Improved Heuristics for Finding Balanced Teams

Daniel Solow, Jie Ning, Jieying Zhu, and Yishen Cai

Department of Operations
Case Western Reserve University
Cleveland, OH 44106
e-mail: daniel.solow@case.edu
phone: (216) 368-3837
fax: (216) 368–6250

February 12, 2020

Abstract

This research addresses the problem of dividing a group of people into a collection of teams that need to be "balanced" across a variety of different attributes. This type of problem arises, for example, in an academic setting where it is necessary to partition students into a number of balanced study teams and also in a youth camp in which children need to be formed into sports teams that are competitive with each other. Recent work has resulted in both linear and nonlinear integer programming models for solving this problem. In the research here, improvements to the models are made together with a linear approximation to the nonlinear objective function that significantly reduce the number of integer variables and constraints. Computational experiments are performed on random instances of the problem as well as on instances for which there are almost perfectly balanced teams, the latter providing a way to determine the quality of the optimal solution obtained by the heuristics. These tests show that the approach developed here almost always obtain better balanced teams than those from prior research.

1 Introduction and Problem Statement

The problem addressed in this work is to divide a group of people—hereafter referred to as the **population**—into some number of teams that need to be as balanced as possible across a variety of different attributes. One example arises when organizing students in a class into study groups that are balanced according to a variety of criteria—such as their mathematical abilities and their communication skills—so that no group has an unfair advantage. A second example is a youth camp in which the children need to be formed into a number of sports teams that are balanced according to their ability in the sport, their age, and their gender so that the teams are all competitive with each other. Indeed, the NFL draft is designed, to some degree, to allow weaker teams to obtain new players that will make their teams more competitive.

Given a division of the population into teams, what is needed is a measure of the degree to which the collection of teams is balanced, or equivalently, unbalanced. In this paper, this measure is a single nonnegative number—hereafter referred to as the **measure of imbalance**—with the property that the closer the number is to 0, the more balanced the collection of teams is. In other words, the **Team Balancing Problem (TBP)** is to divide a group of people into a collection of teams so that the resulting measure of imbalance, however defined, is as close to 0 as possible.

The specific way in which imbalance is measured depends, in part, on the different types of attributes, as described in Rubin and Bai (2015) and on which much of the subsequent explanation is based. Following their discussion of the TBP, three types of attributes arise, each of which is described in more detail now together with other criteria for determining balanced teams.

A **quantitative attribute** is a characteristic measured on a numeric scale—such as a score on the GMAT exam—that reflects each individual's proficiency with respect to that characteristic. For such a quantitative attribute, the goal, when creating balanced teams, is for the numerical values of the people on each team to resemble those of the whole population as closely as possible. For instance, in a perfectly balanced team, the average value of a quantitative attribute over all team members would be exactly equal to the average value of that attribute in the whole population. Thus, a numerical measure of imbalance for a quantitative attribute in a team is determined by how far the team average is from the population average. The specific distance measure used here is the absolute value of the difference of the two foregoing averages.

A **qualitative attribute** is a characteristic of a person for which there are a finite number of possible values that are not numeric in nature. One example of a qualitative attribute is a person's gender, whose possible values are *male* and *female*. As with a quantitative attribute, when creating a balanced team, the goal is for each value of the qualitative attribute of the team members to resemble that of the population as closely as possible. This is typically measured by comparing the *fraction* of members in a team having a particular value for that attribute to the fraction of the population having that same value of the attribute. Thus, for example, in a population of 40% men and 60% women, a perfectly balanced team would also consist of 40% men and 60% women. A measure of how imbalanced a team is with respect to one particular value of a qualitative attribute is determined by how far the fraction of team members having that value of the attribute is from the fraction of the population having that same value. The specific distance measure used here is the absolute value of the difference of the two foregoing fractions. The measures of imbalance for the various possible values of a given qualitative attribute must be combined into a single number, which is done here by summing the absolute values of the differences over all possible values of that attribute.

An **affinity attribute** is similar to a qualitative attribute in that each affinity attribute has a finite number of possible values. However, where the goal with a qualitative attribute in forming balanced teams is to have the team resemble that of the population as a whole, the goal with regard to an affinity attribute is to have each member in the team have exactly the same value for that attribute. For example, in forming teams in an organization, it might be desirable for each team member to live in the same geographical area (city, state, or country) so that they can meet conveniently in person. Another example of an affinity attribute is language, in which the goal is for everyone on a team to be able to speak the same language. In other words, with an affinity attribute, the goal is to create homogeneity within a team by ensuring that each member has the same value for that attribute. There are various different ways to measure imbalance for an affinity attribute (see Rubin and Bai (2015)); however, the measure adopted here is a value of 0, if everyone on that team has the same value for that attribute and some fixed positive number, otherwise.

Combining the Attribute Dissimilarities. With the foregoing methods for measuring how imbalanced a team is with respect to quantitative, qualitative, and affinity attributes, it is still necessary to combine those measures into a single value. One straightforward approach—the one used in the model proposed here—is to take a weighted average, with the decision maker determining appropriate values for the weights. The resulting combined value is referred to hereafter as the **total attribute dissimilarity** of the teams.

Allowing for Differing Team Sizes. An additional imbalance arises in the TBP in the event that the teams have different **team sizes**, that is, different numbers of members. This can happen because it may not be possible to create teams of exactly the same size. For example, with a population of 62 people it is not possible to create 5 teams of equal size. Even if it is possible to create teams of equal size, it is desirable to have the flexibility of not having to do so. One way to allow for differing team sizes is for the decision maker to specify a minimum, maximum, and ideal number of people for each team, say S_{min}, S_{max}, S_{opt}, together with a penalty associated with $|S - S_{opt}|$, where S is the actual size of a team. Additionally, the model allows people not to be assigned to any team, in which case a user-defined penalty per unassigned person is multiplied by the number of unassigned people.

Allowing for Differing Numbers of Teams. In specifying the TBP, it may be the case that the total number of teams itself is not predetermined. To allow for this possibility, in the model proposed here, the decision maker specifies a minimum, maximum, and ideal number of teams, together with a penalty for each team above or below the ideal number.

Additional Considerations. It is also possible to include *together* and *apart* constraints that arise when two or more people must be assigned to the same, or different, teams, respectively. Another feature considered in Rubin and Bai (2015) is a *cluster*, which is a group of individuals each of who has exactly the same values for their numerical, qualitative, and affinity attributes—a situation that is unlikely to occur if there are quantitative attributes measured on a continuous scale. If there are such groups of individuals, then using clusters reduces the size of the model. In the model developed here, however, apart/together constraints and clusters are not explicitly included so as to focus attention on the importance of the other more central factors in forming balanced teams; however, it is possible to modify the models and heuristics to include these items.

A Total Measure of Imbalance. In summary, given all of the foregoing data, a single number serves as the objective function used to measure the degree to which a particular partitioning of the population into teams is unbalanced. The specific objective function used in the model here is the sum of the total attribute dissimilarities in the teams, the team-size penalties, the number-of-teams penalty, and the unassigned-people penalties. The only "hard" constraints that absolutely must be satisfied are that no person be assigned to more than one team and that the team sizes and number of teams be within the specified minimums and maximums.

It is important to note that this problem, as stated, requires the decision maker to assign appropriate weights to each attribute and penalties for differing team sizes, number of teams, and unassigned people, all in relation to each other while also taking into account the different units involved. Following the approach suggested in Rubin and Bai (2015), it is assumed that the user will choose initial values for these parameters and then, based on the resulting teams, readjust the parameters, sequentially solving the problem and adjusting the parameters until achieving a satisfactory set of balanced teams. The heuristics must therefore provide results in a reasonable amount of time, which is chosen here, as in Rubin and Bai (2015), to be 10 minutes. The reason for this choice is to allow the user to solve the problem numerous times, each time varying the penalties, until a satisfactory set of teams is found.

Rubin and Bai (2015) created a nonlinear integer programming model (NLIP) to determine an optimal collection of teams that minimizes the objective function value. To obtain a good solution within 10 minutes, they devised the following three heuristics[1]:

- An equivalent linearized version of the NLIP to which CPLEX is then applied by fixing the value of T sequentially to $T = T_{min}, \ldots, T_{max}$. This is referred to subsequently as the *original linear model* and the *stepwise heuristic*.
- A genetic algorithm for the NLIP.
- An improvement heuristic—referred to as R3—that starts with a random set of teams and, by moving and swapping people amongst teams, attempt to create better balanced teams.

One contribution of the research here is an improvement to the models presented in Rubin and Bai (2015). For example, a version of the NLIP that uses fewer integer variables and constraints is developed (which also results in fewer integer variables and constraints in the linearized version). The main contribution, however, is the replacement of the nonlinear objective function with a linear approximation. The computational experiments in Section 4 show that the teams obtained from using this linear approximation are almost always more balanced than those obtained from the original linear model, with the amount of improvement increasing as the size of the population increases.

Furthermore, Rubin and Bai (2015) were unable to provide any insight as to the quality of the final objective function value obtained by their heuristics. Another contribution

[1] Computational experiments performed with an existing NLIP solver SCIP (https://scip.zib.de/) were found to provide results comparable to the heuristics only when the population size is 40 or less. From this it was concluded that commercial software packages that solve general NLIPs do not perform well on the TBP unless the problem instance is small.

made here is to provide a reasonable estimate for such heuristics. This is accomplished by generating data for which there are almost perfectly balanced teams that provide an effective upper bound on the optimal objective function value. It is then shown that the approach developed here obtains objective function values that are closer to these upper bounds than those obtained by the stepwise heuristic.

The remainder of this paper is organized as follows. A review of previous work on the TBP is presented in Section 2. Models and associated heuristics are developed in Section 3. In Section 4, computational results of using the heuristics are presented and a summary is given in Section 5.

2 Previous Work on the Team Balancing Problem

When it comes to team formation, various subproblems arise when the goal of forming the teams is different. Consequently, researchers have developed heuristics and optimization models to serve different goals. For example, Awal and Bharadwaj (2014) apply the concept of collective intelligence (CI) to maximize the total potential of the team with a set of experts discovered from an expertise social network. Fan et al. (2009) propose a model using knowledge rules and a genetic algorithm for grouping research projects in the sponsorship process. Srba and Bielikova (2015) use a dynamic approach that considers feedback on students' collaboration to improve group formations in the next iterations. Magnanti and Natarajan (2018) develop a discrete optimization model when allocating multidisciplinary capstone projects to students with the requirement that each project involves students from at least two disciplines.

Among all goals in forming teams, a critical one is to create balanced teams so that the teams are similar to each other and are therefore fair. For example, Benincasa et al. (2017) study the golf-director problem and formulate a linear model that incorporates handicaps and the variability of player scores with probability distributions. Another common problem is to form balanced teams of students. Contreras and Salcedo (2017) apply genetic algorithms structuring collaborative groups that balance similarities and differences.

Our work is based on that of Rubin and Bai (2015), where they propose two general mixed-integer programming models—one nonlinear and one linear—that take into account quantitative, qualitative, and affinity attributes as well as the number of teams, the team sizes, and the number of people who do not get assigned to teams. They also include apart/together constraints and address the concept of cluster and symmetry when formulating the model. Several heuristics are implemented, namely, a genetic algorithm, the conventional solver for MILP, and the improvement heuristics. Our model improves upon their paper by eliminating some redundant variables and devising a linear approximation to the original model. These improvements yield better results when the population is large.

There are two other papers that also use (simpler) mathematical models to form balanced teams. Cutshall et al. (2007) use integer programming to form student teams that achieve equity in team members' academic performance, functional diversity, and demographic balance. They define the objective function to minimize the maximum deviation of a team's academic performance. To reduce the complexity of the formulation, they assume that the number of students in class is a multiple of six; if not, they manually adjust the data. Another paper by Drnevich and Norris (2007) assigns civil engineering students to capstone course teams. They formulate an optimization model with the objective function being the sum of the squared deviations weighted by penalties across all teams, which is essentially the quantitative attributes in our model. Although the problem and the method are quite similar, our model can be used for more general and complex problems with more attributes and other constraints.

An important issue in forming balanced teams is the measure of (im)balance for a given team configuration. There is an extensive literature that deals with different measures of equity in various scenarios (Savas 1978, Marsh and Schilling 1994). Our paper uses a straightforward measure of imbalance: the deviation within each team from the population, and focuses on developing efficient heuristics that form balanced teams under this measure.

3 Models and Heuristics for the TBP

To provide a conceptual framework for the team-balancing problem described in Section 1, a network model is first developed. To that end, let:

$$\begin{aligned} N &= \text{the number of people in the population.} \\ T_{min},\, T_{opt},\, T_{max} &= \text{the minimum, ideal, and maximum number of teams.} \\ S_{min},\, S_{opt},\, S_{max} &= \text{the minimum, ideal, and maximum size of a team.} \end{aligned}$$

The N people in the population are represented by the set V of vertices in the graph and each pair of vertices is connected with an edge to represent the possibility of the corresponding two people being assigned to the same team. For a subset V' of V, let $|V'|$ be the number of vertices in V'. A collection of T teams for the TBP corresponds to a partition of the vertices into $T+1$ sets (cliques), $V_1, \ldots, V_T, V_{T+1}$, in which the sets $V_1, \ldots, V_T$ represent the T teams—and are referred to as such—and the set V_{T+1} represents the unassigned people—and is called the **unassigned team**. For such a partition, let $f(V_1, \ldots, V_T, V_{T+1})$ denote the objective function value representing the total measure of imbalance corresponding to the teams created by the partition. Noting that the number of teams T is itself a variable, the TBP, as defined in Section 1, is to find the number of teams, T, and the partition $V_1, \ldots, V_T, V_{T+1}$ of the N vertices into teams so as to minimize the total imbalance, f, that is:

$$\min_{T = T_{min}, \ldots, T_{max}} \quad \begin{array}{ll} \min & f(V_1, \ldots, V_T, V_{T+1}) \\ \text{s.t.} & V_1, \ldots, V_T, V_{T+1} \text{ is a partition of } V \text{ with} \\ & S_{min} \le |V_t| \le S_{max}, \quad (t = 1, \ldots, T) \end{array} \tag{1}$$

A nonlinear optimization model corresponding to (1) was created by Rubin and Bai (2015) in which they introduced, among others, the following integer variables to determine which team each person is assigned to:

$$x_{it} = \begin{cases} 1, & \text{if person } i \text{ is assigned to team } t \\ 0, & \text{otherwise} \end{cases} \qquad i = 1, \ldots, N; t = 1, \ldots, T+1.$$

3.1 Reducing the Number of Variables and Constraints

Two ways are now proposed for reducing the number of integer variables and constraints in the nonlinear model in Rubin and Bai (2015).

3.1.1 Reducing the Number of Variables and Constraints Associated with the Affinity Attributes

The first improvement reduces the number of integer variables and constraints associated with each affinity attribute a whose possible values are in the set $A(a)$. For each team t, Rubin and Bai (2015) use $0-1$ integer variables, v_{at}, whose value is 0, if every person on team t has the same value for this attribute and 1, otherwise, in which case, a penalty of π_a is incurred, so the contribution to the objective function for this affinity attribute in team t is:

$$\pi_a v_{at}. \tag{2}$$

To compute v_{at} correctly, Rubin and Bai (2015) introduce another collection of $0-1$ integer variables, q_{gt}, for each possible value $g \in A(a)$, whose value is 1 if any person on team t has value g for this affinity attribute. These q_{gt} variables are then used to compute the value of v_{at} with the following two groups of constraints, in which $G(g)$ is the set of people in the population who have value g for affinity attribute a:

$$x_{it} \le q_{gt}, \quad \text{for } g \in A(a), i \in G(g) \tag{3}$$

$$\sum_{g \in A(a)} q_{gt} \le (|A(a)| - 1)v_{at} + 1. \tag{4}$$

However, as is now shown, it is possible to replace the integer variables v_{at} and q_{gt} with a single set of $0-1$ variables, v_{gt}, whose value is 0 if everyone on team t has value g for affinity attribute a and 1, otherwise. For affinity attribute a in team t, these variables are defined as follows:

$$v_{gt} \le \sum_{i=1}^{N} x_{it} - \sum_{i \in G(g)} x_{it}, \quad \text{for } g \in A(a) \tag{5}$$

$$S_{max} v_{gt} \ge \sum_{i=1}^{N} x_{it} - \sum_{i \in G(g)} x_{it}, \quad \text{for } g \in A(a) \tag{6}$$

Noting that in team t at most one variable v_{gt} has value 0, the penalty term in the objective function for this affinity attribute is:

$$\pi_a \left(\sum_{g \in A(a)} v_{gt} - |A(a)| + 1 \right). \tag{7}$$

Finally, because the coefficient of π_a in (7) is nonnegative, the constraints in (5) are not needed.

In summary, for each affinity constraint, the result of replacing the original variables v_{at} and q_{gt} and the constraints in (3) and (4) with the new variables v_{gt} and the constraints in (6) reduces the number of integer variables by T from the original formulation and reduces the number of constraints from $T(N|A(a)| + 1)$ in the original formulation to $T|A(a)|$.

3.1.2 Eliminating the Variables Associated with Empty Teams

The linear model in Rubin and Bai (2015) is solved with a "stepwise" approach in which the number of teams is set successively to $T = T_{min}, \ldots, T_{max}$. For a fixed value of T, for each team t, the authors use $0-1$ variables, y_t, whose value is 1, if team t is utilized and 0, if team t is not utilized. In the models proposed here, however, all of these y-variables are eliminated because all teams are utilized.

To see why utilizing all teams is a reasonable heuristic—which would be an exact approach if the program were allowed to obtain an optimal solution—consider the stepwise approach applied to the problem with $T = T_{min}$ teams and suppose, for the moment, that this problem is solved to optimality yielding an optimal objective function value of z^*. Now consider solving the next problem with $T = T_{min}+1$. To obtain an objective function value that is less than z^*, it will be necessary to utilize all $T_{min}+1$ teams. Repeatedly applying this argument means that, if the optimal solutions are sequentially found for $T = T_{min}, \ldots, T_{max}$ teams, then one can require that all teams be utilized in each problem. Although in practice the problem with T teams in the stepwise procedure might not be solved to optimality, it is still reasonable to require all T teams to be utilized in the belief that, after spending time solving the problem for T teams, it would seem unlikely that spending additional time utilizing T out of $T+1$ teams will prove beneficial.

In summary, when using the stepwise approach to solve the models proposed here, it is a reasonable heuristic to require that all T teams be utilized. This heuristic eliminates the T integer variables, y_t. Doing so also eliminates the $T-1$ anti-symmetry constraints $y_{t+1} \geq y_t$ used in the original model.

3.2 A Linear Approximation

Nonlinearities in the model arise in the terms of the objective function associated with the quantitative and qualitative attributes. For example, if α_{ni} is the value of quantitative attribute n for person i, μ_n is the average value of α_{ni} in the population, and π is the penalty per unit of difference in the average of that attribute in the team and in the population, then the total penalty for this attribute in the objective function over all T teams is:

$$\pi \sum_{t=1}^{T} \left| \frac{L_t}{S_t} - \mu_n \right|, \quad \text{where} \quad L_t = \sum_{i=1}^{N} \alpha_{ni} x_{it} \quad \text{and} \quad S_t = \sum_{i=1}^{N} x_{it}. \tag{8}$$

A similar nonlinearity arises in each term of the objective function corresponding to each value of each qualitative attribute. It is possible to linearize the absolute value expression in (8) using two new continuous variables for each team t that represent the amount by which the first term is more than, and less than, the second term, as follows:

$$\begin{aligned} \min \quad & \pi \textstyle\sum_{t=1}^{T} (\delta_t^+ + \delta_t^-) \\ \text{s.t.} \quad & \tfrac{L_t}{S_t} - \delta_t^+ + \delta_t^- = \mu_n \quad (t = 1, \ldots, T). \end{aligned} \tag{9}$$

Eliminating the nonlinearity associated with the division of L_t by S_t in the constraints of (9) is more challenging. One way to do so is to multiply (9) through by S_t before subtracting and adding δ_t^+ and δ_t^-; however, Rubin and Bai (2015), indicate that doing so leads to distorted teams, so instead, they found a clever way to do so by introducing a collection of linear constraints for each team and for each possible integer value of $S_t = K$, for $K = S_{min}, \ldots, S_{max}$ that uses new $0-1$ integer variables y_{Kt}. While doing so faithfully converts the nonlinear constraints to equivalent linear ones, it does so at the cost of using mT_{max} additional integer variables, where $m = (S_{max} - S_{min} + 1)$. Even worse is that the number of constraints increases from T_{max} to mT_{max} for each quantitative attribute and from $d * T_{max}$ to $d * T_{max} * m$ for each qualitative attribute, where d is the number of possible values for the qualitative attribute. Therefore, a different approach is taken here that constitutes one of the main improvements to what was done in Rubin and Bai (2015).

Specifically, multiplying (9) through by S_t yields

$$L_t \quad - \quad \delta_t^+ S_t + \delta_t^- S_t \quad = \quad \mu_n S_t \quad (t = 1, \ldots, T). \tag{10}$$

While the right side of (10) is linear, the left side is not. However, if the variable S_t on the left side of (10) is replaced by a constant, say S_t^*, then the following linear approximation results:

$$L_t \quad - \quad \delta_t^+ S_t^* + \delta_t^- S_t^* \quad = \quad \mu_n S_t \quad (t = 1, \ldots, T). \tag{11}$$

For each fixed value of T, the foregoing linear approximation requires values for the constants S_t^*. These values are chosen as the team sizes that minimize the total team-size penalty and the unassigned penalty. Specifically, thinking of the team sizes $S_1, \ldots, S_T, S_{T+1}$ as variables (in which team $T+1$ consists of the people who are not assigned to any of the T teams), the optimal solution to the following problem provides the values of S_t^* for $t = 1, \ldots, T$:

$$\begin{aligned}
\min \quad & \pi_S \textstyle\sum_{t=1}^{T} |S_t - S_{opt}| + \pi_U S_{T+1} \\
s.t. \quad & \\
& \textstyle\sum_{t=1}^{T+1} S_t = N, \\
& S_{min} \leq S_t \leq S_{max}, \quad t = 1, \ldots, T \\
& \text{All } S_t \geq 0 \text{ and integer.}
\end{aligned}$$

The specific linear version of the foregoing problem that is solved is given in Appendix B and the resulting optimal solution—obtained almost instantly—provides the values for the constants S_t^* for $t = 1, \ldots, T$.

A similar approach is used to linearize the nonlinearity associated with the terms corresponding to each qualitative attribute. The advantage of this linear approximation is that the integer variables y_{Kt} and large number of associated constraints used in Rubin and Bai (2015) are no longer needed. The complete linear model for this approach, together with the reduction in variables and constraints, as described in Section 3.1, is given in Appendix C.

3.3 Time-Allocation Improvements

Recall that the user specifies the minimum, ideal, and maximum number of teams, namely, T_{min}, T_{opt}, and T_{max}, respectively. In the stepwise and R3 heuristics, the 10-minute time limit is allocated to each possible value of the number of teams from T_{min} up to T_{max}.

However, preliminary computational experiments we performed revealed that in the vast majority of cases, the best solution found was relatively near T_{opt}. Based on this observation, another improvement to the stepwise and R3 heuristics is that the total time of 10 minutes is allocated only to the number of teams T being within ± 2 of T_{opt}. In other words, we only solve problems in which the number of teams T is between $T_{opt} - 2$ and $T_{opt} + 2$, thus allowing more time for CPLEX to search those number of teams.

A final improvement—determined by preliminary computational experiments—is to run the linear approximation for 9 minutes and 30 seconds to obtain the best number of teams, T^*, and the corresponding set of teams found so far, whose sizes are, say, $\hat{S}_1, \ldots, \hat{S}_{T^*}$. Then, in the remaining 30 seconds, CPLEX is applied to the *exact* linear problem in which the number of teams is set to T^* and the team sizes are fixed to $\hat{S}_1, \ldots, \hat{S}_{T^*}$, starting with the best set of teams found from the linear approximation.

In summary, the proposed approach to solving a given instance of the TBP in 10 minutes is the following:

Step 1. For each value of $T = T_{opt} - 2, T_{opt} - 1, T_{opt}, T_{opt} + 1, T_{opt} + 2$, find the best set of T teams that CPLEX can obtain in 114 seconds from solving the linear approximation in Appendix C. The linear approximation includes the improvements in Section 3.1 and requires the constants, $S_1^*, \ldots, S_T^*$, in (11) whose values are obtained from an optimal solution to the integer program in Appendix B.

Step 2. Starting with the best set of teams found in Step 1, consisting of T^* teams whose sizes are, say, $\hat{S}_1^*, \ldots, \hat{S}_{T^*}^*$, apply CPLEX for the remaining 30 seconds to the *exact* linear problem obtained from the linear approximation in Appendix B by fixing the number of teams to T^* and the team sizes to $\hat{S}_1^*, \ldots, \hat{S}_{T^*}^*$.

Computational results of using the combined linear approximation together with this final exact linear problem are presented in the next section.

4 Computational Results of Using the Improvements

In this section, computational results are presented for problem instances whose data are generated randomly. Specifically, we compare our heuristic in Section 3 with two heuristics in Rubin and Bai (2015): the "stepwise" heuristic and the "R3" heuristic. The "stepwise" heuristic solves the linear model by setting the number of teams successively to $T = T_{min}, \ldots, T_{max}$ and uses CPLEX to solve each subproblem for a given value of T. The R3 heuristic consists of three phases. In the first phase, an initial set of teams is constructed randomly; in the second phase, improved teams are found by moving a person from one team to another until no further improvement is possible; and in the third phase, further improvements are sought by swapping people on different teams until no improvement is possible. The stepwise and R3 heuristics are chosen to benchmark the performance of our heuristic because they are the best-performing heuristics in Rubin and Bai (2015).

In Section 4.1, the performance of the three heuristics are compared on randomly generated instances. There it is shown that our heuristic performs significantly better than the stepwise and R3 heuristics under different application contexts and intuition is provided for explaining those results.

To assess the effectiveness of using the heuristics, it is desirable to compare the value of the objective function obtained by the heuristics to that of an optimal solution. However, finding an optimal solution in general requires too much computational effort [see Rubin and Bai (2015)]. Another contribution of the work here is the ability to generate data for the TBP for which a nearly optimal solution is known, thus providing the ability to determine approximately how close the objective function value obtained by the heuristics are. Details of how this is done and the results of computational experiments on such instances using the stepwise, R3 and new heuristics are given in Section 4.2.

4.1 Computational Comparisons on Randomly Generated Instances

Arguably, one of the most important measures of a heuristic is its scalability, i.e., its performance as the size of the problem increases. In the context of the TBP, the problem size is largely determined by the population size N and the ideal team size S_{opt}. Intuitively, the larger the population, the more people to be assigned and hence the larger the TBP. For a given population, the larger the ideal team size, the fewer teams that are needed and hence the smaller the TBP. Thus, the scalability of the heuristics can be assessed by varying N and S_{opt} values.

In addition to scalability, another important metric for a TBP heuristic is its performance robustness to the user-input penalties assigned to imbalances in different terms in the objective function. This is because in practice when a TBP is solved, a user is likely to experiment with different penalty values until a desirable team configuration is obtained. If a heuristic performs well only under particular settings of penalty inputs, then its applicability to real-life TBPs would be limited. Thus, our numerical experiments are designed to assess the effects of penalty inputs on heuristic performance.

4.1.1 Experimental Design

In this subsection the experimental design used to study the effect of penalty inputs is explained and the overall experiment design is summarized in Table 1.

Note that the objective function of a TBP with T teams is

$$\begin{aligned} \min \quad & \pi_T|T - T_{opt}| + \pi_S \sum_{t=1}^{T} |S_t - S_{opt}| + \pi_U S_{T+1} \\ & + \sum_{t=1}^{T} \left[\sum_{i}^{n^{qn}} w_i^{qn} \Delta Qn_{it} + \sum_{i}^{n^{ql}} w_i^{ql} \sum_{j}^{d_i^q} \Delta Ql_{ijt} + \sum_{i}^{n^a} p_i^a \Delta A_{it} \right]. \end{aligned} \tag{12}$$

In (12), the three terms in the first line are penalties for imbalances in non-attributes, with:

- π_T being the penalty per unit of deviation from the ideal number of teams, T_{opt}, which is computed as N/S_{opt} (rounded to the nearest integer).
- π_S being the penalty per unit of deviation from the ideal team size in each team.
- π_U being the penalty for each unassigned person.

The second line of (12) is the penalty for imbalances in the attributes, of which, there are n^{qn} quantitative attributes, n^{ql} qualitative attributes, and n^a affinity attributes. The i^{th} qualitative attribute ($i \in \{1, \ldots, n^{ql}\}$) has d_i^q values that are numbered $1, \ldots, d_i^q$. In team $t \in \{1, \ldots, T\}$, ΔQn_{it} denotes the imbalance in the i^{th} quantitative attribute and w_i^{qn} is the penalty per unit of imbalance in this attribute. Similarly, ΔQl_{ijt} denotes the imbalance in value j of the i^{th} qualitative attribute and w_i^{ql} is the penalty per unit of imbalance in this attribute, ΔA_{it} denotes the imbalance in the i^{th} affinity attribute and p_i^a is the per-unit penalty. The formulas for the imbalances ΔQn_{it}, ΔQl_{ijt}, and ΔA_{it} are given in Appendix A.

It is important to note that the objective function in (12) depends on the per-unit penalty as well as the characteristics of the population. For example, the contribution of team-size imbalance depends on the per-unit penalty π_S as well as the number of teams T. The contribution of imbalance in the qualitative attributes depends on the per-unit penalty w_i^{ql}, the size of the attribute domain d_i^q, the number of qualitative attributes n^{ql}, and the number of teams, T.

Thus, the same per-unit penalty inputs may have significantly different optimization implications for different populations. To study the performance robustness of a heuristic, it is necessary to control for these confounding effects from the population. To this end, the per-unit penalties are normalized as follows in the numerical experiments:

$$\pi_T = \pi_S T, \ \pi_U = \pi_S \frac{T}{N}, \ w_i^{qn} = \frac{\pi_a}{n^{qn} R_i}, \ w_i^{ql} = \frac{\pi_a}{n^{ql} d_i^q}, \ p_i^a = \frac{\pi_a}{n^a}, \tag{13}$$

where R_i is the range of the i^{th} quantitative attribute. Normalization in (13) implies that, within each team, the imbalances in number of teams, team size, and unassigned people each receive the same weight that is determined by π_S, and imbalances in the quantitative, qualitative, and affinity attributes each receive the same weight that is determined by π_a.

Under (13), the relative magnitudes of π_S and π_a reflect the relative weights of non-attribute and attribute imbalances for any population. Thus, their ratio

$$\Pi = \frac{\pi_a}{\pi_S} \tag{14}$$

is used to examine the robustness of heuristics.

In all our numerical experiments, the population size N is generated randomly from four groups: (i) $N \in (30, 60)$, (ii) $N \in (60, 90)$, (iii) $N \in (90, 120)$, and (iv) $N \in (120, 150)$. The following table summarizes the key parameters in three numerical experiments that investigate the effects of N, ideal team size S_{opt}, and the penalty ratio Π, respectively.

Besides the population size N, the ideal team sizes $(S_{min}, S_{opt}, S_{max})$, and the penalty ratio Π, other data are generated randomly as follows. The number of quantitative attributes is randomly selected between 2 and 5 with the attribute values uniformly distributed between 1 and 5. The number of qualitative attributes is randomly chosen between 1 and 5 with the attribute values uniformly distributed between 2 and 5. The number of affinity attributes is random between 0 and 2 with the number of values for each attribute uniformly distributed between 2 and 5. In each problem instance, the number of teams are computed from $(S_{min}, S_{opt}, S_{max})$ values: $T_{min} = \lceil N/S_{max} \rceil$ and $T_{max} = \lfloor N/S_{min} \rfloor$.

	Design 1	Design 2	Design 3
Num. of instances in each population group	25	5	5
Team sizes: $(S_{min}, S_{opt}, S_{max})$	$(4, 6, 8)$	$(3, 4, 5)$ $(4, 6, 8)$ $(7, 9, 10)$ $(10, 13, 16)$ $(15, 17, 20)$	$(4, 6, 8)$
Penalty ratio: Π	1	1	$0.02, 0.05, 1, 20, 50$

Table 1: Summary of experiment design

4.1.2 Results

For each generated problem, the new heuristic in Section 3 and the stepwise and R3 heuristics in Rubin and Bai (2015) are applied for 10 minutes each. Let V, V^S, and V^R denote the best objective values found by the three respective methods. The improvement of the new heuristic over the stepwise and R3 heuristics is computed as follows:

$$\alpha^k = \frac{V^k - V}{V^k} \qquad k \in \{S, R\}. \tag{15}$$

Figure 1 shows the result of Design 1 in Table 1 and compares the performance of the new heuristic relative to the stepwise and R3 heuristics. In this figure, each point represent a problem instance and a total of 100 instances are generated, 25 in each population group. The penalty ratio Π is set to one so the non-attribute imbalance has the same weight as the attribute imbalance in the objective.

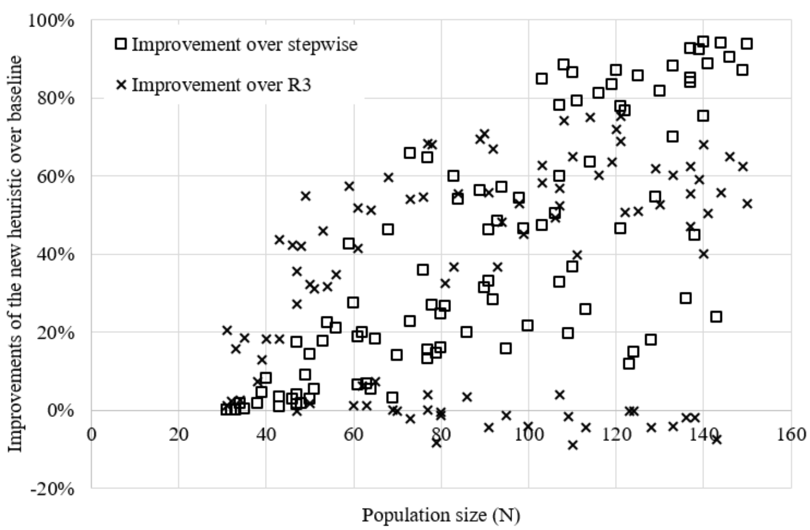


Figure 1: Design 1: Improvement of the new heuristic over stepwise and R3 heuristics with $S_{opt} = 6$ and $\Pi = 1$

Figure 1 shows that the new heuristic scales better than the stepwise and R3 heuristics. The advantage of the new heuristic, measured by α^S and α^R, increases as the population size increases. Specifically, the new heuristic is comparable to the stepwise heuristic when the population is small, i.e., $N < 50$. The improvement α^S, though always nonnegative, is within 10% for the majority of cases. As N increases, α^S increases steadily and can be as high as 94%.

The improvement of the new heuristic over the R3 heuristic, α^R, behaves quite differently. While α^R can be as high as 55% for small populations with $N < 50$, it plateaus and remains stable around 60% after an initial increase in N. The highest improvement for α^R is 75%. Furthermore, α^R is negative in 14 out of 100 simulated instances, with the lowest value being -9%. Thus, the new heuristic performs slightly worse than the R3 heuristic with probability 0.14 (and this probability does not seem to decrease as the population size grows larger).

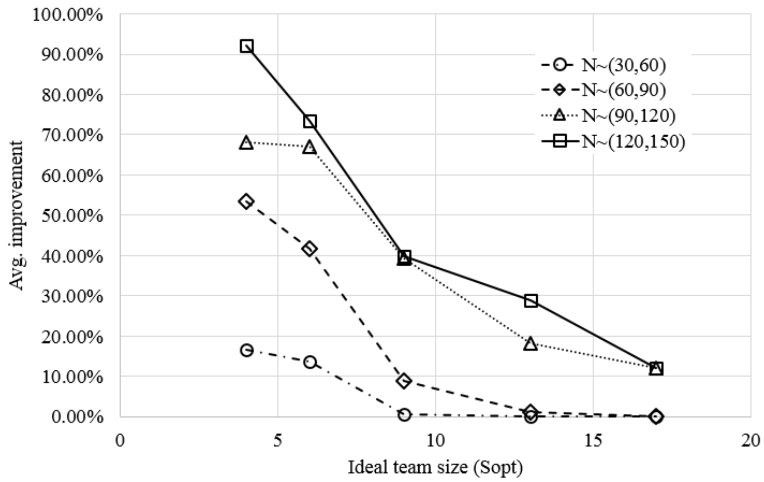
The difference between α^R and α^S reflects the different solution approaches used in the stepwise and R3 heuristics. The former solves $T_{max} - T_{min} + 1$ optimization subproblems in ten minutes. Because $S_{min} = 4$ and $S_{max} = 8$ are fixed in all instances, the number of subproblems that are solved by the stepwise heuristic increases with the population size N, resulting in the fast deterioration of the heuristic performance. In contrast, under the time-allocation improvement (see Section 3.3), our heuristic solves five optimization subproblems, regardless of the population size, thus scaling better. If our heuristic were to solve an increasing number of subproblems, one would expect the improvement α^S to decrease.

The R3 heuristic, in comparison, relies on moving and swapping people across different teams. Holding the team sizes fixed, a larger population implies more teams to move and swap people from, thus leading to worse performance. The effect of expanding the team size, however, is more nuanced. On the one hand, having larger teams reduces the number of alternatives to move and swap people from but results in more people on each team for moving and swapping. Thus, increasing the number of teams should lead to less benefit on the performance of the R3 heuristic than the stepwise heuristic.

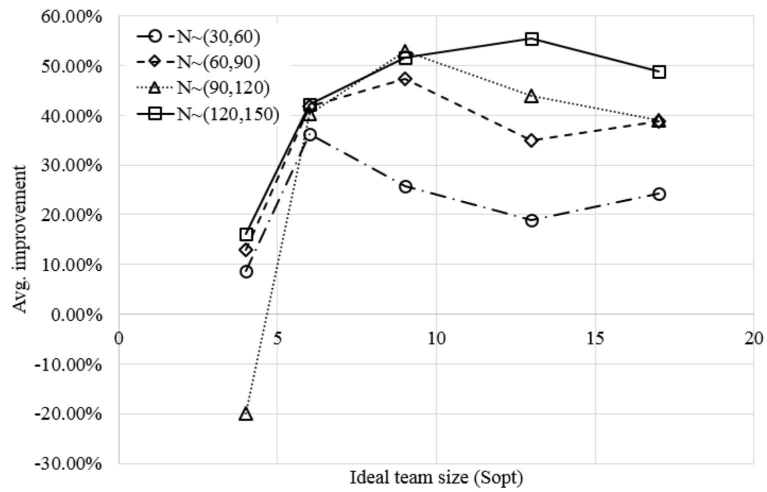
Figure 2 confirms this intuition by varying the team sizes in the problem instances. Specifically, the instances are generated based on Design 2 in Table 1, where five team-size specifications are considered. In each of the four population groups, five random instances are generated and solved by the three heuristics under these five team-size specifications. For each specification, the improvement metrics α^S and α^R are computed so there are five pairs of (α^S, α^R) for each (population, team-size) combination. The average of these improvement values, denoted by $\bar{\alpha}^k$ ($k \in \{S, R\}$), are computed, with $\bar{\alpha}^R$ plotted in Figure 2a and $\bar{\alpha}^S$ in Figure 2b. The penalty ratio Π remains one in all calculations.

From Figure 2a, the stepwise heuristic performs significantly better as the team size increases. For the large population group $N \in (120, 150)$, the average improvement of our heuristic decreases from 92% under the smallest team size to 12% under the largest team size. This result suggests that the additional time made available by our heuristic plays a major role in its superior performance on large population sizes.

Figure 2b illustrates the mixed effect of larger teams for the R3 heuristic. As expected, the R3 heuristic does not benefit as much as the stepwise heuristic from an increase in team size. In fact, it may perform worse with larger, rather than smaller, teams. Under the largest teams, our heuristic is on average more than 20% better than R3 for all population groups.



(a) Average improvement over the stepwise heuristic: $\bar{\alpha}^S$

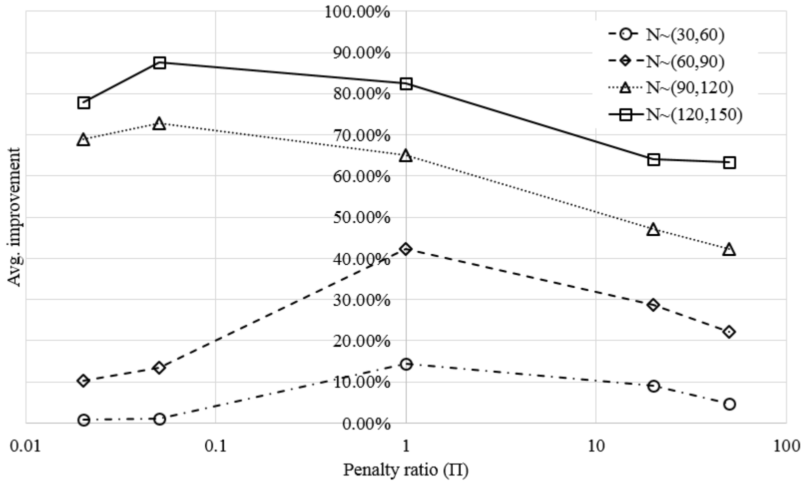


(b) Average improvement over the R3 heuristic: $\bar{\alpha}^R$

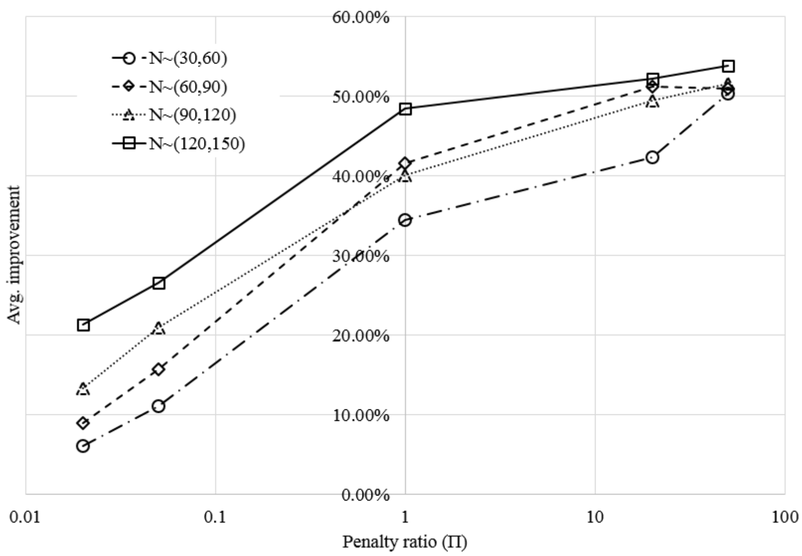
Figure 2: Design 2: Average improvement of the new heuristic with $\Pi = 1$

The performance robustness of the heuristics in user-input penalties is examined in Design 3 of Table 1, where the penalty ratio Π takes five values: 0.02, 0.05, 1, 20, 50. As in Design 2, five random problem instances are generated in each of the four population groups. These problem instances are solved by the three heuristics with each of the foregoing five values of Π. The average improvement $\bar{\alpha}^k$ ($k \in \{R, S\}$) is computed for each (population, penalty) combination. The results for $\bar{\alpha}^S$ and $\bar{\alpha}^R$ are in Figures 3a and 3b, respectively.

From Figure 3a, it is seen that our heuristic is consistently better than the stepwise heuristic for all penalty ratios in all population groups. There is no clear dependence of the improvement on the penalty. This result is quite different in Figure 3b with the R3 heuristic. The advantage of our heuristic over the R3 heuristic grows as more weight is put on the attribute imbalances over the non-attribute imbalances [see (14)]. Intuitively, this is because the R3 heuristic relies on moving and swapping people across teams. The moving step affects the non-attribute imbalances in the number of teams, team sizes, and



(a) Average improvement over the stepwise heuristic: $\bar{\alpha}^S$



(b) Average improvement over the R3 heuristic: $\bar{\alpha}^R$

Figure 3: Design 3: Average improvement of the new heuristic with $S_{opt} = 6$

unassigned people, as well as the attribute imbalances in the quantitative, qualitative, and affinity features of the teams. The swapping step, however, affects only the attribute imbalances. When the penalty ratio Π is low—and thus the non-attribute imbalance contributes much more to the imbalance than the attribute imbalance—the R3 heuristic first achieves a good team configuration through the moving step and then fine-tunes the teams through swapping. This "divide-and-conquer" strategy makes it quite effective. As Π increases, the non-attribute imbalances become less and less important. Thus, the difference between the moving and swapping steps diminishes and R3 can no longer improve the objective value in a sequential systematic way. That heuristic essentially relies on randomly adjusting people in teams and so its performance deteriorates.

To sum up, the numerical experiments demonstrate that our heuristic is consistently superior to the heuristics in Rubin and Bai (2015) under different problem sizes and penalty inputs. The time-allocation improvement of our heuristic enables better scaling as the problem size increases. Its optimization-based nature guarantees an efficient approach for searching for the optimal solution under different penalty inputs.

4.2 Computational Comparisons on Almost Perfectly Balanced Problems

While in Section 4.1 the advantage of the new heuristic for large populations is seen, the results are relative in that they show the amount by which the new heuristic improves over the stepwise and R3 heuristics but not in relation to the true optimum. Unfortunately, it is generally impossible to know how good a heuristic is for the TBP because the true optimum is unobtainable. However, by generating data for which the population has perfectly balanced teams and then adding one extra person to the team that results in the smallest true objective function value, it is possible to get a good estimate of the optimal objective function value for this problem. Even if the objective function value for this "almost perfectly balanced" problem is not the true optimum, it is an upper bound against which it is possible to compare the objective function value obtained from any heuristic. To that end, let $\tilde{\Pi}^b$ denote the objective function value of a problem with almost perfectly balanced teams and let $\tilde{\Pi}^k$ ($k \in \{S, R, N\}$) be the final objective values generated by the stepwise heuristic ($k = S$), R3 heuristic ($k = R$), and our new heuristic ($k = N$) heuristic. Define the *relative deviation* D^k as

$$D^k = \frac{\tilde{\Pi}^k - \tilde{\Pi}^b}{\tilde{\Pi}^b}. \tag{16}$$

Then D^k is an estimate of a heuristic's relative deviation from the true optimum. Note that perfectly balanced populations—in which $\tilde{\Pi}^b_i$ is zero—are avoided so that D^k is well-defined.

It is important to note that the data generated for almost perfectly balanced teams—though created differently from that in the randomize experiments in Section 4.1—do not favor one heuristic over another. By construction, the population with almost perfectly balanced teams has some symmetry: it is a population with perfectly balanced teams plus one individual. However, the stepwise and R3 heuristics do not have any built-in structure that facilitates their solution of a population with symmetry. Likewise, the new heuristic, being an improved version of the stepwise heuristic, also does not exploit symmetry of the data. Thus, one would expect the performance of the heuristics to be consistent with that in Section 4.1, which allows us to use the results in this subsection to understand, in absolute terms, how far the heuristics' objective function values are from that of the true optimal solution.

4.2.1 Generating Data for Which There Are Perfectly Balanced Teams

Generating data for which there are perfectly balanced teams is now described. To that end, consider a TBP with a single quantitative attribute where the goal is to construct T teams with S people on each team from a population of $N = S * T$ people. For a team to be perfectly balanced, the average of the attribute values of the people on the team must be equal to the average attribute values in the population. To achieve this, assign each of the S people on a team a different attribute value in the set $\{1, 2, \ldots, S\}$. Then the average attribute value of the team is $(S + 1)/2$, which is precisely the average attribute value in the whole population, thus resulting in T perfectly balanced teams.

To generate data for a second quantitative attribute so as to ensure that the foregoing perfectly balanced teams remain perfectly balanced, shift the value of the first attribute of each person down to the next person on the team, with the value of the last person on the team becoming that of the first person. Thus, if the values of the first attribute of the S people on a team are $a_1, a_2, \ldots, a_{S-1}, a_S$, then the values of the second attribute for those same S people are $a_S, a_1, a_2, \ldots, a_{S-1}$. A subsequent rotation can be used for a third quantitative attribute, and so on.

Generating data for a qualitative attribute that ensures perfectly balanced teams requires more care. This is because, for a team to be perfectly balanced with respect to a qualitative attribute with d values, say in the set $\{1, \ldots, d\}$, the fraction of people on each team who have each of those d values must equal the fraction of the population that has those same d values. This will be possible when d divides S, for in this case, there is an integer k such that $S = kd$ and so a team with S people can be divided into k subgroups of d people each. In this case, a perfectly balanced team results when each of the d people in a subgroup is assigned a unique qualitative value in the set $\{1, \ldots, d\}$. Additional qualitative attributes can be handled in a similar manner so long as the number of possible values for the attribute divides S.

Generating data for an affinity attribute to ensure perfectly balanced teams is straightforward. If an affinity attribute has k possible values, say in the set $\{1, \ldots, k\}$, then simply assign each of the S people on the first team the value 1 for this attribute, then each of the S people on the second team the value 2 for this attribute, and so on, cycling back to the first value for this attribute, if necessary. Additional affinity attributes are handled in the same way.

4.2.2 Results for Population with Almost Perfectly Balanced Teams

Because the advantage of the new heuristic lies mainly in problems with large populations, 60 almost perfectly balanced instances were generated with populations ranging from 60 to 145. The stepwise heuristic achieves the upper bound, i.e., $D^S = 0$, in 8.3% of the 60 instances; the R3 heuristic achieves $D^R = 0$ in 6.7% of those instances; and the new heuristic achieves $D^N = 0$ in 58.3% of the instances.

Figure 4 shows the relative deviations of the stepwise, R3, and the new heuristics for the 60 test instances, as a function of the size of the population. Similar to Figure 1, the advantage of the new heuristic increases with the population size. Furthermore, from Figure 4a, the relative deviation of the stepwise and R3 heuristics can be as high as 22 and 10, respectively. In contrast, the relative deviation of the new heuristic generally falls within 1.5. To mitigate the effects of outliers, we compute the *mean relative deviation* as the average relative deviations of all the problems with the same population and plot the results in Figure 4b. It can be seen that the mean relative deviation of the new heuristic is below 0.5 whereas the mean relative deviation of the stepwise and R3 heuristics may reach 10.5 and 6 for large instances.

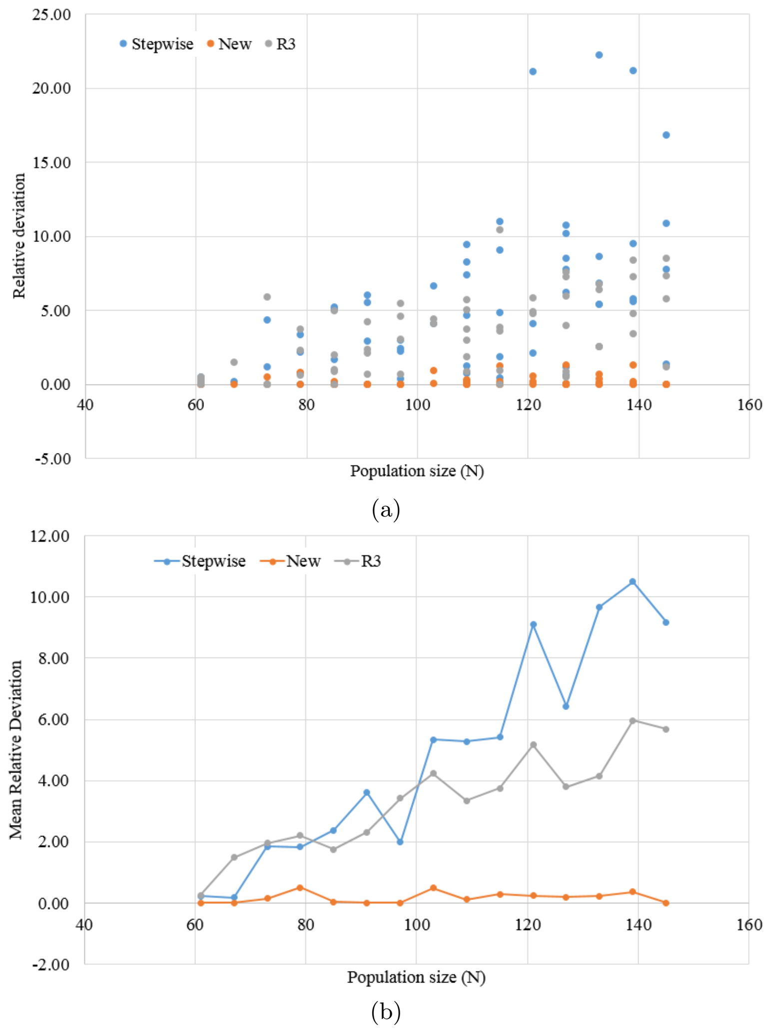


Figure 4: Comparison of the Heuristics in Almost Perfectly Balanced Population

5 Summary

The Team Balancing Problem is to divide a group of people into a collection of teams that are balanced across a number of quantitative, qualitative, and affinity attributes, as measured by a nonnegative objective function whose value represents the amount of imbalance in the teams. To solve this problem, a nonlinear integer programming model was proposed in Rubin and Bai (2015) together with an equivalent linear integer program requiring a large number of integer variables and constraints (the "original" model). This model was solved in Rubin and Bai (2015) using both a "stepwise" heuristic (consisting of a collection of subproblems, each of which has a fixed number of teams) and an "R3" heuristic (consisting of moving and swapping team members so long as improvement in the objective function is obtained).

One contribution of the work here is modifications to the original models that significantly reduce the number of integer variables and constraints. The main reduction in the size of the problem is obtained by replacing the nonlinear objective function with a linear approximation. This, together with some additional time-allocation heuristics, constitutes our new heuristic. Computational experiments are run to compare the final objective function values of the stepwise, R3, and new heuristics on instances of the problem with randomly-generated data and data for almost perfectly balanced teams, the latter providing a way to determine the quality of the optimal solution obtained by the heuristics.

Across all of the random instances that were tested, the new heuristic generally obtained more balanced teams than the stepwise heuristic, with the benefits increasing as the population size increases. Compared to the R3 heuristic, the new heuristic performed better on 86% of the instances and slightly worse on 14% of the instances. As to the quality of the objective function obtained from data in which there are almost perfectly balanced teams, the new heuristic achieves the upper bound of the true optimum in 58% of the 60 instances tested, the stepwise heuristic achieves the upper bound on only 8% of the instances, and the R3 heuristics does so on 7% of the instances. On virtually all of the instances, the optimal objective function value from the new heuristic is closer to the upper bound than either the stepwise or R3 heuristics.

In the heuristic proposed here, the final 30 seconds of the 10 minutes of computer time allocated to solve an instance of the TBP is devoted to the R3 improvement heuristic. Is 30 seconds the optimal amount of time for the R3 heuristic? Future research could involve computational experiments to determine a more effective amount of time. Another possible efficiency in the stepwise approach could result if it were possible to use the best set of T teams found to create a good set of $T + 1$ teams for starting the next problem.

It is also interesting to note that the graph version of the TBP is a generalization of the clique partitioning problem. In the latter problem, given weights for each edge, the objective is to partition the vertices of the graph into cliques so as to minimize the sum of the weights of the edges in the cliques. In the TBP, the contribution to the objective function of a clique is a complex function of the vertices. Another future research topic might be to formulate a more general clique-partitioning problem and to develop appropriate heuristics.

Acknowledgments

The authors would like to acknowledge Ai Zhao who, as an MS student in our department, wrote a preliminary version of the computer code for testing the models developed here. Daniel Solow is also grateful to the Department of Operations Research at the Naval Postgraduate School in Monterey, CA where, as a Visiting Professor he was able to do much of the foundational work for this research. We also gratefully acknowledge the comments of anonymous reviewers and Lizhi Wang who, as editor, gave us the necessary time to produce this significantly improved version.

Appendix A: The Nonlinear Objective Function

In this section, the specific form of the objective function of the network model in (1) presented in Section 3 is given. To that end, the data for the TBP consists of the following:

N	$=$	the number of people in the population.
T_{min}, T_{opt}, T_{max}	$=$	the minimum, ideal, and maximum number of teams.
S_{min}, S_{opt}, S_{max}	$=$	the minimum, ideal, and maximum size of a team.
π_T	$=$	the penalty for each team above or below T_{opt}.
π_S	$=$	the penalty for each member of a team above or below S_{opt}.
π_U	$=$	the penalty for each unassigned person.
n^{qn}	$=$	the number of quantitative attributes.
α_{ik}^{qn}	$=$	the value of quantitative attribute i for person k $(i = 1, \ldots, n^{qn},\ k = 1, \ldots, N)$
n^{ql}	$=$	the number of qualitative attributes.
d_i^q	$=$	the number of values for qualitative attribute i and whose values are numbered $1, \ldots, d_i^q$ $(i = 1, \ldots, n^{ql})$
α_{ijk}^{ql}	$=$	1 if the i-th qualitative attribute of person k takes value j and 0 otherwise $(i = 1, \ldots, n^{ql},\ j = 1, \ldots, d_i^q,\ k = 1, \ldots, N)$
n^a	$=$	the number of affinity attributes.
d_i^a	$=$	the number of values for affinity attribute i and whose values are numbered $1, \ldots, d_i^a$ $(i = 1, \ldots, n^{ql})$
$G^i(g)$	$=$	the set of people in population with value g for affinity attribute i $(i = 1, \ldots, n^a,\ g = 1, \ldots, d_i^a)$
w_i^{qn}	$=$	the penalty for quantitative attribute i $(i = 1, \ldots, n^{qn})$.
w_i^{ql}	$=$	the penalty for qualitative attribute i $(i = 1, \ldots, n^{ql})$.
p_i^a	$=$	the penalty for affinity attribute i $(i = 1, \ldots, n^a)$.
μ_i^P	$=$	the average of quantitative attribute i in the population $(i = 1, \ldots, n^{qn})$
f_{ij}^P	$=$	the fraction of the population whose value for qualitative attribute i is j $(i = 1, \ldots, n^{qn};\ j = 1 \ldots, d_i^q)$

The TBP is to identify an optimal number of teams T with $T_{min} \le T \le T_{max}$ together with a partition of V into $T+1$ teams represented by the sets $V_1, \ldots, V_T, V_{T+1}$ whose sizes are $S_1, \ldots, S_t, S_{T+1}$, where and the set V_{T+1} is the set of unassigned people, so as to

$$\min \pi_T|T - T_{opt}| + \pi_S \sum_{t=1}^{T} |S_t - S_{opt}| + \pi_U S_{T+1} + W. \tag{17}$$

The first term in (17) is the penalty for the number of teams differing from the ideal number. The second term in (17) is the penalty for the team sizes differing from the ideal team size. The third term in (17) is the penalty for the number of unassigned people. The fourth term, W, is the sum of the penalties associated with the teams being unbalanced and is defined as follows:

$$\sum_{t=1}^{T} \left[\sum_{i=1}^{n^{qn}} w_i^{qn} |\mu_i^P - \mu_{it}| + \sum_{i=1}^{n^{ql}} w_i^{ql} \sum_{j=1}^{d_i^q} |f_{ij}^P - f_{ijt}| + \sum_{i=1}^{n^a} p_i^a \chi_{it} \right] \tag{18}$$

in which the following are variables (in addition to the variables $S_t, t = 1, \ldots, T+1$):

$$\mu_{it} = \text{the average of quantitative attribute } i \text{ in team } t \; (i = 1, \ldots, n^{qn};\ t = 1, \ldots, T)$$

$$f_{ijt} = \text{the fraction of team } t \text{ whose value for qualitative attribute } i \text{ is } j \; (i = 1, \ldots, n^{qn};\ j = 1 \ldots, d_i^q)$$

$$\chi_{it} = \begin{cases} 1, & \text{if all members of team } t \text{ have the same value for affinity attribute } i \\ 0, & \text{otherwise} \end{cases}$$

Additional variables and constraints are needed to ensure that the foregoing variables have the meanings indicated. For example,

$$S_t = \sum_{i=1}^{N} x_{it} \quad (t = 1, \ldots, T)$$

where

$$x_{it} = \begin{cases} 1, & \text{if person } i \text{ is assigned to team } t \\ 0, & \text{otherwise} \end{cases}$$

All of the necessary additional variables, together with all of the constraints needed to represent the complete model associated with the linear approximation in Section 3, are presented in Appendices B and C.

Appendix B: Integer Programs for Obtaining Constants for the Linear Approximation

The linear approximation in Section 3.2 requires, for each fixed value of T, values for the constants, S_t^*, for all teams. These values are obtained from an optimal solution to the following integer program, in which team $T+1$ consists of the people who are not assigned to any of the T teams (notation for the data and variables are the same as in Appendix A except that new continuous variables β_t^+ and β_t^- are included here to linearize the following absolute-value term):

$$
\begin{array}{ll}
\min & \pi_S \sum_{t=1}^{T} |S_t - S_{opt}| + \pi_U S_{T+1} = \pi_S \sum_{t=1}^{T} (\beta_t^+ + \beta_t^-) + \pi_U S_{T+1} \\
s.t. & \\
& \sum_{t=1}^{T+1} S_t = N, \\
& S_t - \beta_t^+ + \beta_t^- = S_{opt}, \qquad t = 1, \ldots, T \\
& S_{min} \leq S_t \leq S_{max}, \qquad t = 1, \ldots, T \\
& \text{All } S_t \text{ variables are integer and all } \beta_t^+, \beta_t^- \geq 0.
\end{array}
$$

This problem is solved almost instantly and usually has multiple optimal solutions. After obtaining an optimal solution with objective function value z^*, the following integer program is solved (almost instantly) to make the teams sizes as close to each other as possible:

$$
\begin{array}{ll}
\min & y \\
s.t. & \\
& y \geq S_i - S_j \qquad i, j = 1, \ldots, T \\
& \pi_S \sum_{t=1}^{T} (\beta_t^+ + \beta_t^-) + \pi_U S_{T+1} = z^*, \\
& \sum_{t=1}^{T+1} S_t = N, \\
& S_t - \beta_t^+ + \beta_t^- = S_{opt}, \qquad t = 1, \ldots, T \\
& S_{min} \leq S_t \leq S_{max}, \qquad t = 1, \ldots, T \\
& \text{All } S_t \text{ variables are integer and all other variables } \geq 0.
\end{array}
$$

Appendix C: The Linear Approximation With Reduced Variables and Constraints

The TBP is challenging to solve partly because the number of teams T is a variable. To simplify the problem, the TBP is divided into $T_{max} - T_{min} + 1$ subproblems, each of which has a fixed number of teams between T_{min} and T_{max}. After running CPLEX for a given amount of time on each subproblem, the one achieving the best overall objective function value is chosen as the final solution. The following is the model for a subproblem with T teams, in which team $T+1$ consists of the people who are not assigned to any of the T teams, the data are the same as given in Appendix A, the values of the constants S_t^* for each team t are obtained from the optimal solution to the integer programs in Appendix B, and antisymmetry constraints from Rubin and Bai (2015) are included.

$$\min \quad \pi_T|T - T_{opt}| + \pi_U S_{T+1} + \sum_{t=1}^{T} \left[\pi_S(\beta_t^+ + \beta_t^-) + z_t \right]$$

s.t.

$$S_t = \sum_{k=1}^{N} x_{kt}, \qquad t = 1, \dots, T+1 \tag{19a}$$

$$S_t - \beta_t^+ + \beta_t^- = S_{opt}, \qquad t = 1, \dots, T \tag{19b}$$

$$\sum_{k=1}^{N} \alpha_{ik}^{qn} x_{kt} - S_t^* \gamma_{it}^+ + S_t^* \gamma_{it}^- = \mu_i^P S_t, \quad t = 1, \dots, T; i = 1, \dots, n^{qn} \tag{19c}$$

$$\sum_{k=1}^{N} \alpha_{ijk}^{ql} x_{kt} - S_t^* \delta_{ijt}^+ + S_t^* \delta_{ijt}^- = f_{ij}^P S_t, \; t = 1, \dots, T; i = 1, \dots, n^{ql}; \; j = 1, \dots, d_i^q \tag{19d}$$

$$z_t = \sum_{i=1}^{n^{qn}} w_i^{qn} (\gamma_t^+ + \gamma_t^-) + \sum_{i=1}^{n^{ql}} w_i^{ql} \sum_{j=1}^{d_i^q} (\delta_{ijt}^+ + \delta_{ijt}^-) + \sum_{i=1}^{n^a} p_i^a \chi_{it}, \; t = 1, \dots, T \tag{19e}$$

$$z_t \geq z_{t+1}, \; t = 1, \dots, T-1 \tag{19f}$$

$$\chi_{it} = \sum_{g=1}^{d_i^a} v_{gt}^i - d_i^a + 1, \qquad t = 1, \dots, T; i = 1, \dots, n^a \tag{19g}$$

$$S_{max} v_{igt} \geq S_t - \sum_{k \in G^i(g)} x_{kt}, \qquad t = 1, \dots, T, \; i = 1, \dots, n^a, \; g = 1, \dots, d_i^a \tag{19h}$$

$$\sum_{t=1}^{T+1} x_{kt} = 1, \qquad k = 1, \dots, N \tag{19i}$$

$$S_{min} \leq S_t \leq S_{max}, \qquad t = 1, \dots, T \tag{19j}$$

$$\text{All variables } x_{kt}, v_{igt} \in \{0, 1\} \text{ and all other variables } \geq 0. \tag{19k}$$

References

Awal, Gaganmeet Kaur, K. K. Bharadwaj. 2014. Team formation in social networks based on collective intelligence - an evolutionary approach. *Applied Intelligence* **41**(2) 627–648.

Benincasa, Giacomo, Konstantin Pavlikov, Donald Hearn. 2017. Algorithms and software for the golf director problem. Tech. rep.

Contreras, Ricardo, Pedro Salcedo. 2017. Genetic algorithms as a tool for structuring collaborative groups. *Natural Computing* **16**(2) 231–239.

Cutshall, Rex, Srinagesh Gavirneni, Kenneth Schultz. 2007. Indiana university's kelley school of business uses integer programming to form equitable, cohesive student teams. *Interfaces* **37**(3) 265–276.

Drnevich, Vincent, John Norris. 2007. Assigning civil engineering students to capstone course teams. *Proceedings of the ASEE Annual Conference at Honolulu*.

Fan, Zhi-Ping, Yuan Chen, Jian Ma, Yan Zhu. 2009. Decision support for proposal grouping: a hybrid approach using knowledge rule and genetic algorithm. *Expert systems with applications* **36**(2) 1004–1013.

Magnanti, Thomas L., Karthik Natarajan. 2018. Allocating students to multidisciplinary capstone projects using discrete optimization. *Interfaces* **48**(3) 204–216.

Marsh, MT, DA Schilling. 1994. Equity measurement in facility location analysis: A review and framework. *European Journal of Operations Research* **74**(1) 1–17.

Rubin, Paul A., Lihui Bai. 2015. Forming competitively balanced teams. *IIE Transactions* **47**(6) 620–633.

Savas, SE. 1978. On equity in providing public services. *Management Science* **24**(8) 800–808.

Srba, Ivan, Maria Bielikova. 2015. Dynamic group formation as an approach to collaborative learning support. *IEEE Transactions on Learning Tecnologies* **8**(2) 173–186.