Optimal Allocations of Healthcare Resource

Student ID Number	-	710061748
Candidate Number	-	124611
Word Count	-	2173

Contents

I. Lay Summary

When it comes to resource allocation, demand always exceeds supply, and the healthcare sector is no exception. To effectively deliver healthcare services to patients in need, appropriate strategies and implementation are required. This project aimed to discover such methods. The results indicated that the method produced by this project was better than a random allocation. More work is needed to reinforce this finding.

II. Abstract

Allocations of healthcare resource have always been a tough issue for the administration. A number of resource allocation strategies are used however the majority of them are neither value-based nor need-driven. The goal of the project is to ascertain the optimal allocation that maximises value and minimises inequality among individuals without exceeding the budget available. To solve this multi-objective cost-constraint combinatorial optimisation problem, a metaheuristics method, simulated annealing, was used in this project. The results showed a distinct dominance of simulated annealing over the baseline scenario i.e., a random allocation. In comparison to the existing method of the original project, a significant advantage was not observed. Further research in improving simulated annealing algorithm as well as in data collection of input parameters that reflects the actual data is encouraged to prove that such approach is valuable in healthcare resource allocation.

III. Introduction

Background

With demand exceeding supply, healthcare systems face a difficult choice as to how they distribute their scarce resource to ensure maximum value.

The distributions of financial resources to Integrated Care Boards(ICBs) across England are determined by National Health Service in England (NHS England). ICBs are previously known as Clinical Commissioning Groups(CCGs). ICBs are in turn responsible for hiring service providers on behalf of the respective population they served. (1) While calculating annual budgets, the majority of ICBs in the UK simply add a percentage corresponding with the trend or demographic increase to the prior year's activities to derive contracts and operational plans. (2) Since healthcare resource distribution is not value- and needs-driven, inequality issues result.

The delivery of healthcare services to a population can be improved in a variety of ways. These ultimately relate to the productive and allocative factors of economic efficiency since healthcare is a scarce resource. The former can be explained, whereby enhancements in outcome or cost can be acquired through centralising services, improving treatment scheduling, or using better diagnostic tests. Regarding allocative efficiency, as well as ensuring that the health requirement is assigned to the best suitable activity, the recipients of the demanded activity are an important factor, given that not all demands may be satisfied. The scope on which this project has focused is illustrated in Figure 1.

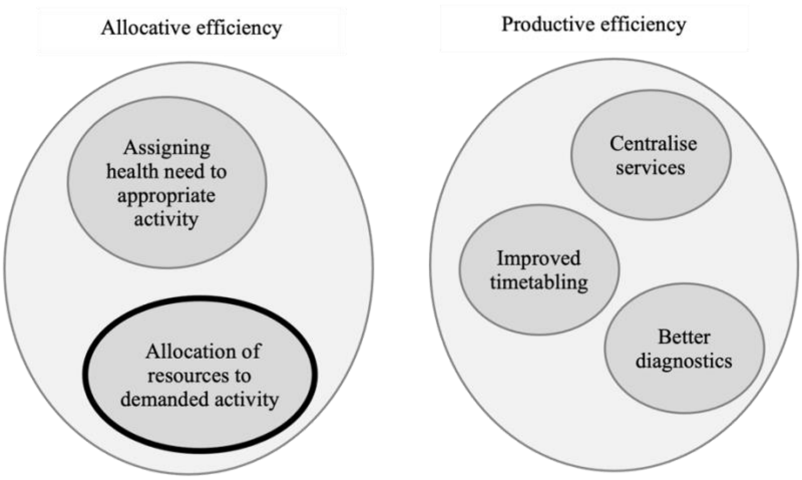

Figure 1 Scope of this paper, in terms of wider allocative and productive economic efficiencies (as represented by solid black outline).

In resource allocation, the typical currency for the resource is ‘activity’; it may be a single outpatient consultation, an inpatient stay, a surgical procedure, or a whole pathway. It is assumed that demand is known, and the task is to allocate it. Since the goal is to optimally allocate the resources to demanded activities within the limited budget while the values and inequality issue are considered, this can be identified as a multi-objective cost-constraint combinatorial optimisation problem. Perhaps, performing the appropriate simulation-based optimisation techniques could be the key to this problem.

The objectives of this project are

- to aid in the transition to a need- and value-driven basis for annual planning
- to develop an 'ideal world' mathematical framework in which activities could be optimally allocated in our system while maximising value and addressing inequalities
- to determine a practically optimal solution through working back from this framework, based on knowledge of what parameters are the most important in order to identify and overcome any particular practical limitations

Approach

Dr Richard Wood et. al. at NHS Bristol, North Somerset and South Gloucestershire ICB have devised five individual-level resource allocation strategies, which are described below. A summary of the key features is provided in Table 1. The details of the strategies are explained in section 1 of the supplementary materials.

Feature	Baseline	Strategy				
		1	2	3	4	5
Maximises total short-term value		✓	✓	✓	✓	✓
Maximises total long-term value			✓	✓	✓	✓
Appreciates unexpressed demand				✓	✓	✓
Minimises unequal value (*) or outcome (**) allocation					✓*	✓**

Table 1 Summary of strategies considered in this study

Baseline Strategy

This strategy naively results in a random allocation of the activity for which there is expressed demand, supposing that overall short-term costs are no larger than the short-term budget. This is done without consideration for outcomes, value, or inequality.

Strategy One

Activity is allocated to maximise total short-term value with no attempt made to account for long-term value or to coerce unexpressed demand. Allocating under this strategy will not take account of outcome deterioration in the future through not allocating activity in the short term nor the additional remedial costs to address such issues. This applies both to activity relating to expressed demand as well as unexpressed demand, with the latter not receiving any proactive targeting under this strategy.

Strategy Two

Activity is allocated to maximise total short and long-term value but with no attempt made to coerce unexpressed demand. This means that the allocation is potentially sub-optimal in terms of value since, even accounting for coercion cost, potentially high outcome or high additional remedial cost activity is being missed.

Strategy Three

Activity is allocated to maximise total short and long-term value while appreciating unexpressed demand and the value of addressing these, even while factoring in the additional costs of coercion. What this strategy fails to do, however, is ensure any level of equality in aggregate value allocated to individuals within the population. Therefore, some may be left with no activity.

Strategy Four

Activity is allocated to maximise total short and long-term value while appreciating unexpressed demand and with some consideration to a more equal distribution of aggregate short-term value to individuals within the population.

Strategy Five

Activity is allocated to maximise total short and long-term value while appreciating unexpressed demand and with some consideration to a more equal distribution of aggregate short-term outcome improvements to individuals within the population, considering the potential importance of the outcome to value from an individual's point of view.

Progress to Date

Based on the above strategies, the team has produced algorithms to find the optimal allocation. For Strategies 1-3, a greedy algorithm was done. From the synthetic data, it selects the activities with the highest value while maintaining the cost under the constraint. For Strategies 4 and 5, an algorithm was written to generate 1,000 random allocations and then select one with the highest value and the lowest variance between individuals. Both performed well and showed significant advantage over the baseline scenario, which is the random allocation.

IV. Methods

Input Parameters

Synthetic data includes three major input components: 1) cohorts 2) short- and long-term outcomes and cost of performing an activity, and 3) points of delivery.

According to BNSSG core segmentation, the population is segmented into five cohort groups, ‘pristine health’, ‘healthy’, ‘unhealthy’, ‘very unhealthy’ and ‘end of life’. The points of delivery include ‘calls to 111’, ‘primary care contact’ and so on. BNSSG core segmentation was done by Richard et. al., which is mentioned in section 3 of the supplementary materials.

In terms of activities, only short-term costs and activity volumes reflect the actual data. The other variables were generated randomly as NHS has no data on them currently.

The tables of the synthetic data are mentioned in section 2 of the supplementary materials.

Selection of an optimisation method

The existing algorithm for Strategies 4 and 5 produced acceptable results but it was not the optimal allocation. To solve this optimisation problem, the exact methods are not possible because the computational power and time to reach the optimal solution are unaffordable.

Metaheuristics, one of the approximate optimisation techniques, has shown remarkable success in tackling combinatorial optimisation problems. They can provide optimal or near-optimal solutions. (3) Among metaheuristics techniques, simulated annealing is chosen as it is convenient to implement and suitable when the search space is discrete. (4) Greedy algorithm and gradient descent are justifiable alternatives however one of the drawbacks is they might be trapped at the local optimum whereas simulated annealing can overcome this barrier and reach the global optimum.

Simulated Annealing

Simulated annealing mimics the metallurgical process of physical annealing. Physical annealing is the procedure of heating a material to an annealing temperature and then gradually cooling it to transform it into the desired structure. The molecular structure of a substance is more brittle and sensitive to change when it is hot. The molecular structure of the material becomes tougher and less changeable as it cools. (5)

In order to thoroughly explore the feasible domain and prevent getting stuck at local optima, simulated annealing features an occasional uphill solution acceptance mechanism while the temperature is high. When the temperature cools down, the search space becomes narrow and concludes at a global optimum. (6)

Resource Allocation

In this project, the output allocation of the existing algorithm was taken as the initial solution for simulated annealing because it was considered to be closer to the optimal solution than a random one. So, it could save computational power and time during the process.

The value gained is calculated respectively from the outcome and cost of performing that activity (value = outcome/cost) to each strategy.

All simulations were done using R programming. (7) The R script is attached in section 4 of the supplementary materials.

V. Results

The computational time spent for 1,000 replications and selecting the allocation with the best value is 2.81 minutes for each strategy whereas that for simulated annealing is 19.14 seconds per strategy.

Comparison between the allocations by 1,000 Replications and Simulated Annealing

As shown in Figures 2 and 3, regarding Strategy 4, the short and long-term value of the allocation produced by the simulated annealing algorithm is better than the other. However, its variance is also larger than the latter. In Strategy 5, the results presented a similar pattern but inversely.

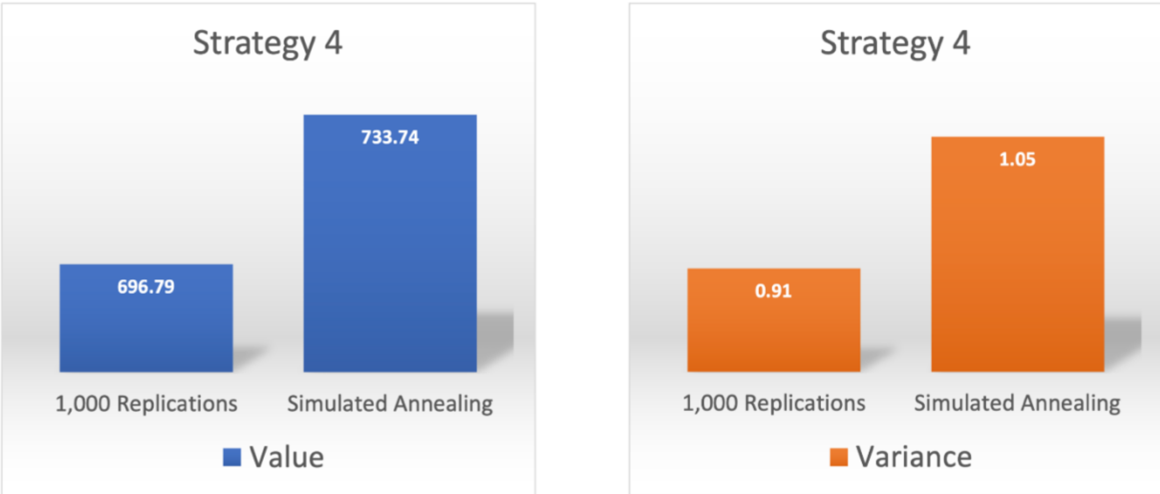

Figure 2 Strategy 4: (left) Comparison of Short- and Long-term Value (right) Comparison of Value Variance between Individuals

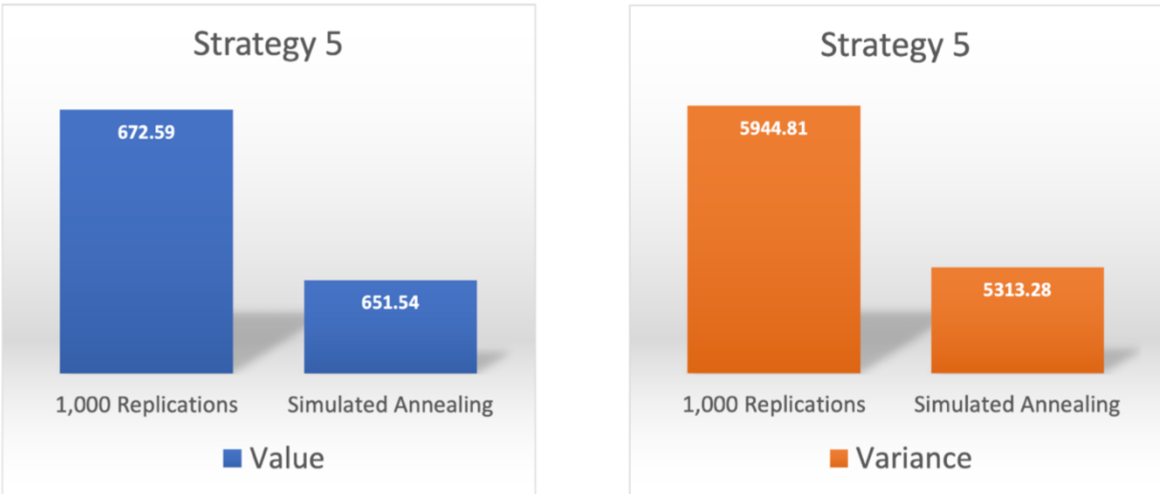

Figure 3 Strategy 5: (left) Comparison of Short- and Long-term Value (right) Comparison of Outcome Variance between Individuals

Comparison between Baseline and Strategies 4 and 5

Figure 4 illustrates that both Strategies 4 and 5 provide better values than the baseline strategy. In Figure 5, the variance of value and outcome in baseline is less than that of the two strategies respectively.

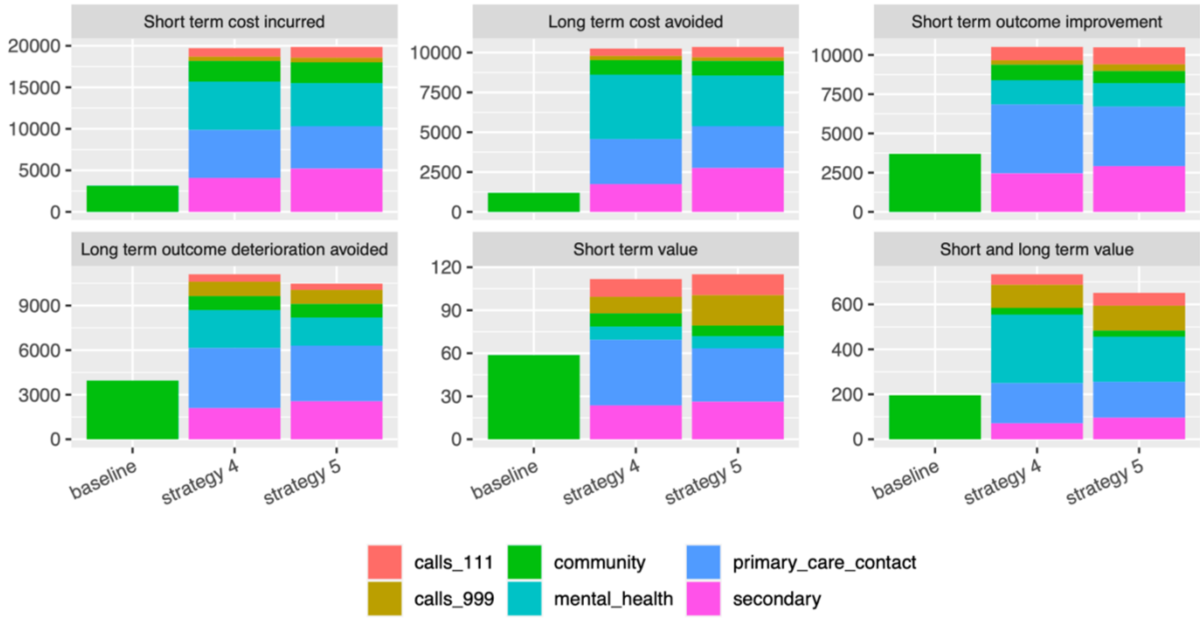

Figure 4 Comparison of Strategies on Cost, Outcome and Value

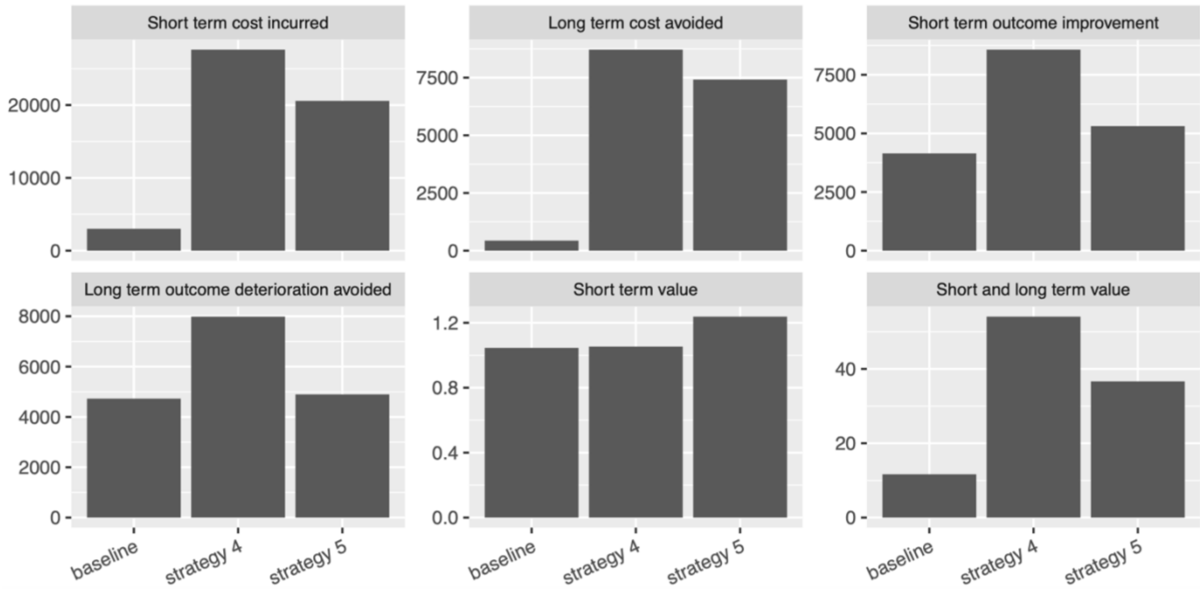

Figure 5 Comparison of Strategies on Variance between Individuals

VI. Discussion

According to the results, comparing the two allocation algorithms, simulated annealing did not show a significant advantage over the other in terms of value and variance. However, the computational time spent for simulated annealing is much lesser. It outperformed while generating similar results. So, this method could potentially replace the original method.

The short- and long-term values and short-term outcome improvements of Strategies 4 and 5 far exceeded those of the baseline scenario. Hence, instead of a random allocation i.e., the baseline scenario, the approaches, that take the patients' demand into consideration, would better fulfil the needs of the patients.

There was a noticeable increase in variance while maximising value. The objective function of the project has two components, to maximise value and to minimise variance. However, the results showed that the algorithm could not manage both components at the same time, so a trade-off happened. It might be due to the weakness in implementing the simulated annealing method or the poor quality of the synthetic data.

Limitations

There are several limitations and room for improvement in this project. First of all, the synthetic data did not reflect the real-world data. Since the aim was to produce an optimisation model, some input parameters were artificially generated to test it. Moreover, in the project, it was assumed that the activities demanded were the same as the activities needed while in reality, the patients' needs might be beyond the demand.

The dataset is also small enough for metaheuristics to show its superiority over other methods. If this approach is proven to be effective by further research and work, it would be beneficial to invest in data collection of required parameters.

The codes to implement simulated annealing could be improved to boost efficiency. In addition, the current mathematical equations do not accurately reflect the objectives. So, it is also needed to investigate the theoretical background.

Relevance and Impact

This project is an ideal world exercise. The aim is to illustrate how this method could be valuable for annual activity commissioning. After successfully developing an optimisation model, we could persuade stakeholders and decision makers to greatly invest in data collection or estimation in order to supply the required parameters to effectively run the algorithm every year.

VII. Conclusion

Referencing the progress so far and the results of the project, the benefits of an optimal resource allocation model are evidently distinct. With a limited healthcare resource in hand, it is essential to utilise its utmost. In this report, different approaches and strategies toward better resource allocation have been mentioned. It is recommended to continue working on this area and ultimately fulfil the health needs of the population.

VIII. Acknowledgements

I would first like to thank my project host supervisor, Dr Richard Wood, and my academic supervisor, Mr John Dennis. The virtual doors to their office were always open whenever I encountered a problem or had a query about my project or report writing. They consistently allowed this report to be my own work but steered me in the right direction whenever they felt I needed it. I also would like to express my gratitude to my supervisors for the helpful comments, remarks, and engagement through the learning process of this master's research project.

Furthermore, I would like to thank my academic tutor, Dr Sean Manzi. Special gratitude goes to Professor Timothy Frayling and Professor Martin Pitt for their support during and even before the course. I am also grateful to all my teachers, Dr Thomas Monks, Dr Andrew Wood, Professor Rob Anderson, Professor William Henley, Dr Eilis Hannon, Dr Julia Frost, Mr Edward Chuah and guest lecturers for their incredible lectures, continuous help, and knowledge sharing. Moreover, my MSc classmates, with whom I experienced a wonderful time as a student, are also to be thanked.

Finally, I must express my profound gratitude to my parents, my elder sister, my beloved, and my friends in Myanmar for providing me with unfailing support and continuous encouragement throughout my study and through the process of researching and writing this report. This accomplishment would not have been possible without them.

I thank you all for your encouragement!

IX. Bibliography

1. Health and Care Act 2022 (Commencement No. 1) Regulations.
2. Sanderson M, Allen P, Osipovic D, Moran V. New models of contracting in the NHS. London: PRUComm; 2019.
3. Laporte G, Osman IH. Metaheuristics in combinatorial optimization. Amsterdam: Baltzer Science; 1996.
4. Delahaye D, Chaimatanan S, Mongeau M. Simulated Annealing: From Basics to Applications. Springer International Publishing; 2019. p. 1-35.
5. Kirkpatrick S, Gelatt CD, Jr., Vecchi MP. Optimization by simulated annealing. Science. 1983;220(4598):671-80.
6. Amine K. Multiobjective Simulated Annealing: Principles and Algorithm Variants. Advances in Operations Research. 2019;2019:1-13.
7. R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.

X. Supplementary Materials

Section (1)

Notation	Description
a	An activity that can be allocated to the short-term planning period
sto_a	The short-term outcome improvement associated with performing activity a
$stcp_a$	The short-term cost associated with performing activity a
$stcc_a$	The short-term cost associated with coercing activity a
lto_a	The long-term outcome deterioration avoided by performing activity a
$ltcp_a$	The long-term additional remedial cost avoided by performing activity a
A	The set of all candidate activities that are possible to be allocated to the short-term planning period (i.e., $a \in A$)
$\dot{A}$	The selected allocation of activities to the short-term planning period (i.e., $\dot{A} \subset A$)
C^*	The total budget available for the short-term planning period

Supplementary Table 1 Description of the notation used to define the resource allocation strategies under consideration within this project

Strategy 1

Strategy 1 allocates activity in order to maximise total short-term value. Under this strategy, no attempt is made to allocate activity to unexpressed demand; that is, the activity for which it is assumed some cost is required to 'coerce' (i.e., $cc_a > 0$). Let the set of all possible candidate activities be represented by $\tilde{A}$ (i.e., $\tilde{A} \subset A: cc_a = 0$). Thus, the (short-term) value for each such activity ($a \in \tilde{A}$) is given by

$$v_a = \frac{sto_a}{stcp_a} \qquad \text{eqn. (1)}$$

Given that $\dot{A}$ represents the activity allocation within the planning period (noting $\dot{A} \subset \tilde{A}$), then the total value of such allocation is given by

$$V(\dot{A}) = \sum_{\forall a \in \dot{A}} v_a = \sum_{\forall a \in \dot{A}} \frac{sto_a}{stcp_a} \qquad \text{eqn. (2)}$$

The total (short-term) costs under such allocation is given by

$$C(\dot{A}) = \sum_{\forall a \in \dot{A}} stcp_a \qquad \text{eqn. (3)}$$

The activity allocation $\dot{A}$ is therefore chosen such that eqn. (2) is maximised subject to ('s.t.') available short-term budget. That is

$$\dot{A} = \mathrm{argmax}_{\dot{A}} V(\dot{A})$$

$$\text{s.t.} \quad C(\dot{A}) \leq C^* \qquad \dot{A} \subset \tilde{A} \qquad \text{eqn. (4)}$$

noting that $\mathrm{argmax}_{\dot{A}} V(\dot{A})$ simply denotes the selection of $\dot{A}$ for which $V(\dot{A})$ is maximised. While this formulation appreciates outcomes and value over the short-term, it fails to capture the long-term implications of allocation decisions.

Strategy 2

Capturing the long-term consequences of *not* allocating a specific activity requires the consideration of the saving, in terms of long-term outcome deterioration and additional remedial cost, of allocating such activity. Thus, augmenting the previous equation for (short-term) value (eqn. (1)),

$$v_a = \frac{sto_a + lto_a}{stcp_a + ltcp_a} \qquad \text{eqn. (5)}$$

The previously considered total value equation (eqn. (2)), is then modified such that

$$V(\dot{A}) = \sum_{\forall a \in \dot{A}} v_a = \sum_{\forall a \in \dot{A}} \frac{sto_a + lto_a}{stcp_a + ltcp_a} \qquad \text{eqn. (6)}$$

Since the long-term cost of *not* performing the activity is borne outside the budget period, no adjustment is required to the previously considered total short-term costs (eqn. (3)). Given that healthcare needs with unexpressed demand remain unaccounted for, the set of possible candidate activities, $\dot{A}$, stays as before, i.e., $\dot{A} \subset \tilde{A}$ where $\tilde{A} \subset A\colon cc_a = 0$.

Strategy 3

Accounting for activity relating to unexpressed demand, requires capturing the additional costs of 'coercing' such demand and derestricting the list of possible candidate activities (i.e., i.e., $\dot{A} \subset A\colon cc_a \geq 0$). Accordingly, the value of activity a is given

$$v_a = \frac{sto_a + lto_a}{stcp_a + stcc_a + ltcp_a}$$

$$v_a = \frac{sto_a}{stcp_a + stcc_a} + \frac{lto_a}{-ltcp_a} \quad \text{eqn. (7)}$$

and thus total value becomes

$$V(\dot{A}) = \sum_{\forall a \in \dot{A}} v_a = \sum_{\forall a \in \dot{A}} \frac{sto_a + lto_a}{stcp_a + stcc_a + ltcp_a} \quad \text{eqn. (8)}$$

The definition of total short-term costs (eqn. (3)) is also modified, such that

$$C(\dot{A}) = \sum_{\forall a \in \dot{A}} stcp_a + stcc_a \quad \text{eqn. (9)}$$

As before, the activity allocation $\dot{A}$ is chosen through solving eqn. (4), albeit with no constraint on $\dot{A}$ to include activity relating only to expressed demand (i.e., removing $\dot{A} \subset \tilde{A}$). While this addresses any inequalities relating to unexpressed demand (levels of which are known to correlate with poorer health outcomes), it does not prevent inequality in terms of the allocation of value afforded to individuals within the population. That is, some individuals may end up with much of their (expressed or unexpressed) demand satisfied, while others may have little-to-none.

Strategy 4

In promoting equality in the amount of value allocated to individuals, the approach taken here is to reduce the variance in aggregate value among the population. First, if activity a relates to person q then $p_a = q$. The aggregate (short-term) value for person $q \in Q$ (where Q is the population) under allocation $\dot{A}$ is given by

$$w_q(\dot{A}) = \sum_{\forall a \in \dot{A}: p_a = q} v_a = \sum_{\forall a \in \dot{A}: p_a = q} \frac{sto_a}{stcp_a + stcc_a} \quad \text{eqn. (10)}$$

Unequal value allocation can be measured by the variance in aggregate person-level value, $\sigma_w^2(\dot{A})$. With mean aggregate person-level value defined as $\bar{w}(\dot{A}) = \frac{\sum_{q \in Q} w_q}{|Q|}$ then $\sigma_w^2(\dot{A}) = \frac{\sum_{q \in Q} (w_q - \bar{w})^2}{|Q|}$. This represents a second objective to be optimised, alongside total value, $V(\dot{A})$. The weight g is used to represent the emphasis placed on optimising each, such that the allocation is chosen through the solution to

$$\dot{A} = \mathrm{argmax}_{\dot{A}} \quad gV(\dot{A}) - (1-g)\sigma_w^2(\dot{A})$$

$$\text{s.t.} \quad C(\dot{A}) \leq C^* \quad \text{eqn. (11)}$$

Strategy 5

It is debatable whether equality should be sought in terms of aggregate person-level value or aggregate person-level improvements in outcome. Arguably, from an individual's perspective, cost (the denominator in the value equation) is unimportant: what matters to he or she is the immediate improvements in outcome, and that the amount of such improvement is distributed *fairly* among all members of the population (notwithstanding that more economically rational or altruistic individuals would favour equality in value over outcome). As such, the above notation for Strategy 4 can be recast with respect to aggregate outcome improvements instead of aggregate value; that is

$$w_q(\dot{A}) = \sum_{\forall a \in \dot{A}: p_a = q} sto_a \qquad \text{eqn. (12)}$$

Section (2)

cohort	number	name	description
1	52	CS1	pristine health
2	25	CS2	healthy
3	13	CS3	unhealthy
4	7	CS4	very unhealthy
5	3	CS5	end of life

Supplementary Table 2 Attributes sheet of Core Segmentation Input file

cohort	pod1	pod2	spec	volume	sto	stcp	stcc	lto	ltcp	detail
1	calls_111	blank	blank	1	79	15	59	17	48	
1	calls_999	blank	blank	1	20	0	26	52	16	
1	community	blank	blank	2	3	63	36	10	45	
1	mental_health	blank	blank	2	50	85	80	84	155	
1	primary_care_contact	blank	blank	3	91	17	95	82	64	
1	secondary	blank	blank	2	96	86	4	90	10	
2	calls_111	blank	blank	1	66	15	66	68	25	
2	calls_999	blank	blank	1	7	0	37	81	18	
2	community	blank	blank	2	74	63	0	79	24	
2	mental_health	blank	blank	2	16	87	71	64	29	
2	primary_care_contact	blank	blank	6	73	17	97	71	53	
2	secondary	blank	blank	3	62	101	50	65	113	
3	calls_111	blank	blank	1	31	15	77	34	13	
3	calls_999	blank	blank	1	7	0	65	2	14	
3	community	blank	blank	4	92	103	91	60	61	
3	mental_health	blank	blank	5	84	114	92	23	86	
3	primary_care_contact	blank	blank	9	40	16	60	81	48	
3	secondary	blank	blank	4	41	98	48	6	60	
4	calls_111	blank	blank	1	55	15	28	67	19	
4	calls_999	blank	blank	1	34	0	12	42	1	
4	community	blank	blank	4	15	106	67	39	37	
4	mental_health	blank	blank	3	34	143	13	56	71	
4	primary_care_contact	blank	blank	12	91	16	58	80	14	

4	secondary	blank	blank	6	68	101	91	44	149	
5	calls_111	blank	blank	1	51	15	44	31	48	
5	calls_999	blank	blank	1	80	0	19	57	6	
5	community	blank	blank	5	25	106	31	82	73	
5	mental_health	blank	blank	4	83	229	12	21	138	
5	primary_care_contact	blank	blank	16	85	17	92	2	63	
5	secondary	blank	blank	7	18	139	25	3	123	

Supplementary Table 3 Activity sheet of Core Segmentation Input file

total.cost
20000

Supplementary Table 4 Cost sheet of Core Segmentation Input file

Section (3)

Title: Introducing the BNSSG Core Segmentation

Date: 9 May 2022

Author: Rich Wood, Fiona Budd, Charlie Kenward, Pete Thomson

---

1. **Background**

The need for a core BNSSG population segmentation has long been recognised. Acting as the standard high-level categorisation of population health needs, the core segmentation would be useful for examining key health and healthcare performance metrics, monitoring population outcomes over time, identifying factors affecting population drift, and serving as the foundation for long-term economic forecasting models on both a do-nothing and do-something basis. While the segmentation would serve as the 'go to' approach, its use would not be expected to remove the need for all bespoke segmentations tailored for specific asks.

BNSSG was one of the first systems to explore segmentation, using a pilot project linking Primary and Secondary Care data for 5 GP practices to comprehensively review 16 different segmentation methods side-by-side (published in a peer-review journal[1]). Work towards a core segmentation began in 2019, with a multidisciplinary cross-system working group agreeing that definition should be driven by both clinical reasoning and a data-based method[2]. The group coalesced on six segments, defined as *End of life*, *Major complexity*, *Frailty*, *Moderate complexity*, *Minor complexity*, and *Relatively healthy*. Any further work on specific definitions were stalled by COVID-19.

In early 2022, the Cambridge Multimorbidity Score[3] was calculated for each individual within the BNSSG population. This score can be used to represent the mortality risk of an individual as well as their future utilisation of healthcare resources. Given these properties, it was considered as a useful basis for the core segmentation approach.

2. Appraisal and selection of the Cambridge Multimorbidity Score

---

[1] https://doi.org/10.1016/j.orhc.2019.100192

[2] The BNSSG core segmentation approach. Briefing Paper for the BNSSG PHM Steering Committee. 2019.

[3] https://doi.org/10.1503/cmaj.190757

People in England can now expect to live for far longer than ever before – but these extra years of life are not always spent in good health, with many people developing conditions that reduce their independence and quality of life[4]. Frailty (rather than age) is an effective way of identifying people who may be at greater risk of future hospitalisation, care home admission or death[5]. As a clinical construct, frailty identifies people whose age-related health status puts them at greater risk than their aging peers[6]. Comorbidity (the concurrent presence of two or more diseases in the same individual; often also referred to as multimorbidity or multiple health conditions) also heightens the risk of disability and mortality[7]. Lifestyle risk factors also contribute significantly to risk of mortality, reduced healthy life expectancy and morbidity[8] [9] [10].

While several scoring systems are used to describe comorbidity, frailty, or risk of mortality, the Cambridge Multimorbidity Score was selected (after clinical review by PT and CK) as the most appropriate to use as a basis for the segmentation for the following reasons:

- *The Cambridge Multimorbidity Score.*
  - Designed for use in system planning, policymaking, allocating resources and research.
  - Excludes age as an independent risk factor, thus avoiding the impact that age stratification can have on more deprived populations, who tend to become more unwell at earlier ages than less deprived populations.
- *The Charlson Comorbidity Index.*
  - Excludes numerous conditions that contribute significantly to healthy life expectancy and healthcare resource utilisation (particularly mental health conditions).
  - Stratifies by age, thus potentially discriminating against more deprived populations.
- *The Electronic Frailty Index.*

4 https://www.longtermplan.nhs.uk/areas-of-work/ageing-well/
5 https://www.england.nhs.uk/ourwork/clinical-policy/older-people/frailty/
6 https://doi.org/10.1038/s43587-021-00099-3
7 https://doi.org/10.1093/gerona/59.3.M255
8 https://doi.org/10.1371/journal.pmed.0050070
9 https://doi.org/10.1371/journal.pone.0111480
10 https://www.gov.uk/government/publications/health-profile-for-england-2018/chapter-3-trends-in-morbidity-and-risk-factors

    - Can only be calculated for patients aged >65 years so unsuitable for a whole population approach.
- *The Clinical Frailty Scale ("Rockwood score").*
    - Not reliably recorded in the System Wide Dataset so is not suitable for a whole population approach.

3. **Segmenting using the Cambridge Multimorbidity Score**

Relative simplicity is key to ensuring a degree of understanding necessary to promote good adoption. As such, it was decided that the BNSSG adult population (18+ years) would be segmented to only five segments, and segment membership would be decided by the individual's Cambridge Multimorbidity Score. The task is therefore to determine the cut-off 'thresholds' on the Cambridge Score for determining which segment an individual falls into. Overall, healthier individuals with lower scores would be contained in the lower-down segments, and less healthy individuals with higher scores would be contained in the higher-up segments.

There is an innumerable combination of thresholds that could be used, with essentially no right or wrong 'answer'. To guide quantification, we strive for the following three properties, satisfaction of which is considered to provide an intuitive and relatable description of the population and its health needs:

1. Segments would halve in size, going from lower to higher order ones.
2. Segments would double in mean spend, going from lower to higher order ones.
3. Any two segments would have the same total spend.

Numerically and logically, it can be seen that satisfaction of any two of these properties ensures satisfaction of the third. Therefore, we seek the solution to only two of the above, namely (1) and (3).

For (1), letting n_i equal the number of individuals in segment i, we therefore seek to minimise departure of the inter-segment ratios from 0.5 (the halving), i.e.

$$\left(\frac{n_2}{n_1} - 0.5\right)^2 + \left(\frac{n_3}{n_2} - 0.5\right)^2 + \left(\frac{n_4}{n_3} - 0.5\right)^2 + \left(\frac{n_5}{n_4} - 0.5\right)^2 \qquad \text{Eqn. (1)}$$

For (3), letting s_i equal the mean spend[11] of individuals in segment i, we therefore seek to minimise the standard deviation of total segment spends, i.e.

$$s.d.(n_1 s_1, n_2 s_2, n_3 s_3, n_4 s_4, n_5 s_5) \qquad \text{Eqn. (2)}$$

These two measures are calculated for each set of considered thresholds. To promote adequate coverage of the vast array of possible threshold combinations, increments of 0.01 were considered for each of the four cut-offs defining the five segments.

The Cambridge Scores are calculated on data as of year-end 2021/22, using the March 2022 cut of the SWD Attributes data. The spend data is from year 2021/22.

Data is available for 762,117 adult (18+ years) individuals, representing approximately 97% of the BNSSG adult population for which data was available[12] due to opt-outs received from Hartwood Healthcare, The Family Practice, The Merrywood Practice, and Birchwood Medical Practice.

Also note that, at the time of the analysis, three conditions required for the Cambridge Score were not present in the SWD Attributes data (sinusitis, prostate disorders, and diverticular disease). However, these conditions have relatively low general outcome weights so would not be expected to lead to significant discrepancy.

4. The BNSSG Core Segmentation

While absolute satisfaction of the three aforementioned properties could not be achieved, a sufficiently good alignment was possible. For full details see the appendix.

Figure 1 presents some key details of the segmentation, including the derived Cambridge Score cut-off thresholds, the proportion of the population per segment (i.e. (1) above), the mean

[11] For all points of delivery in the SWD Activity table, except Maternity.
[12] Recent data for Mendip Vale Medical Practice was not available.

annual per-person spend (2), and the proportion of total spend for that segment (3). Note that were property (3) satisfied absolutely, then the figures would all equal 20%.

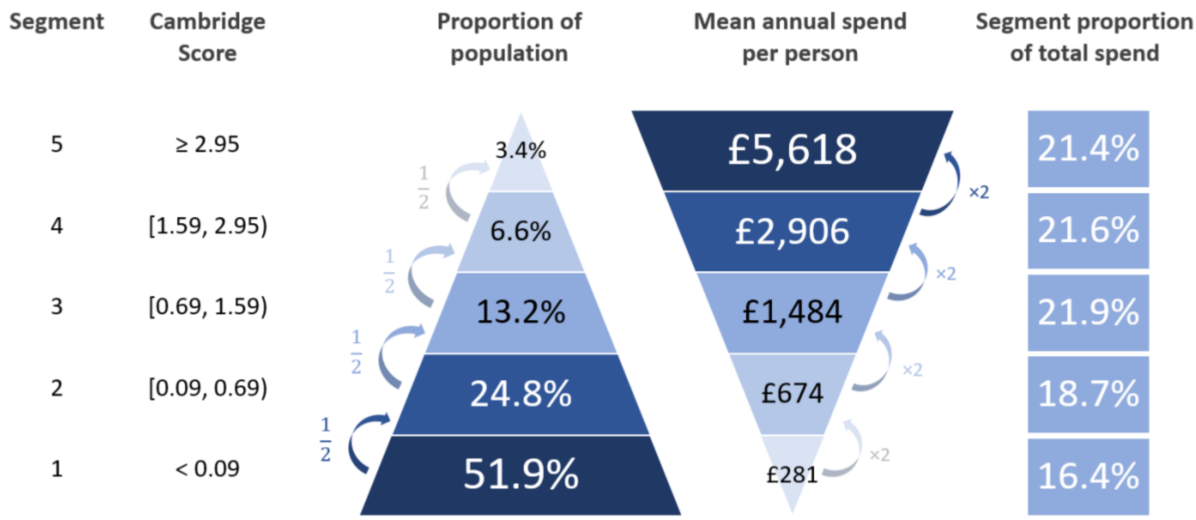

Figure 1. The BNSSG Core Segmentation of the adult (18+ years) population.

In terms of limitations, it should be noted that the three properties are not absolutely satisfied by this segmentation, and that alignment could be improved by including other variables alongside Cambridge Score. However, this would be at the penalty of additional complexity. It should also be noted that while these properties broadly hold on current year 2021/22 data, there is no guarantee they will hold in months and years to come. Such divergence, or segmental 'drift', will be important to monitor. Finally, it should be recalled that, while no specific biases are expected, only 73% of the BNSSG population has been included in this analysis, and there is no guarantee that the properties exhibited in Figure 1 would extend to the whole population.

Appendix

As mentioned in Section 4, the three properties of Section 3 could not be satisfied absolutely. Furthermore, in deciding the best cut-off threshold it was necessary to determine which of the two targeted properties (1) and (3) to favour, given that one 'solution' could not simultaneously yield minimal values of both Eqns (1) and (2). Filtering on a subset of possible solutions, Figure A.1 shows the value of each Eqn corresponding to the considered cut-off set. The chosen solution, i.e. the one reported in Section 4 and Figure 1, is coloured in red (see lower-left of the plot).

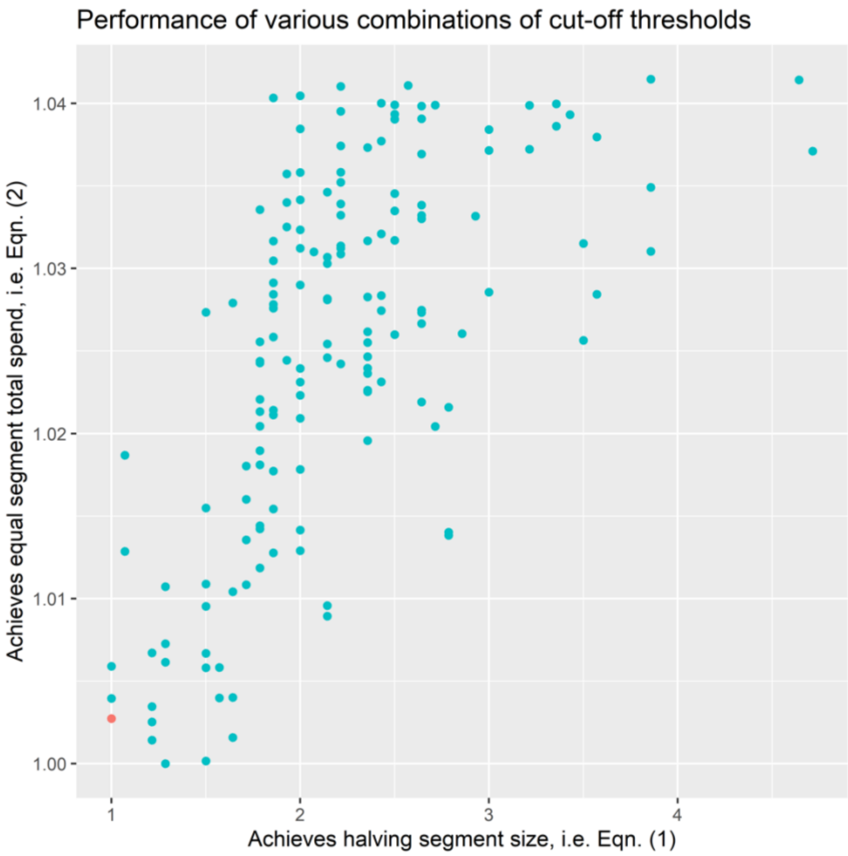

Figure A.1. Extent to which properties (1) and (3) are solved by various top-performing cut-off thresholds on Cambridge Score.

Section 4

The R script of the project are as follows:

```
rm(list=ls())

##########################################################################################################################
### USER INPUTS ###

start_time <- Sys.time() #mark start time

setwd("/Users/silverk/Silver_K/Exeter_Health_Data_Science/HPDM099_Research Project/Temp")

nreps<-1000  #this is the number of replications performed for strategies 4 and 5 (not relevant for strategies 0-3)

g<-0.5 #this is the weight applied to the value component of the objective function for strategies 4 and 5
#^(see code section 1.2.5 and definitions for strategies 4 and 5 in the paper, e.g. eqn. 11)

##########################################################################################################################
# 1. SETUP

# 1.1 PACKAGES

require(readxl)
require(tidyverse)
require(parallel)
require(packcircles)
require(gridExtra)

# 1.2 FUNCTIONS

# 1.2.1 function to capitalise first letter of each word
simpleCap<-function(x) {
  s<-strsplit(x," ")[[1]]
  paste(toupper(substring(s,1,1)),substring(s,2),
        sep="",collapse=" ")
}

# 1.2.2 function to sample one value at random from vector (including vector of length one)
sample1<-function(x) {
  if (length(x)<=1) {



    return(x)
  } else {
    return(sample(x,1))
  }
}

# 1.2.3 function to perform allocations using either greedy method or random
alloc_alg<-function(mlog,seed,strat) {
  set.seed(seed) #this is because there are random numbers used - it enables the same result to be derived each time
(which can help with continuity / debugging)
  acost<-0 #this is the 'accumulated cost', which starts at zero and is added to by the cost of each assigned
activity, at each iteration in the below 'while' loop
  while (any(mlog$status==0)) {     #if all of the mlog statuses =1 then all possible activities have been assigned,
i.e. the assignment loop needs to terminate
    if (strat %in% 0:3) {    # baseline or strategies 1-3
      mlog.row<-mlog %>%
        filter(elig==1 & status==0 & cost<=tcost-acost)     #this filters rows of the mlog file representing
activities that are eligible (see code section 2.3) and
      #have not yet been assigned (i.e. status=0) and whose assignment would not result in the cost constraint being
breached
      if (nrow(mlog.row)==0) break   # if there is no activity that meets the condition then terminate the
assignment loop
      mlog.row <- mlog.row %>%    # otherwise sample...
        slice(sample1(which(value==max(value))))  #only 1 activity can be selected - this ensures that with the
highest 'value' is selected
    } else if (strat %in% 4:5) {    #strategies 4 or 5
      mlog.row<-mlog %>%
        filter(elig==1 & status==0 & cost<=tcost-acost)
      if (nrow(mlog.row)==0) break
      mlog.row <- mlog.row %>%
        sample_n(1)
    }
    if (nrow(mlog.row)==0) break    # if, after applying the above filters, there is no activities that can be
assigned then the assignment loop needs to terminate
    mlog$status[which(mlog$pat.ref==mlog.row$pat.ref & mlog$act.ref==mlog.row$act.ref)]<-1   # otherwise, if there
is an activity that can be assigned, then flip the status from 0 to 1
    # to indicate it is assigned (and therefore cannot be considered for future assignments, as above)
```

```
    acost<-acost+mlog.row$cost   # update the accumulated cost by the cost of the newly assigned activity
  }
  # when the assignment loop has terminated then add on the seed to the mlog table
  mlog<-mlog %>%
    mutate(seed=seed)
  return(mlog)
}

# 1.2.4 function for selecting best allocations for strategies 4 and 5
alloc_sel_s45<-function(res,strat) {
  tres_a<-res %>%
    group_by(seed) %>%
    summarise(objfn_t1a=sum(status*value),cost=sum(status*cost)) %>%
    mutate(objfn_t1b=(max(objfn_t1a)-objfn_t1a)/(max(objfn_t1a)-min(objfn_t1a))) %>%
    mutate(objfn_t1b=ifelse(is.na(objfn_t1b),0.5,objfn_t1b))
  tres_b<-res %>%
    group_by(seed,pat.ref) %>%
    summarise(value=sum(status*var.opt)) %>%
    summarise(objfn_t2a=var(value)) %>%
    mutate(objfn_t2b=(objfn_t2a-min(objfn_t2a))/(max(objfn_t2a)-min(objfn_t2a))) %>%
    mutate(objfn_t2b=ifelse(is.na(objfn_t2b),0.5,objfn_t2b))
  tres<-left_join(tres_a,tres_b,by="seed") %>%
    mutate(objfn_a=g*objfn_t1a-(1-g)*objfn_t2a) %>%
    mutate(objfn_b=g*objfn_t1b+(1-g)*objfn_t2b)
  tres_out<-res %>%
    #filter(seed==tres$seed[which.max(tres$objfn_a)])
    filter(seed==tres$seed[which.min(tres$objfn_b)])
  return(tres_out)
}

# 1.2.5 function for calculating objective function/objfn for simulated annealing
objectiveFunction <- function(res){
  res_a <- res %>%
    mutate(objfn_t1 = sum(status*value)) #calculate first component of objfn
  res_b <- res %>%
    group_by(pat.ref) %>%
    summarise(ind_value = sum(status*var.opt)) %>%
    mutate(objfn_t2 = var(ind_value)) #calculate second component of objfn
  res<-left_join(res_a,res_b,by='pat.ref') %>%
    mutate(objfn = g*objfn_t1 - (1-g)*objfn_t2) #calculate obj function value
  return(list(res$objfn[1], res_a$objfn_t1[1], res_b$objfn_t2[1])) #returns objfn value, short and long term value,
variance)
}

# 1.2.6 Simulated Annealing function for Strategies 4 & 5
simulatedAnnealing<-function(res, seed, niter, step){
  '
  Parameters

  res - dataframe, which is the best allocation from 1000 replications
  seed - random number generator (RNG) state for random number generation
  niter - number of iterations
  step - size of the step in reduction of temperature
  '
  set.seed(seed)
  current_res <- res #assign 'res' as current allocation
  best_res <- current_res #current allocation is the best at the moment
  current_fv <- objectiveFunction(current_res)[[1]] #calculate function value of current allocation
  best_fv <- current_fv #current allocation is the best at the moment
  for (k in 1:as.numeric(niter)) {
    Temp <- (1-as.numeric(step))^k #calculate the temperature from k and step

    nb_res <- current_res #neighbour allocation will be deduced from current one
    nb_tcost <- sum(nb_res$status*nb_res$cost) #total cost of neighbour/current allocation

    nb_alloc <- nb_res %>%
      filter(status==1) #filtering activities included in current allocation
    nb_unalloc <- nb_res %>%
      filter(status==0) #filtering activities NOT included in current allocation

    nb_res.A <- nb_alloc %>%
      sample_n(1) #randomly select one from allocated activities

    j = 0
```

```
    while (j < 10000){
      nb_res.U <- nb_unalloc %>%
        sample_n(1) #randomly select one from allocated activities
      if ((nb_res.U$cost + nb_tcost - nb_res.A$cost) <= tcost) {
        #if swapping the allocated and unallocated ones does not violate the cost constraint, the activity will be
selected.
        #unless, it will loop until such activity is selected
        break
      }
      j = j + 1
    }

    #updating neighbour allocation
    nb_res$status[which(nb_res$pat.ref==nb_res.A$pat.ref & nb_res$act.ref==nb_res.A$act.ref)]<-0
    nb_res$status[which(nb_res$pat.ref==nb_res.U$pat.ref & nb_res$act.ref==nb_res.U$act.ref)]<-1

    nb_fv <- objectiveFunction(nb_res)[[1]] #calculate function value of neighbour allocation

    if (current_fv <= nb_fv){ #allocation with better fv is selected
      current_res <- nb_res
      current_fv <- nb_fv
      if (best_fv <= current_fv){ #comparison with best allocation
        best_res <- current_res
        best_fv <- current_fv
      }
    } else if (exp((current_fv-nb_fv)/Temp) > runif(1,0,1)){
      #if neighbour is not better than current,accept the move with a probability based on the current temperature
      current_res <- nb_res
      current_fv <- nb_fv
    }
  }
  return(list(best_res, best_fv))
}

##############################################################################################################################
# 2. READ DATA AND PRE-PROCESSING

# 2.1 read data
x1<-read_excel("inputs_core_seg.xlsx",sheet="attributes")
x2<-read_excel("inputs_core_seg.xlsx",sheet="activity")
x3<-read_excel("inputs_core_seg.xlsx",sheet="cost")

# 2.2 cost
tcost<-x3$total.cost     # the total cost that cannot be exceeded under and circumstances (this would be an
'infeasible solution')

# 2.3 set up raw attributes and activity log - this is never changed after being defined here
aalog_raw<-left_join(x2,x1,by="cohort") %>%
  uncount(volume) %>%     # this duplicates the activity rows by the provided 'volumes' (e.g. if 2x GP consultation
demanded, then they'll be two rows)
  mutate(act.ref=1:nrow(.)) %>%
  mutate(val_s0=0,cost_s0=stcp,     # see the paper for these, they should be the same equations (all should be
described in the paper)
         val_s1=sto/stcp,cost_s1=stcp,
         val_s2=(sto+lto)/(stcp-ltcp),cost_s2=stcp,
         val_s3=(sto+lto)/(stcp+stcc-ltcp),cost_s3=stcp+stcc,
         val_s4=(sto+lto)/(stcp+stcc-ltcp),cost_s4=stcp+stcc,
         val_s5=(sto+lto)/(stcp+stcc-ltcp),cost_s5=stcp+stcc) %>%
  mutate(elig_s0=ifelse(stcc==0,1,0),
         elig_s1=ifelse(stcc==0,1,0),
         elig_s2=ifelse(stcc==0,1,0),
         elig_s3=1,
         elig_s4=1,
         elig_s5=1)

# 2.4 set up raw main log
mlog_raw_assist<-aalog_raw %>%   # this object simply includes the cumulative count of patients in each of the
cohort...
  group_by(cohort) %>%         # ... it is used just to help in the construction of mlog_raw (below)
  summarise(number=unique(number)) %>%
  mutate(cumnumber=cumsum(number)) %>%
  select(cohort,cumnumber)
mlog_raw<-left_join(aalog_raw,mlog_raw_assist,by="cohort") %>%
  select(cohort,act.ref,number,cumnumber) %>%
```

```
  uncount(number,.remove=FALSE) %>%
  group_by(cohort,act.ref) %>%
  mutate(pat.ref={cumnumber-number+1}:{cumnumber}) %>%
  ungroup() %>%
  select(pat.ref,act.ref) %>%
  mutate(status=0)
# mlog_raw is a table with as many rows as there are distinct patient activities demanded/needed
# it has 3 columns including the pat.ref (1 to the total number of patients, i.e. the sum of column B on the
'attributes' tab of the excel input file)
# ... the act.ref is a reference to the distinct activities included in aalog_raw (column act.ref)
# ... status is whether the activity is currently assigned or not - the default is zero, and it flips to 1 when
assigned in, e.g., alloc_alg in code section 1.2.4

##########################################################################################################################
# 3. PERFORM ALLOCATIONS

strategy_names<-c("baseline","strategy 1","strategy 2","strategy 3","strategy 4","strategy 5")

# 3.1 baseline
aalog_s0<-aalog_raw %>%   # this is taking the raw activity log and selecting only the necessary variables for
making assignments
  select(act.ref,val_s0,cost_s0,elig_s0) %>%   # the necessary variables are the value and cost of the activity,
and whether it is 'eligible'
  #(i.e. whether or not it can be considered for assigment)
  rename(value=val_s0,cost=cost_s0,elig=elig_s0)   # standardises column names so it can be used by the generic
assignment function
mlog_s0<-left_join(mlog_raw,aalog_s0,by="act.ref")   # joins the above to the mlog_raw, thus creating a table with
all possible demanded/needed activities,...
# .... and their various values, costs, and eligibility
res_s0<-alloc_alg(mlog=mlog_s0,seed=1,strat=0)      # this is then passed to the allocation algorithm

# 3.2 strategy 1: greedy algorithm to assign highest SHORT-TERM value activity until cost reached
aalog_s1<-aalog_raw %>%       #same things happening as above for 3.1
  select(act.ref,val_s1,cost_s1,elig_s1) %>%
  rename(value=val_s1,cost=cost_s1,elig=elig_s1)
mlog_s1<-left_join(mlog_raw,aalog_s1,by="act.ref")
res_s1<-alloc_alg(mlog=mlog_s1,seed=1,strat=1)
```

```
# 3.3 strategy 2: greedy algorithm to assign highest SHORT AND LONG-TERM value activity until cost reached
aalog_s2<-aalog_raw %>%
  select(act.ref,val_s2,cost_s2,elig_s2) %>%
  rename(value=val_s2,cost=cost_s2,elig=elig_s2)
mlog_s2<-left_join(mlog_raw,aalog_s2,by="act.ref")
res_s2<-alloc_alg(mlog=mlog_s2,seed=1,strat=2)

# 3.4 strategy 3: greedy algorithm to assign highest SHORT AND LONG TERM value activity APPRECIATING COST-COERSION
until cost reached
aalog_s3<-aalog_raw %>%
  select(act.ref,val_s3,cost_s3,elig_s3) %>%
  rename(value=val_s3,cost=cost_s3,elig=elig_s3)
mlog_s3<-left_join(mlog_raw,aalog_s3,by="act.ref")
res_s3<-alloc_alg(mlog=mlog_s3,seed=1,strat=3)

# 3.5 strategy 4: multiple runs of random greedy algorithms
aalog_s4<-aalog_raw %>%
  mutate(var.opt=sto/(stcp+stcc)) %>%
  select(act.ref,val_s4,cost_s4,elig_s4,var.opt) %>%
  rename(value=val_s4,cost=cost_s4,elig=elig_s4)
mlog_s4<-left_join(mlog_raw,aalog_s4,by="act.ref")
cl<-makeCluster(detectCores()-1)
clusterExport(cl=cl,varlist=c("tcost","sample1"),envir=environment())
clusterEvalQ(cl=cl,c(library(dplyr)))
tres_s4<-do.call("bind_rows",parLapply(cl,1:nreps,alloc_alg,mlog=mlog_s4,strat=4))
stopCluster(cl)
alloc_s4<-alloc_sel_s45(res=tres_s4,strat=4)
res_s4_R<-alloc_s4

# 3.5.1 strategy 4: simulated annealing
best_alloc_s4 <- simulatedAnnealing(res_s4_R, 16, 1000, 0.1)
res_s4 <- best_alloc_s4[[1]] #dataframe

# 3.6 strategy 5: multiple runs of random greedy algorithms
aalog_s5<-aalog_raw %>%
  mutate(var.opt=sto) %>%
```

```
  select(act.ref,val_s5,cost_s5,elig_s5,var.opt) %>%
  rename(value=val_s5,cost=cost_s5,elig=elig_s5)
mlog_s5<-left_join(mlog_raw,aalog_s5,by="act.ref")
cl<-makeCluster(detectCores()-1)
clusterExport(cl=cl,varlist=c("tcost","sample1"),envir=environment())
clusterEvalQ(cl=cl,c(library(dplyr)))
tres_s5<-do.call("bind_rows",parLapply(cl,1:nreps,alloc_alg,mlog=mlog_s5,strat=5))
stopCluster(cl)
alloc_s5<-alloc_sel_s45(res=tres_s5,strat=5)
res_s5_R<-alloc_s5

# 3.6.1 strategy 5: simulated annealing
best_alloc_s5 <- simulatedAnnealing(res_s5_R, 16, 1000, 0.1)
res_s5 <- best_alloc_s5[[1]] #dataframe

##########################################################################################################################
# 4. CALCULATION DATASETS
# essentially, this code is pulling together the results as generated in code section 3, into a common 'calculation
dataset'

res_raw<-do.call("bind_rows",lapply(0:5,function(i) {
  y1<-get(paste0("res_s",i)) %>%
    select(pat.ref,act.ref,status,seed,elig)
  y2<-left_join(y1,aalog_raw %>%
      select(act.ref,pod1,pod2,name,sto,stcp,stcc,lto,ltcp),by="act.ref") %>%
    select(-act.ref) %>%
    mutate(stv=sto/(stcp+stcc),sltv=(sto+lto)/(stcp+stcc-ltcp),
           expr.demand=ifelse(stcc==0,1,0)) %>%
    mutate(strategy=strategy_names[i+1])
  return(y2)
}))

resA_names<-data.frame(measure_id=paste0("v",1:9),
                       measure=c("Expressed_demand_allocated",
                                 "Unexpressed_demand_allocated",
                                 "All_demand_allocated",
                                 "Short_term_cost_incurred",
                                 "Long_term_cost_avoided",
                                 "Short_term_outcome_improvement",
                                 "Long_term_outcome_deterioration_avoided",
                                 "Short_term_value",
                                 "Short_and_long_term_value")) %>%
  mutate(measure=gsub("_"," ",measure)) %>%
  mutate(measure=factor(measure,levels=measure))

resA_values<-res_raw %>%
  group_by(strategy,name,pod1,pod2,pat.ref) %>%
  summarise(v1=sum(status==1 & expr.demand==1),
            v2=sum(status==1 & expr.demand==0),
            v3=sum(status==1),
            v4=sum(stcp*status+stcc*status),
            v5=sum(ltcp*status),
            v6=sum(sto*status),v7=sum(lto*status),
            v8=sum(stv*status),v9=sum(sltv*status),
            nreps=length(unique(seed))) %>%
  mutate_at(vars(starts_with("v")),~./nreps) %>%
  select(-nreps) %>%
  pivot_longer(cols=-c(strategy,name,pod1,pod2,pat.ref),names_to="measure_id",
               values_to="values")

resA_poss1<-res_raw %>%
  group_by(strategy,name,pod1,pod2,pat.ref) %>%
  summarise(v1=sum(expr.demand==1),v2=sum(expr.demand==0),v3=sum(expr.demand>=0),
            v4=sum(stcp+stcc),v5=sum(ltcp),
            v6=sum(sto),v7=sum(lto),
            v8=sum(stv),v9=sum(sltv),nreps=length(unique(seed))) %>%
  mutate_at(vars(starts_with("v")),~./nreps) %>%
  select(-nreps) %>%
  pivot_longer(cols=-c(strategy,name,pod1,pod2,pat.ref),
               names_to="measure_id",values_to="poss1")

resA_poss2<-res_raw %>%
  group_by(strategy,name,pod1,pod2,pat.ref) %>%
  filter(elig==1) %>%
```

```
  summarise(v1=sum(expr.demand==1),v2=sum(expr.demand==0),v3=sum(expr.demand>=0),
            v4=sum(stcp+stcc),v5=sum(ltcp),
            v6=sum(sto),v7=sum(lto),
            v8=sum(stv),v9=sum(sltv),nreps=length(unique(seed))) %>%
  mutate_at(vars(starts_with("v")),~./nreps) %>%
  select(-nreps) %>%
  pivot_longer(cols=-c(strategy,name,pod1,pod2,pat.ref),
               names_to="measure_id",values_to="poss2")

resA<-left_join(resA_values,resA_poss1,
                by=c("strategy","name","pod1","pod2","pat.ref","measure_id"))%>%
  left_join(resA_poss2,
            by=c("strategy","name","pod1","pod2","pat.ref","measure_id")) %>%
  left_join(resA_names,by="measure_id") %>%
  select(-measure_id) %>%
  mutate(poss1=replace_na(poss1,0)) %>%
  mutate(poss2=replace_na(poss2,0))

# calculating short and long term value and variance of allocations by the former method vs simulated annealing
val_s4_SA <- objectiveFunction(res_s4)[[2]]
var_s4_SA <- objectiveFunction(res_s4)[[3]]
val_s4_R <- objectiveFunction(res_s4_R)[[2]]
var_s4_R <- objectiveFunction(res_s4_R)[[3]]

val_s5_SA <- objectiveFunction(res_s5)[[2]]
var_s5_SA <- objectiveFunction(res_s5)[[3]]
val_s5_R <- objectiveFunction(res_s5_R)[[2]]
var_s5_R <- objectiveFunction(res_s5_R)[[3]]

##############################################################################################################################
# 5. SUMMARISE RESULTS
# producing the plots etc from the calculation datasets generated in code section 4

#######################################################
# plot 2: cohort and activity level

pdf("plot_2.0_allocations_by_strategy.pdf",height=6,width=8)
```

```
td1<- subset(resA, strategy=='baseline' |
               strategy == 'strategy 4' |
               strategy == 'strategy 5') %>%
  group_by(strategy,measure,pod1) %>%
  summarise(values=sum(values),poss1=sum(poss1),poss2=sum(poss2))
td2<-subset(resA, strategy=='baseline' |
              strategy == 'strategy 4' |
              strategy == 'strategy 5') %>%
  group_by(strategy,measure) %>%
  summarise(values=sum(values),poss1=sum(poss1),poss2=sum(poss2))
print(td1 %>%
        ggplot(aes(x=strategy,y=values)) +
        geom_bar(stat="identity",aes(fill=pod1)) +
        facet_wrap(~measure,scales="free_y",nrow=4) +
        labs(title="Allocations") +
        xlab("Strategy") +
        theme(legend.position="bottom",
              legend.title=element_blank(),
              axis.title.x=element_blank(),
              axis.title.y=element_blank(),
              axis.text.x=element_text(angle=25,hjust=1),
              strip.text.x=element_text(size=8))
)
dev.off()

#######################################################
# plot 3: variances (person-level)

pdf("plot_3.0_variance_by_strategy.pdf",height=6,width=8)
print(subset(resA, strategy=='baseline' |
               strategy == 'strategy 4' |
               strategy == 'strategy 5') %>%
        group_by(strategy,pat.ref,measure) %>%
        summarise(values=sum(values)) %>%
        group_by(strategy,measure) %>%
        summarise(values=var(values)) %>%
        ggplot(aes(x=strategy,y=values)) +
```

```
        geom_bar(stat="identity") +
        facet_wrap(~measure,scales="free_y",nrow=4) +
        labs(title="Variance between individuals") +
        xlab("Strategy") +
        theme(legend.position="none",
              axis.title.x=element_blank(),
              axis.title.y=element_blank(),
              axis.text.x=element_text(angle=25,hjust=1),
              strip.text.x=element_text(size=8))
)
dev.off()

end_time <- Sys.time() #mark end time
end_time - start_time #time spent
```