Investigating practical team composition problems for a teaching environment, with regard to personality

Submitted September 2022, in partial fulfillment of
the conditions for the award of the degree **MSc Computer Science.**

Angus Robinson
20410493

Supervised by Geert De Maere

School of Computer Science
University of Nottingham

Date 26 / Sep / 2022
EDITED January 2023.

Abstract

Team composition is the problem of dividing a group of people up into fair teams. This thesis looks at the educational context of student group projects, and examines different ways to optimise team allocation with regard to the levels of academic proficiency and personalities of individual students. It compares linear programming approaches to heuristic approaches, and proposes marginal improvements to Andrejczuk et al's SynTeam algorithm, developing SynTeamPlus, which has some promising experimental results on various simulated student cohorts. It also investigates the division of team composition problems into subproblems of manageable sizes, and concludes that this may be a feasible way to use linear programming for larger problems. Finally, it examines two different objective metrics in this context: (1) maximising the Nash product of team proficiencies and (2) minimising the maximum difference in proficiency between any two teams. It finds that within the SynTeam process, the use of the Nash product vastly improves performance.'

Keywords: *Team composition, Personality, Big Five, Optimisation, Education, SynTeam*

Note on the January 2023 edit

This version of the thesis is slightly different from the one I submitted in September 2022, and which was graded. The changes I have made are as follows:

- I fixed half a dozen typos, although I dare say I missed some.
- I reworded a couple of imprecise paragraphs.
- I reran Experiment 2 with more datasets, and replaced the graph, table, and relevant paragraph to make the results clearer.

Angus Robinson

16/01/2023

Acknowledgements

I want to thank Dr Geert De Maere, my supervisor, for helping and putting up with me for the past few months.

Thanks also to ****, a friend, who helped me to write the subdividing script, and my girlfriend **** for humoring my odd work schedule, as well as my sister ****, who provided much needed support over the last few days before submission.

And finally thanks to Dr Ewa Andrejczuk, whose work formed the starting point for all of this research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The aim of this project is to investigate various approaches to team composition problems in a student environment, with personality traits taken into account. A mathematical definition of the problem can be found in the literature, but for the purposes of introduction, it is the old schoolyard question: how do you pick fair teams?

1.1 Motivation

Many university courses, including the MSc in Computer Science at the University of Nottingham, use group projects as a way to teach and assess students. Often the groups are randomly allocated, or students are allowed to pick their own groups. However, these both seem bad ways of partitioning a cohort, if it is desirable to have groups of approximately equal ability.

The reason it is desirable to have groups of equal ability is the premise that no student should be disadvantaged by something outside their own proficiency and work. If there are two students, equal in every respect with regard to their own proficiency and work ethic, and one is assigned to a less capable group than the other, that is tantamount to putting an arbitrary penalty on the former's final grade.

Letting students pick their own teams is likely to be particularly inequitable. Students will be incentivised to join teams with the highest overall proficiency they can, and so the students with the highest degree of choice will be those most individually capable. If there were no other considerations, this would result in teams chosen strictly on ability—the best n students would be on one team, the next best n students on another, right down until the least capable n students in a team of their own. This is unlikely to happen in the real world, where a strict ranking of students is not possible, and students are likely to choose their friends, but it demonstrates that the incentives are likely to draw any team allocation towards rather than away from inequity.

Choosing teams randomly (i.e. with no regard to the overall strength of the team)[1] would not be as bad overall, but is still likely, given a large enough cohort, to result in some teams with significantly lower overall proficiency than some others.

This project therefore addresses ways of partitioning cohorts into teams that minimise the difference in overall proficiency between teams.

1.2 Aims and Objectives

1. This project aims to investigate in the literature the strength of evidence as to the relevance of personality in team composition, and derive approximate metrics by which the strength of a team can be predicted from knowledge of the individual capabilities and personalities of the students.

2. This project aims to devise and test various methods of optimising team composition for students, and evaluate them based on speed and quality of the resulting team allocation.

[1] This project does not explicitly deal with other constraints. The University of Nottingham, for example, has restrictions on whether there can be only one female student in a group.

1.3 Note on Ethics

This section is not an attempt to comprehensively address all the possible ethical considerations of implementing these methods for optimising team composition, some of which will be considered in the Discussion. It is rather the author's holistic defence of the entire endeavour.

Some people may have privacy concerns with using personality data in this way, but there is a simple answer to that objection: you do not need to make make any personality test mandatory to get improvements in team composition. If personality does have an impact on team performance, even partial data should be helpful. A simple solution would be to ask an entire cohort to fill in the personality form, letting them know that the more responses there are, the less likely any of them is to be disadvantaged by the allocation of teams. The missing data from those students who do not wish to fill in the forms can be replaced with typical values, and it will still be an improvement over standard practice. Moreover, the whole process could be automated, so that no administrative staff need even see the data.

There is a broader point to be made here, though—which is that every effort should be made to guarantee fairness in team allocation. Overall grades at schools and universities are awarded to individuals, rather than teams, and they can strongly affect those individuals' futures. Any way those grades might be negatively affected that is outside their own personal control should be avoided if possible, and unfair team allocations is quite clearly one of those ways, and one which is almost standard practice. Data privacy is an important consideration, but it is far from the only ethical consideration with which an education system has to concern itself.

Chapter 2

Background and Related Work

There are two main categories of research relevant to this thesis: firstly the Psychology literature, which can be used to try to understand the effect of personality on team performance; and secondly the Computer Science literature on team composition, particularly those papers which have previously made attempts at using personality to develop solutions to team composition problems.

2.1 Psychology

The Big Five is a tool used in the field of Psychology to assign people personality traits. It originated with Tupes and Christal [28], and after some terminology change, it has become the dominant way of discussing personality in the literature, cited in more than twice as many Psychology articles as its nearest alternative.[6].

The Big Five refers to five characteristics: (1) extraversion, (2) agreeableness, (3) conscientiousness, (4), neuroticism and (5) openness.[14] These are not personality types, but traits, meaning that everybody has a score on each of these scales: somebody can be high on agreeableness, conscientiousness and extraversion, and low on neuroticism and openness, or any other combination. Each trait is derived from a set of highly correlated terms in descriptions of personalities. For instance, people who are called 'kind' are also very likely to be called 'generous', and these might be markers for agreeableness. For more information see 'The Development of Markers For the Big Five Factor Structure'.[14] It is possible for anybody to create a test to determine a Big Five score, using the International Personality Item Pool [15] and the IPIP website.

As well as being the dominant mode of discussing personality among psychologists, it has been suggested that the Big Five already encompasses many other valid personality metrics. Bainbridge et al concluded that 'the Big Five can indeed serve as an organizing framework for a sizable majority of stand-alone psychological trait scales and that many of these scales could reasonably be labeled as facets of the Big Five'[6].

The effect of personality on team performance is a messy experimental issue. Investigations into this under various conditions have been undertaken by LePine [18], Crawford, Rahaman and Sen [9], and Graziano, Hair and Finch [16]. However, most useful are three meta-analyses on the subject, performed by Peeters, van Tuijl, Rutte and Reymen [25], Suzanne Bell [7], and LePine, Buckman, Crawford and Methot [19]. Peeters et al found that the only two personality traits of the Big Five to have any significant bearing upon team performance were conscientiousness and agreeableness. They found that there were significant effects for both elevation and variability—that is, first of all that the more conscientious and more agreeable the members of a team were individually, the better the team performed, and second of all that teams where members had similar levels of agreeableness and conscientiousness performed better than those with widely varying levels. Bell found once again that high agreeableness and high conscientiousness had significant effects on team performance, but only in a field setting, rather than a laboratory setting. She also found that higher extraversion had a weaker, but still significant positive effect. LePine et al reiterated that elevated agreeableness and conscientiousness have the strongest effects, and observed weaker effects for both extraversion and openness, but once again noted that these effects are significantly weaker in a laboratory setting. However, they are more sceptical of Peeters et al's finding that low trait variability was significant, and also of the idea that different traits might interact: 'It is clear, given the lack of extensive work in this area, as well as the somewhat mixed results, that much more work is needed to gain an accurate understand of the complexities of trait interaction, at both the individual-level as well as the team-level.'[19]

2.2 Computer Science

Some specific applications of team composition have been investigated by Farhangian, Purvis, Purvis, and Savarimuthu [13] (for software teams) and Okimoto, Schwind and Clement [23] (for teams robust to losing members). This thesis relies on and responds to one collection of research in particular. This is the body of work done by Ewa Andrejczuk during her PhD Thesis[5], which was titled 'Artificial Intelligence methods to support people management in organisations'. The relevant papers which were released during this research include 'The Composition and Formation of Effective Teams'[2], which compares the practices of Organisational Psychology and previous computer science work into team composition, making the comparison that Organisational Psychology requires a broader definition of capacity as subject to change and relative to motivation, while Computer Science approaches have the strength that they reckon with cost minimisation and 'robustness' (that is, whether a team can perform a task with only a subset of its members), and not just try to maximise the performance of a team. 'Synergistic Team Composition'[4] is the most important paper of the set, as it states a formal version of the team composition problem (slightly different to the one this paper addresses), and introduces the SynTeam algorithm, a heuristic for finding a near-optimal solution to a team composition problem in relatively little time. This is tested in 'Heterogeneous Teams for Homogenous Performance'[3], which compares the complexity and performance of SynTeam with a linear programming model.

There is one other relevant paper in the same vein, by different researchers: Juan Alberola and colleagues published 'An Artificial Intelligence Approach to Team Formation'[1]. In this paper, they present an iterative method for producing a team partition, based on students responding to questionnaires about their classmates after each given task in a series. They present weak experimental success with their method. Unfortunately, this approach requires an environment in which students are repeatedly doing group projects in different teams, and while this may be appropriate for some schools, it is not generally appropriate for the primary use case identified in the Introduction to this thesis, which is in a university setting, where most group projects are likely to last longer, and less likely to come within a sequence of such projects for the same module.

Andrejczuk's work in particular is very useful, but both Alberola and she are insufficiently sceptical of their personality metrics. The Psychology literature, which was identified at the beginning of this literature review, is fairly clear on established best practice. Andrejczuk uses a modified version of the Myers-Briggs Type Indicator, while Alberola uses the Belbin Team Inventory. A good sense of the poor quality of the MBTI framework can be found in David J. Pittenger's 'The Utility of the Myers-Briggs Type Indicator'[26]. Pittenger concludes that 'there is insufficient evidence to support the tenets of and claims about the utility of the test'(p1). He does this on the empirical basis that there is low test-retest reliability (that is, people can often retake the test and get a different answer), and on the theoretical basis that the Types of the theory are adapted from Jungian theory which is lacking in any rigorous scientific underpinning. Furthermore, he argues, to the extent that there are meaningful distinctions, they are already present within the Big Five personality traits.

Chapter 3

Design

3.1 Proficiency calculus

For these models, it is necessary to have some way of estimating the proficiency of teams. Based on the conclusions of the meta-analyses referred to in the literature review[7][25][19], the working assumptions for this project were that the aggregate agreeableness and aggregate conscientiousness of members of a team had a positive effect on team proficiency, but variability had no effect. Therefore each team's proficiency was the sum of the proficiency of each student. The numbers assigned were those of Peeters et al[25]: the personality input towards proficiency was $\frac{6}{11}$ conscientiousness and $\frac{5}{11}$ agreeableness. An assumption was made that academic performance (measured for instance by a mark on a similar course) predicted 80 per cent of a student's proficiency, while the remaining twenty per cent was personality.[1] Thus the following definition, for a team consisting of students 1 to n:

$$Team\ proficiency := \sum_{i=1}^{n} \frac{4}{5}\alpha_i + \frac{6c_i + 5a_i}{55}$$

where α_i, c_i and a_i are respectively defined as the academic performance, conscientiousness and agreeableness of student i, in each case as a percentage.

Ultimately, this definition is easily replaced with any additive definition of team proficiency—that is to say that as long as the team proficiency is defined as the sum of individual pro-

[1]This ratio is easily changed, as depending on the task at hand students will have more or less interaction, and if it is not desirable to use personality at all, it can be set to 100:0.

ficiencies, without regard for matching personality traits between students, the models and algorithms tested in this project will still function identically. Further research could identify more accurate versions of this formula, but the experimental results would still apply.

3.2 Evaluation

There are two different ways of evaluating fairness used in this project: firstly the difference between the most and least proficient teams, and secondly with the Bernoulli-Nash function[22], which multiplies the expected competences of all teams together. The fairest partition by this second metric is the one with the highest overall product, as numbers which are closer together have a higher product than numbers with the same average but further apart. To avoid dealing with large numbers from multiplying many proficiencies together, this project instead adds the natural logarithms of teams' expected proficiency, which can be simply proved as follows to obey the same ordering:

$$\forall u, v, x, y \in \mathbb{R}_+ : uv > xy \iff \ln uv > \ln xy \iff \ln u + \ln v > \ln x + \ln y$$

In this project, each team partition is evaluated with both metrics:

$$Eval1 = \prod_{i=1}^{NumberofTeams} \textit{Proficiency of team } i$$

$$Eval2 = \textit{Proficiency of most proficient team - Proficiency of least proficient team}$$

3.3 Models and Algorithms

In this project, the author used a variety of models, some using linear programming, and others using heuristic algorithms to navigate the search space. In this section, first an outline of the each model and algorithm will be explained thoroughly.

3.3.1 Deviation Model 1 (”dev1”)

Deviation Model 1 is possibly the most naive approach. It uses mixed-integer linear programming to select the partition that minimises the distance to the mean team proficiency: a variable matrix $M \in \mathbb{Z}_2^{A \times B}$keeps track of whether student A is in team B. Other variables work out what the proficiency of each team is for any given partition, the mean team proficiency, the distance from the mean for each team, and whether it is above or below the mean. Constraints make sure that each student is allocated to one and only one team. One variable is called *maxdev*, and it is constrained to be greater than or equal to the largest absolute difference of any single team from the mean team proficiency. The objective function is to minimise this variable. The strength of the model is that it is intuitive, and the objective function delightfully simple. Unfortunately the need for absolute values in linear programming detract from this simplicity.[2]

[2]This model also uses implication, which is not a linear inequality, but CPLEX, IBM’s solver, is happy to linearise it.

Data:	$NStudents \in \mathbb{N}$	
	$NTeams \in \mathbb{N}$	
	$prof_i \in \mathbb{R}^+$	$\forall i \in \mathbb{N}, i \leq NStudents$
	$MinTeamSize \in \mathbb{N}$	
	$MaxTeamSize \in \mathbb{N}$	
Variables:	$y_{ij} \in \{0,1\}$	$i, j \in \mathbb{N}, i \leq NStudents, j \leq NTeams$
	$Uprof_j \in \mathbb{R}^+$	$j \in \mathbb{N}, j \leq NTeams$
	$meanprof \in \mathbb{R}^+$	
	$better_j \in \{0,1\}$	$j \in \mathbb{N}, j \leq NTeams$
	$NAbsdev_j \in \mathbb{R}$	$j \in \mathbb{N}, j \leq NTeams$
	$Absdev_j \in \mathbb{R}^+$	$j \in \mathbb{N}, j \leq NTeams$
	$bd_j \in \mathbb{R}^+$	$j \in \mathbb{N}, j \leq NTeams$
	$maxdev \in \mathbb{R}^+$	$j \in \mathbb{N}, j \leq NTeams$
Minimise:	$maxdev$	
Subject to:	$\sum_{i=1}^{NStudents} y_{ij} \leq MaxTeamSize$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$\sum_{i=1}^{NStudents} y_{ij} \geq MinTeamSize$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$\sum_{j=1}^{NTeams} y_{ij} = 1$	$\forall i \in \mathbb{N}, i \leq NStudents$
	$Uprof_j = \sum_{i=1}^{NStudents} y_{ij} prof_i$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$meanprof = \frac{1}{NTeams} \sum_{j=1}^{NTeams} Uprof_j$	
	$Nabsdev_j = Uprof_j - meanprof$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$Nabsdev_j * small \leq \frac{better_j}{2}$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$-Nabsdev_j * small \leq \frac{1-better_j}{2}$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$better_j = 1 \implies bd_j = NAbsdev_j$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$better_j = 0 \implies bd_j = 0$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$Absdev_j = 2bd_j - Nabsdev_j$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$maxdev \geq Absdev_j$	$\forall j \in \mathbb{N}, j \leq NTeams$

3.3.2 Deviation Model 2 (”dev2”)

In Deviation Model 2, much of the same structure is used as in Deviation Model 1. Instead, however, of optimising for the lowest maximum deviation of any team from the mean team proficiency, it optimises for the minimal difference between the highest and lowest proficiency teams. The model has the same constraints to ensure that no student is allocated to more than one team, and that every team has the correct number of students. Deviation Model 2 keeps the benefit of an intuitive objective function, but unlike Deviation Model 1, there is no need for an absolute value function, as the difference between the proficiencies of the best and worst performing teams, according to the estimate, will always be positive.

Data:	$NStudents \in \mathbb{N}$	
	$NTeams \in \mathbb{N}$	
	$prof_i \in \mathbb{R}^+$	$\forall i \in \mathbb{N}, i \leq NStudents$
	$MinTeamSize \in \mathbb{N}$	
	$MaxTeamSize \in \mathbb{N}$	
Variables:	$y_{ij} \in \{0, 1\}$	$i, j \in \mathbb{N}, i \leq NStudents, j \leq NTeams$
	$Uprof_j \in \mathbb{R}^+$	$j \in \mathbb{N}, j \leq NTeams$
	$maxprof \in \mathbb{R}^+$	
	$minprof \in \mathbb{R}^+$	
Minimise:	$maxprof - minprof$	
Subject to:	$\sum_{i=1}^{NStudents} y_{ij} \leq MaxTeamSize$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$\sum_{i=1}^{NStudents} y_{ij} \geq MinTeamSize$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$\sum_{j=1}^{NTeams} y_{ij} = 1$	$\forall i \in \mathbb{N}, i \leq NStudents$
	$Uprof_j = \sum_{i=1}^{NStudents} y_{ij} prof_i$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$maxprof \geq Uprof_j$	$\forall j \in \mathbb{N}, j \leq NTeams$
	$minprof \leq Uprof_j$	$\forall j \in \mathbb{N}, j \leq NTeams$

3.3.3 Deviation Model 3 ("dev3")

In Deviation Model 3, exactly the same principles are used as in Deviation Model 2. The key difference is the use of additional constraints, called band constraints. Their purpose can be explained as follows: in a large model with lots of students, it is conceivable that there may be very many near-optimal possible solutions, particularly if the data is at low resolution, for instance if every student is only given a mark to the nearest five percent - as this leads to more symmetries (that is, areas of the search space which are equivalent to each other but a computer will exhaustively search anyway). If there are two students with exactly identical numbers in all the relevant categories, any partition which has them on different teams will have an exact pair with the two identical students swapped. However, even without the symmetries, it is likely that there may be many close to optimal solutions, and in order to improve the running time of the programme, the solution space must be cut down. The bands are one way to add extra constraints while keeping many close-to-optimal solutions. Each 'band' represents a fraction of the cohort, identified by how proficient a team-member each student is estimated to be. The constraints ensure that in each team there is no more than one student from each band, with the number of bands chosen so that each team of the smaller possible size having exactly one student from each band, and those from the larger teams having one from every band except one. However, as will be seen, this effort was largely unsuccessful, and the constraint relaxed so that two students from any given band were allowed in any team.

$$
\begin{array}{lll}
\text{Data:} & NStudents \in \mathbb{N} & \\
 & NTeams \in \mathbb{N} & \\
 & prof_i \in \mathbb{R}^+ & \forall i \in \mathbb{N}, i \leq NStudents \\
 & MinSize \in \mathbb{N} & \\
 & MaxSize \in \mathbb{N} & \\
 & Bandmatrix_{ik} \in \{0,1\} & \forall i,k \in \mathbb{N}, i \leq NStudents, k \leq MaxSize \\
 & & \\
\text{Variables:} & y_{ij} \in \{0,1\} & i,j \in \mathbb{N}, i \leq NStudents, j \leq NTeams \\
 & Uprof_j \in \mathbb{R}^+ & j \in \mathbb{N}, j \leq NTeams \\
 & maxprof \in \mathbb{R}^+ & \\
 & minprof \in \mathbb{R}^+ & \\
 & teambands_{jk} \in \{0,1\} & j,k \in \mathbb{N}, j \leq NTeams, k \leq MaxSize \\
 & & \\
\text{Minimise:} & maxprof - minprof & \\
 & & \\
\text{Subject to:} & \sum_{i=1}^{NStudents} y_{ij} \leq MaxSize & \forall j \in \mathbb{N}, j \leq NTeams \\
 & \sum_{i=1}^{NStudents} y_{ij} \geq MinSize & \forall j \in \mathbb{N}, j \leq NTeams \\
 & \sum_{j=1}^{NTeams} y_{ij} = 1 & \forall i \in \mathbb{N}, i \leq NStudents \\
 & Uprof_j = \sum_{i=1}^{NStudents} y_{ij} prof_i & \forall j \in \mathbb{N}, j \leq NTeams \\
 & maxprof \geq Uprof_j & \forall j \in \mathbb{N}, j \leq NTeams \\
 & minprof \leq Uprof_j & \forall j \in \mathbb{N}, j \leq NTeams \\
 & teambands_{jk} = \sum_{i=1}^{NStudents} y_{ij} bandmatrix_{ik} & \forall j,k \in \mathbb{N}, j \leq NTeams, k \leq MaxSize \\
 & teambands_{jk} \leq 1 & \forall j,k \in \mathbb{N}, j \leq NTeams, k \leq MaxSize
\end{array}
$$

3.3.4 Multiplicative Model ("m1")

The multiplicative model follows what might be considered the ideal way of solving these problems, as noted by Andrejczuk et al. Before creating any linear programming model at all, m1 writes out every possible team of either smaller or large size. This list of possible teams is subject to combinatorial explosion, whereby the length of the list increases very quickly as the number of students increases. Once you have this list of teams, the model is very simple in principle - it optimises for the best possible collection of teams (according to the Bernoulli-Nash evaluation method), subject to the constraints that none of the teams can overlap, and every student must be allocated to a team. The difficulty with this model is clearly the huge amount of computational resources needed to use it given a large set of students. For this project, it was not feasible to test it. For someone with those resources and time, it could be useful.

Data:	$A \in (\mathbb{Z}_2^{NStudents})^{NPossibleTeams}$	(List of all possible teams of the right size)
	$NTeams \in \mathbb{N}$	
	$Uprof_i \in \mathbb{R}^+$	$\forall i \in \mathbb{N}, i \leq NPossibleTeams$
	$logprof_i = ln(Uprof_i)$	$i \in \mathbb{N}, i \leq NPossibleTeams$
Variables:	$y_i \in \{0, 1\}$	$i \in \mathbb{N}, i \leq NPossibleTeams$
Maximise:	$\sum$	
Subject to:	$\sum_{i=1}^{NPossibleTeams} y_i = NTeams$	
	$\sum_{i=1}^{NPossibleTeams} y_i A_{ki}$	$\forall k \in \mathbb{N}, k \leq NStudents$

3.3.5 SynTeam

SynTeam is the name of the algorithm for solving team composition problems developed by Eva Andrejczuk and colleagues. They released two different versions of the algorithm, one in a standalone paper [4] and one in Andrejczuk's PhD thesis [5]. For the purposes of this research, 'SynTeam' refers to the former version. It must be noted, however, that the SynTeam algorithm is designed for a specific type of team composition problem: one in which team members have different competences, and the specific subtasks that each team needs to carry out are known prior to the partition. Since that is not the exact framing of the problem in this paper, the version of SynTeam used here has been minimally altered, and so may not be representative of the true SynTeam, nor do the original SynTeam developers claim it is the optimal way of solving this particular version of the team composition problem.

That said, the way that SynTeam works is as follows: start with a random partition of students into the correct team distribution (i.e. the correct number of teams of each size) and keep track of this as the current best partition, then pick two of the teams in that partition, and evaluate every other possible partition of those students into two teams of those size. If the best possible partition is better than the existing partition of those two teams, then replace the old two teams with the new ones. Do this if the new partition is worse than the current best partition of those two teams with some probability that decreases over time. Repeat this process, choosing two new random teams within the current best partition. Each time this step is repeated, subtract a cooling rate from a stored heat variable. By changing the cooling rate and the initial heat, you can control the number of iterations to go through before returning the current best partition as the solution.

1. **Algorithm 1** SynTeam (adapted from Andrejczuk et al (2017))

Require: $NTeams$ ▷ Number of teams

Require: $NStudents$ ▷ Number of students

Require: $heat$ ▷ Initial temperature

Require: $cooling\_rate$

Require: $NSmallTeams$ ▷ Number of smaller teams

Require: $NLargeTeams$ ▷ Number of larger teams

Require: $prof_i \forall i \in \mathbb{N}, i \leq NStudents$ ▷ Students' proficiency

Ensure: $PBest \in \mathbb{Z}_2^{NStudents \times NTeams}$ ▷ Current best partition

$MinTeamSize \leftarrow \lfloor \frac{NStudents}{NTeams} \rfloor$

$MaxTeamSize \leftarrow \lceil \frac{NStudents}{NTeams} \rceil$

$NSmallTeams \leftarrow NTeams \times MaxTeamSize - NStudents$
$NLargeTeams \leftarrow NTeams - NSmallTeams$
$shuffled \leftarrow random.shuffle([0, \ldots, NStudents])$
$Studentindex \leftarrow 0$
$Teamindex \leftarrow 0$
for $i \in \{0, \ldots, NsmallTeams - 1\}$ **do**
  for $j \in \{0, \ldots, MinTeamSize - 1\}$ **do**
    $PBest_{shuffled_{Studentindex} Teamindex} \leftarrow 1$
    $Studentindex \leftarrow Studentindex + 1$
  $Teamindex \leftarrow Teamindex + 1$
for $i \in \{NSmallTeams, \ldots, NTeams - 1\}$ **do**
  for $j \in \{0, \ldots, MaxTeamSize - 1\}$ **do**
    $PBest_{shuffled_{Studentindex} Teamindex} \leftarrow 1$
    $Studentindex \leftarrow Studentindex + 1$
  $Teamindex \leftarrow Teamindex + 1$
$PBesteval \leftarrow 0$
for $i \in \{0, \ldots, NTeams - 1\}$ **do**
  $teamprof \leftarrow 0$
  for $j \in \{0, \ldots, NStudents\}$ **do**
    $teamprof \leftarrow teamprof + (PBest_{ji} \times prof_j)$
  $PBesteval \leftarrow PBesteval + math.log(teamprof)$
while $(heat > 1)$ **do**
  $A_{index} \leftarrow random.randint(0, NTeams - 1)$
  $B_{index} \leftarrow random.randint(0, NTeams - 1)$
  while $(A_{index} == B_{index})$ **do**
    $B_{index} \leftarrow random.randint(0, NTeams - 1)$
  if $(A_{index} > B_{index})$ **then**
    $tmp \leftarrow A_{index}$
    $A_{index} \leftarrow B_{index}$

$B_{index} \leftarrow tmp$

$teamset \leftarrow \{\}$

$teama \leftarrow \{\}$

$teamb \leftarrow \{\}$

for $i \in \{0, \ldots, NStudents\}$ **do**

if $(PBest_{iA_{index}} == 1)$ **then**

$teama \leftarrow teama \cup i$

$teamset \leftarrow teamset \cup i$

if $(PBest_{iB_{index}} == 1)$ **then**

$teamb \leftarrow teamb \cup i$

$teamset \leftarrow teamset \cup i$

$teamprof_A \leftarrow 0$

$teamprof_B \leftarrow 0$

for $i \in teama$ **do**

$teamprof_A \leftarrow teamprof_A + prof_i$

for $i \in teamb$ **do**

$teamprof_B \leftarrow teamprof_B + prof_i$

$ABcurrenteval \leftarrow math.log(teamprof_A) + math.log(teamprof_B)$

if $(|teamset| == 2 \times MinTeamSize)$ **then**

$altA \leftarrow combinations(teamset, MinTeamSize)$

else

$altA \leftarrow combinations(teamset, MaxTeamSize)$

$altB \leftarrow$

for $i \in \{0, \ldots, |altA| - 1\}$ **do**

$altB_i \leftarrow teamset \setminus altA_i$

$ABbesteval \leftarrow 1$

$ABbestindex \leftarrow 0$

for $i \in \{0, \ldots, |altA| - 1\}$ **do**

$temprof_A \leftarrow 0$

$\quad\quad temprof_B \leftarrow 0$

$\quad\quad$ **for** $j \in altA_i$ **do**

$\quad\quad\quad teamprof_A \leftarrow teamprof_A + prof_j$

$\quad\quad$ **for** $j \in altB_i$ **do**

$\quad\quad\quad teamprof_B \leftarrow teamprof_B + prof_j$

$\quad\quad tempeval \leftarrow math.log(teamprof_A) + math.log(teamprof_B)$

$\quad\quad$ **if** $((tempeval > ABbesteval) \wedge (altA_i \neq teama))$ **then**

$\quad\quad\quad ABbesteval \leftarrow tempeval$

$\quad\quad\quad ABbestindex \leftarrow i$

$\quad$ **if** $(ABbesteval > ABcurrenteval) \vee (max(\frac{heat \times (1-(ABcurrenteval - ABbesteval))}{20}, 0) \geq random.random()$ **then**

$\quad\quad$ **for** $i \in altA_{ABbestindex}$ **do**

$\quad\quad\quad PBest_{iAindex} \leftarrow 1$

$\quad\quad\quad PBest_{iBindex} \leftarrow 0$

$\quad\quad$ **for** $j \in altB_{ABbestindex}$ **do**

$\quad\quad\quad PBest_{jAindex} \leftarrow 0$

$\quad\quad\quad PBest_{jBindex} \leftarrow 1$

$\quad heat \leftarrow heat - cooling\_rate$

return $PBest$

3.3.6 SynTeamPlus

SynTeamPlus is this project's attempt to make marginal improvements on SynTeam, at least in the context of dividing students for group projects. This is done by making two changes designed to accelerate the search done by SynTeam:

1. Where SynTeam starts off with students randomly assigned, SynTeamPlus starts off with an initial partition based on students being assigned cyclically to teams in order of their own proficiency.

2. Where SynTeam chooses two random teams in the current partition to take apart and test all possible reassignments, SynTeamPlus, with 80 per cent probability,

chooses the most and least proficient teams.

Both of these measures are designed to speed up the process. (1) means that the starting configuration for the search is already significantly better than random, while (2) means that each iteration of the algorithm looks for improvements where they are most likely to be found.

However, it is not clear that, given sufficient iterations, SynTeamPlus is likely to reach any better a solution. Therefore it will make sense to judge it on how quickly it reaches an equivalent solution, or how optimal the solution is given insufficient time.

Algorithm 2 SynTeamPlus

Require: $NTeams$ ▷ Number of teams

Require: $NStudents$ ▷ Number of students

Require: $heat$ ▷ Initial temperature

Require: $cooling\_rate$

Require: $NSmallTeams$ ▷ Number of smaller teams

Require: $NLargeTeams$ ▷ Number of larger teams

Require: $prof_i \forall i \in \mathbb{N}, i \leq NStudents$ ▷ Students' proficiency

Ensure: $PBest \in \mathbb{Z}_2^{NStudents \times NTeams}$ ▷ Current best partition

$MinTeamSize \leftarrow \lfloor \frac{NStudents}{NTeams} \rfloor$

$MaxTeamSize \leftarrow \lceil \frac{NStudents}{NTeams} \rceil$

$NSmallTeams \leftarrow NTeams \times MaxTeamSize - NStudents$

$NLargeTeams \leftarrow NTeams - NSmallTeams$

$ordered \leftarrow sort(\{prof_1, \ldots, prof_{NStudents}\})$

$Studentindex \leftarrow 0$

$Teamindex \leftarrow 0$

for $i \in \{0, \ldots, NsmallTeams - 1\}$ **do**

  for $j \in \{0, \ldots, MinTeamSize - 1\}$ **do**

    $PBest_{ordered_{Studentindex} Teamindex} \leftarrow 1$

    $Studentindex \leftarrow Studentindex + 1$

  $Teamindex \leftarrow Teamindex + 1$

```
for i ∈ {NSmallTeams, ..., NTeams − 1} do
    for j ∈ {0, ..., MaxTeamSize − 1} do
        PBest_{ordered_{Studentindex} Teamindex} ← 1
        Studentindex ← Studentindex + 1
    Teamindex ← Teamindex + 1
PBesteval ← 0
for i ∈ {0, ..., NTeams − 1} do
    teamprof ← 0
    for j ∈ {0, ..., NStudents} do
        teamprof_i ← teamprof_i + (PBest_{ji} × prof_j)
    PBesteval ← PBesteval + math.log(teamprof)
while (heat > 1) do
    if random.random() < 0.8 then
        A_index ← index(max(teamprof)
        B_index ← index(min(teamprof)
        if (A_index == B_index) then
            break
    else
        A_index ← random.randint(0, NTeams − 1)
        B_index ← random.randint(0, NTeams − 1)
        while (A_index == B_index) do
            B_index ← random.randint(0, NTeams − 1)
    teamset ← {}
    teama ← {}
    teamb ← {}
    for i ∈ {0, ..., NStudents} do
        if (PBest_{iA_index} == 1) then
            teama ← teama ∪ i
```

$teamset \leftarrow teamset \cup i$

if $(PBest_{iB_{index}} == 1)$ **then**

$teamb \leftarrow teamb \cup i$

$teamset \leftarrow teamset \cup i$

$teamprof_A \leftarrow 0$

$teamprof_B \leftarrow 0$

for $i \in teama$ **do**

$teamprof_A \leftarrow teamprof_A + prof_i$

for $i \in teamb$ **do**

$teamprof_B \leftarrow teamprof_B + prof_i$

$ABcurrenteval \leftarrow math.log(teamprof_A) + math.log(teamprof_B)$

if $(|teamset| == 2 \times MinTeamSize)$ **then**

$altA \leftarrow combinations(teamset, MinTeamSize)$

else

$altA \leftarrow combinations(teamset, MaxTeamSize)$

$altB \leftarrow$

for $i \in \{0, \dots, |altA| - 1\}$ **do**

$altB_i \leftarrow teamset \setminus altA_i$

$ABbesteval \leftarrow 1$

$ABbestindex \leftarrow 0$

for $i \in \{0, \dots, |altA| - 1\}$ **do**

$tempprof_A \leftarrow 0$

$tempprof_B \leftarrow 0$

for $j \in altA_i$ **do**

$teamprof_A \leftarrow teamprof_A + prof_j$

for $j \in altB_i$ **do**

$teamprof_B \leftarrow teamprof_B + prof_j$

$tempeval \leftarrow math.log(teamprof_A) + math.log(teamprof_B)$

if $((tempeval > ABbesteval) \wedge (altA_i \neq teama))$ **then**

$ABbesteval \leftarrow tempeval$

$ABbestindex \leftarrow i$

if $(ABbesteval > ABcurrenteval) \vee (max(\frac{heat \times (1-(ABcurrenteval - ABbesteval))}{20}, 0) \geq random.random()$ **then**

for $i \in altA_{ABbestindex}$ **do**

$PBest_{iAindex} \leftarrow 1$

$PBest_{iBindex} \leftarrow 0$

for $j \in altB_{ABbestindex}$ **do**

$PBest_{jAindex} \leftarrow 0$

$PBest_{jBindex} \leftarrow 1$

$heat \leftarrow heat - cooling\_rate$

return $PBest$

3.3.7 Subdivision

One notion of how you might make the problem easier for a given team composition optimisation solver, is to split it up into subproblems. Instead of dividing 200 students into 40 teams of five, you could split them into two groups of 100, and divide each of those groups into twenty teams of five. This will not speed up the process if the solver's method scales linearly with the number of students, as solving two problems half the size will not change the total time needed. However, the linear models presented above are good candidates for methods that will be sped up by such a division. The deviation models have a decision variable for each student-team pair, meaning in the larger example case 8000 (given as y_{ij} in the models). If you split the problem up, you have only 4000 such variables, meaning a significantly more manageable problem.

The trade-off is that the absolute optimal solution may no longer be available, but intuitively you might expect there to be many near-optimal solutions, some of which will be in the new search space. The reason this is likely to be the case is that the larger the number of students, the closer packed they are in terms of proficiency.

This project uses a splitting algorithm that randomly divides students into the new subproblems. The key is that the overall number of teams has to stay constant: if the initial problem is dividing 89 students into eleven teams of three and fourteen teams of four, then the subproblems (by necessity of unequal size) will have to be reconstructed from those teams.

Less naive ways to divide one large problem into smaller subproblems would be better. One would be to assign students cyclically to each subproblem while iterating through them in order of their individual proficiency. This could be a good avenue for further research.

Chapter 4

Implementation

4.1 Data simulation

It would be ideal to use real-life data to test the various methods of partition. However, given the time constraints on the project, especially considering the potential data protection issues arising from collecting students' academic results and personality assessments, it was decided to simulate various datasets for the experiment. This has the benefit that we can perform the experiment at scales implausible for real-world studies. For instance, Andrejczuk et al tested their algorithm SynTeam on two groups of twenty-four students[4]. In this project, datasets of up to two-hundred simulated students were used, and the datasets also varied in resolution—that is, each dataset was tested separately with all values rounded to the nearest percentile, and the nearest vigintile (or 5 per cent), to accurately reflect different kinds of data that might be available in varying contexts. This varying resolution has a practical relevance to optimisation problems, as it likely increases the symmetries of a given problem—in this case this more students will appear to have the same values, and even exactly equivalent students (those with the same values for all collected data) are more likely to appear. Moreover, it is not necessary that the data be exactly representative of the students, as this project makes no attempt whatsoever to make experimental breakthroughs about team performance per se, only about the methods by which teams are selected in given contexts. Nonetheless, every effort was made to produce data that might reflect a real computer science cohort at a

UK university, with the following assumptions:

1. 80 per cent of students are male.[1]

2. Academic performance (e.g. the marks from a previous course) and all the personality traits are normally distributed.

3. Academic performance has a mean of 65, and a standard deviation of 12.5.

4. Personality traits correlate to academic results with the following coefficients:[21]

   - Conscientiousness: 0.27
   - Openness: 0.16
   - Extraversion: 0.01
   - Agreeableness: 0.09
   - Neuroticism: 0.02

5. Personality traits correlate to gender with the following coefficients (when 0 is male, 1 female):[20]

   - Conscientiousness: 0.09
   - Openness: 0.01
   - Extraversion: 0.11
   - Agreeableness: 0.40
   - Neuroticism: 0.34

4.2 Experiments

Three experiments were designed. The first, and main one, was to compare a wide variety of models, given the problem initialised by each of the eight generated datasets (of 30, 50,

[1] Gender, along with some of the personality traits, is not used in the allocation methods of this project, but the data generated could be used by other methods that did take them into account. The use of such a high ratio should not be taken as an endorsement of the lack of women in STEM, but designed to reflect a real issue that course administrators may come across.

150 and 400 students, each with a resolution of 1 or 5 - i.e. marks and personality data rounded to the nearest integer or the nearest multiple of five), with desired team sizes of 3, 5 and 10. If a problem hadn't been solved within 900 seconds, that problem was timed out. Each problem was also run again after being subdivided into smaller problems, intended to be have around thirty students each. The timing mechanism was taken from Chaminda De Alwis.[10]

Practical considerations ruled out running all the models, so Dev1 and Multiplication1, the slowest ones, were withdrawn. The other linear models were given an internal solving time limit of one minute, separate to the overall 900 second limit which included time for the often slow process of model construction. This was after observing that they took an excessively long time to finish, after gaining most of the benefit in the initial seconds.

Experiment 2 pitted SynTeam against SynTeamPlus, where they each had only one minute to produce their best solution. The aim was to see whether the improvements suggested in 3.3.6 actually do give SynTeamPlus a head start in the optimisation race.

Experiment 3 used minor variations on SynTeamPlus. The thought was that, since the goal of the experiment is really to optimise for Eval2 rather than Eval1, wouldn't it be better to internally optimise in the same way, rather than the logarithmic evaluation inherited from SynTeam. As such SynTeamPlusMinus is exactly the same as SynTeamPlus, except replacing all Nash product evaluations with the absolute difference between the best and worst teams. SynTeam80/20 uses the Nash Product 80 percent of the time, but a random 20 per cent of the time uses the absolute difference evaluation.

All three experiments were run from a Python programme on a MacBook Air (2018). Each model and algorithm was also programmed in Python, with the linear models connecting to IBM's CPLEX optimisation software through the docplex module. The simulated data was generated in R.

The setup of each experiment was the same: one programme had a list of each dataset, each method of solving the problem, and a list of teamsizes. It iterated through all of these once without and once with subdividing the problem, and wrote the results to a csv file.

Chapter 5

Results

The concrete questions posed by these experiments might be summarised as follows:

- How do the different methods of team composition optimisation compare with regards to both speed and quality of solution? In particular, in this simulated context, does SynTeamPlus represent an improvement on SynTeam?
- How does the resolution of the data affect the speed and quality of the solution?
- Does dividing a team composition problem up into smaller problems improve speed, and is there a cost in terms of quality of solution?

This chapter will answer those questions, but for the sake of comprehensibility will only give the overview of the results, as well as some visualisation. The full results tables can be found in the Appendices.

5.1 Experiment 1

Experiment 1 was the main focus of this project. It tested four different methods (dev2, dev3, SynTeam and SynTeamPlus) against each of simulated dataset, running each problem both undivided, and divided into smaller subproblems (see Section 4.2 for details).

The first observation from Experiment 1 was that by Evaluation 1 (the Bernoulli-Nash

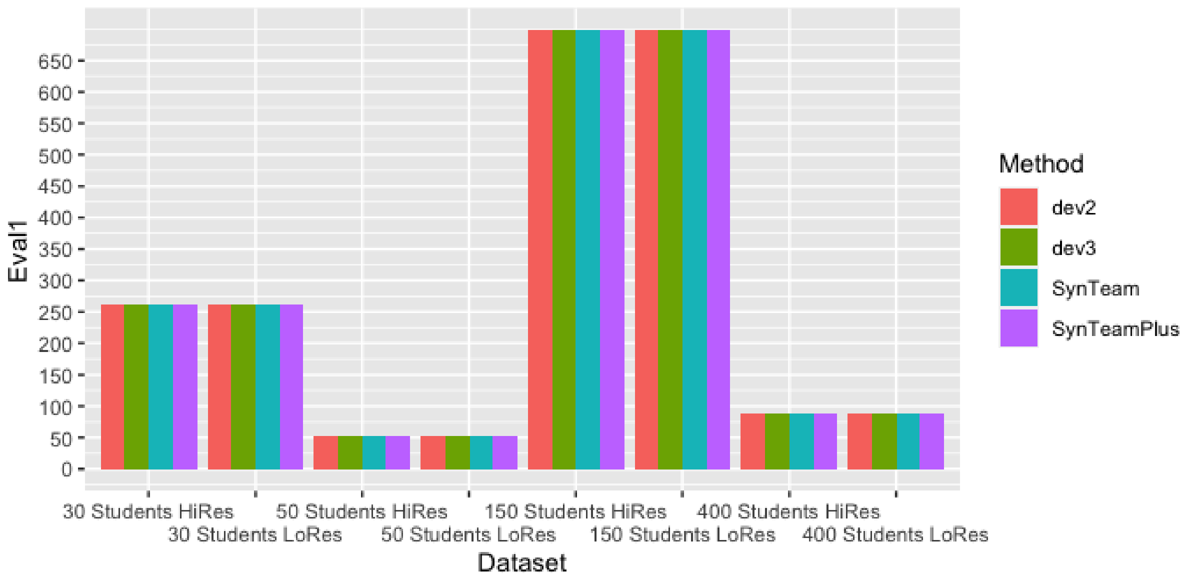

Figure 5.1: The Eval1 (Bernoulli-Nash evaluation) for every method and teamsize is very similar across each dataset.

product), for each given problem, all methods reached almost identically good solutions, as shown in Figure 5.1. This is a result of the fact that a logarithmic function will keep the same ordering of inputs, it will squeeze them together (at least within the relevant range. So while Eval1 may tell us which of two solutions is better, it is not very useful for telling us by how much.

This contrasts strongly with Figure 5.2, the equivalent graph, but for Eval2, in which, as expected, different methods give different a quality solutions.

There are two criteria by which each method must be judged: the quality of the solution it produces (which must, as has been seen, be assessed using Eval2), and the time it took to produce that solution.

The clearest answer to any question posed in this research concerns the latter. Figure 5.3 clearly shows that for teams of 3 and 5 students, both SynTeam and SynTeamPlus produced answers almost instantaneously, whereas for Dev2 and Dev3, the time taken

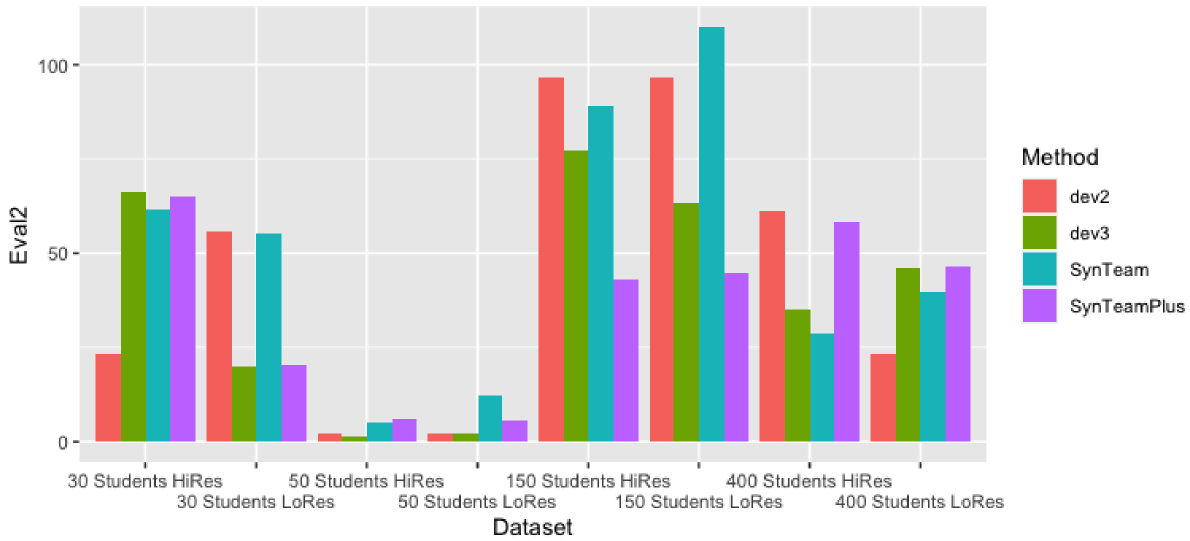

Figure 5.2: The average quality of solutions found by each method for each dataset.

increased linearly with the number of students when the problem was subdivided, and if it was not quickly exceeded a reasonable threshold. The linear programming models behaved the same way for the larger team, but SynTeam and SynTeamPlus suddenly struggled with the undivided problem. Note that with all these models, any attempted solve that lasted longer than 900s was timed out, and recorded as a nominal 1000s, without recording any solution.

Teamsize	UD2	UD3	UST	USTP	SD2	SD3	SST	SSTP
3	1000	1000	0.02	0.81	840.185	815.405	0.07	0.23
5	1000	1000	0.05	0.76	614.655	687.475	0.82	0.905
10	1000	1000	52.36	54.72	518.04	602.9	861.3	635.38

Table 5.1: Average number of seconds before solution for each method on a 400 student problem. UD2 is Undivided Deviation 2, SSTP is Split SynTeamPlus etc.

The reason that SynTeam and SynTeamPlus respond to increasing the size of the teams is in the algorithm design - as part of the iterated loop, two teams in the current best partition are selected, and every possible reassignment of their team members into two teams is tested. For teams of size n, this means listing and evaluating $\binom{2n}{n}$ different partitions each iteration. As a result of the combinatorial explosion mentioned earlier, both SynTeam and SynTeamPlus would be bad choices for picking much larger teams, even

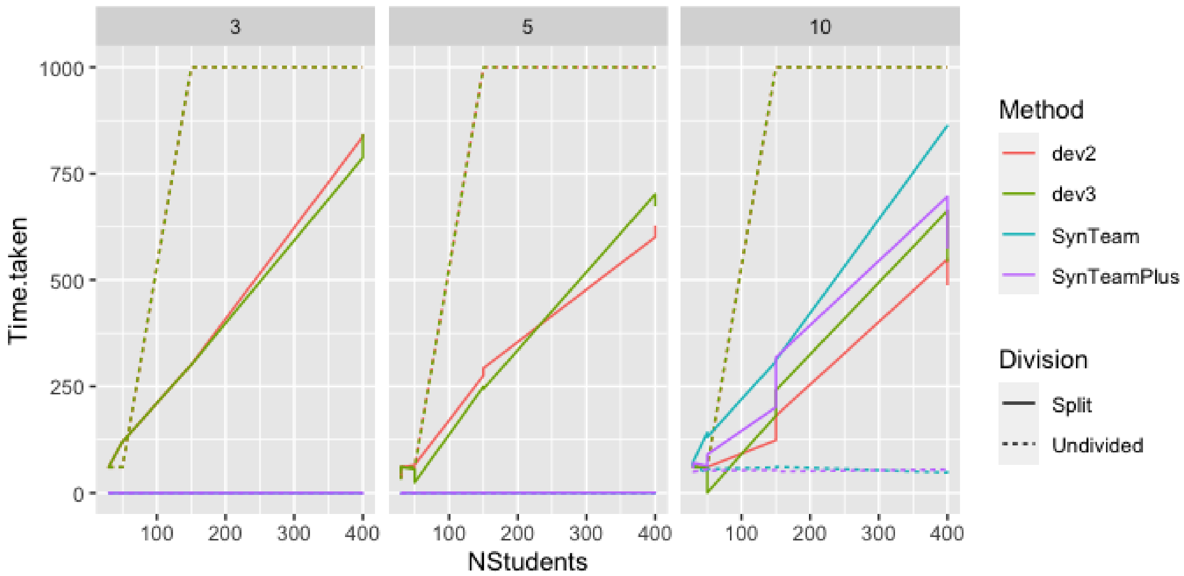

Figure 5.3: How long (in seconds) each problem took to produce a solution, by team sizes indicated above.

out of a smaller cohort.

However, that could all be made irrelevant if some of the models produce significantly lower quality answers. Figure 5.4 shows the equivalent graph but for the evaluation of the answer produced (note that for Eval2, a lower score is better, as it means that the best and worst teams in a given partition are of more equal proficiency).
An important caveat is that there was a time limit on the linear models - no doubt given sufficient time their answers would improve, but this is a practical question, and as noted in Section 4.2, they improve very slowly after the initial moments.

The message is not quite as clear with this one - the datasets were not evenly enough spaced to confirm fully any larger pattern, but tentatively it seems that SynTeamPlus performs well on the quality of submission compare to the other methods, especially with the team sizes of five and ten. For a team size of ten and an undivided problem, the worst solution SynTeamPlus came up with had an Eval2 of 1.64, a negligible amount in the context of team proficiencies in the hundreds.

These results also tentatively show the negative impact on SynTeamPlus on dividing

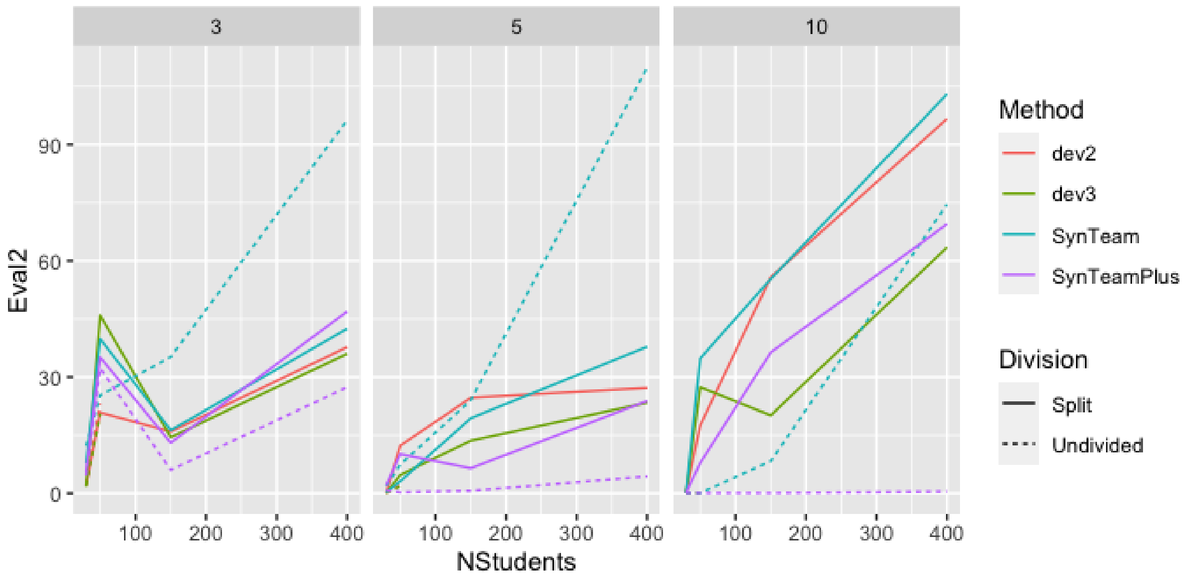

Figure 5.4: Quality of solution given by each method for the team sizes 3, 5 and 10, depending on the number of students.

the problem up, at least in the naive way done in these tests. However the fact that subdivision appears to improve the performance of SynTeam is curious. It is not obvious why these two similar algorithms should react in such opposed ways. For this reason more testing would be needed before putting too much stock in the performance of SynTeam-Plus here.

Figure 5.5 demonstrates the answer to the question as to how much the resolution of the data affect the performance of the models. In Section 4.1 it was proposed that increasing the symmetries of the problem might affect the performance of the linear programming models. This might still be the case, again in the long run, if you let those models run until they stop. But in the practical sense tested here, there is no effect at all: all the models perform almost identically on the high resolution data and low resolution data.

5.2 Experiment 2

Experiment 2 was designed to test SynTeam against SynTeamPlus. Each algorithm had 450 iterations, and was trying to optimise for teams of four, with 16 different datasets ranging between 25 and 400 students. The data shows that for this particular problem,

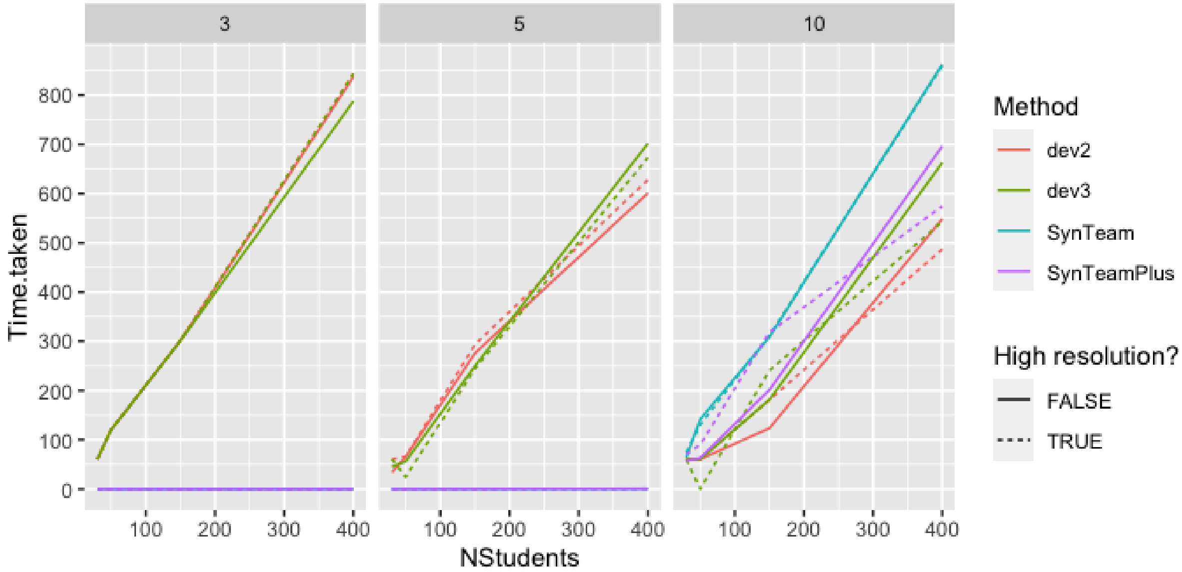

Figure 5.5: Data with high and low resolution were treated alike.

SynTeamPlus is a significant improvement on SynTeam, with the gap growing wider for larger problems, as SynTeamPlus maintains near-optimal performance (close to no difference between the most and least proficient teams) while SynTeam gets worse.

5.3 Experiment 3

Experiment 3 was designed to test different versions of SynTeamPlus - in particular whether it would be better served by internal evaluations of Nash product or absolute difference.

The results are fairly clear that SynTeamPlus is not improved by even the occasional use of Eval2. This is somewhat counter-intuitive, given that it is Eval2 we are ultimately using to assess performance. However, it is a strong endorsement of the initial idea behind SynTeam.

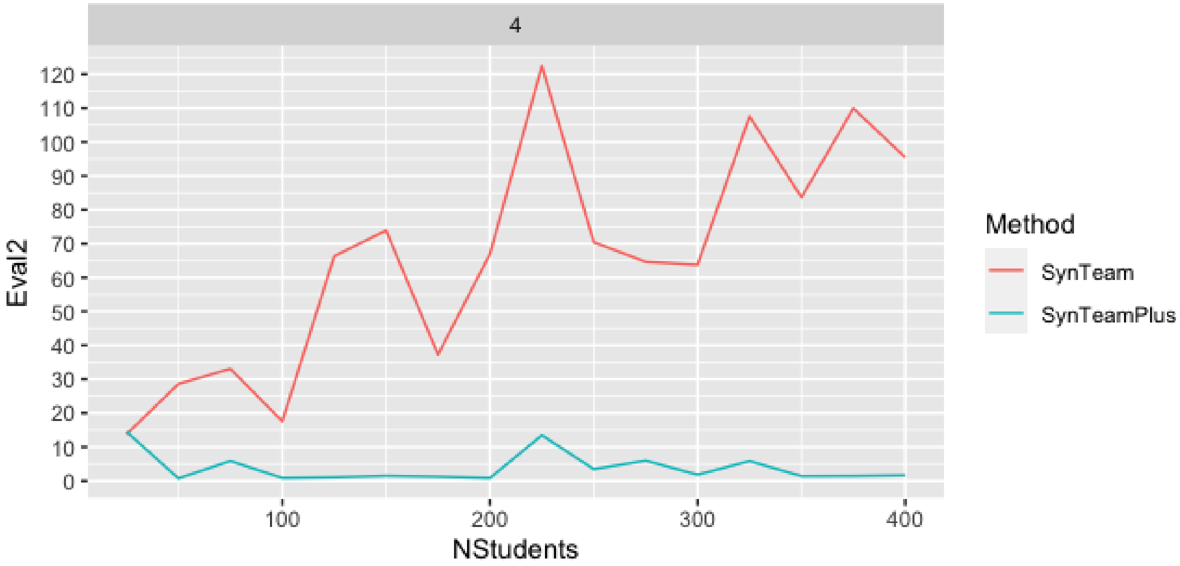

Figure 5.6: Performance of SynTeam vs SynTeamPlus for teams of four across different size datasets.

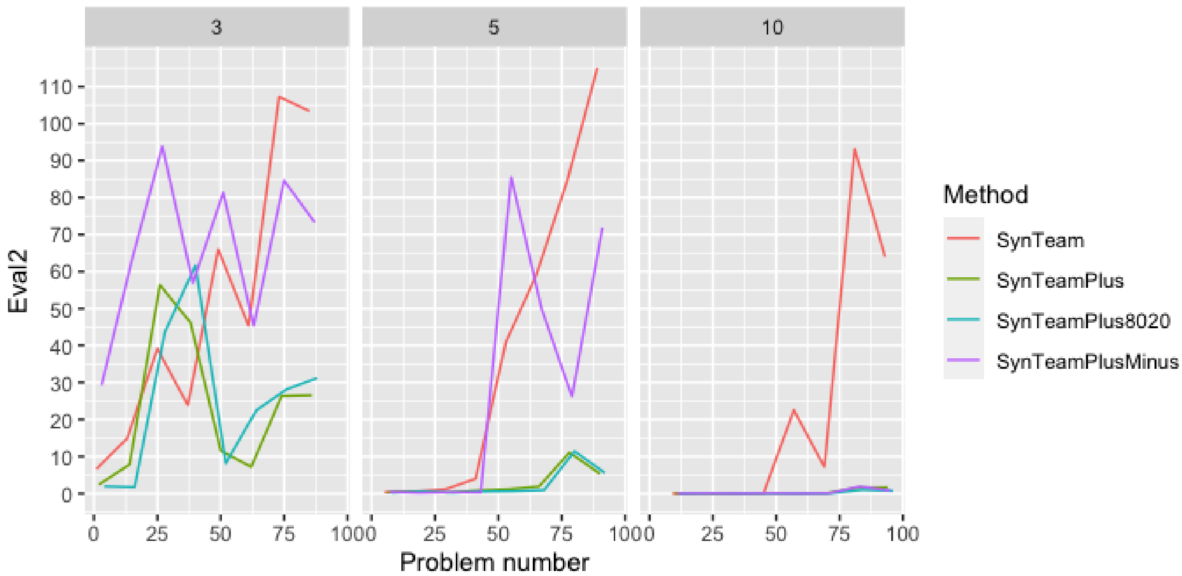

Figure 5.7: SynTeamPlus outperforms all its siblings.

Chapter 6

Summary and Reflections

Building on the work of Ewa Andrejczuk, this project has tested various methods of team composition optimisation on simulated data representing a student cohort. It has proposed some marginal improvements on Andrejczuk's algorithm SynTeam, and performed experiments that show that the new version, SynTeamPlus, may be more suitable in certain conditions. It has demonstrated that splitting a problem into subproblems is a valid way of using linear programming to solve team optimisation, but that it may be actively harmful to algorithms such as SynTeamPlus.

6.1 Potential issues

The most obvious issue with this thesis is the use of completely simulated data. While the author made an effort to make the data as accurate as possible, by making the personality data correlate internally and with the academic data in the way observed in the literature, proof of performance on a real cohort of students would be a much better benchmark.

However, the more important issue with this research is the underlying assumptions about the role that personality plays. As affirmed in the General Ethics section, if it plays a significant role, personality data should be used (with the consent of the students). However, it is not clear that it does play a significant role. First of all, the data in the relevant studies mentioned in the Literature Review is confusing. Peeters et al[25] and Lepine et al[19] disagree on whether there is evidence for variability in personality traits having a significant effect on team performance. Bell [7] notes that personality has almost no observed effect on team performance in a laboratory setting, although plenty has been observed 'in the field' - a fact that does not seem easily explicable. Secondly, there are differences between what an optimal team looks like among students and among workers.[25] There may be many reasons for this, such as different cultures, different social hierarchies - or even different sorts of rooms. Without a clear understanding of these dynamics, it may be unwise to put too much stock in the data. And finally and most importantly, many studies done on the effect of personality on team performance, involve a team of people spending much time face to face. On the other hand, most group projects in a university setting are likely to be months-long, and require only limited (if any) face-to-face contact between members of a given team. It is for these reasons that it may not only be extra effort to gather personality data, but it might also be actively harmful to use it with any weight whatsoever, as opposed to the academic data, for which it is clear that it should be used to level the playing field.

Overall, this research is flexible enough to be able to cope with any advances in the literature in coming years. New formulas for optimal team make-up can be inserted into many of the models without much change in computing resources required. And all of them can be used without any personality data at all, which would be the author's current recommendation.

6.2 Concrete Conclusions

- SynTeamPlus may in some contexts be an improvement on SynTeam, but more experimental data would be needed to confirm this.

- SynTeam and SynTeamPlus are very quick for small team sizes, and slower for larger ones, irrespective of the size of the whole cohort

- An improvement to SynTeam and SynTeamPlus would likely involve replacing the exploration of combination space within the iterated loop.

- Using linear programming may therefore be optimal in such cases, but if the cohort is too large it may be useful to subdivide into smaller problems.
- Within the SynTeam and SynTeamPlus algorithms, the use of the Nash-Bernoulli product as a metric improves performance over taking the maximum difference in proficiency between two teams.

6.3 Reflections

For me, the shame of the project is that I didn't get around to building the online tool to perform the optimisation for teachers, which I had thought would add a real bit of value into the project. I may well still do this, using what I have learnt. I think there are further improvements to SynTeamPlus that can be made, and possibly a completely new approach could be better—one thing that shocked me was how unpredictable the quality of solution could be. There's always room for more and interesting research.

Bibliography

[1] Alberola, J. M., Del Val, E., Sanchez-Anguix, V., Palomares, A., and Teruel, M. D. An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems 101*, 1–14.

[2] Andrejczuk, E., Berger, R., Rodriguez-Aguilar, J. A., Sierra, C., and Marín-Puchades, V. The composition and formation of effective teams: computer science meets organizational psychology. *The Knowledge Engineering Review 33*, e17.

[3] Andrejczuk, E., Bistaffa, F., Blum, C., Rodriguez-Aguilar, J. A., and Sierra, C. Heterogeneous teams for homogeneous performance. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems*, T. Miller, N. Oren, Y. Sakurai, I. Noda, B. T. R. Savarimuthu, and T. Cao Son, Eds., vol. 11224. Springer International Publishing, pp. 89–105. Series Title: Lecture Notes in Computer Science.

[4] Andrejczuk, E., Rodriguez-Aguilar, J. A., Roig, C., and Sierra, C. Synergistic team composition. Number: arXiv:1702.08222.

[5] Andrejczuk, E. D. Artificial intelligence methods to support people management in organisations. 187.

[6] Bainbridge, T. F., Ludeke, S. G., and Smillie, L. D. Evaluating the big five as an organizing framework for commonly used psychological trait scales. *Journal of Personality and Social Psychology 122*, 4, 749–777.

[7] Bell, S. T. Deep-level composition variables as predictors of team performance: A meta-analysis. *Journal of Applied Psychology 92*, 3, 595–615.

[8] Cooke, N. J., and Hilton, M. L. Committee on the science of team science. 281.

[9] Crawford, C., Rahaman, Z., and Sen, S. Evaluating the efficiency of robust team formation algorithms. In *Autonomous Agents and Multiagent Systems*, N. Osman and C. Sierra, Eds., vol. 10002. Springer International Publishing, pp. 14–29. Series Title: Lecture Notes in Computer Science.

[10] De Alwis, C. Timing out of long running methods in python.

[11] DeBruine, L. *faux: Simulation for Factorial Designs*, 2021. R package version 1.1.0.

[12] Dzvonyar, D., and Bruegge, B. Team composition and team factors in software engineering: An interview study of project-based organizations. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, pp. 561–570.

[13] Farhangian, M., Purvis, M., Purvis, M., and Savarimuthu, T. B. R. Agent-based modeling of resource allocation in software projects based on personality and skill. In *Advances in Social Computing and Multiagent Systems*, F. Koch, C. Guttmann, and D. Busquets, Eds., vol. 541. Springer International Publishing, pp. 130–146. Series Title: Communications in Computer and Information Science.

[14] Goldberg, L. R. The development of markers for the big-five factor structure. *Psychological Assessment 4*, 1, 26–42.

[15] Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, C. R., and Gough, H. G. The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality 40*, 1, 84–96.

[16] Graziano, W. G., Hair, E. C., and Finch, J. F. Competitiveness mediates the link between personality and group performance. 15.

[17] Kichuk, S. L., and Wiesner, W. H. The big five personality factors and team performance: implications for selecting successful product design teams. *Journal of Engineering and Technology Management 14*, 3, 195–221.

[18] LePine, J. A. Team adaptation and postchange performance: Effects of team composition in terms of members' cognitive ability and personality. *Journal of Applied Psychology 88*, 1, 27–39.

[19] LePine, J. A., Buckman, B. R., Crawford, E. R., and Methot, J. R. A review of research on personality in teams: Accounting for pathways spanning levels of theory and analysis. *Human Resource Management Review 21*, 4, 311–330.

[20] Lippa, R. A. Gender differences in personality and interests: When, where, and why?: Gender differences in personality and interests. *Social and Personality Psychology Compass 4*, 11, 1098–1110.

[21] Mammadov, S. Big five personality traits and academic performance: A meta-analysis. *Journal of Personality 90*, 2, 222–255.

[22] Nash, J. F. The bargaining problem. *Econometrica 18*, 2, 155–162.

[23] Okimoto, T., Schwind, N., and Clement, M. How to form a task-oriented robust team. *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.)*, 9.

[24] O'Connor, M. C., and Paunonen, S. V. Big five personality predictors of post-secondary academic performance. *Personality and Individual Differences 43*, 5, 971–990.

[25] Peeters, M. A. G., van Tuijl, H. F. J. M., Rutte, C. G., and Reymen, I. M. M. J. Personality and team performance: a meta-analysis. *European Journal of Personality 20*, 5, 377–396.

[26] Pittenger, D. J., and College, M. The utility of the myers-briggs type indicator. *Review of Educational Research 63*, 4, 467–488.

[27] Thalmayer, A. G., and Saucier, G. The questionnaire big six in 26 nations: Developing cross–culturally applicable big six, big five and big two inventories. *European Journal of Personality 28*, 5, 482–496.

[28] Tupes, E. C., and Christal, R. E. Recurrent personality factors based on trait ratings. *Journal of Personality 60*, 2, 225–251.

Appendix A

Note on Additional Materials

The files submitted as additional materials are as follows:

1. General purpose scripts:

    - Experiment1.py (NB Experiment 2 was run from this, just slightly tweaked.)
    - Experiment3.py
    - evaluation.py
    - Split.py

2. Method files (Models and Algorithms)

    - dev2.py
    - dev3.py
    - SynTeam.py
    - SynTeamPlus.py
    - SynTeamPlusMinus.py
    - SynTeam8020.py

3. Simulated datasets

    - simuln30r1.csv
    - simuln30r5.csv

- simuln50r1.csv
- simuln50r5.csv
- simuln150r1.csv
- simuln150r5.csv
- simuln400r1.csv
- simuln400r5.csv

4. R script for generating data

Appendix B

Full Results Tables

Table B.1: Experiment 1 Results

Method	Division	Dataset	Team size	Time taken (s)	Eval1	Eval2
"dev2"	"Undivided"	"simuln30r1.csv"	3	60.35	52.35	2.16
"dev2"	"Split"	"simuln30r1.csv"	3	60.07	52.35	1.36
"dev3"	"Undivided"	"simuln30r1.csv"	3	60.14	52.35	1
"dev3"	"Split"	"simuln30r1.csv"	3	60.08	52.35	1.33
"SynTeam"	"Undivided"	"simuln30r1.csv"	3	0	52.35	5.29
"SynTeam"	"Split"	"simuln30r1.csv"	3	0.01	52.35	3.22
"SynTeamPlus"	"Undivided"	"simuln50r1.csv"	3	0.04	88.36	49.02
"SynTeamPlus"	"Split"	"simuln50r1.csv"	3	0.05	88.36	44.82
"dev2"	"Undivided"	"simuln30r1.csv"	5	35.27	34.48	0.04
"dev2"	"Split"	"simuln30r1.csv"	5	34.12	34.48	0.02
"dev3"	"Undivided"	"simuln30r1.csv"	5	31.6	34.48	0.02
"dev3"	"Split"	"simuln30r1.csv"	5	45.47	34.48	0.02
"SynTeam"	"Undivided"	"simuln30r1.csv"	5	0.08	34.48	0.2
"SynTeam"	"Split"	"simuln30r1.csv"	5	0.08	34.48	0.4
"SynTeamPlus"	"Undivided"	"simuln30r1.csv"	5	0.07	34.48	0.31
"SynTeamPlus"	"Split"	"simuln30r1.csv"	5	0.11	34.48	0.98

”dev2”	”Undivided”	”simuln30r1.csv”	10	60.11	19.32	0.02
”dev2”	”Split”	”simuln30r1.csv”	10	60.12	19.32	0.02
”dev3”	”Undivided”	”simuln30r1.csv”	10	60.11	19.32	0.02
”dev3”	”Split”	”simuln30r1.csv”	10	60.14	19.32	0.02
”SynTeam”	”Undivided”	”simuln30r1.csv”	10	69.89	19.32	0.02
”SynTeam”	”Split”	”simuln30r1.csv”	10	63.56	19.32	0.02
”SynTeamPlus”	”Undivided”	”simuln30r1.csv”	10	50.86	19.32	0.02
”SynTeamPlus”	”Split”	”simuln30r1.csv”	10	57.36	19.32	0.02
”dev2”	”Undivided”	”simuln30r5.csv”	3	60.21	52.09	2.36
”dev2”	”Split”	”simuln30r5.csv”	3	60.1	52.09	2
”dev3”	”Undivided”	”simuln30r5.csv”	3	60.11	52.09	2
”dev3”	”Split”	”simuln30r5.csv”	3	60.14	52.09	1.73
”SynTeam”	”Undivided”	”simuln30r5.csv”	3	0	52.09	12.18
”SynTeam”	”Split”	”simuln30r5.csv”	3	0.01	52.09	7.73
”SynTeamPlus”	”Undivided”	”simuln30r5.csv”	3	0.01	52.09	4.55
”SynTeamPlus”	”Split”	”simuln30r5.csv”	3	0.02	52.09	5.55
”dev2”	”Undivided”	”simuln30r5.csv”	5	60.61	34.32	0.09
”dev2”	”Split”	”simuln30r5.csv”	5	60.13	34.32	0.09
”dev3”	”Undivided”	”simuln30r5.csv”	5	60.5	34.32	0.09
”dev3”	”Split”	”simuln30r5.csv”	5	60.64	34.32	0.09
”SynTeam”	”Undivided”	”simuln30r5.csv”	5	0.06	34.32	1.91
”SynTeam”	”Split”	”simuln30r5.csv”	5	0.08	34.32	0.27
”SynTeamPlus”	”Undivided”	”simuln30r5.csv”	5	0.06	34.32	0.73
”SynTeamPlus”	”Split”	”simuln30r5.csv”	5	0.1	34.32	2.09
”dev2”	”Undivided”	”simuln30r5.csv”	10	60.6	19.24	0.09
”dev2”	”Split”	”simuln30r5.csv”	10	60.2	19.24	0.09
”dev3”	”Undivided”	”simuln30r5.csv”	10	60.23	19.24	0.09
”dev3”	”Split”	”simuln30r5.csv”	10	60.15	19.24	0.09

"SynTeam"	"Undivided"	"simuln30r5.csv"	10	63.36	19.24	0.09
"SynTeam"	"Split"	"simuln30r5.csv"	10	73.63	19.24	0.09
"SynTeamPlus"	"Undivided"	"simuln30r5.csv"	10	50.07	19.24	0.09
"SynTeamPlus"	"Split"	"simuln30r5.csv"	10	70.22	19.24	0.09
"dev2"	"Undivided"	"simuln50r1.csv"	3	60.14	88.38	36.36
"dev2"	"Split"	"simuln50r1.csv"	3	120.2	88.39	24.6
"dev3"	"Undivided"	"simuln50r1.csv"	3	60.56	88.38	31.24
"dev3"	"Split"	"simuln50r1.csv"	3	120.35	88.37	34.93
"SynTeam"	"Undivided"	"simuln50r1.csv"	3	0	88.39	27.62
"SynTeam"	"Split"	"simuln50r1.csv"	3	0.01	88.39	28.69
"SynTeamPlus"	"Undivided"	"simuln50r1.csv"	3	0.04	88.36	49.02
"SynTeamPlus"	"Split"	"simuln50r1.csv"	3	0.05	88.36	44.82
"dev2"	"Undivided"	"simuln50r1.csv"	5	60.12	57.3	2.2
"dev2"	"Split"	"simuln50r1.csv"	5	65.26	57.3	9.31
"dev3"	"Undivided"	"simuln50r1.csv"	5	60.18	57.3	0.98
"dev3"	"Split"	"simuln50r1.csv"	5	55.74	57.3	23.24
"SynTeam"	"Undivided"	"simuln50r1.csv"	5	0.05	57.3	3.09
"SynTeam"	"Split"	"simuln50r1.csv"	5	0.13	57.3	11.64
"SynTeamPlus"	"Undivided"	"simuln50r1.csv"	5	0.1	57.3	0.22
"SynTeamPlus"	"Split"	"simuln50r1.csv"	5	0.19	57.3	3.18
"dev2"	"Undivided"	"simuln50r1.csv"	10	60.1	32.12	0.02
"dev2"	"Split"	"simuln50r1.csv"	10	60.32	32.11	60.98
"dev3"	"Undivided"	"simuln50r1.csv"	10	60.39	32.12	0.04
"dev3"	"Split"	"simuln50r1.csv"	10	60.25	32.12	14.98
"SynTeam"	"Undivided"	"simuln50r1.csv"	10	46.96	32.12	0.02
"SynTeam"	"Split"	"simuln50r1.csv"	10	141.88	32.12	14.62
"SynTeamPlus"	"Undivided"	"simuln50r1.csv"	10	55.32	32.12	0.02
"SynTeamPlus"	"Split"	"simuln50r1.csv"	10	63.02	32.11	58.67

”dev2”	”Undivided”	”simuln50r5.csv”	3	60.2	88.28	23.09
”dev2”	”Split”	”simuln50r5.csv”	3	120.23	88.28	20.82
”dev3”	”Undivided”	”simuln50r5.csv”	3	60.53	88.28	20.55
”dev3”	”Split”	”simuln50r5.csv”	3	120.17	88.25	45.91
”SynTeam”	”Undivided”	”simuln50r5.csv”	3	0	88.28	25.27
”SynTeam”	”Split”	”simuln50r5.csv”	3	0.01	88.26	39.82
”SynTeamPlus”	”Undivided”	”simuln50r5.csv”	3	0.01	88.27	32.27
”SynTeamPlus”	”Split”	”simuln50r5.csv”	3	0.02	88.27	35.09
”dev2”	”Undivided”	”simuln50r5.csv”	5	60.25	57.24	2.18
”dev2”	”Split”	”simuln50r5.csv”	5	65.72	57.24	12.27
”dev3”	”Undivided”	”simuln50r5.csv”	5	60.13	57.24	1.82
”dev3”	”Split”	”simuln50r5.csv”	5	24.65	57.24	4.73
”SynTeam”	”Undivided”	”simuln50r5.csv”	5	0.04	57.24	7.36
”SynTeam”	”Split”	”simuln50r5.csv”	5	0.1	57.24	3.09
”SynTeamPlus”	”Undivided”	”simuln50r5.csv”	5	0.04	57.24	0.27
”SynTeamPlus”	”Split”	”simuln50r5.csv”	5	0.12	57.24	10.09
”dev2”	”Undivided”	”simuln50r5.csv”	10	60.09	32.09	0.09
”dev2”	”Split”	”simuln50r5.csv”	10	60.32	32.09	17.55
”dev3”	”Undivided”	”simuln50r5.csv”	10	60.85	32.09	0.09
”dev3”	”Split”	”simuln50r5.csv”	10	0.22	32.08	27.36
”SynTeam”	”Undivided”	”simuln50r5.csv”	10	56.05	32.09	0.09
”SynTeam”	”Split”	”simuln50r5.csv”	10	130.8	32.08	34.82
”SynTeamPlus”	”Undivided”	”simuln50r5.csv”	10	52.52	32.09	0.09
”SynTeamPlus”	”Split”	”simuln50r5.csv”	10	89.85	32.09	7.82
”dev2”	”Undivided”	”simuln150r1.csv”	3	NA	NA	NA
”dev2”	”Split”	”simuln150r1.csv”	3	301.05	262.49	11.44
”dev3”	”Undivided”	”simuln150r1.csv”	3	NA	NA	NA
”dev3”	”Split”	”simuln150r1.csv”	3	300.86	262.46	21.4

"SynTeam"	"Undivided"	"simuln150r1.csv"	3	0.01	262.4	55.95
"SynTeam"	"Split"	"simuln150r1.csv"	3	0.03	262.48	13.62
"SynTeamPlus"	"Undivided"	"simuln150r1.csv"	3	0.07	262.49	9.87
"SynTeamPlus"	"Split"	"simuln150r1.csv"	3	0.07	262.48	21.53
"dev2"	"Undivided"	"simuln150r1.csv"	5	NA	NA	NA
"dev2"	"Split"	"simuln150r1.csv"	5	275.81	172.82	12.04
"dev3"	"Undivided"	"simuln150r1.csv"	5	NA	NA	NA
"dev3"	"Split"	"simuln150r1.csv"	5	249.68	172.82	9
"SynTeam"	"Undivided"	"simuln150r1.csv"	5	0.05	172.8	61.65
"SynTeam"	"Split"	"simuln150r1.csv"	5	0.27	172.82	11.58
"SynTeamPlus"	"Undivided"	"simuln150r1.csv"	5	0.19	172.82	1.16
"SynTeamPlus"	"Split"	"simuln150r1.csv"	5	0.44	172.81	22.89
"dev2"	"Undivided"	"simuln150r1.csv"	10	NA	NA	NA
"dev2"	"Split"	"simuln150r1.csv"	10	123.58	96.81	23.22
"dev3"	"Undivided"	"simuln150r1.csv"	10	NA	NA	NA
"dev3"	"Split"	"simuln150r1.csv"	10	181.34	96.8	66.04
"SynTeam"	"Undivided"	"simuln150r1.csv"	10	59.42	96.81	36.85
"SynTeam"	"Split"	"simuln150r1.csv"	10	308.56	96.81	20.04
"SynTeamPlus"	"Undivided"	"simuln150r1.csv"	10	53.51	96.81	0.02
"SynTeamPlus"	"Split"	"simuln150r1.csv"	10	201.25	96.8	69.11
"dev2"	"Undivided"	"simuln150r5.csv"	3	NA	NA	NA
"dev2"	"Split"	"simuln150r5.csv"	3	301.18	261.21	15.91
"dev3"	"Undivided"	"simuln150r5.csv"	3	NA	NA	NA
"dev3"	"Split"	"simuln150r5.csv"	3	300.85	261.22	14.45
"SynTeam"	"Undivided"	"simuln150r5.csv"	3	0.01	261.17	35.18
"SynTeam"	"Split"	"simuln150r5.csv"	3	0.07	261.22	16.27
"SynTeamPlus"	"Undivided"	"simuln150r5.csv"	3	0.08	261.23	6
"SynTeamPlus"	"Split"	"simuln150r5.csv"	3	0.05	261.23	13

"dev2"	"Undivided"	"simuln150r5.csv"	5	NA	NA	NA
"dev2"	"Split"	"simuln150r5.csv"	5	293.13	172.05	24.64
"dev3"	"Undivided"	"simuln150r5.csv"	5	NA	NA	NA
"dev3"	"Split"	"simuln150r5.csv"	5	244.16	172.06	13.55
"SynTeam"	"Undivided"	"simuln150r5.csv"	5	0.05	172.06	24
"SynTeam"	"Split"	"simuln150r5.csv"	5	0.35	172.05	19.36
"SynTeamPlus"	"Undivided"	"simuln150r5.csv"	5	0.09	172.06	0.64
"SynTeamPlus"	"Split"	"simuln150r5.csv"	5	0.28	172.06	6.45
"dev2"	"Undivided"	"simuln150r5.csv"	10	NA	NA	NA
"dev2"	"Split"	"simuln150r5.csv"	10	180.98	96.42	55.73
"dev3"	"Undivided"	"simuln150r5.csv"	10	NA	NA	NA
"dev3"	"Split"	"simuln150r5.csv"	10	240.84	96.43	20.09
"SynTeam"	"Undivided"	"simuln150r5.csv"	10	61.17	96.43	8.27
"SynTeam"	"Split"	"simuln150r5.csv"	10	311.87	96.42	55.27
"SynTeamPlus"	"Undivided"	"simuln150r5.csv"	10	50.81	96.43	0.09
"SynTeamPlus"	"Split"	"simuln150r5.csv"	10	318.15	96.43	36.36
"dev2"	"Undivided"	"simuln400r1.csv"	3	NA	NA	NA
"dev2"	"Split"	"simuln400r1.csv"	3	837.02	698.76	46.71
"dev3"	"Undivided"	"simuln400r1.csv"	3	NA	NA	NA
"dev3"	"Split"	"simuln400r1.csv"	3	787.54	698.77	33.09
"SynTeam"	"Undivided"	"simuln400r1.csv"	3	0.02	698.24	89.11
"SynTeam"	"Split"	"simuln400r1.csv"	3	0.07	698.74	52.89
"SynTeamPlus"	"Undivided"	"simuln400r1.csv"	3	0.69	698.74	28.16
"SynTeamPlus"	"Split"	"simuln400r1.csv"	3	0.2	698.73	51.84
"dev2"	"Undivided"	"simuln400r1.csv"	5	NA	NA	NA
"dev2"	"Split"	"simuln400r1.csv"	5	601.01	458.43	42.58
"dev3"	"Undivided"	"simuln400r1.csv"	5	NA	NA	NA
"dev3"	"Split"	"simuln400r1.csv"	5	701.88	458.41	42.24

”SynTeam”	”Undivided”	”simuln400r1.csv”	5	0.05	458.33	89.16
”SynTeam”	”Split”	”simuln400r1.csv”	5	0.8	458.42	39.84
”SynTeamPlus”	”Undivided”	”simuln400r1.csv”	5	0.84	458.48	8.05
”SynTeamPlus”	”Split”	”simuln400r1.csv”	5	0.91	458.42	56.27
”dev2”	”Undivided”	”simuln400r1.csv”	10	NA	NA	NA
”dev2”	”Split”	”simuln400r1.csv”	10	548.72	256.93	96.67
”dev3”	”Undivided”	”simuln400r1.csv”	10	NA	NA	NA
”dev3”	”Split”	”simuln400r1.csv”	10	662.84	256.95	77.31
”SynTeam”	”Undivided”	”simuln400r1.csv”	10	48.12	256.95	74.02
”SynTeam”	”Split”	”simuln400r1.csv”	10	862.87	256.95	69.4
”SynTeamPlus”	”Undivided”	”simuln400r1.csv”	10	54.99	256.97	1.64
”SynTeamPlus”	”Split”	”simuln400r1.csv”	10	696.39	256.95	60.98
”dev2”	”Undivided”	”simuln400r5.csv”	3	NA	NA	NA
”dev2”	”Split”	”simuln400r5.csv”	3	843.35	697.87	37.73
”dev3”	”Undivided”	”simuln400r5.csv”	3	NA	NA	NA
”dev3”	”Split”	”simuln400r5.csv”	3	843.27	697.87	36
”SynTeam”	”Undivided”	”simuln400r5.csv”	3	0.02	697.32	96.27
”SynTeam”	”Split”	”simuln400r5.csv”	3	0.07	697.87	42.45
”SynTeamPlus”	”Undivided”	”simuln400r5.csv”	3	0.93	697.87	27.36
”SynTeamPlus”	”Split”	”simuln400r5.csv”	3	0.26	697.85	46.91
”dev2”	”Undivided”	”simuln400r5.csv”	5	NA	NA	NA
”dev2”	”Split”	”simuln400r5.csv”	5	628.3	457.92	27.18
”dev3”	”Undivided”	”simuln400r5.csv”	5	NA	NA	NA
”dev3”	”Split”	”simuln400r5.csv”	5	673.07	457.93	23.36
”SynTeam”	”Undivided”	”simuln400r5.csv”	5	0.05	457.81	109.91
”SynTeam”	”Split”	”simuln400r5.csv”	5	0.84	457.9	37.82
”SynTeamPlus”	”Undivided”	”simuln400r5.csv”	5	0.68	457.95	4.36
”SynTeamPlus”	”Split”	”simuln400r5.csv”	5	0.9	457.93	23.82

"dev2"	"Undivided"	"simuln400r5.csv"	10	NA	NA	NA
"dev2"	"Split"	"simuln400r5.csv"	10	487.36	256.67	96.64
"dev3"	"Undivided"	"simuln400r5.csv"	10	NA	NA	NA
"dev3"	"Split"	"simuln400r5.csv"	10	542.96	256.68	63.45
"SynTeam"	"Undivided"	"simuln400r5.csv"	10	56.6	256.69	74.55
"SynTeam"	"Split"	"simuln400r5.csv"	10	859.73	256.67	103
"SynTeamPlus"	"Undivided"	"simuln400r5.csv"	10	54.45	256.7	0.45
"SynTeamPlus"	"Split"	"simuln400r5.csv"	10	574.37	256.68	69.45

Table B.2: Experiment 2 Results

Method	Division	Dataset	Team size	Time taken	Eval1	Eval2
SynTeamPlus	Undivided	simuln25r1.csv	4	0.35	37.66	14.51
SynTeam	Undivided	simuln25r1.csv	4	0.01	37.66	13.73
SynTeamPlus	Undivided	simuln50r1.csv	4	0.53	71.21	0.73
SynTeam	Undivided	simuln50r1.csv	4	0.01	71.21	28.6
SynTeamPlus	Undivided	simuln75r1.csv	4	0.68	104.61	5.87
SynTeam	Undivided	simuln75r1.csv	4	0.01	104.6	33.07
SynTeamPlus	Undivided	simuln100r1.csv	4	1.37	137.49	0.85
SynTeam	Undivided	simuln100r1.csv	4	0.03	137.49	17.56
SynTeamPlus	Undivided	simuln125r1.csv	4	1.92	175.98	1.05
SynTeam	Undivided	simuln125r1.csv	4	0.02	175.93	66.31
SynTeamPlus	Undivided	simuln150r1.csv	4	2.44	209.41	1.49
SynTeam	Undivided	simuln150r1.csv	4	0.02	209.33	73.91
SynTeamPlus	Undivided	simuln175r1.csv	4	2.62	242.54	1.24
SynTeam	Undivided	simuln175r1.csv	4	0.03	242.5	37.24
SynTeamPlus	Undivided	simuln200r1.csv	4	3.33	275.59	0.84

SynTeam	Undivided	simuln200r1.csv	4	0.04	275.52	66.96
SynTeamPlus	Undivided	simuln225r1.csv	4	4.6	312.73	13.47
SynTeam	Undivided	simuln225r1.csv	4	0.03	312.53	122.42
SynTeamPlus	Undivided	simuln250r1.csv	4	5.54	346.52	3.45
SynTeam	Undivided	simuln250r1.csv	4	0.02	346.42	70.44
SynTeamPlus	Undivided	simuln275r1.csv	4	6.22	380.54	5.98
SynTeam	Undivided	simuln275r1.csv	4	0.03	380.42	64.65
SynTeamPlus	Undivided	simuln300r1.csv	4	8.01	413.76	1.82
SynTeam	Undivided	simuln300r1.csv	4	0.04	413.66	63.78
SynTeamPlus	Undivided	simuln325r1.csv	4	9.13	452	5.85
SynTeam	Undivided	simuln325r1.csv	4	0.04	451.74	107.56
SynTeamPlus	Undivided	simuln350r1.csv	4	12.39	483.83	1.4
SynTeam	Undivided	simuln350r1.csv	4	0.04	483.63	83.67
SynTeamPlus	Undivided	simuln375r1.csv	4	19.45	518.59	1.45
SynTeam	Undivided	simuln375r1.csv	4	0.08	518.29	109.93
SynTeamPlus	Undivided	simuln400r1.csv	4	16.54	551.94	1.64
SynTeam	Undivided	simuln400r1.csv	4	0.05	551.72	95.47

Table B.3: Experiment 3 Results

Method	Division	Dataset	Team size	Time taken	Eval1	Eval2
SynTeam	Undivided	simuln30r1.csv	3	0.0	52.35	6.6
SynTeamPlus	Undivided	simuln30r1.csv	3	0.01	52.35	2.45
SynTeamPlusMinus	Undivided	simuln30r1.csv	3	0.01	52.34	29.25
SynTeamPlus8020	Undivided	simuln30r1.csv	3	0.01	52.35	1.96
SynTeam	Undivided	simuln30r1.csv	5	0.04	34.48	0.33

SynTeamPlus	Undivided	simuln30r1.csv	5	0.05	34.48	0.27
SynTeamPlusMinus	Undivided	simuln30r1.csv	5	0.04	34.48	0.4
SynTeamPlus8020	Undivided	simuln30r1.csv	5	0.05	34.48	0.2
SynTeam	Undivided	simuln30r1.csv	10	47.62	19.32	0.02
SynTeamPlus	Undivided	simuln30r1.csv	10	46.45	19.32	0.02
SynTeamPlusMinus	Undivided	simuln30r1.csv	10	45.11	19.32	0.02
SynTeamPlus8020	Undivided	simuln30r1.csv	10	46.45	19.32	0.02
SynTeam	Undivided	simuln30r5.csv	3	0.0	52.08	14.82
SynTeamPlus	Undivided	simuln30r5.csv	3	0.01	52.09	7.82
SynTeamPlusMinus	Undivided	simuln30r5.csv	3	0.01	52.04	63.36
SynTeamPlus8020	Undivided	simuln30r5.csv	3	0.01	52.09	1.73
SynTeam	Undivided	simuln30r5.csv	5	0.04	34.32	0.55
SynTeamPlus	Undivided	simuln30r5.csv	5	0.04	34.32	0.55
SynTeamPlusMinus	Undivided	simuln30r5.csv	5	0.04	34.32	0.18
SynTeamPlus8020	Undivided	simuln30r5.csv	5	0.04	34.32	0.55
SynTeam	Undivided	simuln30r5.csv	10	48.94	19.24	0.09
SynTeamPlus	Undivided	simuln30r5.csv	10	55.68	19.24	0.09
SynTeamPlusMinus	Undivided	simuln30r5.csv	10	50.06	19.24	0.09
SynTeamPlus8020	Undivided	simuln30r5.csv	10	50.94	19.24	0.09
SynTeam	Undivided	simuln50r1.csv	3	0.0	88.38	39.13
SynTeamPlus	Undivided	simuln50r1.csv	3	0.01	88.36	56.38
SynTeamPlusMinus	Undivided	simuln50r1.csv	3	0.01	88.29	93.84
SynTeamPlus8020	Undivided	simuln50r1.csv	3	0.02	88.36	43.65
SynTeam	Undivided	simuln50r1.csv	5	0.03	57.3	1.13
SynTeamPlus	Undivided	simuln50r1.csv	5	0.04	57.3	0.27
SynTeamPlusMinus	Undivided	simuln50r1.csv	5	0.03	57.3	0.35
SynTeamPlus8020	Undivided	simuln50r1.csv	5	0.05	57.3	0.2
SynTeam	Undivided	simuln50r1.csv	10	51.34	32.12	0.02

SynTeamPlus	Undivided	simuln50r1.csv	10	50.44	32.12	0.02
SynTeamPlusMinus	Undivided	simuln50r1.csv	10	43.72	32.12	0.02
SynTeamPlus8020	Undivided	simuln50r1.csv	10	45.61	32.12	0.02
SynTeam	Undivided	simuln50r5.csv	3	0.0	88.28	23.91
SynTeamPlus	Undivided	simuln50r5.csv	3	0.01	88.26	46.27
SynTeamPlusMinus	Undivided	simuln50r5.csv	3	0.01	88.23	56.91
SynTeamPlus8020	Undivided	simuln50r5.csv	3	0.01	88.25	61.64
SynTeam	Undivided	simuln50r5.csv	5	0.04	57.24	4.0
SynTeamPlus	Undivided	simuln50r5.csv	5	0.04	57.24	0.82
SynTeamPlusMinus	Undivided	simuln50r5.csv	5	0.03	57.24	0.27
SynTeamPlus8020	Undivided	simuln50r5.csv	5	0.04	57.24	0.55
SynTeam	Undivided	simuln50r5.csv	10	48.27	32.09	0.09
SynTeamPlus	Undivided	simuln50r5.csv	10	46.52	32.09	0.09
SynTeamPlusMinus	Undivided	simuln50r5.csv	10	43.54	32.09	0.09
SynTeamPlus8020	Undivided	simuln50r5.csv	10	45.89	32.09	0.09
SynTeam	Undivided	simuln150r1.csv	3	0.01	262.38	66.02
SynTeamPlus	Undivided	simuln150r1.csv	3	0.08	262.49	11.55
SynTeamPlusMinus	Undivided	simuln150r1.csv	3	0.07	262.28	81.36
SynTeamPlus8020	Undivided	simuln150r1.csv	3	0.07	262.49	8.11
SynTeam	Undivided	simuln150r1.csv	5	0.03	172.81	41.07
SynTeamPlus	Undivided	simuln150r1.csv	5	0.09	172.82	1.15
SynTeamPlusMinus	Undivided	simuln150r1.csv	5	0.08	172.79	85.51
SynTeamPlus8020	Undivided	simuln150r1.csv	5	0.09	172.82	0.64
SynTeam	Undivided	simuln150r1.csv	10	45.61	96.81	22.55
SynTeamPlus	Undivided	simuln150r1.csv	10	53.51	96.81	0.02
SynTeamPlusMinus	Undivided	simuln150r1.csv	10	45.88	96.81	0.02
SynTeamPlus8020	Undivided	simuln150r1.csv	10	47.32	96.81	0.02
SynTeam	Undivided	simuln150r5.csv	3	0.0	261.16	45.36

SynTeamPlus	Undivided	simuln150r5.csv	3	0.08	261.23	7.18
SynTeamPlusMinus	Undivided	simuln150r5.csv	3	0.08	261.1	45.36
SynTeamPlus8020	Undivided	simuln150r5.csv	3	0.08	261.23	22.45
SynTeam	Undivided	simuln150r5.csv	5	0.04	172.05	59.18
SynTeamPlus	Undivided	simuln150r5.csv	5	0.09	172.06	1.91
SynTeamPlusMinus	Undivided	simuln150r5.csv	5	0.08	172.06	50.0
SynTeamPlus8020	Undivided	simuln150r5.csv	5	0.09	172.06	0.91
SynTeam	Undivided	simuln150r5.csv	10	46.6	96.43	7.18
SynTeamPlus	Undivided	simuln150r5.csv	10	46.07	96.43	0.09
SynTeamPlusMinus	Undivided	simuln150r5.csv	10	43.78	96.43	0.09
SynTeamPlus8020	Undivided	simuln150r5.csv	10	45.78	96.43	0.09
SynTeam	Undivided	simuln400r1.csv	3	0.02	698.14	107.24
SynTeamPlus	Undivided	simuln400r1.csv	3	0.63	698.76	26.35
SynTeamPlusMinus	Undivided	simuln400r1.csv	3	0.96	698.26	84.62
SynTeamPlus8020	Undivided	simuln400r1.csv	3	0.96	698.75	28.16
SynTeam	Undivided	simuln400r1.csv	5	0.1	458.31	84.2
SynTeamPlus	Undivided	simuln400r1.csv	5	0.54	458.48	11.0
SynTeamPlusMinus	Undivided	simuln400r1.csv	5	0.56	458.47	26.24
SynTeamPlus8020	Undivided	simuln400r1.csv	5	0.49	458.48	11.36
SynTeam	Undivided	simuln400r1.csv	10	51.51	256.95	93.07
SynTeamPlus	Undivided	simuln400r1.csv	10	52.41	256.97	1.58
SynTeamPlusMinus	Undivided	simuln400r1.csv	10	48.07	256.97	1.91
SynTeamPlus8020	Undivided	simuln400r1.csv	10	51.26	256.97	1.0
SynTeam	Undivided	simuln400r5.csv	3	0.02	697.38	103.36
SynTeamPlus	Undivided	simuln400r5.csv	3	0.62	697.87	26.55
SynTeamPlusMinus	Undivided	simuln400r5.csv	3	0.73	697.37	73.36
SynTeamPlus8020	Undivided	simuln400r5.csv	3	0.81	697.84	31.18
SynTeam	Undivided	simuln400r5.csv	5	0.08	457.8	115.09

SynTeamPlus	Undivided	simuln400r5.csv	5	0.47	457.95	5.36
SynTeamPlusMinus	Undivided	simuln400r5.csv	5	0.43	457.88	71.82
SynTeamPlus8020	Undivided	simuln400r5.csv	5	0.41	457.95	5.55
SynTeam	Undivided	simuln400r5.csv	10	54.75	256.69	64.0
SynTeamPlus	Undivided	simuln400r5.csv	10	50.86	256.7	1.64
SynTeamPlusMinus	Undivided	simuln400r5.csv	10	51.01	256.7	0.82
SynTeamPlus8020	Undivided	simuln400r5.csv	10	56.44	256.7	0.73