| | |
|---|---|
| **1. Module number** | *SET10117* |
| **2. Module title** | *Emergent Computing for Optimisation* |
| **3. Module leader** | *Emma Hart* |
| **4. Tutor with responsibility for this Assessment**<br>Student's first point of contact | *As above* |
| **5. Assessment** | *Report* |
| **6. Weighting** | *60% of overall module total:* |
| **7. Size and/or time limits for assessment** | *Report – 6 pages* |
| **8. Deadline of submission**<br>Your attention is drawn to the penalties for late submission | Hand-in: by midday on 30th November |
| **9. Arrangements for submission** | Via Moodle |

| | |
|---|---|
| **10. Assessment Regulations**<br>All assessments are subject to the University Regulations**.** | |
| **11. The requirements for the assessment** | *Please see attached document* |
| **12. Special instructions** | *See attached document* |
| **13. Return of work** | *within 3 weeks of submission.* |
| **14. Assessment criteria** | *See attached document*<br><br>*Normal academic conventions for acknowledging sources should be followed.* |

**SET10117 Optimisation of a Fantasy Football Team**

In the online game of Fantasy Football, players have to select a high-quality football team from a large pool of potential players, purchased using a fixed budget. Each player has a "points" score associated with them, derived from large numbers of statistics gathered from previous games (for example, goals scored, man of the match, 'clean-sheet', red cards, penalty scored/saved, etc). They also have a purchase cost. From a computational perspective, it's an NP-hard combinatorial optimisation problem to solve to find the best possible selection while remaining in budget.

As well as the constraint regarding the maximum budget that can be spent, there are a number of additional constraints that need to be satisfied. In fact, there are 7 constraints in total:

1. The total cost of the team must be less than or equal to **£100**
2. There must be exactly 11 players in the team
3. You can't pick the same player more than once (i.e. all players in a team are unique)
4. There must be exactly 1 goal-keeper (GK)
5. There must be at least 3 but no more than 5 defenders (DEF)
6. There must be at least 3 but no more than 5 midfielders (MID)
7. There must be at least 1 but no more than 3 strikers (STR)

**Hence, the goal is to select a team that maximises the total points score of the team, remains within budget and satisfies all of the constraints.**

**What to do**

Your goal is to write an algorithm to find the best possible solution to this problem (i.e. a team that maximises the point score, remains within budget and does not break any constraints) using any meta-heuristic or local search algorithm (or a hybrid method). You can use any of the methods covered in the module, but you are also free to do some research and use another method if it falls under the category of a stochastic search algorithm*. If you want to use a method not covered in the lecture material, please check with me before going ahead.*

You can implement your algorithm using the DEAP libraries using the knowledge you have learned in class, but if you prefer to write in another language then that is fine. Note there are no marks associated with the style of the code or for its efficiency or for choosing one language over another.

A data-file is provided that contains real information from the 17/18 soccer season. The file contains cost, points and position information for 523 players from which you have to select the optimal 11.

You will have to design a representation to use with a stochastic search algorithm (or algorithms) of your choice. You might want to experiment with different representations, algorithms, and/or the parameters and operators of an algorithm. You will need to carefully consider how to deal with the constraints imposed by the problem.

You need to write a report that documents your approach and identifies the single best solution found by your method. The report will be evaluated according to the following criteria (the marks associated with each are also shown):

**Approach (15 marks)**: Explain the representation(s) used to represent an individual solution, including a description of why you chose this. Also, explain the choice of algorithm(s) (EA, local search, hybrid) you will use to solve the problem. Clearly set out what your investigation covers, for example comparing operators/compare algorithms/..

**Algorithm/Operator Design (25 marks)**: For the algorithm you chose, describe the operators used and any customisation you might have made to an algorithm. Clearly describe any custom operators designed and the rationale behind their design. If you use the off-the-shelf operators provided in DEAP without modification you should state which ones used but you don't need to describe how they work. However you should describe why you chose them.

**Experimental Design & Analysis (25 marks)** : describe any experiments conducted to test version(s) of your algorithm. Give sufficient detail that experiments could be reproduced, including parameter settings. Present results of experiments in an appropriate form using graphs and/or tables, and use statistical significance testing where appropriate and give a commentary that describes results/highlights interesting findings.

**Solution Quality (10 marks)**: you must include the points score of the best solution found, and the associated cost. If it breaks any constraints, you must also state this in your report. *You must check the best solution found using the **constraint_checker()** function supplied and include a screenshot of the output from this as an appendix to your report.*

**Evaluation (15 marks)**: provide a reflective commentary on the results obtained and the approach taken, highlighting any strengths or weaknesses and making suggestions for future work.

**Clarity/overall style (10 marks)**: the document should be written in a scientific style in the format of an academic paper. Be careful to label graphs/tables clearly (including axes on graphs) and use references where appropriate. The report should be a maximum of 6 pages, using Arial font, minimum size 11. **Do not** use any appendices other than the single one required with the screen shot of the checker function.

A marking rubric included below

| | <40 | 40-50 | 50-60 | 60-70 | 70+ |
|---|---|---|---|---|---|
| Approach | Inappropriate choices; no explanation provided/explanation incorrect; no clear questions or plan defined | adequate choices of method and/or representation; some basic attempt at explanation but partially incorrect/choices that do not demonstrate understanding of ECO concepts | Good choice of algorithm/representation; good attempt at explanation but missing some depth or has some inaccuracies; choices demonstrate understanding of core material | Very good approach with explanations that demonstrate v. good understanding of material, but perhaps missing some detail or depth | Excellent approach, very clearly justified, demonstrating excellent understanding of core concepts of course, perhaps going beyond taught material |
| Algorithm and Operator design and customisation | Algorithm does not produce a solution; design shows lack of understanding of basic concepts | Uses off-the-shelf operators/algorithms with no attempt at developing any customisation to make methods suitable to solving the problem | Some attempt to provide customisation of at least one operator to enforce constraints | Very good attempt to adapt algorithm to problem taking account of constrained nature; may | Excellent; bespoke customisations that demonstrate excellent understanding of the nature of the problem |

| | | | | | |
|---|---|---|---|---|---|
| | | | | have customised multiple operators or shown very good insight into particular operator design | and deep insights into the algorithm; may draw on literature not covered in the course |
| Experimental design and analysis | Insufficient experiment; poor presentation of results, no analysis | Experimental design meets minimum requirements, but very limited in scope or ha some flaws; presentation adequate but lacks statistical analysis | Good design, that includes some experimentation of parameters/methods. Presentation could perhaps be improved; at least some attempt at basic statistical analysis provided | Very good design that covers a range of factors; results well presented, appropriate use of statistics | Excellent design, wide ranging or very thorough investigation; thoughtful presentation or results, thorough use of statistics, excellent analysis |
| Solution quality | No solution given | Solution provided but breaks most/all of the constraints | Solution provided but breaks a few constraints, at top end of this range, solutions are very close to being feasible | Solution doesn't break any constraints, good quality re total point score | Excellent quality of solution re total point score, does not break any constraints |
| Evaluation and future work | None provided or demonstrates significant misunderstandings | minimal reflection; suggestions for future work either minimal or flawed | Provides some reflection with some attempt to highlight strengths/weaknesses with at least one sensible suggestion for future work but perhaps not well linked to weaknesses | Very good and critical reflection that shows insight into domain and methods. Focused suggestions for future work linked to weaknesses | In-depth reflection, shows deep insights in both methods and the domain, with excellent suggestions for future work that address highlighted issues |
| Clarity/overall style | Below standard expected at this level | adequate | good | Very good | excellent |

## Technical Information

You are provided with a program in Python that has some basic functions:

- It reads in the data file and creates some useful arrays, such as lists of strikers/goalkeepers etc, as well as a constraint checker function.

- You must provide a screen shot of the output of the *constraint_checker()* function run on your best solution in your report:
    - The constraint checker function is passed a binary array of length 523, where each bit is set to 0 or 1 to indicate if the player at bit *i* is included in the team. *You can represent a solution in any way you want in order to solve this problem. If not using a binary representation, you will need to write your own checker to suit your representation.* **However, you must convert your final solution to the form stated above in order to check the constraints of the final solution you produce and provide its output in your report**. **<u>If you do not provide this, you will not get any marks for Solution Quality</u>.**
    - I may rerun the checker on your solution – therefore you should not sort the input file provided in any way as I will apply the checker to the data read from the file provided.