Open access • Journal Article • DOI:10.1080/00207543.2015.1061221

On the design of custom packs: grouping of medical disposable items for surgeries

Brecht Cardoen, Jeroen Belien, Mario Vanhoucke

Institutions: Katholieke Universiteit Leuven, University College London

Published on: 09 Jul 2015 - International Journal of Production Research (Taylor & Francis)

Topics: Programming paradigm and Integer programming

International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

On the design of custom packs: grouping of medical disposable items for surgeries

Brecht Cardoen[ab], Jeroen Beliën[c] & Mario Vanhoucke[ade]
[a] Technology & Operations Management, Vlerick Business School, Gent, Belgium
[b] Faculty of Business and Economics, Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium
[c] Campus Brussels, Center for Information Management, Modelling and Simulation, KU Leuven, Brussel, Belgium
[d] Faculty of Economics and Business Administration, Department of Management Information and Operations Management, Universiteit Gent, Gent, Belgium
[e] Management Science and Innovation, University College London, London, UK
Published online: 09 Jul 2015.

To cite this article: Brecht Cardoen, Jeroen Beliën & Mario Vanhoucke (2015): On the design of custom packs: grouping of medical disposable items for surgeries, International Journal of Production Research, DOI: 10.1080/00207543.2015.1061221

To link to this article: http://dx.doi.org/10.1080/00207543.2015.1061221

International Journal of Production Research, 2015
http://dx.doi.org/10.1080/00207543.2015.1061221

On the design of custom packs: grouping of medical disposable items for surgeries

Brecht Cardoen[a,b]*, Jeroen Beliën[c] and Mario Vanhoucke[a,d,e]

[a]*Technology & Operations Management, Vlerick Business School, Gent, Belgium;* [b]*Faculty of Business and Economics, Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium;* [c]*Campus Brussels, Center for Information Management, Modelling and Simulation, KU Leuven, Brussel, Belgium;* [d]*Faculty of Economics and Business Administration, Department of Management Information and Operations Management, Universiteit Gent, Gent, Belgium;* [e]*Management Science and Innovation, University College London, London, UK*

(*Received 25 April 2014; accepted 28 May 2015*)

A custom pack combines medical disposable items into a single sterile package that is used for surgical procedures. Although custom packs are gaining importance in hospitals due to their potential benefits in reducing surgery setup times, little is known on methodologies to configure them, especially if the number of medical items, procedure types and surgeons is large. In this paper, we propose a mathematical programming approach to guide hospitals in developing or reconfiguring their custom packs. In particular, we are interested in minimising points of touch, which we define as a measure for physical contact between staff and medical materials. Starting from an integer non-linear programming model, we develop both an exact linear programming (LP) solution approach and an LP-based heuristic. Next, we also describe a simulated annealing approach to benchmark the mathematical programming methods. A computational experiment, based on real data of a medium-sized Belgian hospital, compares the optimised results with the performance of the hospital's current configuration settings and indicates how to improve future usage. Next to this base case, we introduce scenarios in which we examine to what extent the results are sensitive for waste, i.e. adding more items to the custom pack than is technically required for some of the custom pack's procedures, since this can increase its applicability towards other procedures. We point at some interesting insights that can be taken up by the hospital management to guide the configuration and accompanying negotiation processes.

Keywords: health care services; combinatorial optimisation; integer linear programming; case study

1. Introduction

The provision of accessible and cost-efficient health care is seen as one of the major service challenges of today's society. In 2012, member countries of the Organisation for Economic Co-operation and Development spent on average 9.5% of their gross domestic product on health care (OECD 2012). Hospitals contribute for about one-third of these national health expenditures (Poisal et al. 2007). One unit that is of particular interest within hospitals is the operating theatre. Since this facility generates about 42% of a hospital's revenue and a proportionate share of its costs, it has a significant impact on financial performance. Improving the surgical throughput by just one additional case per day per operating room may generate additional revenue of 4–7 million dollar for an average-sized hospital (HFMA 2005).

Multiple strategies exist to increase patient throughput in the operating room. First, one can adjust the planning and scheduling practice within the operating theatre and its related facilities. Zhang et al. (2009) show how an improved assignment of capacity to specialties reduces inpatient's length of stay (LOS) waiting for the surgery. By reducing unnecessary LOS and increasing operating theatre utilisation, they positively impact patient throughput. We refer to Cardoen, Demeulemeester, and Beliën (2010), Guerriero and Guido (2011) and May et al. (2011) for recent literature reviews on surgery planning and scheduling contributions. Second, one can impact the operative time. Technological advances such as robotics may shorten the required surgical time, which enables the steady shift from inpatient surgery to outpatient surgery (Toftgaard and Parmentier 2006). Third, one can focus on the non-operative time in the operating theatre. Harders et al. (2006) define non-operative time as the room turnover time plus anaesthesia induction and emergence time, in which the room turnover time encompasses the time to clean and ready an operating room for the next case. Improvements in non-operative time can be obtained, for instance, by reassigning tasks amongst nursing and logistics personnel, by introducing lean techniques and therefore excluding unnecessary steps and actions in the according processes, or – as explained in this paper – by introducing custom procedure trays or *custom packs* during the setup of the surgery.

*Corresponding author. Emails: brecht.cardoen@vlerick.com, brecht.cardoen@kuleuven.be

In its strict sense, a custom pack is defined as a single, sterile, customised, disposable pack that contains all the supplies needed for a particular surgical procedure. It is assembled by a vendor to the specifications of the health care facility (Gellman 1988). However, custom packs may also include only a subset of the supplies needed for a specific surgery. In this case, single-item picking or even a selection of other custom packs may complete the required set of medical items for the surgery. Moreover, custom packs do not have to be developed for a single surgical procedure: they can be used for a set of surgical procedures, or for a specific surgeon instead of a specific procedure. This study deals with the configuration of custom packs for a given set of surgical procedures, surgeons performing these procedures and medical items required for the procedures so that the physical contact between staff and the medical materials during setup of the surgery is minimised. We will measure this physical contact through what we will refer to as points of touch. Obviously, the minimum requirement of medical items for a procedure using a pack should be present at the time of surgery (i.e. included into the custom pack or additionally picked). Also, the number of custom packs to be configured is limited and therefore custom packs should be shared as much as possible among different surgeons and/or procedures.

The custom pack decision problem incorporates a set covering optimisation problem, which is known to be NP-hard (e.g. Garey and Johnson 1979). In Section 2, we describe how we extend the traditional set covering problem due to the allowance of excess items to custom packs. The introduction of waste is a counter-intuitive feature that seems of little use in optimisation problems where the objective is defined by efficiency. Making this feature explicit in the model, though, we show when and to what extent this strategy might prove useful. Although the optimisation problem is difficult, we describe easy modifications to an exact mixed integer linear programming (MILP) approach and hence contribute to a better understanding of heuristics based on linear programming (LP) on top of commonly applied metaheuristics (such as simulated annealing (SA), tabu search or genetic algorithms) or methods like branch-and-price.

The remainder of this paper is organised as follows. In Section 2, we further describe the concept of custom packs, show that it is both medically and financially interesting to examine, and provide a focused literature review. A mathematical formulation of the optimisation problem is introduced in Section 3, followed by the development of solution approaches in Section 4. We test the solution approaches both using a real case and a set of scenarios derived from the case to understand the sensitivity of the results. Section 5 introduces the test setting and reports on the computational performance of the solution procedures, whereas Section 6 is mainly directed to the managerial learning that stems from the case and the comparison of the scenarios. Section 7 concludes this paper and addresses some directions for future research.

2. Literature review

The introduction of custom packs into the operating theatre is beneficial for the clinical quality of care (see, e.g. Baines et al. 2001). Since custom packs directly reduce material handling effort, they decrease the risk that nurses contaminate sterile products. Also, less unwrapping of medical items is required, which results in a more stable air flow within the operating room and hence less impact of, for instance, dust particles. Custom packs also allow for a quick response to material intensive emergency services, such as a Caesarean section, due to speeding in surgery setup. Yet, the most important driver for using custom packs is the gain in operational efficiency and the resulting cost savings (Baines et al. 2001; Birk 2009; Gellman 1988; Normén and Evans 2013). As mentioned, custom packs are effective in reducing the time needed for surgery setup, for mainly two reasons. First, items are no longer individually packaged so that unwrapping the custom packs replaces the individual unwrapping of all medical items that are included. Second, medical items do not have to be sorted anymore since they are organised and sequenced within the custom pack, irrespective of the nurse who is in charge of this process. A decrease in non-operative time may translate in reduced operating room staffing costs (Dexter et al. 2003) or additional surgical caseload (e.g. Kharraja, Albert, and Chaabane 2009; Krupka, Sathaye, and Sandberg 2008). Less material handling is not only beneficial in the operating room during setup, but also reduces the effort of storing and picking disposable items. Moreover, the chance of not picking a requested item or picking wrong items is strongly reduced. Custom packs also seem to simplify tracking and tracing of materials as a single registration of the pack suffices for adequate billing, whereas this task can be cumbersome in the case of a single-pull system and hence prone to errors or omissions.

Unfortunately, not all consequences of custom pack usage are perceived as a clear advantage. Although custom packs reduce waste related to packaging as there is no need to package individual items, they often trigger the replacement of reusable items with disposable items. Linen aprons, for instance, are replaced by paper ones, ready to be thrown away after surgery. Another example of waste is mentioned by Gellman (1988) who states that if a custom pack proves to be unsterile (e.g. due to transportation), all items included into the custom pack are wasted. Also, waste may stem from exceeding the required number of medical items in a custom pack for a given procedure (Akridge 2005). Although one may consider this as bad practice, it can broaden the reach of a custom pack and increase its applicability over different procedures. We will elaborate on this idea and explicitly incorporate this option in our problem definition (see Section 3). Through a single-site, observational study conducted on the surgical instrumentation at a large academic medical centre in Chicago, Stockert and Langerman (2014) found that the per cent use of instruments across surgical specialties and multiple tray types is extremely low (13–22%). A similar observation was made by Chin et al. (2014) who found that the average instrument utilisation rate at the Department of Otolaryngology- Head and Neck Surgery at St Joseph's Hospital (London, ON, Canada) was less than 30%. The authors conclude that attention to tray composition may result in immediate and significant cost savings. Similar to the discussion on waste, it is unclear whether custom packs really contribute to the inventory holding process of hospitals. On the one hand, custom packs may reduce the single-pull inventory (working capital reduction) and reduce the number of stock keeping units, freeing up some of the limited space and reducing the effort of stock counting. On the other hand, the risk-averse character of medical institutions will not entirely eliminate single-pull inventory, so that space conditions may even deteriorate despite the often frequent delivery of packs. It should be noted that the vendor takes up an important role in the custom pack design and delivery. Since they sell their packs as custom-made, it is hard for hospitals to quickly substitute suppliers. This dependency also makes hospitals vulnerable towards out-of-stock. Therefore, vendors typically accumulate inventory of custom packs, though this action implies that requests of hospitals for changes to the content may result in a long depletion lead time until the new configuration will be put in place (Gellman 1988). Although vendors are normally considered to be sole suppliers of custom packs, Birk (2009) recently reports on an initiative in which several hospitals and facilities join forces to produce their custom packs in-house.

The configuration process of custom packs involves many stakeholders and consecutive phases in order to agree upon the content and the variety of custom packs to be developed. Akridge (2005) concludes that the degree to which disposable items can be standardised is crucial to the success of custom packs. In other words, to what extent can different surgeons use a uniform set of medical items to perform the same procedure type. Three important questions here are (i) is the medical item really needed (ii) is the number of a particular item appropriate and (iii) can we rationalise items with the same functionality but with, for instance, a different brand or size. Standardisation as such is a difficult task as stakeholders have different interests: surgeons prefer to use their own and familiar disposable items, the hospital prefers cost efficiency while vendors try to standardise over multiple hospitals to achieve economies of scale. Not only agreeing upon standardisation seems problematic, also assessing the efficiency impact of standardisation and the identification of potentially interesting custom pack configurations is difficult and time consuming. This paper introduces a model that deals with the underlying optimisation problem and hence may contribute in facilitating this negotiation process.

Literature provides a range of contributions in the field of product platform architecture (Jiao, Simpson, and Siddique 2007) that are related to our problem when we think of platforms being custom packs, products being procedures and components being medical items. While these contributions generally focus on a manufacturing or technological environment with a clear focus on a production cost function minimisation, we provide a service setting and introduce a proxy that, among other, indirectly includes costs, namely the points of touch. Our main reason to do so is that many outcomes in our surgery setting are financially difficult to estimate, such as clinical quality improvement realised by using custom packs. Also, contributions in the field of platform architecture see platforms for immediate use and benefit of supplier's in-house production, whereas in our case platforms are developed from a customer's point of view.

The optimisation problem at hand is related to the set covering problem, which we can define as follows (e.g. Bautista and Pereira 2007): Given a set or universe M, $|M| = m$ and n subsets $S_j \subseteq M$, $j \in N$, $N = \{1, \ldots, n\}$ each with a non-negative cost c_j, the objective is to find a minimum cost family of subsets S_j such that each element $i \in M$ belongs to at least one subset of the family. Set covering problems appear across different sectors, including health care (see, e.g. Stolletz and Brunner (2011) and Beliën et al. (2013) for recent examples). The custom pack problem, however, modifies the traditional set covering problem due to the restrictions on waste, originating from adding excess items to custom packs. This makes that solution approaches to the set covering problem are not readily applicable and justifies the development of an alternative solution approach. Considerable attention has been directed towards the development of efficient solution approaches for the set covering problem (Caprara, Toth, and Fischetti 2000). Given the complexity and relation to platform architecture, metaheuristic solution approaches such as SA (e.g. Agard and Penz 2009), tabu search (e.g. Khalaf, Agard, and Penz 2011) or genetic algorithms (e.g. Qu et al. 2011) might prove useful. Yet, we show that also LP can contribute to finding adequate solutions and hence provide an LP-based heuristic to solve large instances.

Recently, the application of optimisation techniques applied to the configuration of custom packs has been addressed by Dobson et al. (2014). They present a column generation approach and a heuristic algorithm for finding a low-cost trays configuration taking into account surgeons' preferences and surgical schedules. Another paper that uses a cost-driven objective function was introduced by Reymondon, Pellet, and Marcon (2008). The authors propose a methodology for grouping reusable medical devices (RMD) into packages so that costs, related to the sterilisation of medical equipment, are minimised. Note that this context differs from ours as custom packs only consider disposable items, eliminating the need to relate the problem to the sterilisation unit of the hospital. Reymondon, Pellet, and Marcon (2008) formulate a monetary multiobjective cost function comprising four elements: costs of RMD storage, cost of box package storage (i.e. type of wrapping), costs of process times and costs of non-used RMD. This cost is similar to our (disposable) waste cost, though we incorporate waste through a constraint and include that costs may vary according to the item that is not used. In doing this, we obtain a single-objective function that is not expressed in monetary value but in countable movements (i.e. the points of touch) and hence introduce an alternative to a cost-driven objective function. Interviews with field experts of multiple hospitals in Flanders indicate that estimating the true economic value of time of resources in the operating theatre or perioperative setting (e.g. storage processes) – and especially clinical outcome measures -may turn out to be difficult in practice for our custom pack setting. Using points of touch as a performance measure provides a means to impact many objectives at once. Also, including waste as a constraint and having a single objective can facilitate the translation of results to managerial insights (see Section 6) for our particular setting. Our objective does not include inventory, though we control for the number of custom packs to be configured through a constraint. Again, this may lead to interesting insights and also provides a mechanism to assess the marginal value of adding one more pack while keeping track of the increase in stock keeping units. Reymondon, Pellet, and Marcon (2008) mention a two-stage approach though only discuss the first stage in which the problem size is reduced by exploiting sharing potential of RMD, which they test on real size, though not real case, data. The second step, introducing an optimisation method, constitutes future research, though they already indicate that SA may prove useful based on previous work (Reymondon and Marcon 2005). Having both an exact and two heuristic approaches, we are able to numerically assess the heuristic performance. Also, we apply all solution methods to a real case, which should add value to the managerial insights stemming from the computational experiment.

3. Problem definition

Let x_{ip} denote an integer decision variable that equals the number of a medical item i in pack p. The pack p is used for procedure j when its binary decision variable y_{pj} is equal to 1. The binary consequence variable w_p equals 1 if at least one procedure is using pack p. We also introduce a parameter *weight* to define the trade-off between points of touch and the introduction of one additional custom pack. We represent the number of single-pulled items of type i for procedure j by the integer decision variable z_{ij} and the number of excess items of type i for procedure j by the integer consequence variable red_{ij}. Furthermore, let D_j be the annual demand for procedure j, C_i the cost of one item of type i, N_{ij} the required number of items of type i for procedure j and B the total budget that is allowed to be spent on redundant or excess items. We refer to Table 1 for an overview of all indices, sets, parameters and variables.

Now we can formulate the model as follows:

$$\text{Minimise} \sum\nolimits_p \sum\nolimits_j \left(D_j \cdot y_{pj}\right) + \sum\nolimits_i \sum\nolimits_j \left(D_j \cdot z_{ij}\right) + weight \sum\nolimits_p w_p \tag{1}$$

$$\text{S.T.} \sum\nolimits_i \sum\nolimits_j C_i \cdot \mathrm{red}_{ij} \cdot D_j \leq B \tag{2}$$

$$\sum\nolimits_p w_p \leq |P| \tag{3}$$

Table 1. Overview of the indices, sets, parameters and variables.

$i \in I$	Index and set	Disposable items
$p \in P$	Index and set	Custom packs
$j \in J$	Index and set	Procedures
y_{pj}	Variable	Equals 1 if pack p is used for procedure j, 0 otherwise
x_{ip}	Variable	Number of items i in pack p ($\in \mathbb{N}$)
w_p	Variable	Equals 1 if pack p is in use, 0 otherwise
z_{ij}	Variable	Number of items i single-pulled for procedure j ($\in \mathbb{N}$)
red_{ij}	Variable	Number of excess items i for procedure j ($\in \mathbb{N}$)
D_j	Parameter	Demand for procedure j
C_i	Parameter	Cost of item i
N_{ij}	Parameter	Required number of items i needed to perform procedure j
B	Parameter	Allowed cost associated with excess items
weight	Parameter	Trade-off between points of touch and an additional custom pack

$$w_p - y_{pj} \geq 0 \quad \forall p \in P, \quad j \in J \tag{4}$$

$$\sum\nolimits_p \left(x_{ip} \cdot y_{pj} \right) + z_{ij} - \text{red}_{ij} = N_{ij} \quad \forall i \in I, \quad j \in J \tag{5}$$

$$y_{pj} \in \{0, 1\} \quad \forall p \in P, \quad j \in J \tag{6}$$

$$w_p \in \{0, 1\} \quad \forall p \in P \tag{7}$$

$$x_{ip} \in \mathbb{N} \quad \forall i \in I, \quad p \in P \tag{8}$$

$$z_{ij}, red_{ij} \in \mathbb{N} \quad \forall i \in I, \quad j \in J \tag{9}$$

Expression (1) introduces the objective function in which the annual number of points of touch over all procedures is minimised. Points of touch may originate from the use of custom packs and from single-pull inventory. Next to the points of touch, also the number of custom packs to be developed and configured can be minimised whenever the weight is set to a non-zero positive number. This weight should then reflect how many touch points the introduction of an extra custom pack is worth. Unfortunately, in reality, this trade-off is difficult to assess. For the remainder of this paper, we assume this weight to be zero and solely focus on the points of touch resulting from the custom pack design choices. Inequality (2) restricts the annual spent on redundant items to be less than or equal to the foreseen budget. Equation (3) fixes the maximum number of packs to be configured to $|P|$. Note that there is no use in setting $|P| > |J|$, as all procedures would already have their own, tailored pack. Constraint set (4) ensures that, if a custom pack is used for a given procedure, this pack has to be configured. Equation set (5) imposes that for each procedure at least the minimum requirement of all of its medical items should be covered, either through the use of custom packs and/or single-pull units. Redundant items occur when this supply exceeds the medically required number of the item, though redundancy will reasonably only materialise when no single-pull units for the item are needed, as enforced for the optimal solution by the objective function. The domains of the different decision and consequence variables are described by Expressions (6)–(9).

The model described by Expressions (1)–(9) is non-linear due to Expression (5), which exhibits a multiplication of two decision variables x_{ip} and y_{pj}. Let v_{ipj} denote a new decision variable to represent the number of items i in a pack p that is used for procedure j. In order to obtain a linear model, we can now substitute Expression (5) by Expressions (10)–(14):

$$\sum\nolimits_p v_{ipj} + z_{ij} - red_{ij} = N_{ij} \quad \forall i \in I, \quad j \in J \tag{10}$$

$$v_{ipj} \leq x_{ip} \quad \forall i \in I, \quad p \in P, \quad j \in J \tag{11}$$

$$v_{ipj} \leq y_{pj} \cdot M \quad \forall i \in I, \quad p \in P, \quad j \in J \tag{12}$$

$$v_{ipj} \geq x_{ip} - M + y_{pj} \cdot M \quad \forall i \in I, \quad p \in P, \quad j \in J \tag{13}$$

$$v_{ipj} \in \mathbb{R}^+ \quad \forall i \in I, \quad p \in P, \quad j \in J \tag{14}$$

Expression (10), similar to Expression (5), implies that for each procedure at least the minimum requirement of all of its medical items should be covered. Excess supply is captured by the consequence variable red_{ij}. The relationship between the variables x_{ip}, y_{pj} and v_{ipj} is described by Expressions (11)–(13). If a pack p is used for a procedure j (equivalently, if $y_{pj} = 1$), the corresponding v_{ipj} will take the value of x_{ip}. Else, if $y_{pj} = 0$, all corresponding v_{ipj} will be set to 0. Note that we cannot discard the x_{ip} variables as they ensure that the number of an item i in pack p is constant over all procedures for which the pack is used. The linear transformation provides a MILP formulation which will serve as a basis for the development of a two-phase heuristic in Section 4.

4. Solution methodology

We propose a two-phase heuristic that builds upon the strengths of mathematical programming without engaging into advanced modelling techniques. Despite the fact that heuristics cannot prove optimality of solutions, they might provide an adequate answer to finding good solutions when the exact procedure would get computationally too time consuming. In essence, the heuristic generates in a first phase a set of custom packs that might prove useful in minimising the points of touch. In a second phase, a MILP is defined which selects the best combination of custom packs defined in the first phase, while keeping the flexibility to still configure some custom packs completely from scratch during optimisation. In doing so, we include an opportunity for the algorithm to discover a custom pack configuration which we might have missed during the first phase and would contribute to the overall solution. As we only incrementally allow for additional packs in the second phase, we limit the number of variables and constraints needed due to linear transformation (see Section 5) and increase the probability to find a qualitative solution within a limited time frame. *Pseudocode 1* illustrates the heuristic procedure and describes in further detail how to determine a set of predefined custom packs in the first phase, which actually is a structured repetition of the exact method for smaller subproblems of the original problem. The number of subproblems or subsets $|S|$ is determined by a random procedure that divides the original set of procedures into 2, 3 or 4 subsets (e.g. in case of 2 subsets and 16 procedures to be allocated over the subsets, a subset will contain on average about 8 procedures). The more procedures are grouped into a subset, the better the interplay between procedures can be taken into account. However, this comes at a higher computational cost as the number of variables of the MILP problem to be solved will be higher. However, when too few procedures are grouped, computational effort will be low but the interplay might be lost as one procedure might be very dominant given its volume and item usage compared to the others. The random repartition hence creates all kinds of subsets and provides the opportunity to find a diverse set of interesting custom pack configurations. Note that each subproblem is only solved once, keeping the number of custom packs to be configured low (random, but maximum 4 custom packs) to increase computational tractability. The first phase of the heuristic is limited to 60 s, and the time for solving the MILP of a subproblem is limited to 5 s. The parameters of the algorithm were set by insights resulting from a factorial design (see Section 5). When the set of predefined packs is generated, the second-phase optimisation loop is initiated for the original full problem. Initially, the problem is solved using the set of predefined packs plus one additional pack that can be configured from scratch. Upon completion of the MILP search, and if time is allowing, we add another additional pack to be configured from scratch (so two packs from scratch by now, next to the fixed set of predefined packs of phase 1) and repeat the MILP search, and so forth.

Pseudocode 1: Predefining potentially interesting custom pack configurations (phase 1) and the global MILP problem loop (phase 2).

```
First phase:
procedure ← 1;
while (procedure ≤ |J|) do
   register the custom pack p that exactly matches the item requirements for the procedure (easy to retrieve from data file);
end while
elapsed_time ← 0;
start_time ← time;
while (elapsed_time − start_time < 60 seconds) do
   split set of procedures J randomly into |S| smaller subsets;
   subset ← 1;
while (subset ≤ |S| and elapsed_time − start_time < 60 seconds) do
      calculate total annual cost of required items of procedures in the subset;
      waste ← 0% total annual cost;
      while (waste ≤ 2% of total annual cost) do
         nr_packs_to_be_configured ← random integer in [1, max(4, # procedures in subset)]
         solve exact MILP model with time limit of 5 seconds;
         register any new custom pack p retrieved from solution;
         waste ← waste + 1% of total annual cost;
      end while
      elapsed time ← time;
      subset ← subset + 1;
   end while
end while

Second phase:
additional_packs ← 1;
while (elapsed_time − start_time < 300 seconds) do
   solve adjusted MILP model with predefined and additional packs for global problem;
   register best solution found;
   additional_packs ← additional_packs + 1;
end while
```

We will compare the quality of the two-phase heuristic with solutions generated by a SA approach. As mentioned in Section 2, SA seems to be a promising methodology for problems showing some similarity with our setting (see, e.g. Reymondon and Marcon 2005). We restrict the technical discussion of the SA algorithm and only aspire here to convey its main decision steps. A matrix describing the pack–procedure combinations is at the base of the algorithm. In order to determine the content of a given pack for a given (set of) procedure(s), the algorithm checks the maximum number of each item to be included in the pack so to avoid any waste and ensuring feasibility. If possible and allowed, the number of a particular item in the pack can be further increased (this means introducing waste for at least one of the procedures using the pack). This is done in an iterative way as long as budget allows, starting with adding items that exhibit the highest marginal contribution, determined by the gain in points of touch relative to the cost of adding waste. The SA updates its best solution found whenever it finds a solution that exhibits a lower number of points of touch. Depending on the cooling scheme (linear), it could be that the configuration of a pack is accepted although it is leading to a worse solution in terms of points of touch. The SA applied to our problem is a multistart algorithm, meaning that it restarts with an empty pack-procedure matrix when there is no further improvement after a predetermined number of iterations.

Next to the SA, we will also test two priority rules which are easy-to-implement in practice. The first rule (Rule 1) sequences the procedures in decreasing order based on their annual volume of surgeries. If a custom pack is introduced, it will be assigned to the first procedure of the list and the content of the pack will exactly match the material requirements of the particular procedure. If a second pack is introduced, it will be assigned to the second procedure on the list, and so forth. The second rule (Rule 2) is similar to Rule 1, except for the sequencing system. In Rule 2, procedures are sequenced in decreasing order of the multiplication of the annual volume with the total number of required items for that procedure. In other words, the order is determined by the annual volume of items needed for a particular procedure.

5. Computational results

In Section 5.1, we discuss the data gathering phase and the construction of the test set, whereas we report on the performance (in terms of gap analysis) of the solution approaches in Section 5.2. All experiments were performed on an 2.67 GHz PC with 4 GB RAM and the windows 7 Operating system. The algorithm is coded using MS Visual C++ 2012 Express and is linked with the IBM ILOG CPLEX 12.6 library to execute the optimisation.

5.1 *Data gathering*

We obtained data from a medium-sized Belgian hospital that provides a wide range of medical disciplines. The hospital currently consists of two campuses that will be merged into a single facility on the short term. Both campuses have an operating theatre, but only one currently uses custom packs. With the upcoming restructuring, the hospital is particularly interested whether they can also use the current set of custom packs in the new setting or if they have to rethink their configurations. We limit the scope to orthopaedic surgery types for which a pack is currently used or viable (standardised procedures for which many medical items are required). The main problem consists of 16 different procedures, which we define as different ‘surgeon-surgery type’ combinations. In total, four surgery types (Hip replacement, Knee replacement, Arthroscopy knee and Arthroscopy shoulder), 7 surgeons and 137 different medical items with associated unit prices are incorporated. The annual total number of cases incorporated by the test problem equals 2715, which represents about 18% of the total surgical volume of the hospital, leading to a maximum of 83,100 individual items needed to be picked for surgery.

The data gathering phase was executed in multiple phases so to overcome various problems. One major issue was that the required sets of medical items were not electronically available but had to be identified using paper picking lists. This process, however, resulted in listing items that were already discarded from the hospital’s SKU-portfolio, or having duplicate names for the same item. Corrections, in cooperation with the operating theatre head nurses, reduced the number of different medical items significantly. In addition to that, some unit prices were not always readily available. For those items without unit price, we set the price to zero and ensured that no waste is allowed in solutions (so no excess items). It is currently unclear whether custom packs, as they provide additional service to the organisation, are always more expensive than buying single items. Based on our data, we noticed that the custom pack of the knee arthroscopy was about 8% more expensive than buying its constituting items on an individual basis, whereas the custom pack of the shoulder arthroscopy was about 20% cheaper than the reference.

In testing the case, we differentiate between three different waste levels which can respectively amount to 0%, 1% (€2854) or 2% (€5708) of the total budget required for medical material. Solving the case with 1 up to 16 custom packs, for three different waste settings, eventually results in solving $16*3 = 48$ instances. Since the case exhibits a low commonality of medical item usage between procedures, a large discrepancy between the annual number of medical items to be picked for procedures, and a set of medical items for which the cost per item can significantly differ, we developed next to the real setting some alternative scenarios to assess the sensitivity of results following these three characteristics. This implies that the test set consists of 2 (high or low commonality) * 2 (unequal or approximately equal annual volume of medical items per procedure) * 2 (unequal or equal cost of medical items) = 8 scenarios, for which each time 48 instances are solved. In generating the scenarios, we did control for the total number of surgeries to be performed (2715) and the total number of medical items needed to do so (83,100). Balancing the annual volume of items per procedure was achieved by changing the number of surgeries per procedure, not by changing the number of medical items needed for the procedure. Equal cost of medical items implies assigning the average cost (€3.5). In the remainder of the paper, we refer to a scenario as e.g. LEU – standing for low commonality (L), equal annual volume of items among procedures (E) and unequal item cost (U), or e.g. HUE – standing for high commonality (H), unequal annual volume of items among procedures (U) and equal item cost (E). A scenario name therefore always consists of three letters describing the characteristics in a fixed order.

5.2 *Comparison of the solution approaches*

We will express performance in terms of a relative solution gap, defined as the ratio of (i) the difference between a solution's number of points of touch and the optimal number of points of touch, and (ii) the difference between the single-pull solution's number of points of touch (i.e. 83,100) and the optimal number of points of touch. If we were unable to identify the optimal points of touch for an instance within 24 h of runtime (less than 20% of the instances, all of them with allowance of waste and listed under the high commonality scenarios), we did opt to use the current best solution found as an approximation of the optimal one since the lower bound that was returned by the MILP approach appears to be weak and unrealistically low. In Section 6, we graphically show that this seems to be an acceptable policy. Note that a gap of 100% would imply that the method was unable to identify a solution that performs better than the one in a single-pull system, hence leading to 83,100 points of touch. In this computational testing of the solution methods, the parameters of the two-phase heuristic were validated using a factorial design. We examined the impact of changes in time split between the two phases (60/240; 150/150; 240/60) and the maximum runtime in seconds of a MILP problem in phase 1 (5;30;60). This exercise included 216 instances in which 8 custom packs had to be configured. Instances stem from all different scenarios and waste policies. From Table 2, we conclude that the smallest average solution gap occurs when the time in phase 1 is limited to 60 s, and the maximum time for solving a MILP problem in phase 1 does

Table 2. Average solution gap (%) of factorial design for two-phase heuristic

		Time limit (s) for solving a MILP model in phase 1		
		5 s	30 s	60 s
Time phase 1/time phase 2 (s)	60 s/240 s	**0.10%**	0.15%	0.16%
	150 s/150 s	0.14%	0.14%	0.14%
	240 s/60 s	0.12%	0.11%	0.12%

Note: The parameter settings of the bold value equal those of the computational experiment.

Table 3. Performance of solution methods over all scenarios (runtime limited to 300 s).

		Zero solution gap	Positive solution gap		
Method	Waste (%)	% instances	% instances	Average gap (%)	Standard deviation gap (%)
Exact	0	59	41	21.8	28.5
	1	15	85	39.1	42.4
	2	17	83	35.9	41.7
Two-phase	0	97	3	1.5	0.9
	1	70	30	0.7	0.7
	2	60	40	0.7	1.0
SA	0	16	84	7.1	7.6
	1	10	90	5.8	5.7
	2	9	91	6.2	6.7

Table 4. Performance of solution methods according to the different scenarios (runtime limited to 300 s).

				Zero solution gap	Positive solution gap		
		Methods	Waste (%)	% instances	% instances	Average gap (%)	Standard deviation gap (%)
LUU	Instances (%) with known optimal solution: 100%	Exact	0	81	19	42.8	44.2
			1	25	75	23.1	36.5
			2	19	81	12.4	26.7
		Two-phase	0	100	0	NA	NA
			1	88	13	0.4	0.3
			2	81	19	0.9	0.5
		SA	0	19	81	6.9	5.9
			1	13	88	5.4	4.4
			2	13	88	5.2	4.1
LUE	Instances (%) with known optimal solution: 100%	Exact	0	81	19	42.8	44.2
			1	13	88	58.1	48.3
			2	13	88	46.5	43.8
		Two-phase	0	100	0	NA	NA
			1	88	13	0.0	0.0
			2	81	19	0.1	0.0
		SA	0	13	88	7.2	7.3
			1	13	88	6.7	4.7
			2	13	88	5.6	4.9
LEU	Instances (%) with known optimal solution: 100%	Exact	0	75	25	60.0	19.9
			1	6	94	31.8	35.7
			2	13	88	26.1	35.6
		Two-phase	0	100	0	NA	NA
			1	63	38	0.3	0.3
			2	50	50	0.4	0.3
		SA	0	19	81	14.3	10.0
			1	13	88	10.4	6.7
			2	6	94	12.5	7.9
LEE	Instances (%) with known optimal solution: 100%	Exact	0	75	25	60.0	19.9
			1	19	81	59.2	41.0
			2	13	88	48.0	46.2
		Two-phase	0	100	0	NA	NA
			1	100	0	NA	NA
			2	75	25	0.2	0.2
		SA	0	13	88	15.2	9.4
			1	13	88	12.8	7.5
			2	13	88	15.1	8.7
HUU	Instances (%) with known optimal solution: 71%	Exact	0	50	50	4.4	8.8
			1	6	94	5.4	5.4
			2	25	75	5.9	4.6
		Two-phase	0	100	0	NA	NA
			1	25	75	0.7	0.6
			2	25	75	0.6	0.5
		SA	0	13	88	2.3	1.9
			1	6	94	1.6	1.4
			2	13	88	1.5	1.3
HUE	Instances (%) with known optimal solution: 56%	Exact	0	50	50	4.4	8.8
			1	13	88	45.0	43.9
			2	19	81	52.1	46.6
		Two-phase	0	100	0	NA	NA
			1	100	0	NA	NA
			2	63	38	0.1	0.1
		SA	0	19	81	3.1	2.1
			1	6	94	2.0	1.3
			2	6	94	2.3	2.0
HEU	Instances (%) with known optimal solution: 63%	Exact	0	31	69	14.8	22.2
			1	25	75	23.5	36.1

(*Continued*)

Table 4. (*Continued*).

		Methods	Waste (%)	Zero solution gap % instances	Positive solution gap % instances	Average gap (%)	Standard deviation gap (%)
			2	25	75	32.9	40.8
		Two-phase	0	100	0	NA	NA
			1	38	63	1.2	0.8
			2	31	69	1.6	1.6
		SA	0	13	88	3.2	2.3
			1	6	94	2.8	1.6
			2	6	94	1.9	1.2
HEE	Instances (%) with known optimal solution: 60%	Exact	0	31	69	14.8	22.2
			1	13	88	66.2	45.1
			2	13	88	58.5	45.3
		Two-phase	0	75	25	1.5	0.9
			1	56	44	0.4	0.5
			2	75	25	0.9	1.0
		SA	0	19	81	4.3	2.3
			1	13	88	5.7	3.1
			2	6	94	5.8	3.1

not exceed 5 s. Note, however, that the impact of changing parameters on the solution performance overall remains limited (and not statistically different at 95% confidence).

Table 3 lists the performance of both the exact and the heuristic methods (two-phase heuristic and SA) over all scenarios after 300 s of runtime, whereas Table 4 splits the results over the different scenarios. Both tables differentiate between instances for which the methods found a solution value that equals the optimal value (zero solution gap) and those instances exhibiting a gap between the returned solution and the optimum (positive solution gap). The average gap and the standard deviation of the gap are calculated only including the instances that do not return the optimal solution value. The results in Table 3 clearly indicate that the two-phase heuristic outperforms both the exact method and the SA in finding ‘zero gap solutions’, regardless of the waste policy. Even if the two-phase heuristic did not retrieve a ‘zero gap solution’, it outperforms the other methods in two ways. First, it returns a gap that is stable and characterised by very small standard deviations (hence providing a reliable method). Second, the average gap that is obtained is very small and on average only marginally above zero (hence providing a high-quality method). Although the exact approach succeeds in optimally solving small-scale instances (see Appendix 1 which lists results for individual instances), its performance clearly shows large average gaps and corresponding standard deviations when more custom packs are allowed to be introduced and waste is allowed to be included into the packs. The main reason why the exact method often fails to deliver good solutions stems from the curse of dimensionality: MILP approaches often deteriorate in solution quality when the number of variables and constraints gets large. The heuristic does not suffer from this drawback thanks to its decomposition approach in which only small MILP problems need to be solved. Table 3 also shows that the SA encounters difficulties in finding ‘zero gap solutions’. Although SA cannot match the performance of the two-phase heuristic, it clearly outperforms the exact method in terms of average solution gap and standard deviation for those instances for which no ‘zero gap solution’ was returned. Please note that regardless the method, it seems easier to obtain ‘zero gap solutions’ under the 0% waste policy, which limits the flexibility of the viable configurations and restricts the search space.

From Table 4, we see that the above (aggregated) conclusions on computational efficiency and solution quality in general hold when we differentiate between the eight scenarios. Based on the amount of ‘zero gap solutions’ and the solution quality of the remaining instances, it seems that the two-phase heuristic is not sensitive to the scenario that is underlying the optimisation process. Again, this increases its reliability and improves its applicability in practice: the difference over all scenarios between the lowest and highest average solution gap is only 1.6 percentage points. A similar reasoning holds when verifying the difference for the standard deviation of the solution gap, which is also at most 1.6 percentage points. This insight does not seem to hold for the SA, where both the average solution gap and the standard deviation are smaller for the scenarios with high commonality compared to those with low commonality. Table 4 also shows that the exact method seems to perform better than average under the HUU scenario, partly triggered by the absence of inferior solutions for the instances in which a high number of custom packs is allowed to be developed (see

Appendix 1 for further details on results for individual instances). In this scenario, the procedures have a lot of similarity regarding content but differences in terms of volume, which also increases the potential gain of adding waste (see Section 6). When we limit the focus in Table 4 to the two-phase heuristic, we see that it is less powerful in solving instances leading to a zero solution gap under the HUU scenario, at least for those instances in which redundancy of items is allowed (1% waste or 2% waste). This observation also holds for the HEU scenario. When the items have an unequal cost (U), in combination with high commonality (H), more configurations will seem viable and need exploration, therefore impacting the efficacy of the search. Remark that the HEE scenario is the only one in which the two-phase heuristic could not solve all instances (but still 75%) to a zero solution gap.

6. Managerial insights

The structure of the test set allows for some interpretation on how the number of custom packs actually impact the number of points of touch for the different scenarios. Figures 1 and 2 visualise this impact for the LUU (base case) and the HEU scenario, respectively. In the interest of the paper, we do not visualise all scenarios as many insights can already be deduced from the two selected scenarios. Both figures show the optimal points of touch curves, ranging from the single-pull solution to the case in which all procedures have their own dedicated pack. Recall from Section 5.2 that for some of our instances under the high commonality scenarios we did use the current best solution known as we were unable to verify optimality within 24 h of runtime. This can be seen in Figure 2: the markers of some solutions (between 4 and 11 custom packs under waste policies 1 and 2%) are not filled, in contrast to optimal solutions which have a solid fill marker. From Figure 2, one can see that the markers without fill do follow the expected pattern that is stipulated by the markers with solid fill, for which optimality has been proven. Hence, we feel this approximation is valid for discussing our findings.

If we focus in Figures 1 and 2 on the curves that are depicting the points of touch under the no waste policy (Opt 0%), we clearly see that the introduction of custom packs decreases the points of touch, though at a decreasing or non-linear pace. In other words, with only few packs the points of touch can already be significantly lowered. Considering all scenarios, the gain of the first few custom packs is even more explicit and steeper under the high commonality scenarios and the scenarios with an unequal annual volume of medical items. If there are already many packs in place, the marginal gain of adding one more seems almost negligible. Comparing these results to the priority rules, which also do not allow for the use of waste, we clearly see that Rule 2 outperforms Rule 1, though they are far off the optimal points of touch that could be reached. Since scenario LUU, depicted in Figure 1, is reflecting the real data settings, we are able to compare the outcome of the 0% Opt curve with the current hospital result. The room for improvement is apparent: either the hospital should keep its 9 packs but reconfigure its content, which would lead to a decrease of more than 40,000 points of touch, or it should reduce the number of custom packs from 9 to 2, resulting in the same points of touch though with far less packs.

A comparison of the policy without waste with the 1% waste (Opt 1%) or 2% waste (Opt 2%) policy learns that allowing for redundant items in packs can significantly lower the points of touch for the same number of custom packs, especially if this number of custom packs is limited. The impact of waste seems to be bigger for those scenarios where the cost of medical items is unequal and procedures share many items (high commonality). Regardless the waste policy (0, 1 or 2%), having high commonality between the procedures increases the impact of custom pack usage, it is key for hospitals to standardise the usage of their medical items over different procedures and surgeons. This is in line with the findings of, e.g. Robertson and Ulrich (1998), which indicate the importance of commonality to achieve economies of scale. Although standardisation seems viable and advantageous to all stakeholders, this is often a cumbersome exercise in practice and difficult to achieve when dealing with medical responsibilities. Is a compress of size 10 × 20 really different from a compress of size 10 × 25? Is a compress of size 10 × 20 of brand A really different from one of brand B? Although this standardisation effort should be executed prior to the actual custom pack configuration, it should be clear that our algorithm can help in assessing the potential gain of such actions. This, however, is out of the research scope. From Figures 1 and 2, it furthermore appears that the return of adding more waste, i.e. shifting from the 1% waste to the 2% waste policy, is positive but decreasing. This implies that only a limited budget for waste could already bring substantial gains in terms of points of touch. Note that the inclusion of redundancy implies that we can already achieve minimal picking (i.e. 2715) with less than 16 custom packs. In Figure 2, for instance, one can see that the markers of the 1 and 2% waste policy when 14 or 15 custom packs need to be configured are on a perfect horizontal line with the 16 pack solution under the 0% waste policy, resulting in 2715 points of touch.

Comparing Figures 1 and 2, we find that a custom pack will sooner being shared and thus impact multiple procedures when the procedures exhibit a high commonality, especially in case no waste is allowed. However, one can foster the sharing of packs significantly by adding redundant items, even for low commonality settings. In line with the

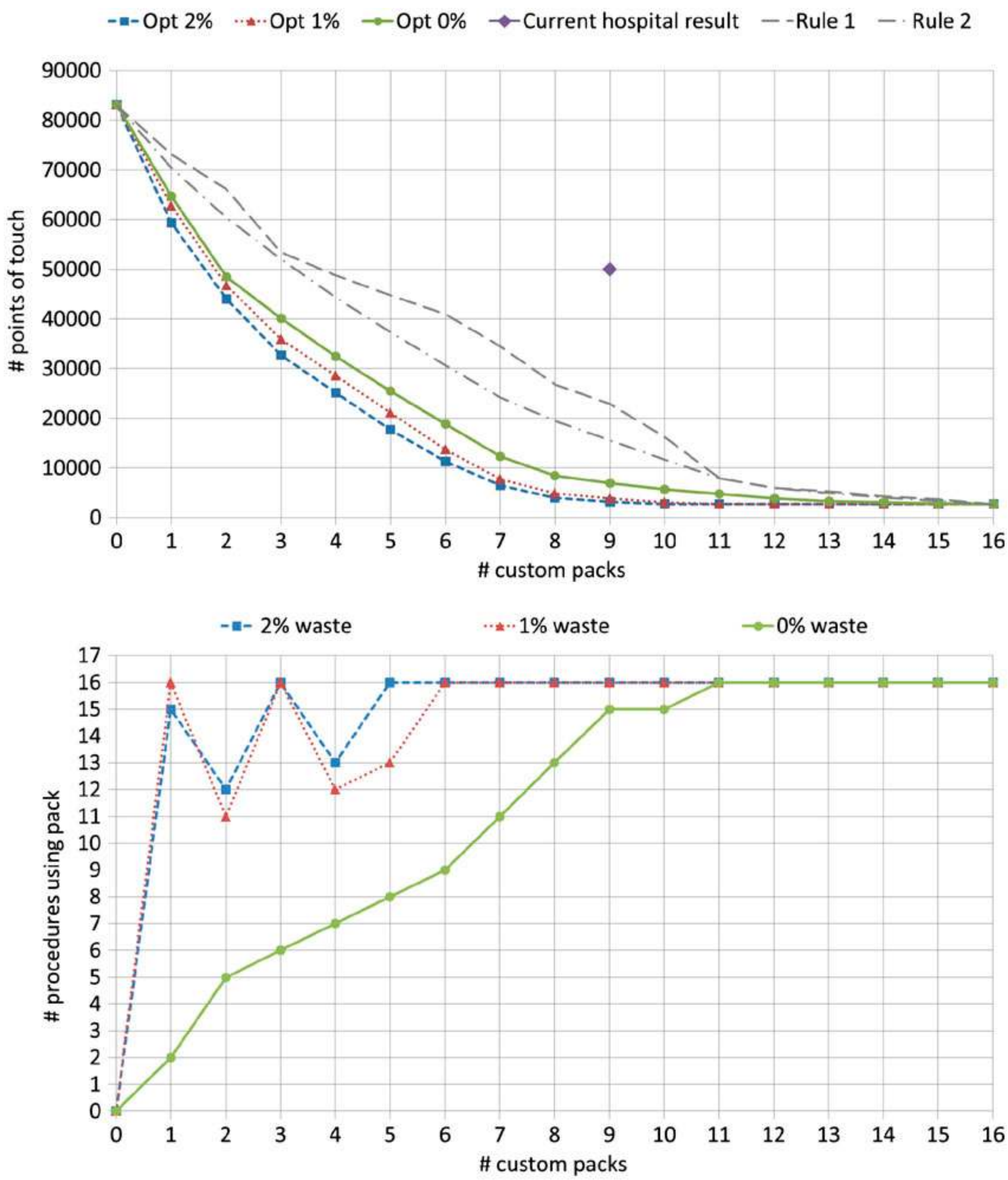

Figure 1. Impact of the number of custom packs on points of touch and the extent to which packs are shared, for scenario LUU (low commonality, unequal volume and unequal cost).

findings on the number of points of touch, waste seems to be especially beneficial for sharing packs when a limited number of custom packs is used and the waste cost is unequal (many cheap items can cover the item gaps within the limited budget). Although not visible in the figures, we should add that our instances hardly return solutions in which more than one custom pack is simultaneously used for one particular procedure. As such, the concept of modularity is far from dominant based on our data settings. Note that the pattern in number of procedures using packs is quite different between Figures 1 and 2. The content of a custom pack when only one pack is allowed to be configured can be very different from the content when two packs can be configured and therefore also might result in very different procedures that will use the custom packs. In other words, there is no guarantee that a procedure which is using a custom pack when only one pack can be configured, will also use a custom pack when two packs are allowed to be configured. It could be, but it also could be very different. This is an outcome of the interplay between procedures and (if allowed)

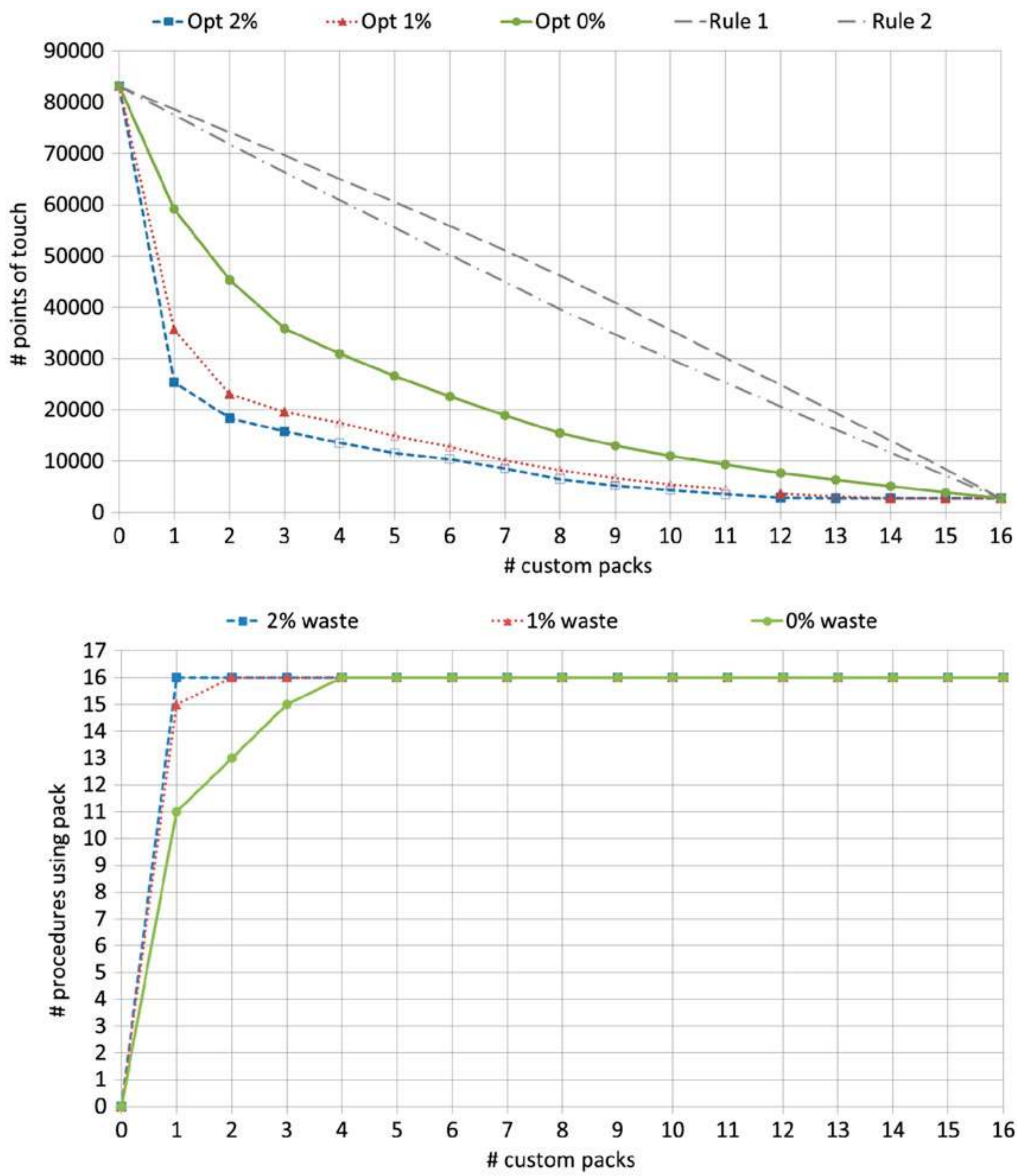

Figure 2. Impact of the number of custom packs on points of touch and the extent to which packs are shared, for scenario HEU (high commonality, equal volume and unequal cost).

the inclusion of waste. This irregular behaviour will always smooth out when more custom packs are allowed to be configured.

7. Conclusion

Based on a real case and a structured test set, this research shows that the introduction of custom packs can significantly decrease the points of touch needed to get medical materials ready for surgery. From the computational testing, the two-phase MILP heuristic was shown to be a powerful solution method to solve the complex combinatorial optimisation problem that describes the configuration of custom packs. In assessing how these custom packs should be configured to achieve the best returns, we found that a substantial improvement could already be realised using only few packs. Also, though perhaps counterintuitive when it comes to efficiency, we did show to what extent the inclusion of redundant medical items can foster the sharing of a pack among procedures, creating economies of scale. Similar to the number of custom packs to be introduced, it appears that already a limited budget dedicated to redundant items can be beneficial for decreasing the points of touch.

Many opportunities for future research can be listed and hence build upon the findings of this project. One major area for further initiatives concerns the objective function to be optimised. Instead of dealing with a single unit (points of touch), one could further explore how to define trade-offs and to evolve towards a meaningful multiobjective approach. Also, instead of fixing the budget of waste to be incorporated, one can think of delineating the budget that ideally should be spent. In order to achieve this, though, again a trade-off mechanism has to be specified. Also, we would like to find out whether the heuristic approach would also support different contexts and settings and still return near-optimal solutions as an alternative to the common set of metaheuristic procedures.

Acknowledgements

We would like to thank all hospital members and partners of the MINOZ Research Centre of the Vlerick Business School for sharing their knowledge about the design and the use of custom packs. Their suggestions and experience undoubtedly contributed to the practical value of this research. We are particularly grateful to the hospital willing to share their data and for the particular efforts to improve the quality. The authors furthermore acknowledge the support of the Vlerick Academic Research Fund, partially subsidised by the Flemish government, for their financial support to execute this research project.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Agard, B., and B. Penz. 2009. "A Simulated Annealing Method Based on a Clustering Approach to Determine Bills of Materials for a Large Product Family." *International Journal of Production Economics* 117 (2): 389–401.

Akridge, J. 2005. "How to Pack for Surgery? Managing Custom Procedure Packs is Key to Optimization." *Healthcare Purchasing News*, May, 20–23.

Baines, R., G. Colquhoun, N. Jones, and R. Bateman. 2001. "The Benefits of Using Customised Procedure Packs." *British Journal of Perioperative Nursing* 11 (1): 34–39.

Bautista, J., and J. Pereira. 2007. "GRASP Algorithm to Solve the Unicost Set Covering Problem." *Computers & Operations Research* 34: 3162–3173.

Beliën, J., L. De Boeck, J. Colpaert, S. Devesse, and F. Van den Bossche. 2013. "Optimizing the Facility Location Design of Organ Transplant Centers." *Decision Support Systems* 54 (4): 1568–1579.

Birk, S. 2009. "Customised Packs Yield Major Savings." *Materials Management in Health Care* 18 (4): 34–36.

Caprara, A., P. Toth, and M. Fischetti. 2000. "Algorithms for the Set Covering Problem." *Annals of Operations Research* 98: 353–371.

Cardoen, B., E. Demeulemeester, and J. Beliën. 2010. "Operating Room Planning and Scheduling: A Literature Review." *European Journal of Operational Research* 201: 921–932.

Chin, C., L. Sowerby, A. John-Baptiste, and W. Rotenberg. 2014. "Reducing Otolaryngology Surgical Inefficiency via Assessment of Tray Redundancy." *Journal of Oltolaryngology-Head & Neck Surgery* 43. Article Number: 46.

Dexter, F., A. Abouleish, R. Epstein, C. Whitten, and D. Lubarsky. 2003. "Use of Operating Room Information System Data to Predict the Impact of Reducing Turnover times on Staffing Costs." *Anesthesia and Analgesia* 97: 1119–1126.

Dobson, G., A. Froix, A. Seidmann, and V. Tilson 2014. "Configuring Surgical Instrument Trays to Reduce Costs." http://ssrn.com/abstract=2524426 orhttp://dx.doi.org/10.2139/ssrn.2524426.

El Hadj Khalaf, R., B. Agard, and B. Penz. 2011. "Simultaneous Design of a Product Family and Its Related Supply Chain Using a Tabu Search Algorithm." *International Journal of Production Research* 49 (19): 5637–5656.

Garey, M., and D. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: W. H. Freeman.

Gellman, C. 1988. "OR Procedure Packs." *AORN Journal* 48 (1): 72–75.

Guerriero, F., and R. Guido. 2011. "Operational Research in the Management of the Operating Theatre: A Survey." *Health Care Management Science* 14 (1): 89–114.

Harders, M., M. Malangoni, S. Weight, and T. Sidhu. 2006. "Improving Operating Room Efficiency through Process Redesign." *Surgery* 140 (4): 509–516.

HFMA. 2005. *Achieving Operating Room Efficiency through Process Integration*. Technical Report. Health Care Financial Management Association.

Jiao, J., T. Simpson, and Z. Siddique. 2007. “Product Family Design and Platform-based Product Development: A State-of-the-art Review.” *Journal of Intelligent Manufacturing* 18: 5–29.

Kharraja, S., P. Albert, and S. Chaabane. 2009. “The Impact of Induction Room on Operating Theatre Performances.” *Studies in Computational Intelligence* 189: 207–220.

Krupka, D., S. Sathaye, and W. Sandberg. 2008. “Reducing Non-operative Time: Methods and Impact on Operating Room Economics.” *International Journal of Healthcare Technology and Management* 9 (4): 325–352.

May, J., W. Spangler, D. Strum, and L. Vargas. 2011. “The Surgical Scheduling Problem: Current Research and Future Opportunities.” *Production and Operations Management* 20 (3): 392–405.

Normén, A., and A. Evans. 2013. A Practical Approach to OR Efficiency – Switching to Customised Procedure Trays in Ramsay Health Care Hospitals. *Journal of Perioperative Practice – Procurement Guide* 2 (1): 9–10.

OECD. 2012. *Health Data*. Organisation for Economic Co-operation and Development.

Poisal, J., C. Truffer, S. Smith, A. Sisko, C. Cowan, S. Keehan, and B. Dickensheets. 2007. “Health Spending Projections through 2016: Modest Changes Obscure Part D’s Impact.” *Health Affairs* 26 (2): w242–w253.

Qu, T., S. Bin, G. Q. Huang, and H. D. Yang. 2011. “Two-stage Product Platform Development for Mass Customisation.” *International Journal of Production Research* 49 (8): 2197–2219.

Reymondon, F., and E. Marcon. 2005. “Optimization of Hospital Sterilization Costs Acting on Grouping Choices of Medical Devices into Packages, Using the Simulated Annealing Method.” International Conference on Industrial Engineering and System Management, Marrakech, Maroc.

Reymondon, F., B. Pellet, and E. Marcon. 2008. “Optimization of Hospital Sterilization Costs Proposing New Grouping Choices of Medical Devices into Packages.” *International Journal of Production Economics* 112: 326–335.

Robertson, D., and K. Ulrich. 1998. “Planning for Product Platforms.” *Sloan Management Review* 39 (4): 19–31.

Stockert, E. W., and A. Langerman. 2014. “Assessing the Magnitude and Costs of Intraoperative Inefficiencies Attributable to Surgical Instrument Trays.” *Journal of the American College of Surgeons* 219 (4): 646–655.

Stolletz, R., and J. Brunner. 2011. “Fair Optimization of Fortnightly Physician Schedules with Flexible Shifts.” *European Journal of Operational Research* 219 (3): 622–629.

Toftgaard, C., and G. Parmentier. 2006. “International Terminology in Ambulatory Surgery and Its Worldwide Practice.” In *Day Surgery – Development and Practice*, 35–59. International Association for Ambulatory Surgery.

Zhang, B., P. Murali, M. Dessouky, and D. Belson. 2009. “A Mixed Integer Programming Approach for Allocating Operating Room Capacity.” *Journal of the Operational Research Society* 60: 663–673.

Appendix 1

This appendix lists the solution gap (as explained in Section 5.2) for all individual instances of the test set. It is structured according to the solution method, scenario, number of custom packs to be configured and the waste policy. Gaps of instances for which the optimal solution value is unknown are italicised. All gaps are expressed in percentage.

#	%	Exact								Two-phase heuristic								Simulated annealing							
packs	waste	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE
1	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.09	0.04	0.03	0.01	0.00
3	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.11	0.26	0.18	0.02	0.04	0.03	0.04
4	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.07	0.17	0.21	0.22	0.04	0.08	0.05	0.03
5	0	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.06	0.22	0.33	0.07	0.05	0.04	0.06
6	0	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.22	0.29	0.16	0.04	0.05	0.02	0.05
7	0	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.18	0.19	0.21	0.01	0.03	0.08	0.07
8	0	0.00	0.00	0.00	0.00	0.02	0.02	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.11	0.24	0.27	0.02	0.03	0.05	0.06
9	0	0.00	0.00	0.00	0.00	0.01	0.01	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.01	0.15	0.21	0.02	0.03	0.04	0.07
10	0	0.00	0.00	0.00	0.00	0.02	0.02	0.21	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.03	0.13	0.16	0.01	0.03	0.06	0.06
11	0	0.00	0.00	0.00	0.00	0.02	0.02	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.14	0.02	0.01	0.03	0.04
12	0	0.00	0.00	0.00	0.00	0.01	0.01	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.06	0.09	0.01	0.02	0.01	0.04
13	0	0.02	0.02	0.70	0.70	0.26	0.26	0.74	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.08	0.02	0.01	0.00	0.01	0.02
14	0	0.00	0.00	0.44	0.44	0.00	0.00	0.35	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.00	0.00	0.01	0.01
15	0	0.37	0.37	0.43	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
16	0	0.90	0.90	0.83	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.06	0.00	0.00	0.01	0.01	0.00
2	1	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.05	0.06	0.10	0.02	0.03	0.05	0.10
3	1	0.01	0.03	0.02	0.00	0.02	0.02	0.02	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.03	0.00	0.09	0.03	0.21	0.13	0.04	0.04	0.04	0.06
4	1	0.05	0.02	0.02	0.02	0.07	0.06	*0.08*	0.03	0.00	0.00	0.00	0.00	0.01	0.00	*0.02*	0.00	0.07	0.08	0.22	0.26	0.05	0.04	*0.04*	0.07
5	1	0.04	0.04	0.14	0.25	0.15	0.14	*0.06*	0.12	0.00	0.00	0.01	0.00	0.02	0.00	*0.01*	0.02	0.07	0.11	0.14	0.21	0.03	0.03	*0.05*	0.09
6	1	0.19	0.02	0.31	0.29	0.07	*0.18*	*0.12*	*0.11*	0.01	0.00	0.00	0.00	0.02	*0.00*	*0.01*	*0.00*	0.14	0.15	0.15	0.22	0.03	*0.03*	*0.03*	*0.08*
7	1	0.08	0.09	0.11	0.12	*0.06*	*0.24*	*0.13*	*0.11*	0.00	0.00	0.00	0.00	*0.01*	*0.00*	*0.01*	*0.00*	0.13	0.11	0.14	0.19	*0.03*	*0.02*	*0.04*	*0.09*
8	1	0.05	0.07	0.16	0.23	*0.09*	*0.48*	*0.12*	*0.92*	0.00	0.00	0.00	0.00	*0.01*	*0.00*	*0.01*	*0.00*	0.07	0.14	0.05	0.19	*0.01*	*0.03*	*0.04*	*0.07*
9	1	0.10	0.90	0.07	0.22	*0.09*	*0.08*	*0.20*	*0.95*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.03	0.06	0.14	0.09	*0.02*	*0.03*	*0.04*	*0.06*
10	1	0.05	0.97	0.22	0.57	*0.02*	*1.00*	*1.00*	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.04	0.08	0.15	0.12	*0.01*	*0.02*	*0.03*	*0.06*
11	1	0.00	1.00	0.17	1.00	*0.03*	*0.07*	*1.00*	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.04	0.07	0.07	0.11	*0.01*	*0.02*	*0.01*	*0.05*
12	1	0.21	1.00	0.06	1.00	*0.01*	*1.00*	*0.04*	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.01	0.03	0.02	0.03	*0.01*	*0.01*	*0.01*	*0.04*
13	1	1.00	1.00	0.81	1.00	*0.01*	*0.05*	*0.05*	1.00	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	0.00	0.01	0.03	0.03	0.08	*0.00*	*0.01*	*0.02*	0.01
14	1	0.00	1.00	0.88	1.00	0.02	*1.00*	0.00	*1.00*	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	*0.00*	0.01	0.01	0.02	0.04	0.00	*0.00*	0.01	*0.01*
15	1	1.00	1.00	1.00	1.00	0.00	*1.00*	0.00	*1.00*	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	*0.00*	0.00	0.00	0.00	0.02	0.00	*0.00*	0.01	*0.01*
16	1	0.00	1.00	0.80	1.00	0.17	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.15	0.00	0.01	0.00	0.01	0.00
2	2	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.03	0.01	0.14	0.15	0.03	0.05	0.01	0.11
3	2	0.01	0.01	0.03	0.06	0.04	0.00	0.03	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.03	0.02	0.12	0.02	0.19	0.19	0.04	0.05	0.04	0.07
4	2	0.07	0.02	0.02	0.07	0.03	*0.05*	*0.05*	*0.02*	0.01	0.00	0.00	0.00	0.01	*0.00*	*0.03*	*0.00*	0.09	0.09	0.18	0.27	0.04	*0.06*	*0.03*	*0.07*
5	2	0.08	0.16	0.09	0.14	0.09	*0.00*	*0.12*	*0.08*	0.01	0.00	0.01	0.00	0.02	*0.00*	*0.02*	*0.00*	0.09	0.10	0.27	0.22	0.03	*0.04*	*0.04*	*0.09*
6	2	0.18	0.12	0.01	0.05	*0.05*	*0.08*	*0.16*	*0.07*	0.00	0.00	0.01	0.00	*0.01*	*0.00*	*0.01*	*0.01*	0.12	0.08	0.06	0.30	*0.02*	*0.04*	*0.02*	*0.07*
7	2	0.05	0.00	0.12	0.03	*0.03*	*0.18*	*0.10*	*0.14*	0.00	0.00	0.00	0.00	*0.01*	*0.00*	*0.01*	*0.00*	0.09	0.12	0.22	0.23	*0.02*	*0.02*	*0.02*	*0.07*
8	2	0.01	0.08	0.16	0.20	*0.15*	*0.10*	*0.14*	*0.26*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.05	0.12	0.16	0.19	*0.01*	*0.01*	*0.03*	*0.07*
9	2	0.04	0.28	0.12	0.04	*0.04*	*0.29*	*0.18*	*0.61*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.01*	*0.00*	0.04	0.13	0.15	0.16	*0.01*	*0.03*	*0.02*	*0.09*
10	2	0.01	0.56	0.18	0.17	*0.05*	*1.00*	*0.15*	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.02	0.07	0.14	0.14	*0.01*	*0.02*	*0.02*	*0.07*

(*Continued*)

#	%	Exact								Two-phase heuristic								Simulated annealing							
packs	waste	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE	LUU	LUE	LEU	LEE	HUU	HUE	HEU	HEE
11	2	0.06	0.29	0.17	0.94	*0.03*	*1.00*	*1.00*	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	*0.00*	*0.00*	0.02	0.02	0.08	0.09	*0.01*	*0.01*	*0.01*	*0.06*
12	2	0.00	1.00	0.00	1.00	*0.15*	*1.00*	1.00	*1.00*	0.00	0.00	0.00	0.00	*0.00*	*0.00*	0.01	*0.00*	0.01	0.02	0.05	0.03	*0.00*	*0.00*	0.01	*0.04*
13	2	0.04	1.00	0.90	1.00	0.05	*1.00*	0.03	*1.00*	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	*0.00*	0.01	0.00	0.02	0.05	0.00	*0.00*	0.01	*0.02*
14	2	1.00	1.00	0.82	1.00	0.00	*1.00*	1.00	*1.00*	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	*0.00*	0.00	0.01	0.03	0.05	0.00	*0.00*	0.00	*0.02*
15	2	0.05	1.00	0.02	1.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.00	0.02
16	2	0.01	1.00	1.00	1.00	0.00	0.08	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00