\}$ with $S_p = \{\, i \in N : (\boldsymbol{\sigma}_p)_i = 1 \,\}$ for $p = 1, 2, \ldots, r$. We obviously have $\bigcup_{p=1}^{r} S_p = N$ and $S_p \cap S_q \neq \emptyset$ iff $p = q$ for $p, q = 1, 2, \ldots, r$.

Assume that a fuzzy coalition structure $\tilde{\mathcal{S}} = (\tilde{S}_p)_{p \in \mathcal{R}}$ has crystallized. It means that the fuzzy coalitions $\tilde{S}_p \tilde{\subseteq} N$, for $p \in \mathcal{R}$, have emerged and exist. We interpret the fact that $0 \leq (\boldsymbol{\sigma}_p)_i \leq 1$ for $p \in \mathcal{R}$ so that the player i is involved in the coalition $\tilde{S}_p$ for "part-time job" in general; that is, the player is not involved in the coalition at all if $(\boldsymbol{\sigma}_p)_i = 0$, the player is involved for "full-time job" if $(\boldsymbol{\sigma}_p)_i = 1$, and the player is involved for "part-time job" in the remaining cases. Moreover, we understand the fact that formally the same coalition $\tilde{S}_p = \tilde{S}_q$, for $p, q \in \mathcal{R}$ with $p \neq q$, can be present several times in the coalition structure $\tilde{\mathcal{S}}$ so that the coalitions $\tilde{S}_p$ and $\tilde{S}_q$ are actually *distinct* and they endeavour in different branches of industry in general. Given this interpretation, it follows that the total profits achieved by the distinct coalitions $\tilde{S}_p$ and $\tilde{S}_q$, both of which exist at the same time, may be distinct too in general.

Based on these considerations, we propose a new model of cooperative fuzzy game with transferable utility. We propose that the cooperative fuzzy game is given by a pair of functions $V \colon \mathcal{R} \to \mathbb{R}$ and $v \colon [0,1]^N \to \mathbb{R}$ with $v(\boldsymbol{\chi}^{\emptyset}) = 0$. The first function V assigns the total profit of $V(p)$ units of some transferable utility to any fuzzy coalition $\tilde{S}_p$ of the present fuzzy coalition structure $\tilde{\mathcal{S}}$ for $p \in \mathcal{R}$; that is, the total profit $V(p)$ is assigned to any coalition $\tilde{S}_p$ that presently exists and is active and endeavouring in some branch of industry. (This approach loosely resembles that of Thrall and Lucas [10].) Now, a new fuzzy coalition $\tilde{K} \tilde{\subseteq} N$ may take the opportunity and form, leave the present coalition structure $\tilde{\mathcal{S}}$, and start to endeavour in a new branch of industry. This is the reason why we consider the second function v. It assigns the total profit of $v(\boldsymbol{\kappa})$ units of the transferable utility to the fuzzy coalition $\tilde{K} \tilde{\subseteq} N$ that decides to take the opportunity and leave the present coalition structure $\tilde{\mathcal{S}}$.

(We remark that the above model can easily be adapted to include the case of restricted cooperation: Let $\mathcal{A} \subseteq [0,1]^N$ be the collection of the membership vectors that correspond to the feasible fuzzy coalitions. We then define the function v on the collection $\mathcal{A}$ only ($v \colon \mathcal{A} \to \mathbb{R}$) and adapt the below given considerations accordingly.)

Now, again, the purpose is that the players within each fuzzy coalition $\tilde{S}_p$ divide the total profit $V(p)$ of their coalition among themselves for $p \in \mathcal{R}$. The division of the profit will be described by the *payoff matrix* which is any matrix $\boldsymbol{A} \in \mathbb{R}^{N \times \mathcal{R}}$, where a_{ip} is the profit apportioned to player i in coalition $\tilde{S}_p$ for $i \in N$ and for $p \in \mathcal{R}$. Moreover, we set $a_{ip} := 0$ for $i \in N$ and for $p \in \mathcal{R}$ such that $(\boldsymbol{\sigma}_p)_i = 0$; that is, the player i is not involved in the fuzzy coalition $\tilde{S}_p$ at all. (The total profit of player i achieved via all the player's involvements in the coalitions is the row sum $\pi_i = \sum_{p \in \mathcal{R}} a_{ip}$ for $i \in N$.) Our purpose is to extend the classical concept of the core to the present setting. Thus, consider a payoff matrix $\boldsymbol{A} \in \mathbb{R}^{N \times \mathcal{R}}$. We agree that, if $\boldsymbol{A}$ belongs to the core, then the equations $\sum_{i \in N} a_{ip} = V(p)$, which express the feasibility and efficiency or group rationality, must hold for all $p \in \mathcal{R}$. Regarding the group stability, assume that a fuzzy coalition $\tilde{K} \tilde{\subseteq} N$ with membership vector $\boldsymbol{\kappa} \in [0,1]^N$ takes the opportunity and deviates from the present coalition structure $\tilde{\mathcal{S}}$. Then the coalition $\tilde{K}$ endeavouring in a new branch industry will achieve its total profit of $v(\boldsymbol{\kappa})$ units of the utility. We stipulate that each player $i \in N$ must have left some coalitions so that the sum of the players "part-time jobs" exceeds κ_i. Mathematically speaking, we stipulate that there exists an index subset $\mathcal{K} \subseteq \mathcal{R}$ such that $\sum_{p \in \mathcal{K}} \boldsymbol{\sigma}_p \geq \boldsymbol{\kappa}$. Though the index subset $\mathcal{K} \subseteq \mathcal{R}$ could be infinite in general, we shall assume that the index subset $\mathcal{K}$ is finite to obtain a simple definition of balancedness below. Then the inequalities which prevent the fuzzy coalition $\tilde{K} \tilde{\subseteq} N$ from the deviation from the coalition structure $\tilde{\mathcal{S}}$ are $\sum_{i \in N} \sum_{p \in \mathcal{K}} a_{ip} \geq v(\boldsymbol{\kappa})$ for every finite $\mathcal{K} \subseteq \mathcal{R}$ such that $\sum_{p \in \mathcal{K}} \boldsymbol{\sigma}_p \geq \boldsymbol{\kappa}$.

To conclude, we define the *core* of the cooperative fuzzy TU-game given by its fuzzy coalition structure $\tilde{\mathcal{S}} = (\tilde{S}_p)_{p \in \mathcal{R}}$, the coalition of this fuzzy coalition structure function $V \colon \mathcal{R} \to \mathbb{R}$ and the fuzzy coalition function $v \colon [0,1]^N \to \mathbb{R}$ with $v(\boldsymbol{\chi}^{\emptyset}) = 0$ to be the set

$$\begin{aligned} \mathcal{C} = \big\{ \boldsymbol{A} \in \mathbb{R}^{N \times \mathcal{R}} : (\boldsymbol{\sigma}_p)_i = 0 \implies a_{ip} = 0 \qquad & \text{for} \quad i \in N \text{ and for } p \in \mathcal{R}, \\ \textstyle\sum_{i \in N} a_{ip} = V(p) \qquad & \text{for} \quad p \in \mathcal{R}, \\ \textstyle\sum_{p \in \mathcal{K}} \sum_{i \in N} a_{ip} \geq v(\boldsymbol{\kappa}) \qquad & \text{for} \quad \boldsymbol{\kappa} \in [0,1]^N \text{ and} \\ & \text{for finite } \mathcal{K} \subseteq \mathcal{R} \text{ such that } \textstyle\sum_{p \in \mathcal{K}} \boldsymbol{\sigma}_p \geq \boldsymbol{\kappa} \big\} \end{aligned}$$

Notice that, if $\boldsymbol{A} \in \mathcal{C}$, then each of the variables a_{ip} is bounded below and above for $i \in N$ and for $p \in \mathcal{R}$. Indeed, if $i \in N$ and $p \in \mathcal{R}$ are such that $(\boldsymbol{\sigma}_p)_i = 0$, then $a_{ip} = 0$. Consider now $i \in N$ and $p \in \mathcal{R}$ are such that $(\boldsymbol{\sigma}_p)_i > 0$. Take the membership vector $\boldsymbol{\kappa} \in [0,1]^N$ such that $\kappa_i = (\boldsymbol{\sigma}_p)_i$ and $\kappa_j = 0$ for $j \in N \setminus \{i\}$. Then $a_{ip} \geq v(\boldsymbol{\kappa})$, which is a lower bound. Let $\underline{a}_{ip}$ be a lower bound of a_{ip} for $i \in N$ and for $p \in \mathcal{R}$. Consider again $i \in N$ and $p \in \mathcal{R}$ such that $(\boldsymbol{\sigma}_p)_i > 0$. We then have $a_{ip} + \sum_{j \in N \setminus \{i\}} \underline{a}_{jp} \leq \sum_{j \in N} a_{jp} = V(p)$, whence $a_{ip} \leq V(p) - \sum_{j \in N \setminus \{i\}} \underline{a}_{jp}$, which is an upper bound. Let $\overline{a}_{ip}$ be an upper bound of a_{ip} for $i \in N$ and for $p \in \mathcal{R}$.

Let us suppose wlog that $\underline{a}_{ip} \leq \overline{a}_{ip}$ for $i \in N$ and for $p \in \mathcal{R}$. (Should we have $\underline{a}_{ip} > \overline{a}_{ip}$, then let $\underline{a}_{ip} := \overline{a}_{ip}$, say.) Then the closed interval $[\underline{a}_{ip}, \overline{a}_{ip}]$, endowed with the usual Euclidean topology, is compact, therefore the product $\mathcal{X} = \prod_{i \in N} \prod_{p \in \mathcal{R}} [\underline{a}_{ip}, \overline{a}_{ip}]$, endowed with the product topology, is a compact topological space by Tychonoff's Theorem. Notice that the core $\mathcal{C} \subseteq \mathcal{X}$.

It is easy to see that the core $\mathcal{C}$ is non-empty if and only if the following system of linear inequalities, where a_{ip} are variables, has a solution:

$$\begin{aligned} \textstyle\sum_{i \in N, (\boldsymbol{\sigma}_p)_i > 0} a_{ip} &\leq V(p) \qquad \text{for} \quad p \in \mathcal{R}, \\ -\textstyle\sum_{i \in N, (\boldsymbol{\sigma}_p)_i > 0} a_{ip} &\leq -V(p) \qquad \text{for} \quad p \in \mathcal{R}, \end{aligned} \tag{1}$$

$$
\begin{aligned}
-\textstyle\sum_{p\in\mathcal{K}}\sum_{i\in N,(\boldsymbol{\sigma}_p)_i>0} a_{ip} \leq -v(\boldsymbol{\kappa}) \qquad &\text{for} \quad \boldsymbol{\kappa}\in[0,1]^N \ \text{and} \\
&\text{for finite } \mathcal{K}\subseteq\mathcal{R} \text{ such that } \textstyle\sum_{p\in\mathcal{K}}\boldsymbol{\sigma}_p \geq \boldsymbol{\kappa}.
\end{aligned}
\tag{2}
$$

Notice that there is a finite number of variables on the left-hand side of each inequality in (1)–(2). Moreover, it is easy to see that, for any finite subset $\mathcal{I}\subseteq N\times\mathcal{R}$ and for any constant $c\in\mathbb{R}$, the halfspace $F=\big\{\boldsymbol{A}\in\mathbb{R}^{N\times\mathcal{R}} : \sum_{(i,p)\in\mathcal{I}} a_{ip} \leq c\big\}$ is a closed set in the product topology of the space $\mathcal{X}$. It follows that the core $\mathcal{C}$ is the intersection of (possibly infinitely many) closed halfspaces. Since the space $\mathcal{X}$ is compact, we conclude that the core $\mathcal{C}$ is non-empty if and only if every finite subsystem of (1)–(2) has a solution; that is, for any natural numbers $r,s\in\mathbb{N}$, for any $p_1,p_2,\dots,p_r\in\mathcal{R}$, for any $\boldsymbol{\kappa}_1,\boldsymbol{\kappa}_2,\dots,\boldsymbol{\kappa}_s\in[0,1]^N$ and for any finite $\mathcal{K}_1,\mathcal{K}_2,\dots,\mathcal{K}_s\subseteq\mathcal{R}$ such that $\sum_{p\in\mathcal{K}_q}\boldsymbol{\sigma}_p\geq\boldsymbol{\kappa}_q$ for $q=1,\,2,\,\dots,\,s$, the following system of linear inequalities, where a_{ip} are variables, has a solution:

$$
\begin{aligned}
\textstyle\sum_{i\in N,(\boldsymbol{\sigma}_{p_\rho})_i>0} a_{ip} &\leq \ \ V(p_\rho) &&\text{for} \quad \rho=1,2,\dots,r,\\
-\textstyle\sum_{i\in N,(\boldsymbol{\sigma}_{p_\rho})_i>0} a_{ip} &\leq -V(p_\rho) &&\text{for} \quad \rho=1,2,\dots,r,\\
-\textstyle\sum_{p\in\mathcal{K}_q}\sum_{i\in N,(\boldsymbol{\sigma}_p)_i>0} a_{ip} &\leq -v(\boldsymbol{\kappa}_q) &&\text{for} \quad q=1,2,\dots,s.
\end{aligned}
\tag{3}
$$

The following result is useful:

Gale's Theorem of the alternative [4, 5]**.** *Let $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ be a matrix and let $\boldsymbol{b}\in\mathbb{R}^m$ be a vector. Then there exists a solution $\boldsymbol{x}\in\mathbb{R}^n$ to the system of linear inequalities*

$$\boldsymbol{A}\boldsymbol{x}\leq\boldsymbol{b} \tag{4}$$

if and only if

$$\forall\boldsymbol{\lambda}^{\mathrm{T}}\in\mathbb{R}^{1\times m},\ \boldsymbol{\lambda}^{\mathrm{T}}\geq\boldsymbol{0}^{\mathrm{T}}\colon\ \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{0}^{\mathrm{T}} \implies \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{b}\geq 0\,. \tag{5}$$

By identifying system (4) with (3), the condition (5) and some calculations yield the concept of balancedness of the cooperative fuzzy TU-game.

It will be useful to introduce the operation of rounding up. A number $\sigma\in[0,1]$ is rounded up as follows: we let $\lceil\sigma\rceil=0$ if $\sigma=0$, and $\lceil\sigma\rceil=1$ if $\sigma>0$. Given a row membership vector $\boldsymbol{\sigma}\in[0,1]^N$, the operation $\lceil\cdot\rceil$ is applied to the vector componentwise; that is, we have $\lceil\boldsymbol{\sigma}\rceil\in\{0,1\}^N$ and $\lceil\boldsymbol{\sigma}\rceil_i=0$ or $\lceil\boldsymbol{\sigma}\rceil_i=1$ if $\sigma_i=0$ or $\sigma_i>0$, respectively, for $i\in N$.

Recall that the fuzzy coalition structure $\tilde{\mathcal{S}}=(\tilde{S}_p)_{p\in\mathcal{R}}$ consists of fuzzy coalitions $\tilde{S}_p\mathrel{\tilde{\subseteq}}N$ with membership vectors $\boldsymbol{\sigma}_p\in[0,1]^N$ for $p\in\mathcal{R}$. We say that a collection $\{\tilde{K}_1,\tilde{K}_2,\dots,\tilde{K}_s\}$ of fuzzy coalitions $\tilde{K}_1,\tilde{K}_2,\dots,\tilde{K}_s\mathrel{\tilde{\subseteq}}N$ with membership vectors $\boldsymbol{\kappa}_1,\boldsymbol{\kappa}_2,\dots,\boldsymbol{\kappa}_s\in[0,1]^N$ along with a collection $\{\mathcal{K}_1,\mathcal{K}_2,\dots,\mathcal{K}_s\}$ of finite index sets $\mathcal{K}_1,\mathcal{K}_2,\dots,\mathcal{K}_s\subseteq\mathcal{R}$ such that $\sum_{p\in\mathcal{K}_q}\boldsymbol{\sigma}_p\geq\boldsymbol{\kappa}_q$ for $q=1,\,2,\,\dots,\,s$ is *balanced* with respect to the fuzzy coalition structure $\tilde{\mathcal{S}}=(\tilde{S}_p)_{p\in\mathcal{R}}$ if and only if

$$\sum_{q=1}^{s}\sum_{p\in\mathcal{K}_q}\lambda_q\lceil\boldsymbol{\sigma}_p\rceil=\sum_{\rho=1}^{r}\mu_{p_\rho}\lceil\boldsymbol{\sigma}_{p_\rho}\rceil$$

for some balancing weights $\lambda_1, \lambda_2, \ldots, \lambda_s \geq 0$, for some natural number $r \in \mathbb{N}$, for some indices $p_1, p_2, \ldots, p_r \in \mathcal{R}$, and for some $\mu_{p_1}, \mu_{p_2}, \ldots, \mu_{p_r} \geq 0$ such that $\mu_{p_1} + \mu_{p_2} + \cdots + \mu_{p_r} = 1$.

Finally, we say that the given cooperative fuzzy TU-game is *balanced* with respect to the fuzzy coalition structure $\tilde{\mathcal{S}} = (\tilde{S}_p)_{p\in\mathcal{R}}$ if and only if

$$\sum_{q=1}^{s} \sum_{p\in\mathcal{K}_q} \lambda_q v(\boldsymbol{\kappa}_q) \leq \sum_{\rho=1}^{r} \mu_{p_\rho} V(p_\rho)$$

for every balanced collection $\{\tilde{K}_1, \tilde{K}_2, \ldots, \tilde{K}_s\}$ of fuzzy coalitions along with the corresponding collection $\{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_s\}$ of the finite index sets.

By combining all the facts together, we come to the main result of this paper:

Bondareva-Shapley Theorem, generalized version. *Let a fuzzy cooperative TU-game; that is, a fuzzy coalition structure $\tilde{\mathcal{S}} = (\tilde{S}_p)_{p\in\mathcal{R}}$, a function $V\colon \mathcal{R} \to \mathbb{R}$ of the coalition of this fuzzy coalition structure and a fuzzy coalition function $v\colon [0,1]^N \to \mathbb{R}$ with $v(\boldsymbol{\chi}^{\emptyset}) = 0$ be given. Then the core* $\mathcal{C} = \big\{ \boldsymbol{A} \in \mathbb{R}^{N\times\mathcal{R}} : \sum_{i\in N} a_{ip} = V(p)$ for $p \in \mathcal{R}$, and $\sum_{p\in\mathcal{K}} \sum_{i\in N} a_{ip} \geq v(\boldsymbol{\kappa})$ for $\boldsymbol{\kappa} \in [0,1]^N$, and also $a_{ip} = 0$ if $(\boldsymbol{\sigma}_p)_i = 0$ for $i \in N$ and for $p \in \mathcal{R} \big\}$ *is non-empty if and only if the game is balanced.*

Acknowledgement

This work was supported by the Czech Science Foundation under grant number GAČR 21-03085S.

References

1. Aumann, R. J., Dreze, J. H.: Cooperative Games with Coalition Structures. International Journal of Game Theory **3**(4), 217–237 (1974)
2. Bartl, D.: On the Non-Emptiness of the Core of a Cooperative Game. In: Proceedings of the 25th International Conference on Mathematical Methods in Economics 2007, pp. 9-11. Ostrava: VŠB–Technical University of Ostrava. Faculty of Economics (2007) ISBN 978-80-248-1457-5
3. Bondareva, O. N.: Some Applications of Linear Programming Methods to the Theory of Cooperative Games (in Russian). Problemy Kybernetiki **10**, 119–139 (1963)
4. Fan, K.: On Systems of Linear Inequalities. In: Kuhn, H. W., Tucker, A. W. (eds.) Linear Inequalities and Related Systems, Annals of Mathematics Studies, vol. 38, pp. 99–156. Princeton: Princeton University Press (1956)
5. Gale, D.: The Theory of Linear Economic Models. New York: McGraw-Hill (1960)
6. Gillies, D. B.: Solutions to General Non-zero-sum Games. In: Tucker, A. W., Luce, R. D. (eds.) Contributions to the Theory of Games, Vol. IV, Annals of Mathematics Studies, vol. 38, pp. 47–85. Princeton: Princeton University Press (1959)
7. Kannai, Y.: The Core and Balancedness. In: Aumann, R. J., and Hart, S. (eds.) Handbook of Game Theory with Economic Applications, Vol. I, pp. 591–667. Amsterdam: North-Holland (1992)
8. Shapley, L. S.: Markets as Cooperative Games. RAND Corporation Paper P-629 (1955)
9. Shapley, L. S.: On Balanced Sets and Cores. Naval Research Logistics Quarterly **14**, 453–460 (1967)
10. Thrall, R. M., Lucas, W. F.: N-person games in partition function form. Naval Research Logistics Quarterly **10**, 281–298 (1963)

Dependence of supplier efficiency on the choice of input-output criteria in Data Envelopment Analysis (DEA)

Imre Dobos[1], Gyöngyi Vörösmarty[2]

[1] Budapest University of Technology and Economics, Budapest, Hungary
[2] Corvinus University of Budapest, Budapest, Hungary
vocal@p-graph.org

Abstract. When using Data Envelopment Analysis (DEA) to support management decisions the splitting of criteria into input and output variables is often incidental because the decision variables are difficult to interpret from a management point of view. This means that all criteria can be considered as input but also output variables if the criteria are interpreted correctly. In this presentation, we analyze three model versions of the Data Envelopment Analysis (DEA). In the first model we define the input and output variables together to be maximized, in the other two models we consider each variable as a criterion to be minimized/maximized. Accordingly, we examine a classical DEA model, a DEA model with explicit input and explicit output, respectively. The efficiency results of the models are presented in a supplier selection numerical example. For the calculations normalized data will be used.

Keywords: Supplier Selection, DEA Without Explicit Input (WEI), DEA Without Explicit Output (WEO), Common Weight Analysis (CWA).

1 Introduction

The Data Envelopment Analysis (DEA) model is a well-known tool for assessing efficiency. It is used to support economic decisions in many areas. One of the features of the model is that it defines so-called input and output criteria. In their original work, Charnes, Cooper and Rhodes [1] give the example of a school to interpret these: for the inputs to training, they give the working hours of teachers as an example, while for the outputs they give the competencies of students. One group is therefore the set of resources that are used (input criteria), and performance improves if less is used, while the other group is the result achieved (output criteria), the higher the indicators the better. Although a very large number of articles have been published on the development and application of DEA [2], relatively few publications have addressed the nature of the data used. These have mainly dealt with the issue of undesired outputs [3] and the nature of the data (e.g. negativity [4]). However, this problem is less addressed in the literature. Our article will examine how the three known model variants of DEA yield results, and how the definition of inputs and outputs affects the final result. The results of three examples of supplier evaluation will be compared.

2 Three basic DEA models

The DEA model was initiated by Charnes, Cooper and Rhodes [1]. After the model was set up, countless model versions were developed. When using different models of DEA, for each of the criteria, the given criterion is appropriate if it is as high as possible, so the criteria can be considered as output. As there are no criteria to be minimized, no input criteria exist. The latter model is called in the literature DEA model Without Explicit Inputs (DEA/WEI) and/or the DEA-type Composite Indicators (DEA/CI) method, as they correspond to the above description [5], [6].

The DEA model without explicit input was first applied by Fernandez-Castro and Smith to real-world problem [7] and then by Despotis [8] and Liu et al. [9]. Due to the shape of the model, it was also used to create composite indicators, as mentioned above.

The DEA model without explicit output was investigated by Toloo [10]. The DEA efficiencies in this case form a fractional programming model with one in the counter. This also means that the denominator will be the value to be determined, the reciprocal of which is efficiency.

We first present the basic model of DEA shown by formulas (1) to (3). In equation (1) the output criteria are in the numerator and the input criteria are in the denominator. Let number m be the number of outputs and number n be the number of inputs. It means that $m + n$ criteria are examined in this DEA model. The output criteria are maximized, while the inputs are minimized. If the output variable is to be minimized, the input should be maximized, we multiply the criterion by minus one and perform a data alteration.

$$\boldsymbol{u}\cdot\boldsymbol{y}_j \,/\, \boldsymbol{v}\cdot\boldsymbol{x}_j \leq 1;\ j = 1,2,\dots,p. \tag{1}$$

$$\boldsymbol{u} \geq \boldsymbol{0},\ \boldsymbol{v} \geq \boldsymbol{0}. \tag{2}$$

$$\boldsymbol{u}\cdot\boldsymbol{y}_i \,/\, \boldsymbol{v}\cdot\boldsymbol{x}_i \rightarrow max. \tag{3}$$

If we have to maximize all the criteria, we have to solve the (1_{WEI}) to (3_{WEI}) model. In this case, since there is no input, there is no denominator, or its value is one. Then we have to solve a smooth linear programming (LP) problem, which can be solved with a simple simplex method, even with MS Excel.

$$\boldsymbol{u}\cdot\boldsymbol{y}_j \leq 1;\ j = 1,2,\dots,p. \tag{1_{WEI}}$$

$$\boldsymbol{u} \geq \boldsymbol{0}. \tag{2_{WEI}}$$

$$\boldsymbol{u}\cdot\boldsymbol{y}_i \rightarrow max. \tag{3_{WEI}}$$

If we have to minimize all the criteria, we have to solve the (1_{WEO}) to (3_{WEO}) model. In this case, since there is no output, there is no numerator, or its value is one. Since there is no numerator, we can calculate the reciprocal of the denominator, which is shown by the formula (1_{WEO}). Then we have to solve a simple linear programming problem, which can be solved as before with MS Excel.

$$\boldsymbol{v}\cdot\boldsymbol{x}_j \geq 1;\ j = 1,2,\ldots,p. \tag{1_{WEO}}$$

$$\boldsymbol{v} \geq \boldsymbol{0}. \tag{2_{WEO}}$$

$$\boldsymbol{v}\cdot\boldsymbol{x}_i \rightarrow min. \tag{3_{WEO}}$$

The three models now described are solved in the next section, defining the DEA efficiencies.

3 Determining the DEA efficiency solving the three LP problems

The initial data used for supplier selection are shown in Table 1 [11]. We assume that the criteria can be divided into input and output variables. Among the input, i.e. management criteria, we need to maximize quality, while among the green variables, we need to minimize CO_2 emissions. This also means that to solve the (1) to (3) model, we have to multiply the two variables by minus one to satisfy the conditions of the DEA model.

Table 1. Basic data.

Basic Data	**Management criteria**			**Environmental criteria**	
Min/Max	MIN	MAX	MIN	MAX	MIN
Supplier	**Lead time** (Day)	**Quality** (%)	**Price** ($)	**Reusability** (%)	**CO_2 Emission** (g)
1	2	80	2	70	30
2	1	70	3	50	10
3	3	90	5	60	15
4	1.5	85	1	40	20
5	2.5	75	2.5	65	35
6	2	95	4	90	25
7	3	80	1.5	75	15
8	1.5	85	3.5	85	20
9	1	70	3.5	55	10
10	2.5	75	4	45	10
11	3.5	90	2.5	80	25
12	2	65	1.5	50	20
13	3	85	3	75	15
14	1.5	70	4.5	85	20
15	1	65	2	75	15

The data in Table 1 are normalized to the three models. This procedure was used by Gnaldi and Ranalli [12]. The method is the following

Criteria for maximization: $n_{ij} = \frac{r_{ij}-r_{min}}{r_{max}-r_{min}}$,

Criteria for minimization: $n_{ij} = \frac{r_{max}-r_{ij}}{r_{max}-r_{min}}$.

Tables 3-5 in the Appendix summarize the transformed data used for the current DEA model.

Table 2. DEA efficiencies of the three models.

Supplier	I-O	WEI	WEO
1	0.648	0.917	0.800
2	0.079	0.500	0.459
3	1.000	1.000	1.000
4	0.266	0.667	0.528
5	1.000	1.000	1.000
6	1.000	1.000	1.000
7	0.461	0.853	0.603
8	0.535	0.900	0.686
9	0.126	0.636	0.500
10	0.089	0.750	0.614
11	1.000	1.000	1.000
12	0.278	0.500	0.504
13	0.673	0.919	0.825
14	0.511	1.000	1.000
15	0.239	0.700	0.459

Table 2 shows the DEA efficiencies obtained for the three models. It is clear that the DEA model without input, resp. output, it gave a very similar solution. The same suppliers will be effective in both cases. In the original DEA model, only the 14th supplier will not be efficient, while the other four supplier, i.e. the 3rd, 5th, 6th, and 11th, will remain. It can also be seen the other DEA efficiencies are lower for the original model than for the two models.

4 Conclusion

The DEA model is a well-known tool for assessing efficiency, which splits criteria into input and output criteria. The model is often recommended in the literature for supplier evaluation. However, input and output criteria are often difficult to interpret in these management decisions. Therefore, our article has examined the criteria groups in three cases. Our examples show that the results are similar but not identical. Further interpretation and analysis of the results is an important further task.

References

1. Charnes, A., Cooper, W. W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2(6), 429–444 (1978).
2. Emrouznejad, A., Yang, G. L.: A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016. Socio-economic planning sciences 61, 4–8 (2018).
3. Liu, W. B., Meng, W., Li, X. X., Zhang, D. Q.: DEA models with undesirable inputs and outputs. Annals of Operations Research 173(1), 177–194 (2010).
4. Pastor, J. T., Ruiz, J. L.: Variables with negative values in DEA. In: Zhu, J., Cook, W. D. (eds.) Modeling data irregularities and structural complexities in data envelopment analysis, pp. 63–84. Springer (2007).
5. Cherchye, L., Moesen, W., Rogge, N., Van Puyenbroeck, T., Saisana, M., Saltelli, A., Liska, R., Tarantola, S.: Creating composite indicators with DEA and robustness analysis: the case of the Technology Achievement Index. Journal of the Operational Research Society 59(2), 239–251 (2008).
6. Dobos, I., Vörösmarty, G.: Green supplier selection and evaluation using DEA-type composite indicators. International Journal of Production Economics 157, 273–278 (2014).
7. Fernandez-Castro, A., Smith, P.: Towards a general non-parametric model of corporate performance. Omega 22(3), 237–249 (1994).
8. Despotis, D. K.: A reassessment of the human development index via data envelopment analysis. Journal of the Operational Research Society 56(8), 969–980 (2005).
9. Liu, W. B., Zhang, D. Q., Meng, W., Li, X. X., Xu, F.: A study of DEA models without explicit inputs. Omega 39(5), 472–480 (2011).
10. Toloo, M.: Data Envelopment Analysis with Selected Models and Applications. SAEI, vol. 30. Ostrava: VŠB-TU Ostrava (2014).
11. Vörösmarty, G., Dobos, I.: Supplier evaluation with environmental aspects and common DEA weights. Periodica Polytechnica Social and Management Sciences 27(1), 17–25 (2019).
12. Gnaldi, M., Ranalli, M. G.: Measuring university performance by means of composite indicators: A robustness analysis of the composite measure used for the benchmark of Italian universities. Social Indicators Research 129(2), 659–675 (2016).

Appendix

Table 3. Data for (1) – (3).

Basic Data	**Management criteria**			**Environmental criteria**	
Min/Max	MIN	MIN	MIN	MAX	MAX
Supplier	**Lead time (Day)**	**Quality (%)**	**Price ($)**	**Reusability (%)**	**CO$_2$ Emission (g)**
1	0.600	0.500	0.750	0.600	0.800
2	1.000	0.833	0.500	0.200	0.000
3	0.200	0.167	0.000	0.400	0.200
4	0.800	0.333	1.000	0.000	0.400
5	0.400	0.667	0.625	0.500	1.000
6	0.600	0.000	0.250	1.000	0.600
7	0.200	0.500	0.875	0.700	0.200
8	0.800	0.333	0.375	0.900	0.400
9	1.000	0.833	0.375	0.300	0.000
10	0.400	0.667	0.250	0.100	0.000
11	0.000	0.167	0.625	0.800	0.600
12	0.600	1.000	0.875	0.200	0.400
13	0.200	0.333	0.500	0.700	0.200
14	0.800	0.833	0.125	0.900	0.400
15	1.000	1.000	0.750	0.700	0.200

Table 4. Data for (1$_{\mathrm{WEI}}$) – (3$_{\mathrm{WEI}}$).

Basic Data	**Management criteria**			**Environmental criteria**	
Min/Max	MAX	MAX	MAX	MAX	MAX
Supplier	**Lead time (Day)**	**Quality (%)**	**Price ($)**	**Reusability (%)**	**CO$_2$ Emission (g)**
1	0.400	0.500	0.250	0.400	0.200
2	0.000	0.167	0.500	0.800	1.000
3	0.800	0.833	1.000	0.600	0.800
4	0.200	0.667	0.000	1.000	0.600
5	0.600	0.333	0.375	0.500	0.000
6	0.400	1.000	0.750	0.000	0.400
7	0.800	0.500	0.125	0.300	0.800
8	0.200	0.667	0.625	0.100	0.600
9	0.000	0.167	0.625	0.700	1.000
10	0.600	0.333	0.750	0.900	1.000
11	1.000	0.833	0.375	0.200	0.400

12	0.400	0.000	0.125	0.800	0.600
13	0.800	0.667	0.500	0.300	0.800
14	0.200	0.167	0.875	0.100	0.600
15	0.000	0.000	0.250	0.300	0.800

Table 5. Data for (1WEO) – (3WEO).

Basic Data	**Management criteria**			**Environmental criteria**	
Min/Max	MIN	MIN	MIN	MIN	MIN
Supplier	**Lead time (Day)**	**Quality (%)**	**Price ($)**	**Reusability (%)**	**CO_2 Emission (g)**
1	0.400	0.500	0.250	0.600	0.800
2	0.000	0.167	0.500	0.200	0.000
3	0.800	0.833	1.000	0.400	0.200
4	0.200	0.667	0.000	0.000	0.400
5	0.600	0.333	0.375	0.500	1.000
6	0.400	1.000	0.750	1.000	0.600
7	0.800	0.500	0.125	0.700	0.200
8	0.200	0.667	0.625	0.900	0.400
9	0.000	0.167	0.625	0.300	0.000
10	0.600	0.333	0.750	0.100	0.000
11	1.000	0.833	0.375	0.800	0.600
12	0.400	0.000	0.125	0.200	0.400
13	0.800	0.667	0.500	0.700	0.200
14	0.200	0.167	0.875	0.900	0.400
15	0.000	0.000	0.250	0.700	0.200

Large scale performance analysis of international kidney exchange programmes by the ENCKEP simulator

Kristóf Druzsin[1,2], Péter Biró[1,3], Rita Fleiner[2], and Xenia Klimentova[4]

[1] KRTK, Budapest, Hungary
[2] Óbuda University, Budapest, Hungary
[3] Corvinus University, Budapest, Hungary
[4] INESC TEC, Porto, Portugal

Abstract. We conduct performance analysis for national and international kidney exchange programmes (KEPs) with the simulator developed in the ENCKEP COST Action. Most of the large European countries operate kidney exchange programmes where kidney patients can exchange their willing but immunologically incompatible donors with each other. The ENCKEP simulator can mimic the operation of the European national and international KEPs by generating realistic datasets for a given time period, and then conducting regular matching runs by using the hierarchic optimisation criteria of the countries considered. In this new study we conduct large number of simulations to obtain robust findings on the performance of specific national programmes and on the possible benefits of international collaborations.

Keywords: kidney exchanges · computer simulation · hierarchical optimisation

1 Introduction

Kidney exchange programmes (KEPs) have been established in many countries to facilitate the exchanges of the donors. The European practices have been surveyed in [1] and the optimisation aspects of the European KEPs were described in [2], as the results of a COST Action called European Network for Collaboration on Kidney Exchange Programmes (ENCKEP).

International kidney exchanges have been conducted first in between Vienna and Prague in 2016 [3], followed by the collaboration of Spain, Portugal and Italy. The recent Handbook [4] of Working Group 3 and 4 of the ENCKEP COST Action has studied the practice of international KEPs, the modelling possibilities including results from [5] and [6], and the description of a simulation and evaluation tool developed by these working groups.

2 Computer simulations

Following up on our previous results presented in [8], we conducted multiple computer simulations for the three largest KEPs currently operating in Europe, namely the national programmes of the UK, Netherlands and Spain. We simulated 5 years in all cases, where the time interval between matching runs were 3 months, since this is the setting used in real practice. In order to obtain robust results, we conducted 1000 simulations per collaboration policy, and we used different input datasets for each of these runs. We present our results in this paper, however due to lack of space, the simulation results for the Netherlands is not included.

We used the ENCKEP Simulator tool [4] for conducting the simulations. For a survey on KEP simulators, see [7].

3 UK

In the national KEP of the UK, matching runs are conducted every 3 months, and the upper length limit both for exchange cycles and chains is 3. We used the same settings, and allowed internal recourse in the simulations in order to search for embedded cycles to implement in case of arc or node failure. As for the optimisation policy, we used the same set of criteria that we described in [8] (see [4] for further details about the scoring functions).

The frequency distribution of final levels in the 20.000 optimisation runs conducted is depicted on Fig. 1.

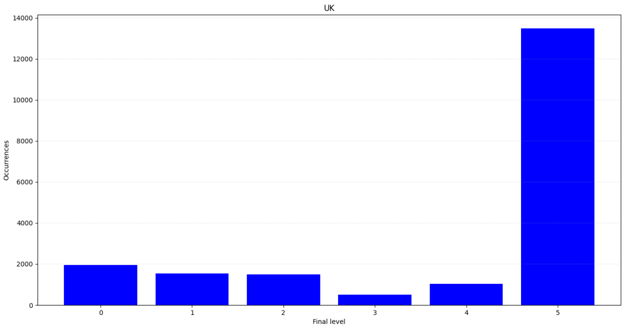

Fig. 1. Final level of optimisation runs for UK

Level 0 means that there were no possible cycles found in that optimisation run, and the 5th level represents the weighted optimisation level. As we can see on the figure, the occurrence of level 0 is relatively high, which might be because of the fact that we had to reduce the size of pools to be generated, since the optimisation process took too long to practically conduct 1000 simulations using the original pool sizes. However, recently we have managed to speed up this optimisation process, so further research can be conducted without these limitations.

Most importantly, we can see on Fig. 1 that the weighted optimisation level was reached in the majority of the cases, so the criteria defined for this level very often had an important effect on the solution selected for implementation. Since there are multiple criteria used on the weighted level of optimisation in the UK as well, checking which one had the biggest impact can be useful to precisely adjust the scoring functions. These results are shown on Fig. 2.

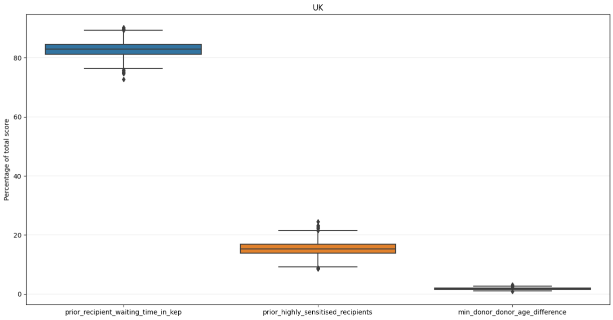

Fig. 2. Weighted criteria impact in UK

We measured these scores only if the weighted optimisation level was reached, because this is when the weighted criteria have an effect on the selection of the solution. The criterion for prioritising based on waiting time produced most of the scores given in weighted optimisation runs, while much less scores were given for prioritising highly sensitised recipients. Minimising the donor-donor age differences seems to be the final discriminator with small amounts of scores given in case of the other weighted criteria resulting in approximately the same amount of scores for two or more solutions, as it was intended for real practice.

4 Spain

The national KEP operating in Spain sets the limit for maximum length of exchange cycles to 3. They do not use length constraint on chains, but since we have to set an upper bound for this in the simulations (which should be reasonable to limit run-time), we used 4. We also allowed internal recourse in the simulations. The same set of optimisation criteria was used here as in [8].

To mimic real practice, we tried to set realistic relative pool sizes, which meant that the Spanish pool was set to be the smallest and the pool of the UK to be the largest. Since this pool size used for the UK was already limited, the initial pool of the Spanish KEP was often generated to be very small. As we can see on Fig. 3, this resulted in an increased number of optimisation runs where there were no possible exchange cycles in the virtual compatibility graph, and in case there were some, then the first criteria has already made the solution unique.

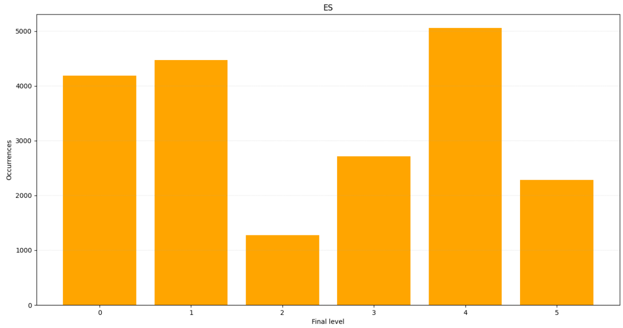

Fig. 3. Final level of optimisation runs for ES

In those cases, where the optimisation reached the weighted level, the impact of the different optimisation criteria used is depicted on Fig. 4 for the Spanish KEP.

The priority for the same blood-group transplants produced the highest scores in weighted optimisation runs, while the other 3 criteria resulted in way less scores in comparison. Also, in contrast to the UK, the scores given based on the waiting time of recipients in KEP produced the least amount of scores here.

5 International collaboration

The ENCKEP Simulator tool can simulate three different collaboration policies as follows (see [4] for further details):

- Individual policy: Each participating pool will have its own matching run separately.
- Consecutive policy: First, in each matching run, there will be an optimisation run for each pool separately. Then, the pairs who are still in the separate

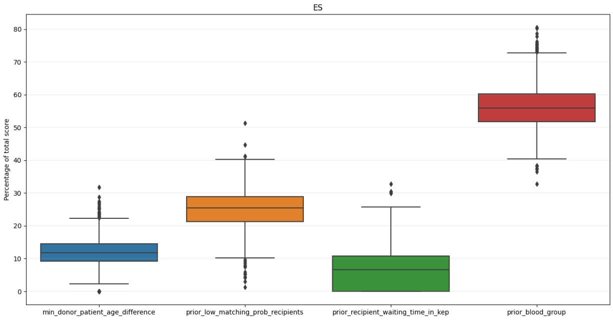

Fig. 4. Weighted criteria impact in ES

pools after this, will be merged into one joint pool, and there will be an optimisation run for this pool as well.

- Joint policy: All the participating pools will be merged into one pool immediately, and this merged pool will be used in the matching runs.

The simulator software allows us to set a customised optimisation policy for each pool in the simulations just as in real practice, so we used this feature with the corresponding settings described in [8]. Also, for the merged pools, we used the upper cycle and chain length limits and the optimisation policy that is used in the UK. The effectiveness of different collaboration policies studied is depicted on Fig. 5.

The results show that for each KEP, the joint collaboration policy was the most beneficial in the simulations regarding the number of transplants conducted. In case of the Netherlands, the benefit of the joint policy and the difference between the collaboration policies in general seems to be small, and the reason behind this is probably that the upper length limit for cycles used in the Netherlands is 4, while in merged pools we used the policy of the UK, which meant that these upper length limits were reduced to 3 in case of the KEP of the Netherlands.

In our previous study [8], we used one simulation instance to obtain a similar performance analysis, and we concluded that the joint policy was the most effective way of collaboration between KEPs. In this paper, we presented that robust results based on large scale performance analysis using 1000 simulation instances per collaboration policy led to the same conclusion. In further research, we would like to analyse the effectiveness of these collaboration and optimisation policies from the perspective of transplant quality as well, using quality indices [1].

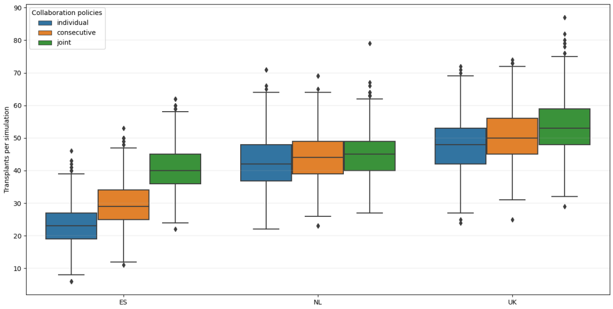

Fig. 5. Effectiveness of different collaboration policies

References

1. P. Biró, B. Haase, and J. van de Klundert et al., Building kidney exchange programmes in Europe – An overview of exchange practice and activities, *Transplantation* 103:1514–1522, 2019.
2. P. Biró, J. van de Klundert, D.F. Manlove et al., Modelling and optimisation in European Kidney Exchange Programmes, *European Journal of Operational Research* 291:447–456, 2021.
3. G. A. Böhmig, J. Fronek, A. Slavcev, G. F. Fischer, G. Berlakovich, and O. Viklicky, Czech-Austrian kidney paired donation: first European cross-border living donor kidney exchange, *Transplant International* 30:638–639, 2017.
4. X. Klimentova, and et al., International Kidney Exchange Programmes in Europe: Practice, Solution Models, Simulation and Evaluation tools, *Handbook of Working Group 3 and 4 of the ENCKEP COST Action* 2021.
5. X. Klimentova, A. Viana, J. P. Pedroso, and N. Santos, Fairness models for multi-agent kidney exchange programmes, *Omega* 102:102333, 2021.
6. R. S. Mincu, P. Biró, M. Gyetvai, A. Popa and U. Verma, IP solutions for international kidney exchange programmes, *Central European Journal of Operations Research*, 29(2):403–423, 2021.
7. N. Santos, P. Tubertini, A. Viana and J. P. Pedroso, Kidney exchange simulation and optimization, *Journal of the Operational Research Society* 68:1521–1532, 2017.
8. K. Druzsin, P. Biró, R. Fleiner, X. Klimentova, Simulations for measuring efficiency of international kidney exchange programmes, *16th International Symposium on Operational Research in Slovenia*, 2021.

Dynamic programming approaches for line balancing problems

András Éles[1], István Heckl[1]

[1] Department of Computer Science and Systems Technology,
University of Pannonia, Veszprém, Hungary
eles@dcs.uni-pannon.hu

Abstract. In the present work, solution methods are presented which provide solutions to different classes of assembly line balancing problems. Line balancing consists of the sequencing of production tasks on one or more assembly lines, and assigning them to working personnel, usually to maximize productivity or minimize costs. This is a complex combinatorial problem. However, in certain circumstances, dynamic programming can be used to provide globally optimal solutions in polynomial time, for example, when task order is fixed. Two investigated scenarios are presented in this work: when the given production lines and workforce must be distributed among different products, and when workstations consisting of multiple workers can execute tasks in parallel. The effectiveness of the proposed dynamic programming algorithms is demonstrated on a few examples.

Keywords: Optimization, Dynamic Programming, Line Balancing.

1 Introduction

Assembly or production line balancing seeks to evenly distribute workload among workforce at an assembly line to maximize productivity. There are many different formulations for this generally NP-hard optimization problem [1]. Approaches depend on different assembly line types, product variety and assembly line synchronization [2]. In simpler cases, there is a straight assembly line with a fixed pace and data are deterministic [3]. Mathematical programming approaches had also been proposed for line balancing [4].

The present work is motivated by Bartos and Bertók [5] who developed a P-Graph based model [6], resulting in a 20-25% productivity improvement in a real-world line balancing problem instance. A core assumption in this work was that the order of tasks was decided a priori. Under this assumption, even faster and more robust solution techniques become possible, and a wider range of line balancing problems can be covered. Our primary goal was to explore extensions of the problem formulation and design algorithmic approaches to achieve better results and a more generally applicable solution technique. Since our previous work [7], algorithms for more general problems using dynamic programming had been developed and tested.

2 Line balancing problems and solutions methods

2.1 Single line case

The first investigated line balancing problem is the following. Given a product with N tasks to be performed in a fixed order and K workers. Assign each task to a single worker such that each worker has consecutive tasks assigned and the cycle time is minimal. The cycle time is the highest of the workers' working times. The working time of a worker is the time needed to execute all tasks assigned to that worker on a single product item. The cycle time is the average time elapsed between product items being ready, which defines the overall production rate.

$$Z^{opt} = \min_{0=x_1 \leq x_2 \leq \cdots \leq x_{K+1}=N} \left(\max_{1 \leq k \leq K} T_{k,x_k,x_{k+1}} \right) \tag{1}$$

Equation (1) displays the optimal cycle time, where x_k is the first task the kth worker performs, and $x_{k+1} - 1$ is the last one. Parameter $T_{k,x_k,x_{k+1}}$ denotes the time worker k spends by performing tasks from x_k to $x_{k+1} - 1$. Note that tasks are indexed from 0 to $N - 1$, while workers are indexed from 1 to K.

A solution algorithm for this problem with dynamic programming relies on the definition of subproblems $F_{k,n}$, which denotes the optimal cycle time if only the first k workers and the first n tasks are considered. We can set $F_{k,0} = 0$ and $F_{0,n} = \infty$, otherwise the recursive formula shown in Equation (2) holds.

$$F_{k,n} = \min_{0 \leq x_k \leq n} \left(\max\left\{ F_{k-1,x_k}, T_{k,x_k,n} \right\} \right) \quad \forall\, 1 \leq k, n \tag{2}$$

In the formula, x_k is the choice made in subproblem $F_{k,n}$ for the kth worker. Then $F_{k,n}$ from $1 \leq k \leq K$ and $1 \leq n \leq N$ can be recursively calculated. Although only $F_{K,N}$ is the final answer, $F_{k,N}$ is obtained for all $1 \leq k \leq K$ in the procedure. In other words, the answers for all different worker counts up to K are obtained, which is a practically useful property of this solution technique.

If for all $F_{k,n}$ the optimal choice $x_k = x_{k,n}^{opt}$ is recorded, then the optimal task assignment can be reproduced from the table of $x_{k,n}^{opt}$ values in $O(k)$ time for any $F_{k,n}$.

2.2 Implementation considerations

Even the straightforward calculation of all $F_{k,n}$ can be done in $O(N^2 K)$ time which is fast enough for a single problem. However, if this optimization step is to be used multiple times, in an optimization procedure of a wider scope, any improvement can be helpful. The following was used in our implementation.

If $T_{k,a,b}$ is increasing as interval $[a, b[$ expands, that is, $\max\left\{T_{k,a+1,b}, T_{k,a,b-1}\right\} \leq T_{k,a,b}$, then F_{k-1,x_k} is increasing and $T_{k,x_k,n}$ is decreasing with x_k increasing. Then it can be proven that $x_{k,n+1}^{opt} \geq x_{k,n}^{opt}$, and moreover, $x_{k,n+1}^{opt} \in \{x - 1, x\}$, where x is the smallest index $x \geq x_{k,n}^{opt}$ for which $T_{k,x,n+1} \leq F_{k-1,x}$. This gives the possibility to find all $x_{k,n}^{opt}$ for a given k in $O(N)$ time, reducing the overall complexity to $O(NK)$.

The condition of $T_{k,a,b}$ being increasing in this desired way can be guaranteed based on problem assumptions. For example, a simple definition of $T_{k,a,b}$ is the sum of estimated time requirements c_i for all tasks i as shown in Equation (3).

$$T_{k,a,b} = \sum_{i=a}^{b-1} c_i \quad \forall\, k, a, b \tag{3}$$

Based on this starting definition of $T_{k,a,b}$, additional problem formulations were implemented. Some of these are guaranteed to preserve the condition on $T_{k,a,b}$, for example penalties for performing particular consecutive tasks or too many tasks with the same worker, or not identical workers. Some may break the condition, for example penalty for assigning particular consecutive tasks to *two different* workers.

Note that if workers are not assumed to be identical in the modeling point of view, then this dynamic programming approach can only be maintained if the order of workers is also assumed to be fixed.

3 Multiple products and lines

A more general problem was formulated and solved with dynamic programming. Here not only a single line is considered, but the total revenue from producing many different products on many lines is maximized. Given the number L of accessible assembly lines, and P different products, each with a revenue r_p, maximize total revenue generated per unit time. Let $M_{p,l,k}$ denote the optimum assuming that only the first p products, and a total of the first l lines and k workers are available. The recursive formula shown in Equation (4) holds. The decisions w and u correspond to the number of lines and workers dedicated to the p^{th} product.

$$M_{p,l,k} = \max_{\substack{w,u:\, w=u=0 \,\vee \\ 1\le w\le l, 1\le u\le k}} \left(M_{p-1,l-w,k-u} + r_p \cdot U_{p,w,u} \right) \quad \forall\, p, l, k \tag{4}$$

Here, $U_{p,l,k}$ denotes the maximum production rate if exactly k workers are assigned on l different lines to produce the p^{th} product. This value can also be obtained by its own recursive formula for each product individually, as shown in Equation (5). The decision u corresponds to the number of workers assigned to the l^{th} line.

$$U_{p,l,k} = \max_{1\le u\le k} \left(U_{p,l-1,k-u} + \frac{1}{F^{p}_{u,N_p}} \right) \quad \forall\, p, l, k \tag{5}$$

Here F^{p}_{u,N_p} is the optimal cycle time for product p and exactly u workers. These values are directly obtained from the single line problem for all products.

A straightforward implementation requires $O(PL^2K^2 + PN^2K)$ time to evaluate $M_{P,L,K}$ and all subproblems below, which is still low enough for practical examples. Note that in this problem formulation, assembly lines and workers were assumed to be identical. Considering one different type of worker would add a new dimension to the subproblem space.

In our implementation, additional constraints were considered in the problem formulation, including limits on worker and line counts, and even lower bounds for production rate of a single product. The latter allows to formulate a composition of a cost minimization and a throughput maximization problem. In general, any constraint prohibiting a particular subproblem, that can be unambiguously evaluated for a particular (w, u) choice and its result $M_{p,l,k}$ in Equation (4), or an u choice and its result $U_{p,l,k}$ in Equation (4), could be added to the problem while maintaining the structure of the aforementioned dynamic programming approach.

4 Parallel execution of tasks

In the previous formulations, workstations – which are responsible for a sequence of tasks – consisted of a single worker. In the following formulation, this assumption is relaxed, and multiple workers can be present on a single workstation, which allows an even distribution of working times of the task sequence among the workers involved. This can be advantageous in practice, for example if both sides of the assembly line are occupied, or the line simply transfers multiple product items to multiple workers at the same time.

The single line problem with possible parallel executions was considered as the problem formulation. The objective is cycle time minimization as before.

The average working time $Y_{s,a,b}$ for product items in a workstation with s identical workers performing tasks a to $b-1$ is shown in Equation (6), based on $Y_{1,a,b}$. Note that $Y_{s,a,b}$ is different from $T_{k,a,b}$, because u is a single worker with possibly special attributes, but s is the number of identical workers.

$$Y_{s,a,b} = \frac{Y_{1,a,b}}{s} + B(s) \quad \forall\, s, a, b \tag{6}$$

Here $B(s)$ is a penalty factor which should be non-zero. $B(s) = 0$ would lead to an extreme case where there is a single workstation with all workers. For this reason, our implementation supports arbitrary values for $B(s)$ for which $B(s+1) \geq B(s)$ and $B(1) = 0$. Note that $B(s) = \infty$ effectively sets an upper limit on the number of workers on a single workstation.

The subproblems are the same as in the original single line case, but a choice is added on the number of parallel workers s for the last station, see Equation (7).

$$F_{k,n} = \min_{\substack{0 \leq x_k \leq n \\ 1 \leq s \leq k}} \left(\max\left\{F_{k-s,x_k}, Y_{s,x_k,n}\right\}\right) \quad \forall\, k, n \tag{7}$$

The complexity of evaluating $F_{K,N}$ with a straightforward calculation is $O(N^2KS)$, where $S \leq K$ is an upper bound for worker count on a single workstation. Again, if $\max\left\{Y_{1,a+1,b}, Y_{1,a,b-1}\right\} \leq Y_{1,a,b}$ holds, then this can be reduced to $O(NKS)$.

5 Computational results

The problem formulations considering multiple products could be combined with the extension of parallel executions allowed on a single line, although we tested these two problem formulations individually. Implementations were C++ programs requiring only the standard libraries. Some of the tests are presented, executed on a Lenovo E5470 Laptop with Intel i7-6600 CPU and 16 GB RAM, on Ubuntu 20.04.4 LTS.

Table (1) shows some runtimes with various values of task count (N), available line count (L), worker count (K) and product count (P). The rest of the parameters were randomly generated. It seems that for these instances, the algorithm is most sensitive to the increasing of L and K and less to N and P. Inspecting the actual assignments shows that in most cases, one or two products are used. The reason behind this is that the product with the highest revenue to time ratio tends to be dominating, and only remainder workers are assigned to other options.

Table 1. Running times for the formulation with multiple products and lines.

N	L	K	P	Runtime (ms)
100	5	20	10	0.38
100	5	40	10	0.74
100	5	100	10	2.62
100	5	200	10	14.27
100	5	500	10	41.16
300	5	500	10	42.50
100	15	500	10	194.60
100	5	1500	10	206.81
100	5	500	30	77.92

Table 2. Optimal cycle times and runtimes for the formulation with parallel execution allowed.

N	K	p_{avg}	Objective	Runtime (ms)
20	15	0.5	8.99	0.34
50	15	0.5	22.15	0.94
100	15	0.5	39.60	2.99
200	15	0.5	77.67	13.09
500	15	0.5	186.39	66.98
500	15	0	184.97	66.64
500	15	1.5	187.75	66.75
500	15	∞	189.40	64.46
1000	15	0.5	366.40	254.95
500	30	0.5	93.80	78.77
500	60	0.5	47.90	86.81
500	200	0.5	15.39	297.41

Table (2) shows optimal cycle times and runtimes for randomly generated problem instances with various values of task count (N), worker count (K), and average value of $B(s+1) - B(s)$ denoted as p_{avg}, where $B(s)$ is the penalty factor for parallel workers. Among many findings, we can observe that for a large N, the objective is less sensitive to p_{avg}. The effect of N on runtime seems to be greater than of K, which is counter-intuitive and is subject to future research.

Both algorithms are very fast. Note that in the original real-world instances, there were only dozens of tasks and workers. A favourable property of these dynamic programming approaches compared to mathematical programming and P-Graph implementations is the strong guarantees for complexity. Although a potential disadvantage is difficulty to extend with additional constraints.

6 Conclusions

Algorithms for various line balancing optimization problem formulations were developed, particularly for revenue optimization with multiple products and allowing parallel execution on workstations. The approaches use dynamic programming, and guarantee robust and globally optimal solutions. Our tests have shown that these algorithms are very fast to be used not only for problem instances, but as part of possible optimization procedures in the future, targeting a wider scope.

A possible direction of future research is the investigation of several worker types and flexible task orders. These make the problem much more difficult. The algorithms shown can be of good use for this purpose by providing heuristics due to their speed.

Acknowledgement

This work has been implemented by the TKP2020-NKA-10 project with the support provided by the Ministry for Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the 2020 Thematic Excellence Programme funding scheme.

References

1. Sivasankaran, P., Shahabudeen. P.: Literature review of assembly line balancing problems. Int J Adv Manuf Technol 73, 1665–1694 (2014).
2. Kumar, N., Mahto, D.: Assembly line balancing: a review of developments and trends in approach to industrial application. Glob J Res Eng 13 (2013).
3. Becker, C., and Scholl., A.: A survey on problems and methods in generalized assembly line balancing. Eur J Oper Res 168, 694–715 (2006).
4. Deckro, R. F., Rangachari., S.: A goal approach to assembly line balancing. Comput Chem Eng 17 509–521 (1990).
5. Bartos, A., Bertok, B.: Production line balancing by P-graphs. Optim Eng (2019).
6. Friedler, F., Tarjan, K., Huang, Y. W., Fan, L. T.: Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems. Chem Eng Sci 47, 1973–1988 (1992).
7. Eles, A., Heckl, I.: Solving and extended line balancing optimization problem using dynamic programming. In: Vassanyi, I. (ed.) Proceedings of the Pannonian Conference on Advances in Information Technology (PCIT'2020), pp. 31–37. (2020).

Input design - from convex to non-convex problems and vice versa

László Gerencsér[1], György Michaletzky[2], and Zsuzsanna Vágó[3]

[1] ELKH SZTAKI, Institute for Computer Science and Control, Budapest, Hungary.
[2] Eötvös Loránd University, Budapest, Hungary.
[3] Pázmány Péter Catholic University, Budapest, Hungary
gerencser.laszlo@sztaki.hu, michaletzky@caesar.elte.hu,
vago.zsuzsanna@itk.ppke.hu

Abstract. We describe an optimization problem arising in the identification and control of linear stochastic systems, the design of optimal input. This leads to a convex problem in $\mathbb{R}^{p\times p}$, however, the generation of an input signal may lead to a non-convex problem. We discuss various approaches of convexification, and point out issues for further research.

Keywords: Experiment design · Active learning · Generalized moment problem · Sparse estimation

1 Technical setup

We consider discrete time, single input single output linear stochastic control system with input u external noise e and output y. The identification of the dynamics under frequency-weighted energy constraints on the input is of fundamental importance in most industrial control systems, see [1]. Optimal design of the input in order to minimize losses due to uncertainty is a fundamental theoretical problem, an a key component in practical adaptive input design, [8], [9]. More recently the problem of input design has become central in machine learning, see [2].

Consider a discrete time, single input single output linear stochastic control system with input u, external noise e and output y, defined in $-\infty < n < +\infty$

$$y = H^u\, u + H^e\, e. \tag{1}$$

Here H^u and H^e are rational functions of the backward shift operator q^{-1}. The associated transfer functions obtained when replacing q^{-1} by $e^{-i\omega}$, are assumed to belong to a parametric family of rational transfer functions of fixed degrees, say $H^u(\theta,\, e^{-i\omega})$ and $H^e(\theta,\, e^{-i\omega})$. The true parameter will be denoted by θ^*. The input process (u_n) and the noise process (e_n) are jointly wide sense stationary (w.s.st.) stochastic processes. Moreover, (e_n) is a martingale difference process with constant conditional variance. Finally we assume that u is orthogonal to e, written as $u \perp e$. It follows that $e = (e_n)$ is a w.s.st. orthogonal process.

The spectral distribution measure of u will be denoted by $\mathrm{d}\Phi^u(\omega)$, with $-\pi \leq \omega \leq \pi$. It is a finite measure, and for a real-valued process u it is symmetric w.r.t. 0, i.e. $\mathrm{d}\Phi^u(\omega) = \mathrm{d}\Phi^u(-\omega)$, defining the set of $\mathcal{S}^u$. Under standard conditions, see [3] or [1], the off-line prediction error (PE) estimator of θ^* is known to have the asymptotic covariance matrix $\Sigma_{\theta\theta} = \sigma^2 M^{-1} = \sigma^2 \left(M^u + M^e\right)^{-1}$, where

$$M^u = \int_{-\pi}^{+\pi} D^u(e^{-i\omega})\, D^{u\top}(e^{i\omega})\, \mathrm{d}\Phi^u(\omega) \tag{2}$$

$$M^e = \int_{-\pi}^{+\pi} D^e(e^{-i\omega})\, D^{e\top}(e^{i\omega})\, \mathrm{d}\omega \cdot \sigma^2, \tag{3}$$

and $D^u(e^{-i\omega})$ and $D^e(e^{-i\omega})$ are $\mathbb{C}^p$-valued rational function explicitly known up to the unknown true parameter θ^* :

$$D^u(e^{-i\omega}) = -H^e(\theta^*, e^{-i\omega})^{-1}\, H^u_\theta(\theta^*, e^{-i\omega}) \tag{4}$$

$$D^e(e^{-i\omega}) = -H^e(\theta^*, e^{-i\omega})^{-1}\, H^e_\theta(\theta^*, e^{-i\omega}). \tag{5}$$

The set of feasible matrices M^u, denoted by $\mathcal{M}^u$, is defined by taking a preselected, bounded, continuous weight-function $w(e^{i\omega}) \geq 0$ and imposing

$$\int_{-\pi}^{+\pi} w(\omega)\, \mathrm{d}\Phi^u(\omega) \leq K. \tag{6}$$

It is easily seen that $\mathcal{M}^u$ is a compact, convex set in $\mathbb{R}^{p\times p}$.

The quality of the input design is quantified via a performance index, penalizing uncertainty. Assuming that the performance index is sufficiently smooth in θ, and its Hessian at $\theta = \theta^*$, denoted by P, is positive definite, the primary input design problem can be formulated as

$$\min_{M^u \in \mathcal{M}^u} \operatorname{tr}\left(M^{-1} P\right) \tag{7}$$

$$\text{subject to} \quad M = M^u + M^e \in \mathbb{R}^{p\times p}_+. \tag{8}$$

Now, $\operatorname{tr}\left(M^{-1} P\right)$ is a strictly convex function for $M \in \mathbb{R}^{p\times p}_+$, and thus the optimization problem above has a unique solution M^{u*}. The question remains how to construct a spectral distribution measure $\mathrm{d}\Phi^u$ that generates M^{u*} via (2).

2 Input design - the associated moment problem

Thus we have a generalized moment problem at hand. A classic approach, due to engineering insights, is that for any feasible M^u a spectral measure concentrated to a finite number of frequencies can be constructed, see [4]. Indeed, for

$$M^u = 2 \int_0^{+\pi} \Re\left(D^u(e^{-i\omega})\, D^{u\top}(e^{i\omega})\right)\, \mathrm{d}\Phi^u(\omega), \tag{9}$$

the real, symmetric matrices $\Re\left(D^u(e^{-i\omega})\, D^{u\top}(e^{i\omega})\right)$ span a vector-space of dimension at most $s := p(p+1)/2$. Hence we can select at most s frequencies and corresponding weights, $0 \leq \omega_1 < \ldots < \omega_s \leq \pi$ and $\bar{\Phi}^u(\omega_k)$ so that

$$M^u = 2\sum_{k=1}^{s} \Re\left(D^u(e^{-i\omega_k})\, D^{u\top}(e^{i\omega_k})\right) \bar{\Phi}^u(\omega_k). \tag{10}$$

For any k a *real-valued* w.s.st. process with spectral measure concentrated at $\pm\omega_k$, with energy 1 at both, is obtained by setting

$$u_{k,n} := 2\cos(\varphi_k + \omega_k n), \tag{11}$$

where the random phase φ_k is uniform in $[-\pi, \pi]$. Choosing a set of independent random phases φ_k and energy levels σ_k^2 associated with $\pm\omega_k$, the process

$$u_n := \sum_{k=1}^{s} \sigma_k u_{k,n} = \sum_{k=1}^{s} 2\sigma_k \cos(\varphi_k + \omega_k n) \tag{12}$$

is a w.s.st. process with the discrete spectral measure assigning the weight σ_k^2 to each frequency $\pm\omega_k$. The signal $u = (u_n)$ is called a *multi-sine.* Letting $\alpha_k := \sigma_k^2$ the covariance matrix M^u will be linear in the α_k-s. Unfortunately, the cost function $\mathrm{tr}\left(M^{-1}\, P\right)$ is *non-convex* in ω_k, $k = 1, \ldots, s$.

However, an equivalent convex problem can be formulated by letting $\mathrm{d}\Phi^u(\omega)$ vary over the set of discrete symmetric spectral measures concentrated at any finite number of pairs of frequencies $\pm\omega_k$. This (convex) set will be denoted by $\mathcal{S}^d$. Let t be arbitrary, and for a set of frequencies ω_k and associated energy levels α_k, with $k = 1, \ldots, t$, consider the matrix

$$M^u = 2\sum_{k=1}^{t} \alpha_k\, \Re\, \left(D^u(e^{-i\omega_k})\, D^{u\top}(e^{i\omega_k})\right). \tag{13}$$

Then the optimization problem (7) becomes *convex* over the space of discrete measures $\mathcal{S}^d$. However its solution is a challenging task.

A heuristic approach to find a sub-optimal solution is to take a large t and a dense mesh of frequencies, and replacing the energy constraint by a linear penalty term in the objective function in the hope of enforcing sparsity of the solution, echoing the ideas of Lasso estimates, see [5], [6], [7]. To simplify the notation let $d_k = D^u(e^{-i\omega_k})$, and let $\gamma > 0$ be a scaling parameter of the penalty term. Then a sub-optimal input design is obtained via the minimization problem

$$\min_{\alpha}\, \mathrm{tr}\, \left[\left(2\Re\sum_{k=1}^{t} \alpha_k\, d_k \overline{d}_k^{\top} + M_e\right)^{-1} P\right] + \gamma\sum_{k=1}^{t} \alpha_k w(\omega_k),$$

subject to $\alpha_k \geq 0$. It can be shown that the above convex problem has a unique solution such that the number of non-zero coefficients is at most $s = p(p+1)/2$ for wide class of weight functions, such as band-pass filters. Experimental results regarding the above method with real data will be presented.

3 An analytic approach

An nice analytic approach for the characterization input spectra $\mathrm{d}\varPhi^u(\omega)$ solving the generalized momentum problem, allowing *vector-valued* inputs, was given in [10] in the special case when the filter $D^u(e^{-i\omega})$ is realized by a linear stochastic system $x_{n+1} = Ax_n + Bu_n$, i.e.

$$D^u(e^{-i\omega}) = (e^{i\omega}I - A)^{-1}B. \tag{14}$$

The key technical tool is the extension of what is called half-spectra, from the unit circle $\lambda = e^{-i\omega}$, to to the open unit disc of the complex plane. Let $\mathcal{M}$ denote the set of $p \times p$ matrix-valued Hermitian non-negative finite measures on $[-\pi, \pi]$. For any $\mathrm{d}\varPhi^u := \mu \in \mathcal{M}$ consider the auto-covariance matrices

$$\mathbb{E}[u(m+n)\, u^\top(m)] =: R(n) = \int_{-\pi}^{\pi} e^{in\omega}\, \mathrm{d}\mu(\omega). \tag{15}$$

It is well-known and easily seen that the sequence $R(n)$ is positive semi-definite, and if it is absolutely summable then $\mathrm{d}\mu(\omega)$ is absolutely continuous, and

$$\mathrm{d}\mu(\omega)/\,\mathrm{d}\omega = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} R(n) e^{-in\omega} \geq 0. \tag{16}$$

The left hand side is the spectral density of u. It follows that the *half spectrum*

$$F(e^{-i\omega}) := \frac{1}{2}R(0) + \sum_{n=1}^{\infty} R(n) e^{-in\omega} \tag{17}$$

is positive real, i.e. $\Re F(e^{i\omega})$ is positive semidefinit. Substituting $R(n)$ from (15), and replacing ω in (17) by ω', we get

$$F(e^{-i\omega'}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1 + e^{i(-\omega'+\omega)}}{1 - e^{i(-\omega'+\omega)}}\, \mathrm{d}\mu(\omega). \tag{18}$$

Replacing $e^{-i\omega'}$ by λ and letting λ vary in the open unit disc $\mathbb{D} = \{\lambda : |\lambda| < 1\}$ of the complex plane we can define the function

$$F(\lambda) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1 + \lambda e^{i\omega}}{1 - \lambda e^{i\omega}}\, \mathrm{d}\mu(\omega). \tag{19}$$

The function $F(\lambda)$ can be shown to be analytic and positive real for all $\lambda \in D$. The set of such functions will be denoted by $\mathcal{F}$. Thus we established a one-to-one correspondence between the set of measures $\mathcal{M}$ and the elements of $\mathcal{F}$ (modulo a constant additive skew-Hermitian term) irrespective of the nature of $\mathrm{d}\mu(\omega)$. In the scalar case for a multi-sine of the form (12) we have

$$F(\lambda) = \frac{1}{2\pi} \sum_{k=-s}^{s} \sigma_k^2 \frac{1 + \lambda e^{i\omega_k}}{1 - \lambda e^{i\omega_k}}. \tag{20}$$

To summarize the tools developed in [10] within the notations of the paper consider (14), and set $G(e^{-i\omega}) := D^u(e^{-i\omega})$. Substituting $e^{-i\omega}$ by λ we obtain

$$G(\lambda) = \lambda(I - \lambda A)^{-1}B. \tag{21}$$

A beautiful algebraic result of [10] provides a characterization of the set of covariance matrices $\varSigma := M^u$ that can be realized via (2):

Proposition 1. *Assume that the pair* (A, B) *is controllable with* A *having its eigenvalues in the open unit disc* $\mathbb{D}$. *Then a symmetric, positive semi-definit matrix* $M^u = \varSigma$ *is the stationary state covariance matrix of the linear system defined via* (14) *if and only if the matrix equation*

$$\varSigma - A\varSigma A^\top = BH + H^\top B^\top \tag{22}$$

has a solution H.

Now, the set of measures yielding a given covariance matrix $\varSigma$ can be characterized via their transforms in $\mathcal{F}$. First we note that is (A, B) is controllable, then we can find a suitable linear state-space transformation such that

$$AA^\top + BB^\top = I. \tag{23}$$

Let A, B normalized as above. Then the matrix $[A,\ B]$ can be completed to a an orthonormal matrix

$$U := \begin{pmatrix} A & B \\ C & D \end{pmatrix}. \tag{24}$$

Let us fix such a C and D, and consider the transfer function

$$V(e^{-i\omega}) = D + C(e^{i\omega}I - A)^{-1}B \tag{25}$$

and its extension to $\mathbb{D}$ obtained by substituting $\lambda = e^{-i\omega}$. It is well-known that $V(\lambda)$ is an *all-pass* or *inner* function: it is bounded analytic in the open unit disc, and has unitary radial limits on the unit circle. For a simple proof see [10].

Proposition 2. *Assume that* (23) *is satisfied. Then* $F(\lambda) \in \mathcal{F}$ *is a solution of the generalized moment problem if and only if it can be written in the form*

$$F(\lambda) = F_0(\lambda) + Q(\lambda)V(\lambda), \tag{26}$$

where $F_0(\lambda) = HG(\lambda)$ *and* $Q(\lambda)$ *is analytic in* $\mathbb{D}$.

Thus the set of solutions of the generalized moment problem is linearly parametrized by a convex set of analytic functions $Q(\lambda)$, subject to the constraint that $\Re F(\lambda) \geq 0$. For fixed $\varSigma$ the function $Q(\lambda)$ is uniquely determined by $F(\lambda)$. The actual design parameter is the matrix H defining $\varSigma$ implicitly via (22) subject to $\varSigma \geq 0$. The full exploitation of this results for solving the original optimization problem (7) - (8), and its extension to transfer functions of the form $D^u(e^{-i\omega}) = C(e^{i\omega}I - A)^{-1}B$ is an attractive open problem.

Acknowledgement. László Gerencsér was supported by the Ministry of Innovation and Technology NRDI Office (National Research, Development and Innovation Office) within the framework of the Autonomous Systems National Laboratory Program.

References

1. H. Hjalmarsson, “System identification of complex and structured systems,” *European Journal of Control*, vol. 15, no. 3-4, pp. 275–310, 2009.
2. A. Wagenmaker and K. Jamieson, “Active learning for identification of linear dynamical systems,” in *Conference on Learning Theory*, pp. 3487–3582, PMLR, 2020.
3. L. Gerencsér, “On the martingale approximation of the estimation error of arma parameters,” *Systems & Control Letters*, vol. 15, no. 5, pp. 417–423, 1990.
4. M. B. Zarrop, *Optimal experiment design for dynamic system identification*. Springer, 1979.
5. R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
6. D. L. Donoho, M. Elad, and V. N. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2005.
7. J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” *IEEE transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
8. L. Gerencsér and H. Hjalmarsson, “Adaptive input design in system identification,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 4988–4993, IEEE, 2005.
9. L. Gerencsér, H. Hjalmarsson, and L. Huang, “Adaptive input design for lti systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2390–2405, 2016.
10. T. T. Georgiou, “The structure of state covariances and its relation to the power spectrum of the input,” *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1056–1066, 2002.

Testing re-optimisation strategies in international kidney exchange programmes by the ENCKEP simulator

Lilla Matyasi[1], Péter Biró[2,3]

[1] Budapest University of Technology and Economics, Budapest, Hungary
[2] Institute of Economics, KRTK, Budapest, Hungary
[3] Corvinus University of Budapest, Budapest, Hungary
lilla.matyasi@gmail.com, peter.biro@krtk.hu

Abstract. We tested re-optimisation strategies for international kidney exchange programmes using the simulator developed by the ENCKEP COST Action. Kidney exchange programmes (KEPs) are operating in most of the European countries to facilitate the exchange of kidney donors for the recipients with incompatible living donors. The optimal solutions for national and international KEPs are typically selected in every three months based on the compatibilities estimated on the individual immunological data. However, these estimations are not always accurate, and if a positive crossmatch is found in the laboratory crossmatch tests then the corresponding exchange cycle must be cancelled. Depending on the matching process, the coordinators may use different re-optimisation strategies to repair the failed solutions. We examine the effects of using multiple rounds of re-optimisation with different optimisation strategies, such as fixing good cycles in the intermediate solutions or prioritising arcs with negative crossmatch tests in previous rounds. In the context of international KEPs we also consider the possibility of testing and prioritising internal arcs in the solutions. We measure the performance of these policies regarding the number of transplants and the number of compatibility tests conducted in the given time period.

Keywords: kidney exchange programmes, integer programming, computer simulation.

1. Introduction

Kidney exchange programmes (KEPs) help patients who have willing donors, who would donate their kidneys, however, the transplantation is not feasible due to immunological incompatibility. The goal of a KEP is to perform exchanges (or chains triggered by altruistic donors) between incompatible patient-donor pairs so that each patient can receive a compatible kidney. For example, if the recipient of the first pair can receive a kidney from the donor of the second pair, and the recipient of the second pair can receive a kidney from the donor of the first pair, a pairwise kidney exchange can be performed between them.

The immunological compatibility of patients and donors in a KEP is checked by crossmatch tests. In the first round, virtual crossmatch tests are performed based on the immunological data of the donors and recipients. These are quick and inexpensive to perform, however, these tests are not completely accurate. Based on the results of these virtual crossmatch tests, the transplants are selected by an optimisation algorithm.

A laboratory crossmatch test must also be performed prior to the actual implantation, which is much more costly, time consuming, and if it gives a positive result, the transplant cycle in which the particular patient-donor pair was involved cannot be accomplished. One of the biggest challenges in connection with KEPs is how to deal with such failures. The related strategies are called recourse, failure-aware or re-optimisation policies, and they have been extensively studied in the literature [4,5,7].

KEPs have been set up in many countries over the past two decades. At least ten countries have operating programmes in Europe [1,2] the largest are located in the United Kingdom [9], in the Netherlands and in Spain. These programmes use significantly different methods and optimality criteria to find the optimal solution [2]. The differences are usually due to legal constraints or a different structure of organisations that coordinate kidney exchanges, and to the fact that it is not entirely clear what solutions do the experts consider optimal.

Multi-country collaborations have also been established in several region, in between Vienna and Prague since 2016 [3], and in Spain, Portugal and Italy since 2018 [11], and for Sweden and Denmark by Scandiatransplant from 2019. International cooperation offers many new opportunities, but also new problems to solve [10].

For our research, we used and further developed the kidney exchange simulator of the ENCKEP (European Network for Collaboration on Kidney Exchange Programmes) COST Action [6,12]. This simulator can take real historical or generated instances of national or international KEPs as input and conduct the match runs for a given time period under various collaboration policies and optimisation criteria. In this paper we present our results on testing different re-optimisation strategies for national and international KEPs on ten generated instances for each simulation.

2. Re-optimalisation

In the set of solutions selected on the basis of virtual crossmatch tests, errors may be revealed after laboratory crossmatch tests. When a failed cycle or chain is detected, we can deal with the problem in different ways, that in general we call as re-optimisation.

A traditional way of fixing errors is called recourse, when we use only the internal arcs of a failed cycle to accomplish at least some of the transplantations, if possible. This strategy is used in the UK, where only one round of laboratory testing takes places, since the shipment in between the HLA-labs is time consuming, so the internal arcs are also tested together with the arcs selected for transplants. This feature was already implemented in the ENCKEP simulator. Our simulations showed that an improvement of 2% can be achieved in the number of transplants in the period of five years.

To consider more sophisticated recourse strategies, we implemented new features in the ENCKEP simulator. The most important is the possibility of re-optimisation, that is to find a new optimal solution after some failure occurs. We simulated various re-optimisation policies for national programmes by using the UK optimisation rules and for international KEPs by using the pools and rules of the UK, Netherlands and Spain.

2.1. Re-optimisation for one pool

In the first case we ran simulations from one up to five optimisation rounds, which is a general practice in the Netherlands, where multiple laboratory test runs can take place as the blood samples are stored in one central HLA-lab. In each round the simulator finds the best possible solution based on the result of the previous rounds. By the end of the fifth round in all cases the algorithm found the best possible solution.

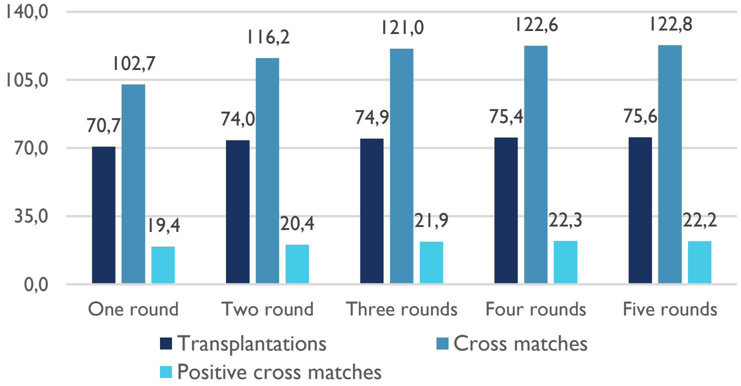

Fig. 1. The effect of conducting multiple re-optimisation rounds on the number of transplants and cross match tests.

We implemented and tested two new re-optimisation features for national KEPs, that we just summarise here due to lack of space. The first is the possibility of fixing the good cycles and chains in the intermediate solutions. This is a practice used in Spain, where the correct cycles and chains are kept after the first laboratory tests, and in the second re-optimisation only some additional cycles and chains are selected for the rest of the pool. When compared with the previous full (Dutch) re-optimisation policy this strategy resulted slightly less (~1%) transplants, but much less (~15%) laboratory crossmatch tests when conducting four re-optimisation rounds in each match run over the five years period.

The other possible improvement idea that we implemented and tested with the simulator is to track the results of the laboratory crossmatch rounds, since if one arc was found to be good in one round then one can assume that this transplant will remain possible in the future as well. In the simulations we added the maximisation of the number of good arcs right after the first objective of maximising the number of transplants. As a result, the number of transplants did not change significantly, but the number of crossmatch tests decreased slightly, by 2%. So, we found that this feature had no significant effect overall for the setting used.

2.2. Re-optimisation for multiple pools

When testing the effect of re-optimisation for multiple pools, we can think about possible testing and re-optimisation strategies for international KEPs. However, our findings can also be interesting for a national KEP that has multiple HLA-labs (e.g. UK and Spain), where the internal arcs correspond to transplant possibilities belonging to the same HLA-labs, and the external arcs correspond to transplant possibilities in between donors and recipients belonging to different HLA-labs. We may assume that the internal arcs (that belong to the same country or HLA-lab) can be tested in the laboratories quicker and cheaper, whilst the external transplants are more time consuming and expensive to get lab-tested due to the shipments of blood samples.

As the first feature we implemented and tested the possibility of conducting laboratory testing for separate pools. In order t to be able to simulate any testing strategy with the simulator, we made it possible to specify for which pools the software should test the solution in between the re-optimisation rounds. Intuitively, testing the internal arcs for pool after pool may be beneficial compared to testing these arcs together, since any failure is repaired immediately with re-optimisation.

An example for the testing and re-optimisation order:

"optimization_rounds":[["NL"],["UK","NL"], ["UK"], ["NL"], ["general"]]

In this example there are five re-optimisation rounds, in the first round only the internal arcs of the Dutch program will be tested, in the second the internal arcs of the United Kingdom and the Netherlands, while in the last round all of them will be tested.

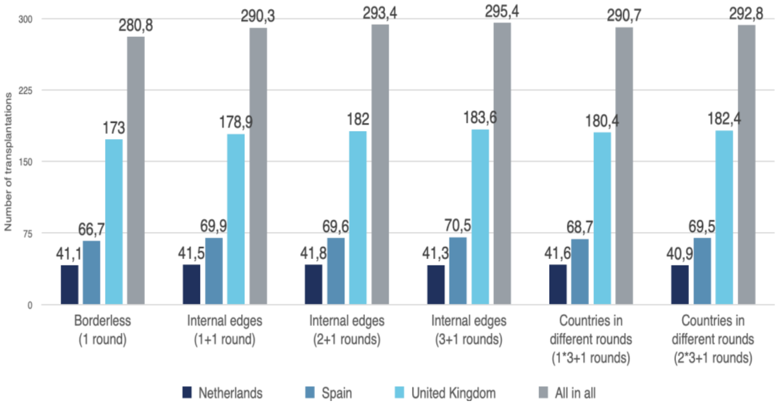

Fig. 2. The effect of the order of re-optimisation rounds on the number of transplants

In the results we can see that allowing one additional round of testing of internal arcs improved the number of transplants by more than 3%, however, we see no significant difference if these internal tests are conducted sequentially in each country.

In order to make international KEPs as successful as possible, it may be beneficial to minimise the number of international crossmatch-tests. We can do this by giving preference to the internal arcs, by maximising their total number in the solution, as the second objective after the maximisation of the number of transplants. As we can see in Figure 3., the total number of transplants has slightly increased, and the number of international tested has drastically decreased.

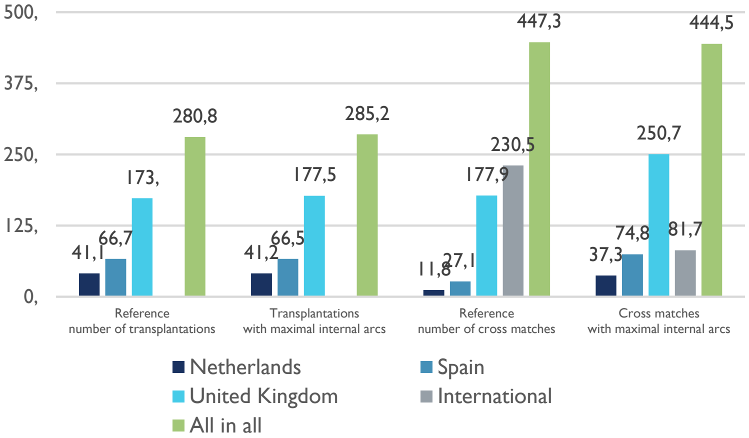

Fig. 3. The effect of the maximisation of internal arcs on the number of transplants and the number of internal and external crossmatch tests.

Finally, we tested the combination of maximisation of internal arcs, and conducting additional testing on internal arcs, which resulted in additional transplants.

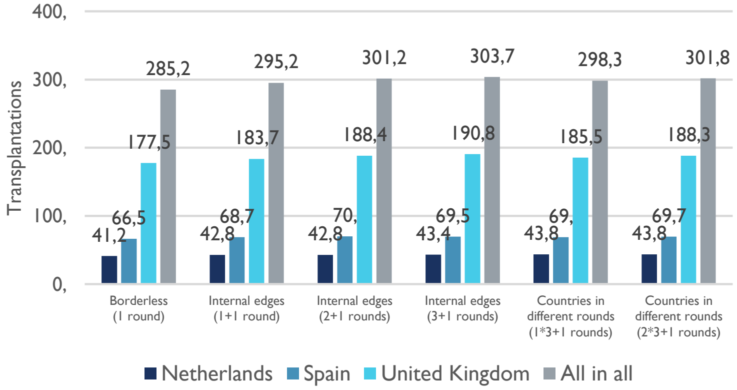

Fig. 4. The effect of the maximisation of internal arcs and additional testing runs of internal arcs on the number of transplants and the number of internal and external crossmatch tests.

3. Conclusion and future work

We extended the ENCKEP-simulator for testing re-optimisation policies for national and international KEPs. We could quantify the effects of various strategies for re-optimisation, including multiple testing rounds, multiple testing round for the internal arcs only, fixing good cycles in the solutions, and prioritising internal arcs.

As future work, we are planning to conduct much more simulations for each case, at least 1000, instead of 10, to get more robust findings. We will also consider further strategies and their combinations. Most importantly, we would also like to test these strategies on real datasets and understand better the nature of failures. In this paper we assumed that the failures are coming by incorrect virtual crossmatches, as traditional in the literature, that reflected past practices. However, the modern medical techniques create different types of failures: the high-resolution HLA-typing is more accurate, so the virtual crossmatch tests are almost as reliable as the laboratory tests, but the new possibility of HLA or ABO incompatible transplants might also lead to failures.

References

1. P. Biró, B. Haase, J. vd Klunder. et.al.: Building kidney exchange programmes in Europe - An overview of exchange practice and activities. Transplantation, 103(7):1514-1522 (2019)
2. P. Biró, J. vd Klunder, D.F. Manlove. et.al.: Modelling and optimisation in European Kidney Exchange Programmes. European Journal of Operations Research, 291:447-456 (2020)
3. G.A. Böhmig, J. Fronek, A. Slavcev, G.F. Fischer, G. Berlakovich, O. Viklicky. Czech-Austrian kidney paired donation: first European cross-border living donor kidney exchange. Transplantation International, 30/6:638-639 (2017)
4. M. Carvalho, X. Klimentova, K. Glorie, A. Viana, M. Constantino: Robust models for the kidney exchange problem. INFORMS Journal on Computing, 33(3), 861-881 (2021)
5. J.P. Dickerson, A.D. Procaccia, T. Sandholm: Failure-aware kidney exchange. In Proceedings of EC-2013 (pp. 323-340).
6. X. Klimentova et al.: International Kidney Exchange Programmes in Europe: Practice, Solution Models, Simulation and Evaluation tools, Handbook of Working Group 3 and 4 of the ENCKEP COST Action.
7. X. Klimentova, J.P. Pedroso, A. Viana: Maximising expectation of the number of transplants in kidney exchange programmes. Computers & Operations Research, 73, 1-11 (2016)
8. X. Klimentova, A. Viana, J. P. Pedroso, and N. Santos: Fairness models for multi-agent kidney exchange programmes. Omega, 102, 102333 (2021)
9. D.F. Manlove D, G. O'Malley. Paired and Altruistic Kidney Donation in the UK: Algorithmss and Experimentation. ACM Journal of Experimental Algorithmics, 19/2, art. 2.6. (2014)
10. R.S. Mincu, P. Biró, M. Gyetvai, A. Popa, U. Verma: IP solutions for international kidney exchange programmes. Central European Journal of Operations Research, 29(2), 403-423. (2021)
11. M.O. Valentin, M. Garcia, A.N. Costa, et al.: International cooperation for kidney exchange success. Transplantation, 103(6), e180-e181. (2019)
12. Website of the CA15210 COST Action : European Network for Collaboration on Kidney Exchange Programmes. http://www.cost.eu/COST_Actions/ca/CA15210

On Double Roman Domination

Darja Rupnik Poklukar[1] and Janez Žerovnik[1,2]

[1] Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva 6, 1000 Ljubljana, Slovenia;

[2] Institute of Mathematics, Physics and Mechanics, Jadranska 19, 1000 Ljubljana, Slovenia

janez.zerovnik@fs.uni-lj.si

Abstract. A double Roman dominating function on a graph $G = (V, E)$ is a function $f : V \to \{0, 1, 2, 3\}$ satisfying the condition that every vertex u for which $f(u) = 0$ is adjacent to at least one vertex assigned 3 or at least two vertices assigned 2, and every vertex u with $f(u) = 1$ is adjacent to at least one vertex assigned 2 or 3. The weight of f equals $w(f) = \sum_{v \in V} f(v)$. The double Roman domination number $\gamma_{dR}(G)$ of a graph G equals the minimum weight of a double Roman dominating function of G. We summarize the known results on double Roman domination number of generalized Petersen graphs $P(ck, k)$ including some very recent improvements based on discharging method and graph covers.

Keywords: Double Roman domination · Generalized Petersen graph · Discharging method · Graph cover · Double Roman graph.

1 Introduction

Double Roman domination of graphs was first studied in [6], motivated by a number of applications of Roman domination in present time and in history [7]. The initial studies of Roman domination [17, 23] have been motivated by a historical application. In the 4th century, Emperor Constantine was faced with a difficult problem of how to defend the Roman Empire with limited resources. His decision was to allocate two types of armies to the provinces in such a way that all the provinces in the empire will be safe. Some military units were well trained and capable of moving rapidly from one city to another in order to respond to any attack. Other legions consisted of a local militia and they were permanently positioned in a given province. The Emperor decreed that no legion could ever leave a province to defend another if in this case they left the province undefended. Thus, at some provinces two units were stationed, a local militia units were stationed at others, and some provinces had no army. While the problem is still of interest in military operations research [4], it also has applications in cases where a time-critical service is to be provided with some backup. For example, a fire station should never send all emergency vehicles to answer a call.

Similar reasoning applies in any emergency service. Hence positioning the fire stations, first aid stations, etc. at optimal positions improves the public services without increasing the cost. A natural generalization, in particular in the case of emergency services, is the k-Roman domination [11], where in each district k emergency teams are expected to be quickly available in case of multiple emergency calls. Special case $k = 2$, the double Roman domination, is considered here. It is well-known that the decision version of the double Roman domination problem (MIN-DOUBLE-RDF) is NP-complete, even when restricted to planar graphs, chordal graphs, bipartite graphs, undirected path graphs, chordal bipartite graphs and to circle graphs [15, 1, 5]. It is therefore of interest to study the complexity of the problem for other families of graphs. For example, linear time algorithms exist for interval graphs and block graphs [5], for trees [28], for proper interval graphs [16] and for unicyclic graphs [15].

Another avenue of research that is motivated by high complexity of the problem is to obtain closed expressions for the double Roman domination number of some families of graphs. This will be the topic of the talk at VOCAL 22 conference. In particular, generalized Petersen graphs and certain subfamilies of generalized Petersen graphs have been studied extensively in recent years. The results listed among related previous work include closed expressions for the double Roman domination number of some, and tight bounds for other subfamilies [8, 12, 20, 21]. For more results on double Roman domination we refer to recent papers [2, 14, 25] and the references there. For more details on the recent results to be discussed we refer to [18, 19].

2 Formal definitions

Let $G = (V, E)$ be a graph without loops and multiple edges. As usual, we denote with $V = V(G)$ the vertex set of G and with $E = E(G)$ its edge set.

A set $D \subseteq V(G)$ is a dominating set if every vertex in $V(G) \setminus D$ has at least one neighbor in D. The domination number $\gamma(G)$ is the cardinality of a minimum dominating set of G. A double Roman dominating function (DRDF) on a graph $G = (V, E)$ is a function $f : V \to \{0, 1, 2, 3\}$ with the following properties:

(1) every vertex u with $f(u) = 0$ is adjacent to at least one vertex assigned 3 or at least two vertices assigned 2, and

(2) every vertex u with $f(u) = 1$ is adjacent to at least one vertex assigned 2 or 3 under f.

Define $f(U) = \sum_{u \in U} f(u)$ as the weight of f on an arbitrary subset $U \subseteq V(G)$. Then, the weight of f equals $w(f) = f(V(G)) = \sum_{v \in V(G)} f(v)$. The double Roman domination number $\gamma_{dR}(G)$ of a graph G is the minimum weight of a double Roman dominating function of G. A DRD function f is called a γ_{dR}-function of G if $w(f) = \gamma_{dR}(G)$.

For any double Roman dominating function f, defined on G we define a partition of the vertex set $V = V_0 \cup V_1 \cup V_2 \cup V_3$, where $V_i = V_i^f = \{u \mid f(u) = i\}$.

Domination in graphs with its many varieties has been extensively studied in the past [9, 10]. For more references we refer to full versions of the related papers [18, 19].

Petersen graphs are among the most interesting examples when considering nontrivial graph invariants. The generalized Petersen graph $P(n,k)$ is a graph with vertex set $U \cup V$ and edge set $E_1 \cup E_2 \cup E_3$, where $U = \{u_0, u_1, \cdots, u_{n-1}\}$, $V = \{v_0, v_1, \cdots, v_{n-1}\}$, $E_1 = \{u_i u_{i+1} \mid i = 0, 1, \ldots, n-1\}$, $E_2 = \{u_i v_i \mid i = 0, 1, \ldots, n-1\}$, $E_3 = \{v_i v_{i+k} \mid i = 0, 1, \ldots, n-1\}$, and subscripts are reduced modulo n. Thus, we identify integers i and j iff $i \equiv j \bmod n$. (As usual, $m \equiv r \bmod n$ means that $m = kn + r$, or equivalently, $m - r = kn$ for some integer $k \in \mathbb{Z}$.

It is well known that the graphs $P(n,k)$ are 3-regular unless $k = \frac{n}{2}$ and that $P(n,k)$ are highly symmetric [27, 22]. As $P(n,k)$ and $P(n, n-k)$ are isomorphic, it is natural to restrict attention to $P(n,k)$ with $n \geq 3$ and k, $1 \leq k < \frac{n}{2}$.

3 Related previous Work

Results of the previous work are gathered in the following tables.

Table 2. Previously known results on double Roman domination number of generalized Petersen graphs $P(n,k)$ for small k.

$n \geq 3$: $\gamma_{dR}(P(n,1)) = \begin{cases} \frac{3n}{2}, & n \equiv 0 \bmod 4, \\ \frac{3n+3}{2}, & n \equiv 1, 3 \bmod 4, \\ \frac{3n+4}{2}, & n \equiv 2 \bmod 4 \end{cases}$	Shao et al. [20]
$n \geq 5$: $\gamma_{dR}(P(n,2)) = \begin{cases} \lceil \frac{8n}{5} \rceil, & n \equiv 0 \bmod 5, \\ \lceil \frac{8n}{5} \rceil + 1, & n \equiv 1, 2, 3, 4 \bmod 5. \end{cases}$	Jiang et al. [12]
If G is a graph of maximum degree $\triangle \geq 1$: $\gamma_{dR}(G) \geq \left\lceil \frac{3\vert V(G)\vert}{\triangle+1} \right\rceil$. $\gamma_{dR}(P(n,k)) \geq \lceil \frac{3n}{2} \rceil$.	Shao et al. [20]

General upper bound is due to Gao et al. [8]. Let $k \geq 3$. Then

$$\gamma_{dR}(P(n,k)) = \frac{3n}{2}, \quad k \equiv 1 \bmod 2,\ n \equiv 0 \bmod 4.$$

and

$$\gamma_{dR}(P(n,k)) \leq \begin{cases} \frac{3n}{2} + \frac{5k+5}{4}, & k \equiv 1 \bmod 4, n \not\equiv 0 \bmod 4, \\ \frac{3n}{2} + \frac{5k+7}{4}, & k \equiv 3 \bmod 4, n \not\equiv 0 \bmod 4, \\ \frac{3n}{2}\frac{(3k+2)}{(3k+1)}, & k \equiv 0 \bmod 4, \\ & n \equiv 0 \bmod (3k+1), \\ \lceil \frac{3n}{2}\frac{(3k+2)}{(3k+1)} \rceil + \frac{5k+4}{4}, & k \equiv 0 \bmod 4, \\ & n \not\equiv 0 \bmod (3k+1), \\ \frac{3n}{2}\frac{(3k)}{(3k-1)}, & k \equiv 2 \bmod 4, \\ & n \equiv 0 \bmod (3k-1), \\ \lceil \frac{3n}{2}\frac{(3k)}{(3k-1)} \rceil + \frac{5k+6}{4}, & k \equiv 2 \bmod 4, \\ & n \not\equiv 0 \bmod (3k-1). \end{cases}$$

Double Roman domination of families $P(ck,k)$ has been studied recently for small c, including $c = 3$, 4, and 5. The results are summarized below.

Table 2. Previously known results on double Roman domination number of generalized Petersen graphs $P(ck,k)$ for small c.

$\gamma_{dR}(P(3k,k)) = \begin{cases} 5k+1, & k \in \{1,2,4\} \\ 5k, & \text{otherwise} \end{cases}$	Shao et al. [21]
$k \geq 1$: $6k \leq \gamma_{dR}(P(4k,k)) \leq \begin{cases} 6k; & k \equiv 1 \bmod 2 \\ 6k+3; & k \equiv 0 \bmod 2 \end{cases}$	
$k \geq 2$: $8k \leq \gamma_{dR}(P(5k,k)) \leq \begin{cases} 8k, & k \equiv 2, 3 \bmod 5 \\ 8k+2, & \text{otherwise} \end{cases}$	Rupnik P. et al. [18]

Table 3. Some special cases.

$\gamma_{dR}(P(4,1)) = 6,\ \gamma_{dR}(P(8,2)) = 14,\ \gamma_{dR}(P(10,2)) = 16$	Jiang et al. [12]
$\gamma_{dR}(P(12,3)) = 18,\ \gamma_{dR}(P(5,1)) = 9$	Shao et al. [20]
$23 \leq \gamma_{dR}(P(15,3)) \leq 26$	Gao et al. [8]

4 New results to be presented

Our recent results were obtained by application of the discharging method [20, 18] that is very well known due to its use in the famous proof of the four color theorem [3]. Another well known concept that can be very useful is the notion of graph covers [24, 13]. Very recently, we have proved the next theorem.

Theorem 1. *[19]*

1. *If* $c \equiv 0 \bmod 4$ *and* k *odd, then*

$$\gamma_{dR}(P(ck,k)) = \frac{3}{2}ck$$

2. *If* $c \not\equiv 0 \bmod 4$ *and* k *odd, then*

$$\frac{3}{2}ck \le \gamma_{dR}(P(ck,k)) < \begin{cases} \frac{3}{2}(c+\frac{1}{2})k, \ c \equiv 1,3 \bmod 4, \\ \frac{3}{2}(c+\frac{2}{3})k, \ c \equiv 2 \bmod 4, \end{cases}$$

3. *For* k *even, it holds*

$$\frac{3}{2}ck \le \gamma_{dR}(P(ck,k)) \le \begin{cases} \frac{3}{2}c(k+\frac{1}{2}), & c \equiv 0 \bmod 4, \\ \frac{3}{2}(c+\frac{1}{2})(k+\frac{1}{2}), & c \equiv 1,3 \bmod 4, \\ \frac{3}{2}(c+\frac{2}{3})(k+\frac{1}{2}), & c \equiv 2 \bmod 4, \end{cases}$$

References

1. Ahangar, H.A.; Chellali, M.; Sheikholeslami, S.M. On the double Roman domination in graphs. *Discret. Appl. Math.* **2017**, *232*, 1–7.
2. Amjadi, J.; Nazari-Moghaddam, S.; Sheikholeslami, S.M.; Volkmann, L. An upper bound on the double Roman domination number. *J. Comb. Optim.* **2018**, *36*, 81–89.
3. Appel, K.; Haken, W.; Every planar map is four colorable. I. Discharging. Illinois Journal of Mathematics 21 **1977** 429–490, doi:10.1215/ijm/1256049011.
4. Arquilla, J.; Fredricksen, H. “Graphing”—An Optimal Grand Strategy. Mil. Oper. Res. **1995**, *1*, 3–17.
5. Barnejee, S.; Henning, M.A.; Pradhan, D. Algorithmic results on double Roman domination in graphs. *J. Comb. Optim.* **2020**, *39*, 90–114.
6. Beeler, R.A.; Haynes, T.W.; Hedetniemi, S.T. Double Roman domination. *Discrete Appl. Math.* **2016**, *211*, 23–29.
7. Cockayne, E.J.; Dreyer, P.A., Jr.; Hedetniemi, S.M.; Hedetniemi, S.T. Roman domination in graphs. *Discret. Math.* **2004**, *278*, 11–22.
8. Gao, H.; Huang, J.; Yang, Y. Double Roman Domination in Generalized Petersen Graphs. *Bull. Iran. Math. Soc.* **2021**, 1–10. https://doi.org/10.1007/s41980-021-00551-5
9. Haynes, H.W.; Hedetniemi, S.; Slater, P. *Fundamentals of Domination in Graphs*; Marcel Dekker: New York, NY, USA, 1998.
10. Haynes, H.W.; Hedetniemi, S.; Slater, P. *Domination in Graphs: Advanced Topics*; Marcel Dekker: New York, NY, USA, 1998.
11. Hening, M.A. Defending the Roman Empire from Multiple Attacks. *Discret. Math.* **2003**, *271*, 101–115.

12. Jiang, H.; Wu, P.; Shao, Z.; Rao, Y.; Liu, J. The double Roman domination numbers of generalized Petersen graphs $P(n, 2)$, *Mathematics* **2018**, *6*, 206–217.
13. Malnič, A.; Pisanski, T.; Žitnik, A. The clone cover. *Ars Math. Contemp.* **2015**, *8*, 95–113.
14. Mobaraky, B.P.; Sheikholeslami, S.M. Bounds on Roman domination numbers of graphs. *Mat. Vesnik* **2008**, *60*, 247–253.
15. Poureidi, A.; Rad, N.J. On algorithmic complexity of double Roman domination, *Discret. Appl. Math.* **2020**, *285*, 539–551.
16. Poureidi, A. A linear algorithm for double Roman domination of proper interval graphs. *Discret. Math. Algorithms Appl.* **2020**, *12*, 2050011.
17. ReVelle, C.S.; Rosing, K.E. Defendens Imperium Romanum: A classical problem in military strategy. *Am. Math. Mon.* **2000**, *107*, 585–594.
18. Rupnik Poklukar D, Žerovnik J. On the Double Roman Domination in Generalized Petersen Graphs $P(5k, k)$. *Mathematics* 10 (2022) 119. https://doi.org/10.3390/math10010119
19. Rupnik Poklukar D, Žerovnik J. Double Roman Domination in Generalized Petersen Graphs $P(ck, k)$. *submitted.*
20. Shao, Z.; Wu, P.; Jiang, H.; Li, Z.; Žerovnik, J.; Zhang, X. Discharging approach for double Roman domination in graphs. *IEEE Acces* **2018**, *6*, 63345–63351, doi:10.1109/ACCESS.2018.2876460.
21. Shao, Z.; Erveš, R.; Jiang, H.; Peperko, A.; Wu, P.; Žerovnik, J. Double Roman graphs in $P(3k, k)$. *Mathematics* **2021**, *9*, 336, doi:10.3390/math9040336.
22. Steimle, A.; Staton, W. The isomorphism classes of the generalized Petersen graphs. *Discret. Math.* **2009**, *309*, 231–237.
23. Stewart, I. Defend the Roman Empire! *Sci. Am.* **1999**, *281*, 136–138.
24. Gross, J.L.; Tucker, T.W. *Topological Graph Theory*; Wiley-Interscience: New York, NY, USA, 1987.
25. Volkmann, L. Double Roman domination and domatic numbers of graphs. *Commun. Comb. Optim.* **2018**, *3*, 71–77.
26. Wang, H.; Xu, X.; Yang, Y. On the Domination Number of Generalized Petersen Graphs $P(ck, k)$. *ARS Comb.* **2015**, *118*, 33–49.
27. Watkins, M.E. A theorem on Tait colorings with an application to the generalized Petersen graphs. *J. Comb. Theory* **1969**, *6*, 152–164.
28. Zhang, X.; Li, Z.; Jiang, H.; Shao, Z. Double Roman domination in trees. *Inf. Process. Lett.* **2018**, *134*, 31–34.
29. Zhao, W.; Zheng, M.; Wu, L. Domination in the generalized Petersen graph $P(ck, k)$. *Util. Math.* **2010**, *81*, 157–163.

Parallelizing zero-one linear programs using graph theoretical considerations

Sándor Szabó[1] and Bogdán Zavalnij[2]

[1] University of Pecs, Institute of Mathematics and Informatics sszabo7@hotmail.com
[2] Renyi Institute of Mathematics bogdan@renyi.hu

Abstract. In this short note we will present a method to parallelize certain zero-one linear programs in which graph theoretical arguments play the most important role. The ready availability of microprocessors with a dozen or a few dozens of cores is our main motivation. In other words we are aiming moderate scale parallelization. For supercomputers with thousands of cores one should turn to different parallelization ideas. In order to illustrate the suggested parallelization method we will work out a small toy size numerical example in details. In addition we will introduce preconditioning methods to simplify the resulting linear programs. These preconditioning techniques are also motivated by graph theoretical considerations.

Keywords: zero-one linear program, weighted maximum clique problem

1 Introduction

Let F be a finite family of subsets $A_1, \ldots, A_n$ of a finite set $U = \{u_1, \ldots, u_m\}$. Let $c_1, \ldots, c_n$ be non-negative weights assigned to $A_1, \ldots, A_n$, respectively. The maximum weight packing problem is asking for locating pair-wise disjoint members of F such that the sum of the weights of these elements is as large as possible. This maximum weight packing problem can be stated as a zero-one linear program P.

$$\mathbf{cx} \to \max$$

$$\mathbf{Ax} \leq \mathbf{b}$$

Here $\mathbf{x} = [x_1, \ldots, x_n]^{\mathrm{T}}$, $\mathbf{c} = [c_1, \ldots, c_n]$, $\mathbf{b}$ is the m by 1 matrix $[1, \ldots, 1]^{\mathrm{T}}$. Finally, $\mathbf{A}$ is an m by n matrix whose typical entry is $\beta_{i,j}$. The entry $\beta_{i,j}$ is equal to 1 if $u_i \in A_j$ and it is equal to 0 otherwise. The meaning of the decision variable x_i is the following. The value of x_i is equal to 1 if the subset A_i is part of a packing and x_i is equal to 0 otherwise.

The maximum weight packing problem can be formulated as a maximum weight clique problem as well.

Let G be a finite graph without any loop and double edge. A subset Δ of the nodes of G is a clique if any two distinct nodes are adjacent in Δ. Suppose that a non-negative weight is assigned to each node of G. The maximum weight clique problem is asking for locating a clique Δ in G such that the sum of the weights of the nodes of Δ is as large as possible.

We construct an agreement graph G whose nodes are the subsets $A_1, \ldots, A_n$. Two sets A_i and A_j are adjacent in G whenever A_i and A_j are disjoint. To the node A_i we assign c_i as a weight. The basic observation we need is the following. The cliques in G and the feasible solutions of the linear program P are in a one-to-one correspondence.

The maximum weight packing problem instance can be solved using a zero-one linear program solver or alternatively using a maximum weight clique solver. On certain instance the linear program solver is more efficient than the clique solver while for other instance the situation may be reversed.

In this short note we use the linear program solver but we wish to use informations provided by the clique approach. In particular, we will use intimate information from the clique reformulation to divide the linear program into smaller parts what can be solved independently of each other. Secondly, the clique approach will help to establish upper bounds of the optimal value of the objective function and consequently will help in reducing the search space of the procedure. Finally the clique approach suggests preconditioning methods to simplify the linear programs before feeding them into a solver.

Let W_1, W_2, W_3 be pair-wise disjoint subsets of the nodes of the graph G and assume that V is the set of nodes of G. Suppose that $V = W_1 \cup W_2 \cup W_3$. If in addition there is no edge of G is going from W_1 to W_3, then we say that the triple (W_1, W_2, W_3) is a splitting partition of G.

Let G_1, G_2 be the subgraphs of G induced by the set of nodes $W_1 \cup W_2$, $W_2 \cup W_3$, respectively.

Lemma 1. *If Δ is a maximum weight clique in the graph G, then Δ is a maximum weight clique in either G_1 or in G_2.*

This observation is proved in [3]. Lemma 1 allows to divide a maximum clique problem instance into two smaller instances. It also allows to divide the program P into two smaller programs P_1 and P_2. The programs P_1, P_2 are constructed from P by equating the variables corresponding to the nodes in W_3, W_1 to zero, respectively. (The variable x_i and the node A_i are associated with each other.)

2 Preconditioning methods

It is a known fact that establishing tight upper bounds for the optimal value of the objective function greatly reduces the searching space in the course of solving a zero-one linear program. In the first part of this section we will show that the agreement graph associated with the linear program may help finding useful upper bounds for the value of the objective function.

Let G be the agreement graph associated with the zero-one linear program P. We assign colors the nodes of G such that each node of G receive exactly one color and adjacent nodes never receive the same color. This coloring is referred as legal coloring of the nodes of G. A color class is a set of nodes receiving the same colors. Suppose that the nodes of G are legally colored using s colors and $C_1, \ldots, C_s$ are the colors classes. Let K be the following number. From each colors class choose a representative node whose weight is maximum in its colors class and add up the weight of these representative nodes.

Lemma 2. *For each feasible solution of the program P the value of the objective function is at most K.*

Proof. The feasible solutions of the program P and the cliques in the graph G are in a one-to-one correspondence of each other. Let Δ be a clique in G such that the sum of the weights of the nodes of Δ is maximum. Each color class contains at most one nodes from Δ. It follows that the sum of the weights of the nodes of Δ is at most K.

With a little extra work we may establish better upper bounds. For this purpose we introduce a color index for each vertex v of the graph G. We compute the following number K. The neighbors of v are distributed in the color classes $C_1, \ldots, C_s$. It can happen that a color class does not contain any neighbor of v. We toss out these color classes. From each remaining color class we choose a representative neighbor whose weight is maximum in its color class and add up the weights of these representative neighbors. Finally, we add the weight of v to this sum. This number K is assigned to v and we refer to it as the color index of v. The reader can verify the following basic observation about the color index.

Lemma 3. *If Δ is a clique in G whose total weight is K, then the color index of each node of Δ must be at least K.*

It is a corollary to Lemma 3 that the largest of the color indices of the nodes of the agreement graph G is an upper bound for the objective function of the program P for each feasible solution of P.

We introduce a color index for each edge $e = \{u, v\}$ of the agreement graph G. We compute the following number K. The common neighbors of the nodes u, v are distributed in the color classes $C_1, \ldots, C_s$. It can happen that a color class does not contain any common neighbor of u, v. We sort out these color classes. From each remaining color class we choose a representative common neighbor whose weight is maximum in its color class and add up the weights of these representative neighbors. Finally, we add the weights of u and v to this sum. This number K is assigned to edge e and we refer to it as the color index of e. The reader can verify the following basic observation about the color index.

Lemma 4. *If Δ is a clique in G whose total weight is K, then the color index of each edge of Δ must be at least K.*

When we construct a legal coloring of the nodes of the agreement graph G we are not trying to use the optimum number of colors. The reason is that finding the chromatic number of a given graph is an NP hard optimization problem itself. We use a simple greedy coloring procedure which can be completed in relatively quickly and we accept the fact that the number of the colors is not optimal.

In the remaining part of this section we describe methods to delete nodes or edges from the agreement graph G. Deleting nodes helps to set a certain variable x_i to be zero in the linear program P without changing the value of the optimal solution. Deleting edges may help to find a better legal coloring of the nodes of G or may help to find a better splitting partition of the nodes.

Let u, v be non-adjacent nodes of G. We say that u dominates v if $N(v) \subseteq N(u)$, $\mathrm{weight}(v) \leq \mathrm{weight}(u)$ hold. Here $N(u)$, $N(v)$ are the set of neighbors of u, v, respectively. Let $e = \{u, v\}$, $f = \{v, w\}$ be distinct edges of G. We say that e dominates f if the nodes u, w are not adjacent in G and $[N(w) \cap N(v)] \subseteq [N(v) \cap N(u)]$, $\mathrm{weight}(w) \leq \mathrm{weight}(u)$ hold. The proof of the following lemma can be found in [3].

Lemma 5. *If node u dominates node v, then node v can be deleted from the agreement graph G when we are looking for a maximum weight clique in G. Similarly, if edge e dominates edge f, then edge f can be deleted from the agreement graph G when we are looking for a maximum weight clique in G.*

Lemma 5 tells us that it is reasonable to inspect the nodes and edges and delete dominated nodes and edges. (For more sophisticated preconditioning methods see [2].)

3 A toy size small example

Let us start with a zero-one linear program P with 8 variables and 6 constraints given by Table 1. The reader will notice that the program P is an instance of

Table 1. The given zero-one linear program P in the toy example.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$\in$	$\{0,1\}$
	1	3	2	1	4	2	2	3	$\rightarrow$	max
(1)	1		1			1			$\leq$	1
(2)			1		1			1	$\leq$	1
(3)		1			1	1			$\leq$	1
(4)		1		1	1				$\leq$	1
(5)		1					1	1	$\leq$	1
(6)			1			1	1		$\leq$	1

the maximum weight set packing problem about 8 subsets

$$A_1 = \{u_1\}, \quad A_2 = \{u_3, u_4, u_5\}, \quad A_3 = \{u_1, u_2, u_6\}, \quad A_4 = \{u_4\},$$

$$A_5 = \{u_2, u_3, u_4\}, \quad A_6 = \{u_1, u_3, u_6\}, \quad A_7 = \{u_5, u_6\}, \quad A_8 = \{u_2, u_5\}$$

of a 6 elements universal set $U = \{u_1, \ldots, u_6\}$. Non-negative integer weights 1,3,2,1,4,2,2,3 are assigned to these sets, respectively. We are looking for a pairwise disjoint collection of the subsets such that the sum of the associated weights is as large as possible.

The subsets can be described by an 6 by 8 incidence matrix depicted in Table 2. The rows are labeled by the elements of U and the columns are labeled by the subsets $A_1, \ldots, A_8$. The incidence matrix gives the coefficient matrix of the linear program P.

Table 2. The incidence matrix of the subsets $A_1, \ldots, A_8$ and the adjacency matrix of the agreement graph G.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
u_1	●		●			●		
u_2			●		●			●
u_3		●			●	●		
u_4		●		●	●			
u_5		●					●	●
u_6			●			●	●	

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	×	●		●	●		●	●
A_2	●	×	●					
A_3		●	×	●				
A_4	●		●	×		●	●	●
A_5	●				×		●	
A_6				●		×		●
A_7	●			●	●		×	
A_8	●			●		●		×

We construct an agreement graph G whose nodes are the subsets $A_1, \ldots, A_8$. The adjacency matrix of G is in Table 2. A possible geometric representation of G can be seen in Figure 1.

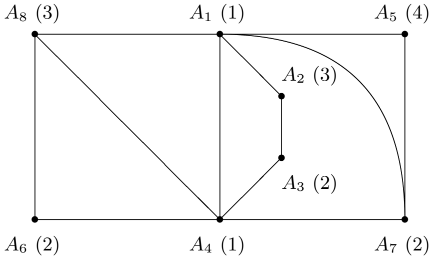

Fig. 1. A graphical representation of the agreement graph G in the toy example. The numbers in parenthesis are the weights.

Going through all cliques in the agreement graph G we located the clique with nodes A_1, A_5, A_7 whose weight is maximum. Of course, in case of a larger graph this exhaustive inspection of the cliques is prohibitively time consuming. We may conclude that $x_1 = x_5 = x_7 = 1$, $x_2 = x_3 = x_4 = x_6 = x_8 = 0$ is an optimal solution of the original P program. (For maximum weight clique algorithms see [1].)

The reader can notice that there are no edges going from the set of nodes $\{A_2, A_3\}$ to the set of nodes $\{A_5, A_6, A_7, A_8\}$ in the graph G and so the set

$$W_1 = \{A_2, A_3\}, \quad W_2 = \{A_1, A_4\}, \quad W_3 = \{A_5, A_6, A_7, A_8\}$$

form a splitting partition of the node set of the graph G. Consequently, the original linear program P can be replaced by two smaller programs P_1 and P_2. These smaller program are contained in Table 3. A more systematic way to locate splitting partitions can be found in [4].

Table 3. The two smaller zero-one linear programs P_1 and P_2 we get from the program P in the toy example.

	x_1	x_2	x_3	x_4	$\in$	$\{0,1\}$
	1	3	2	1	$\to$	max
(1)	1		1		$\leq$	1
(2)			1		$\leq$	1
(3)		1			$\leq$	1
(4)		1		1	$\leq$	1
(5)		1			$\leq$	1
(6)			1		$\leq$	1

	x_1	x_4	x_5	x_6	x_7	x_8	$\in$	$\{0,1\}$
	1	1	4	2	2	3	$\to$	max
(1)	1			1			$\leq$	1
(2)			1			1	$\leq$	1
(3)			1	1			$\leq$	1
(4)		1	1				$\leq$	1
(5)					1	1	$\leq$	1
(6)				1	1		$\leq$	1

Acknowledgments: This research was supported by National Research, Development and Innovation Office - NKFIH Fund No. SNN-135643.

References

1. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo M.: The Maximum Clique Problem. In: Handbook of Combinatorial Optimization Vol. 4, Kluwer Academic Publisher, 1999.
2. Gellner, A., Lamm, S., Schulz, Ch., Strash, D., Zavalnij, B.: Boosting Data Reduction for the Maximum Weight Independent Set Problem Using Increasing Transformations. In: Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX21) SIAM (2021) pp. 128–142.
3. Szabó, S.: Parallel algorithms for finding cliques in a graph. Journal of Physics Conference Series. **268** 2 (2011) 012030.
4. Szabó, S., Zavalnij, B.: Splitting partitions and clique search algorithms, In: Middle-European Conference on Applied Theoretical Computer Science. Editors: A Brodnik, G. Galambos, B. Kravsek, Ljubljana, 2019. pp. 75–78.

Solving a dynamic route planning problem in the area of patient transport with mixed integer linear programming

Martin Tóth[1], Tamás Hajba[2] and Adrián Horváth[3]

[1] Széchenyi István University, Győr, Hungary
[2] Széchenyi István University, Győr, Hungary
[3] Széchenyi István University, Győr, Hungary
martin.toth0127@gmail.com

Abstract: Consumer behavior has undergone a major transformation in recent years. On-demand has become one of the most important expectations. There is a very short time available in passenger and freight areas to process, order and meet incoming needs. In order for service providers to meet this through the efficient use of available resources, they need to take a new planning approach to maintain their efficiency and maintain a high standard of service. Within passenger transport, there is a sub-sector that has received less emphasis in recent decades. And this is patient transport. Sick people have the opportunity to use patient transport services, but the route organization of these services is not efficient enough, the service is expensive. Yet there would be a need for a service that can help people travel to hospitals at the right level. In this article, we present the solutions available abroad and a suitable mathematical model that can solve the transport organization task quickly and efficiently in the case of a vehicle. We also take into account the fact that patients can set their own maximum travel time and it is not up to the operating company to decide. We also pay attention for that there are patient who needs more space on the vehicle (e.g. wheelchair users), so the passengers are heterogeneous in this system. The usability of the model is also illustrated through an example.

Keywords: dial a ride problem, ridesharing, patient transport, mixed integer linear programming

Introduction

Expectations regarding the quality of passenger and freight transport services have been growing and changing in recent years. One reason for this is that transportation providers have introduced new services that have increased user convenience in order to gain a competitive advantage. Typical examples of this there are a possibility to given several delivery address, modification of transport addresses and the shortening of delivery time. In the field of passenger transport, various taxi companies, public transport services, while in the field of freight transport, parcel carriers and courier companies are the main stakeholders in these changes. While the new services provide users with a high degree of convenience, they incur significant additional costs for operators when planning and managing their implementation using traditional procedures. In order to achieve efficient operation and resource utilization, companies providing passenger and freight transport strive in some way to optimize the route of their vehicles and maximize their capacity utilization. Traditional route planning procedures are characterized by a large time difference between the planning and execution of transportation tasks and in the regard of passenger transportation, the organization of one shipment to one vehicle. However, this is inflexible to new consumer expectations and not effective enough. (Gillett, 1974).

This special route planning problem in the field of passenger transport is called the Dial a Ride problem (Daganzo et al., 2019).

The Dial-a-Ride Problem (DARP) consists of designing vehicle routes and schedules for n users who specify pickup and delivery requests between origins and destinations. Very often the same user will have two requests during the same day: an outbound request from home to a destination, and an inbound request for the return trip. In the standard version, transport is supplied by a fleet of m identical vehicles based at the same depot. The aim is to plan a set of minimum cost vehicle routes capable of accommodating as many requests as possible, under a set of constraints (Jean-François Cordeau, 2007). The most common example arises in door-to-door transportation services for elderly or disabled people (see, e.g., Toth and Vigo, 2002; Andrew Lim, 2018; Yves Molenbruch, 2017; Timothée Chane-Haï, 2021).

1. Problem description

One specific area of passenger transport is patient transport. While there is an increasing emphasis on different passenger services, this is not true with patient transport. Patient transport services in Hungary use inflexible planning procedures and focus on only 1-1 patients. At present, patient transport services in Hungary have the following characteristics:

- Door-to-door service
- The patient travels alone (Family members may be in the vehicle)
- Transport for the disabled and inpatients is also possible
- Patients can determine when they want to arrive at the hospital
- The amount of the round trip is between approx. 15,000 and 30,000 HUF (approx. 40 - 80 EUR). It is cheaper within the city, more expensive when traveling to or from the countryside.

These options provide a high-quality service to patients, but are very expensive. These services are mostly used by those who have some serious illness. On the other hand, people who only go for tests choose a different solution. These solutions can be provided by public transport, using your own vehicle, using different taxi companies, or approaching the hospital on foot or by bicycle.

At present, there is no patient transport service in Hungary that collects patients and transports them together to the hospital.

In our study we are looking for solution for a system, where patients are heterogeneous (normal patients and patients with reduced mobility (eg wheelchair users)). These characteristics must be taken into account when using the vehicle's capacity (eg wheelchair uses need more space on the vehicle). And we have to pay attention for, that the patients have different need about travelling time. So in this system they can they can determine the maximum amount of travel time they are willing to spend on the vehicle. Generally, this time is set uniformly for passengers by the operating companies.

2. Presentation of the model used for route planning

There is a depot, a hospital, and set of N locations from which patients must be transported to the hospital. Patients are transported by 1 vehicle. At the first delivery task it leaves the depot, the start point of the other tasks is the hospital. Then returns to the depot after taking the last patient to the hospital. The depot has an opening hours; the vehicle must leave the depot and return to the depot within this time interval. It is known how many patients and wheelchair users the vehicle can carry at one time. For each (i, j) pair, where the elements of i and j (depot, hospital, locations) are known, the time from i to j and the distance between i and j are known. The following are known for each location:

- the number of patients and disabled patients awaiting removal
- a date before which patients may not be admitted to the vehicle at that location
- a time by which patients must arrive at the hospital at the latest;
- maximum travelling time which patients cannot be longer on the vehicle.
- the time of boarding the vehicle for patients at that location

It is assumed that the number of patients and disabled patients at any location does not exceed the capacity of the vehicle. We also stipulate that if a vehicle goes to a particular location, it will pick up all the patients in a wheelchair waiting there. The goal is to determine the route of the vehicle that minimizes the total travel distance.

2.1. Parameters

N:	Number of pick up locations
0:	Depot (starting point)
N+1	Hospital
K:	Number of delivery tasks
D_{ij}:	Travel distance between points i and j.
$T_{i,j:}$	Travel time between points i and j.
C_p:	Maximum capacity of the vehicle for patients
C_{dp}:	Maximum capacity of the vehicle for disabled patients

- I_i: Maximum travel time for different patients.
- p_i: The number of patients in the point i.
- dp_i: The number of disabled patients in the point i.
- a_i: Boarding time in the point i.
- A_0: The opening time of the service company
- B_0: Closing time of the service company
- $t_{req:}$ Earliest admission date for patients
- $t_{arr:}$ Patient requested arrival time at the hospital

2.2. Variables

- $start^k$: Starting time of K task.
- end^k: Ending time of K task.
- $p_{num\ i}$: The number of patients in the vehicle after new patients got into the vehicle at point i.
- $dp_{num\ i}$: The number of disabled patients in the vehicle after new patients got into the vehicle at point i.
- $x_{i,j}^k$: Binary variable, value 1, if the vehicle is will go to point j after point i. otherwise 0.
- w_i: Travel time is for patient of i.
- m_i: Time of arrival at point j.

2.3. Objective function

The goal of the mathematical model is to minimize the travel distance, what the vehicle has to do.

$$\min \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} \sum_{k=1}^{K} D_{i,j} \ * x_{i,j}^{k} \tag{1}$$

2.4. Restrictive conditions

Vehicle only go once to each point:

$$\sum_{i=0}^{N+1} \sum_{k=1}^{K} x_{i,j}^{k} = 1 \qquad \forall\ 1 \leq j \leq N \tag{2}$$

Customers do not be start or end points:

$$\sum_{j=0}^{N+1} x_{i,j}^{k} - \sum_{J=0}^{N+1} x_{j,i}^{k} = 0 \qquad \forall\ 1 \leq j \leq N \tag{3}$$

The starting point for the first delivery task has to be the depot:

$$\sum_{i=1}^{N} x_{0,i}^{1} = 1 \qquad (4)$$

Do not arrive at the depot in the first trip, but return to the depot in the last trip:

$$\sum_{i=1}^{N} x_{i,0}^{1} = 0 \qquad (5\ a)$$

$$x_{n+1,0}^{k} = 0 \qquad 1 \leq k \leq K \qquad (5\ b)$$

Hospital can not be the starting point for vehicle in the first round:

$$\sum_{i=1}^{N} x_{n+1,i}^{1} = 0 \qquad (6)$$

Vehicle has to arrive at the hospital in the first round:

$$\sum_{i=1}^{N} x_{i,n+1}^{1} = 1 \qquad (7)$$

From the 2nd round the depo can not be the starting and the end point:

$$\sum_{i=1}^{N} x_{0,i}^{k} = 0 \qquad 2 \leq k \leq K \qquad (8)$$

Starting from round 2, the depot cannot be an endpoint:

$$\sum_{i=1}^{N} x_{i,0}^{k} = 0 \qquad 2 \leq k \leq K-1 \qquad (9)$$

Starting from round 2, hospital is the starting and the end point:

$$\sum_{i=1}^{N} x_{i,n+1}^{k} = \sum_{i=1}^{N} x_{n+1,i}^{k} \qquad 2 \leq k \leq K \qquad (10)$$

A k. the vehicle can only go to the hospital once per round:

$$\sum_{i=1}^{N} x_{i,n+1}^{k} \leq 1 \qquad 2 \leq k \qquad (11)$$

The change in the number of patients in the vehicle at the loading points:

$$p_{num\ i} + p_j + M * \left(x_{i,j}^{k} - 1\right) \leq p_{num\ j}$$
$$0 \leq i \leq N+1;\ 1 \leq j \leq N;\ 1 \leq k \leq K \qquad (12)$$

Change in the number of patients with reduced mobility in the vehicle at the loading points:

$$dp_{num\ i} + dp_j + M * \left(x_{i,j}^k - 1\right) \leq dp_{num\ j}$$
$$0 \leq i \leq N+1;\ 1 \leq j \leq N;\ 1 \leq k \leq K \qquad (13)$$

When the vehicle begins the transportation task, it must be ensured in the model that the number of vehicle capacity for both passengers and patients with reduced mobility starts from 0. Transport tasks can start from the depot and the hospital, so the conditions are needed for these starting points:

$$p_{num\ 0} = 0 \qquad (14)$$

$$p_{num\ n+1} = 0 \qquad (15)$$

$$dp_{num\ 0} = 0 \qquad (16)$$

$$dp_{num\ n+1} = 0 \qquad (17)$$

The number of patient in the vehicle cannot higher than the capacity of the vehicle:

$$p_{num\ i} \leq Cp \qquad (18)$$
$$dp_{num\ i} \leq Cp \qquad (19)$$

In the model, care shall be taken to ensure that the vehicle is fitted with the i. the time in point j shall be less than that in point j. date is specified in point. Thus, the transport tasks that arise take into account the passengers' requested pick-up times. However, time factors that affect the travel time between points i and j (eg boarding time in point i) must also be taken into account. In addition, when determining transport tasks, the desired time of arrival at the hospital should not be compromised:

$$m_i + T_{i,j} + a_i + M * \left(x_{i,j}^k - 1\right) \leq m_j$$
$$1 \leq i \leq N;\ 1 \leq j \leq N;\ 1 \leq k \leq K \qquad (20)$$

$$m_i + T_{i,n+1} + a_i + M * \left(x_{i,n+1}^k - 1\right) \leq end^k \qquad 1 \leq k \leq K \qquad (21)$$

$$start^1 + T_{0,i} + M * \left(x_{0,i}^1 - 1\right) \leq m_i \qquad 1 \leq i \leq N \qquad (22)$$

$$start^k + T_{n+1,i} + M * \left(x_{n+1,i}^k - 1\right) \leq m_i \qquad 2 \leq k \leq K \qquad (23)$$

Consideration of patient-requested boarding times:

$$t_{req\ i} \leq m_i \qquad 1 \leq i \leq N \qquad (24)$$

The i. Include the travel time and boarding times for patients admitted under (i) and (j) in the travel time of patients admitted under:

$$a_i + T_{i,n+1} + M * \left(\sum x_{i,n+1}^k - 1\right) \leq w_i$$
$$1 \leq i \leq N;\ 1 \leq j \leq N+1;\ 1 \leq k \leq K \qquad (25\ a)$$
$$m_j - m_i + w_j + M * \left(\sum x_{i,j}^k - 1\right) \leq w_i$$
$$1 \leq i \leq N;\ 1 \leq j \leq N+1;\ 1 \leq k \leq K \qquad (25\ b)$$

Patients should also be required to determine the length of maximum time they would like to be in the vehicle:

$$I_i \geq w_i \tag{26}$$

The start of the first round cannot be bigger than the opening date of the company:

$$start^1 \geq A_0 \tag{27}$$

The start time of the last round and the travel time of the vehicle from the hospital to the depot cannot be bigger than the closing time of the company:

$$end^k + T_{n+1,0} \leq B_0 \tag{28}$$

The start time of k+1 circle should be bigger than at the end of the round k:

$$start^{k+1} \geq end^k \qquad 1 \leq k \leq K-1 \tag{29}$$

The start time of the k circle must be less than endpoint of k circle:

$$start^k \leq end^k \qquad 1 \leq k \leq K \tag{30}$$

The vehicle departing from the i-th place can not go to the i-th place:

$$x_{i,i}^k = 0 \qquad 0 \ \leq \ \mathrm{i} \ \leq \ \mathrm{N}+1;\ 1 \leq k \leq K \tag{31}$$

Modell must pay attention for the patients requested arrival time

$$t_{arr\ i} \geq m_i + w_i \tag{32}$$

The customers have to arrive to the destination point between the requested time and before it with 15 minutes:

$$t_{arr\ i} - 15 \leq m_i + w_i \tag{33}$$

3. Conclusion

In this paper, we presented mathematical model, which is based on mix integer linear programming and tested in IBM ILOG CPLEX Optimization Studio. The result shows, that we can solve such a Dial – a – Ride problems, where customers can determine to earliest loading time, latest delivery time and the maximum length of travel time. The customer needs were known in advance, but this model can also use for dynamic system too, just we have to use Sequential planning.

The mentioned model can lead to better and more cost-effective solutions, because it can guarantee optimal solution of delivery tasks on faster way, than with traditional on-demand systems. Operating companies can maximize the capacity utilization of their fleet, and this will also reduce CO2 emissions too. Furthermore, customer get a such service, which they can be modified to their own convenience. This can help to companies to sell their services against others.

We will also focus on the following aspects later, in the interest of this model would be used by companies. These aspects are:

- Passenger conflicts (eg passengers with an infectious could not travel with other patients)
- Vehicle cleaning time (eg after certain transport tasks the vehicle has to stop for cleaning and disinfection)
- More vehicles be in the system
- The model has to also determine delivery tasks from hospital, not only to there.
- The starting point of the transport task should be the last end point of the previous transport task (this is relevant if the vehicle is already picking up passengers from the hospital).

This requires needed further research.

Acknowledgement

Project no. TKP2021-NKTA-48 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NKTA funding scheme.

References:

1. B. E. Gillett, L. R. Miller: A heuristic algorithm for the Vehicle-Dispatch Problem. Operational research, (1974).
2. Jean-François Cordeau, Gilbert Laporte: The dial-a-ride problem: models and algorithms, Springer Science+Business Media (2007).
3. C. F. Daganzo, Y. Ouyang: A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride, Transportation Research, (2019).
4. Zhixing Luo, Mengyang Liu, Andrew Lim: „A Two-Phase Branch-and-Price-and-Cut for a Dial-a-Ride Problem in Patient Transportation," Transportation Science, (2018).
5. Yves Molenbruch, Kris Braekers: An Caris, Greet Vanden Berghe, „Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation," Computers & Operations Research, pp. 58-71, (2017).
6. T. Chane-Hai., S. Vercraene, T. Monteiro: „Sharing a ride time constraint in a multi-trip dial-a-ride problem. An application to the non-urgent patient transportation problem," HAL, (2021).
7. Daniele Vigo, Paolo Toth: The Vehicle Routing Problem, Philadelphia: Society for Industrial and Applied Mathematics (2002).

The effect of problem complexity on the optimal solution of Simple Assembly Line Balancing Problems with learning

Zakaria Zine El Abidine[1], Imre Dimény[2], and Tamás Koltai[3]

Budapest University of Technology and Economics, Department of Management and Business Economics, Budapest, Hungary
zakariazineelabidine@edu.bme.hu, dimeny.imre@gtk.bme.hu, koltai.tamas@gtk.bme.hu

Abstract. The objective of simple assembly line balancing problem type1 is to minimize the number of workstations organized to perform tasks with precedence constraints. In assembly lines, where manual tasks are executed repeatedly, learning can have a significant effect on the operation. In this case, when the minimal number of workstations is determined, the change of task times as a consequence of learning must be considered. This paper investigates the effect of learning on the minimum number of workstations when simple assembly line balancing problems are solved. A modified simple assembly line balancing model incorporating the learning effect is formulated, and two benchmark problems with different characteristics are analysed.

Keywords: Assembly line balancing, simple assembly line, learning curve, mathematical programming.

1. Introduction

Assembly lines, which demand manual labor, are an example of a manufacturing environment where learning effects are essential to determine task time. Balancing an assembly line implies assigning assembly tasks to workstations to maximize a specific performance measure without violating precedence constraints. Most techniques focus on simple assembly line balancing problems (SALBP) of types 1 and 2 [2]. The type1 problem (SALBP-1) aims to minimize the number of workstations required to meet a specific cycle time, whereas the type2 problem (SALBP-2) aims to minimize the cycle time for a fixed number of workstations [1].

Because of workers' learning, task times may decrease. That is, the lower the task times, the higher the quantity of assembled products. As a result of task time decrease the best balancing solution changes over time. In this paper SALP-1 with the learning effect is investigated.

This research aims to solve the simple assembly line balancing problem type1(SALBP-1) by incorporating a learning curve (LC) into the basic model. For this purpose, the Wright's power function was selected [5]. The main contribution of the paper is the formulation of a modified SALBP model, which can be used to determine the minimum number of workstations when the learning effect occurs. The effects of the learning curve, the learning rates, and the problem complexity on the optimal solutions are investigated.

The rest of this paper is organized as follows. Section 2 outlines the formulation of the modified SALBP-1 model. Section 3 describes the benchmark problems. Section 4 illustrates the application of the modified model and discusses the obtained results. Finally, the main results are summarized, and some conclusions are given.

2. Formulation of the modified SALB-1 model

In this section, the modified SALBP-1 model used in this paper is presented. The applied notations are summarized in table 1. Tasks are numbered in a continuously increasing order. The number i assigned to a task is called the task index. We refer to a task either by its name or task index. Those tasks which are not succeeded by any other task are called last tasks. The index set of last tasks is denoted by F.

Workstations are also numbered in a continuously increasing order. The first workstation is numbered 1, and the last workstation is numbered J. The number j assigned to a workstation is called the workstation index. Workstations are referred to in this paper by the workstation index. Similarly, workers are numbered in a continuously increasing order. We refer to a worker by its index k, which is a number between 1 and K. is the number of available workers.

The assignment of tasks and workers to workstations is expressed with the x_{ijk} binary decision variable. If task i and worker k are assigned to the same workstation j, then x_{ijk} = 1, otherwise x_{ijk}= 0. Similarly, the assignment of workers to workstations is expressed with the y_{jk} binary decision variable. If worker k is assigned to workstation j, then y_{jk} = 1, otherwise y_{jk} = 0. The following integer linear programming formulation of the modified SALBP-1 is used in this paper,

$$Min\ (N) \tag{1}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K} t_{ik}x_{ijk} \leq T_c \qquad \forall j \tag{2}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i \tag{3}$$

$$\sum_{j=1}^{J} j\left(x_{qjk} - x_{pjk}\right) \geq 0 \qquad \forall (p,q) \in R \tag{4}$$

$$x_{ijk} \leq y_{jk} \qquad \forall\ i,j,k \tag{5}$$

$$\sum_{j=1}^{J} y_{jk} \leq 1 \qquad \forall k \tag{6}$$

$$\sum_{k=1}^{K} y_{jk} \leq 1 \qquad \forall j \qquad (7)$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} jx_{ijk} \leq N \qquad \forall i \in F \qquad (8)$$

3. Problems' description

To illustrate the performance of the presented model, let us consider two SALBP-1 benchmark problems (named by the authors Mansoor and Rosenberg) with different characteristics and complexity levels taken from Scholl (1995) [4]. The precedence graphs of the two problems are as follows:

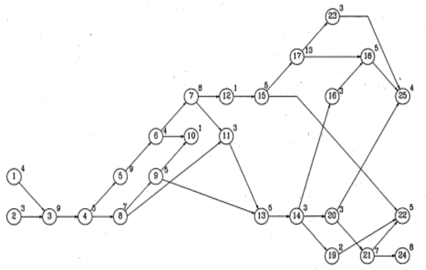

Fig.1. Rosenberg precedence graph.

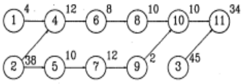

Fig.2. Mansoor precedence graph.

Table 2 summarizes the characteristics and the complexity measures for each of the two problems.

Table 2. Benchmark problems' characteristics and complexity measures.

Benchmark Problem	Characteristics		Complexity measures		
	T_c	N^*	I	$SO(\%)$	WR
Rosenberg	14	10	25	71.7	2.5
Mansoor	48	4	11	60	2.75

Comparing the complexity measures of the two problems, we notice that Rosenberg problem has a higher number of tasks (25 against 11) and strength order (71.1 against 60) with a smaller west ratio (2.5 against 2.75). It can be concluded that Rosenberg problem is more complex than Mansoor problem.

Table 1. Summary of notation applied in the model.

Summary of notations:

Indices:

i = index of tasks (i=1 ,…, I),
p = index of subtasks,
q = index of subtasks,
j = index of workstations (j=1,.., J),
k = index of workers (k=1,…, K),

Parameters:

I = number of tasks,
J = maximum number of workstations,
N^* = minimum number of stations (the result of the station number minimization of the basic SALBP-1 model),
K = number of available workers ($K>=N^*$),
t_i = initial task time of task i,
T_c = cycle time,
Q_{ik} = rank of the part on which task i is performed by worker k,
t_{ik} = time necessary to perform task i by worker k on Q_{ik} ($t_{ik} = t_i \, . \, (Q_{ik})^b$),
b = power of the learning curve function,
L = learning rate ($L = 2^b$),
SO = strength order of the precedence graph ($SO = I.(I-1)/2$),
WR = west ratio (I/N^*),

Sets:

F = set of final tasks, $i \in F$, if task i does not precede any other task,
R = set of pair of indices which belong to tasks with precedence relations, that is, $(p; q) \in R$, if task p immediate precedes task q,

Decision variables:

N = objective function variable for the number of workstations,
N^* = minimal number of workstations,
x_{ijk} = 0-1 decision variable, if x_{ijk}=1, then task i is performed by worker k at workstation j, otherwise x_{ijk}=0,
y_{jk} = 0-1 decision variable; if y_{jk} =1, then worker k is assigned to workstation j, otherwise
y_{jk} = 0

4. Modified SALBP-1 model application and results

4.1 The effect of learning on the optimal number of WSs

The Wright learning curve was chosen to analyze the effect of learning on the optimal number of stations. In this case, three learning rates were selected: 0.85, 0.9, and 0.95, which gives the corresponding three values of b respectively: -0.23, -0.15, and -0.07.

For the sake of simplicity, it is assumed that work-in-process inventory cannot accumulate between stations. This implies that the worker assigned to station $(j + 1)$ cannot start the execution of task(s) unless the worker assigned to station j has finished the tasks of the upcoming part.

The solution of the modified SALBP-1 model defined for the chosen learning curve demands a flexible mathematical modeling tool. Such a tool is provided by the AIMMS Prescriptive Analytics Platform, which is often applied for solving commercial optimization problems [3].

To track the evolution of the minimum number of workstations as the workers' learning evolves, the model is solved after the production of each unit. This means that when a unit exits the last station, the station times are updated according to the learning curve, and the problem is resolved. Since the learning effect is dominant only at the beginning of the production run, the solution is repeated for the first 100 produced units.

Rosenberg problem. Figure 3 shows the minimum number of WSs in function of the output units Q while the learning effect occurs. It can be seen that for all the three learning rates (L=0.85/0.9/0.95), the minimum number of workstations after the production of the first unit(Q=1) is 10. At this stage, the workers have not progressed yet along the learning curve, since the learning effect has not started yet.

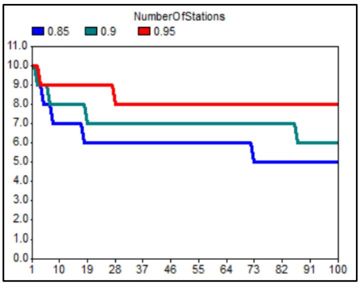

Fig. 3. Rosenberg problem: Evolution of the minimum number of WSs during learning.

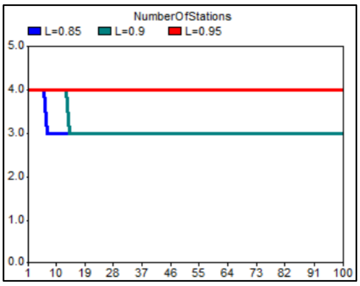

Fig. 4. Mansoor problem: Evolution of the minimum number of WSs during learning.

As the workers repeat the assigned tasks, task times decrease exponentially causing the station times to decrease. The station times keep dropping until fewer stations are required to satisfy the predetermined cycle time constraint. For instance, for L=0.85 the minimum number of WSs dropped from 10 to 9 at Q=2, to 8 at Q=4, to 7 at Q=7, to 6 at Q=17, and finally to 5 at Q=72. In the case of L=0.9 and 0.95, we could only reach 7 and 8 stations, respectively, because of the slow learning rates.

Mansoor problem. Figure 4 shows that the minimum number of WSs starts at 4 for the three learning rates (L=0.85/0.9/0.95). As the learning effect evolves, for L=0.85 and 0.9 the minimum number of WSs decreased to 3 at Q=6 and 13, respectively. However, for L=0.95, no drop in the minimal number of WSs is witnessed because of the slow learning effect.

5. Conclusion

In this paper, a modified SALBP-1 model incorporating the learning effect is formulated and applied. The modified SALBP-1 model developed in section 2 is illustrated with two benchmark problems with different characteristics and complexity levels in section 4. The Wright learning curve was selected for our model because of its extended use in practice. The model, however, can be adapted to any other learning curve.

In the illustrative examples, three different learning rates were chosen: 0.85, 0.9, and 0.95. The results showed that regardless of the learning rate, the minimum number of workstations at the start of production (Q=1 unit) is the same as the basic SALBP-1 model. However, as production progresses, it decreases along with the output units due to the learning effect. Additionally, the higher the learning rate, the slower the minimum number of workstations decrease.

The results also revealed that a problem with higher complexity level requires more optimal number of stations at the start of production and therefore has a higher potential to witness more stations' drops along with the produced units due to the learning effect.

References

1. Gansterer, M., & Hartl, R. F. One-and two-sided assembly line balancing problems with real-world constraints. International Journal of Production Research, 56(8), 3025-3042 (2018).
2. Lolli, F., Balugani, E., Gamberini, R., & Rimini, B. Stochastic assembly line balancing with learning effects. IFAC-PapersOnLine, 50(1), 5706-5711 (2017).
3. Roelofs, M., Bisschop, J: AIMMS the User's Guide. AIMMS B.V (2018).
4. Scholl, Armin. Data of assembly line balancing problems. Techn. Hochsch., Inst. für Betriebswirtschaftslehre (1995).
5. Tamás, A., Koltai, T. Application of Learning Curves in Operations Management Decisions, Periodica Polytechnica Social and Management Sciences, 28(1), pp. 81–90 (2020).