

All in Good Team: Optimising Team Personalities for Different Dynamic Problems and Task Types

Soo Ling Lim^{1,2} and Peter J. Bentley^{1,2}

¹Braintree Research Lab, Braintree Limited, United Kingdom

²Department of Computer Science, University College London, United Kingdom
s.lim@cs.ucl.ac.uk

Abstract

Change is inevitable in this fast-moving world. As the environment and people's needs continuously change, so must the project. In our previous work, we developed an agent-based model of human collaboration that incorporates individual personalities. In this work, we applied a genetic algorithm to select the optimal personality combinations of a team in order to cope with different types of project change. We studied change in the context of three types of tasks: disjunctive (team performance is the performance achieved by the best performing individual), conjunctive (team performance is the performance achieved by the worst performing individual), and additive (team performance is the total performance of the group). Results reveal that different compositions of team personalities are suitable for different dynamic problems and task types. In particular, optimal personalities found for static problems differ from optimal personalities found for dynamic problems.

Introduction

Dynamically changing problems are a fact of life. Teams of people face change every day. Software developers frequently find themselves chasing a moving target as clients change their minds about features to be implemented (Lim & Finkelstein, 2011). Architects and builders must change their solutions as regulators decide there have been infringements of rules. Doctors must modify their treatments as illnesses follow unexpected courses. Engineers must change their processes as new technologies emerge. Companies must alter their products as markets change. People are remarkable in their adaptability and ability to cope with such change. Yet not all people are suited for all types of change. In every team, some may be better suited to cope with certain forms of change compared to others, and this may be related to their personalities.

In a pioneering study of group processes and productivity, Steiner (1972) found that tasks can be classified based on how individual contributions of members of a group are combined. For example, in *disjunctive* tasks, team performance is the performance achieved by the best performing individual (e.g., mathematicians proving a theorem), in *conjunctive* tasks, team performance is the performance achieved by the worst performing individual (e.g., a factory assembly line), and in *additive* tasks, team performance is the total performance of the group (e.g., a relay race or tug of war).

In our previous work, we developed an agent-based model of human collaboration and studied the effectiveness of different personalities at solving problems with different levels of noise (Lim & Bentley, 2018). We also used the model to investigate the effects of diversity in background and personality on team performance (Lim & Bentley, 2019).

In this work, we hypothesise that different combinations of personalities are needed for different dynamic problems and task types. We modify the model in order to simulate dynamic problems and types of task, and add a genetic algorithm in order to optimise the best team personalities to solve each task type and dynamic problem.

The rest of the paper is organised as follows. The next section describes the background, the section after that describes the agent-based model of human collaboration. Then we describe the modifications we made to the model for this work, followed by the experiments, results and conclusions.

Background

Agent-based models have been used to study the effects of human personalities in collaborative work, such as termites gathering food (Salvit & Sklar, 2012), ant colony (Ahrndt et al., 2015) and crowd movement (Durupinar et al., 2011). In these models, each agent is provided with a human personality, which determines how it behaves and interacts with other agents.

Agent-based models have been optimised in previous work using genetic algorithms (GAs). Heppenstall et al. (2007) used a GA to optimise an agent-based model of a retail market. They model petrol stations as agents and integrated additional system behaviour through the use of spatial interaction model. A GA is then used to optimise the model, producing results that match those derived by expert analysis through rational exploration.

Wang et al. (2009) used a GA to optimise partner selection for virtual enterprises that reduces their collaboration time and cost. They consider three types of collaborations (logistics, information transmission and capital flow) and two task allocation scenarios (allocating all tasks to one partner and allocation each task to different partners).

Knoester et al. (2013) used the AVIDA platform (Lenski et al., 2003) to study the evolution of consensus, a cooperative behaviour in which members agree on information sensed in

their environment. They used a form of evolutionary computation where a population of digital organisms is subject to instruction-level mutations and natural selection, and placed them into groups with fitness determined by their ability to perform consensus. Their experiments found that while genetic heterogeneity within groups increases the difficulty of the consensus task, a surprising number of groups were able to evolve this cooperative behaviour.

Lim and Bentley (2019) used a GA to investigate the effects of differing initial knowledge within team members as they collaborate on a shared task. The GA is used to evolve the optimal starting positions of each team member (representing their initial idea about the solution) and diversity is measured as the distance between their starting positions. The experiments found that diversity in team members' initial knowledge improves team performance, although teams with diverse personalities are more resilient to effects of diversity.

In this work we model teams collaborating to solve dynamically changing problems. Most research on dynamic optimisation has so far concentrated on tracking the moving optima as closely as possible (Jin, 2011). In practice, this is costly, if not impossible. To address dynamic optimisation problems more practically, Jin et al. (2013) introduced an optimisation algorithm that aims to find an acceptable (optimal or sub-optimal) solution that changes slowly over time, rather than the moving global optimum. A local approximation model is constructed using its neighbouring historical data in the database to estimate a solution's past performance.

In summary, despite related work touching on many aspects of this topic, there do not exist any agent-based models that attempt to understand which combinations of personalities are optimal for dynamic problems or for different kinds of task. Given that evolutionary algorithms have been shown to be effective at optimising agent-based models, we use this approach to optimise the agent-based model in this work.

Agent-based Model of Collaboration

The model used in this work, proposed in (Lim & Bentley, 2018), is designed to model human behaviour as team members collaborate to solve a shared task. The agent-based model uses a unique variant of particle swarm optimisation (PSO) to simulate the differing behaviours of people according to their personalities. The model has the following key abstractions:

- **Problem.** The shared goal of all agents is abstracted as the shared task to optimise a function (i.e., find the values of \mathbf{x} such that $f(\mathbf{x})$ is maximised).
- **Agent psychology.** The current “mental state” of each agent is modelled by giving it a position in the solution space (denoting the solution its mind has found so far), a velocity vector (denoting the direction and speed of its thought process), and acceleration vectors (representing the force of ideas and influences that modify the direction and speed of thought), the latter determined by its personality (behaviour algorithm).
- **Agent communication.** The distribution of information between agents is modelled as they each try to solve the

same problem. The exact type of information perceived by each agent and its use is determined by its personality.

Figure 1 shows the algorithm of the model, and the following sections describe each component in detail.

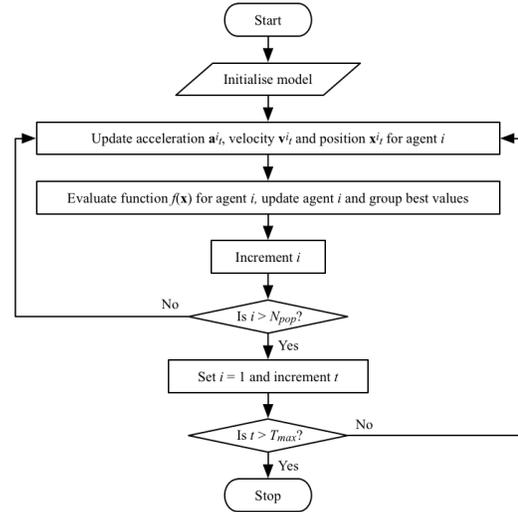


Figure 1: Algorithm of the model (Lim & Bentley, 2018).

Initialise

The model is initialised with:

- a problem space $\mathbf{D} \in \mathbb{R}^n$
- an objective function $f(\mathbf{x})$
- the number of timesteps T_{max} to run the model
- a population of agents N_{pop} , each agent $i \in \{1, \dots, N_{pop}\}$ is initialised with:
 - a personality type \mathbf{P}^i (one of the MBTI personality types, defined by a corresponding genotype (see later))
 - a random position $\mathbf{x}^i_0 \in \mathbf{D}$: $\mathbf{x}_{min} \leq \mathbf{x}^i_0 \leq \mathbf{x}_{max}$
 - a random velocity $\mathbf{v}^i_0 \in \mathbb{R}^n$: $-\mathbf{v}_{init} \leq \mathbf{v}^i_0 \leq \mathbf{v}_{init}$
 - personal best $f^i_{best} = f(\mathbf{x}^i_0)$
- Group best for timestep 0, $f^g_{best_0}$ (see later)

Update

For each timestep $t \in \{1, \dots, T_{max}\}$, each agent i 's position \mathbf{x}^i_t is updated using Equation 1:

$$\mathbf{x}^i_t = \mathbf{x}^i_{t-1} + \mathbf{v}^i_t \quad (1)$$

with the velocity \mathbf{v}^i_t calculated using Equation 2:

$$\mathbf{v}^i_t = \mathbf{v}^i_{t-1} + \mathbf{a}^i_t \quad (2)$$

If $|\mathbf{v}^i_t| > \mathbf{v}_{max}$, it is scaled to equal \mathbf{v}_{max} , in order to prevent excessive speed (an individual with high velocity would literally become too “set in their ways” and would find it impossible to change its direction of thought into a useful direction).

Acceleration \mathbf{a}^i_t is used to change the direction and speed of thought, as determined by the agent's personality – one of the

16 MBTI personality types (Myers, 1962). The interpretation provided here is designed to enable each personality to have an equally good chance of finding the solution. Interpretations were created to represent MBTI personality types appropriately and were not tuned in order to achieve any specific result in later experiments.

The MBTI consists of 16 personality types based on a person’s preferences on four opposing dichotomies: Extraversion (E) – Introversion (I), Sensing (S) – Intuition (N), Thinking (T) – Feeling (F), and Judging (J) – Perceiving (P) (Myers, 1962). J – P defines the person’s preferred manner (either S–N or T–F) of dealing with the outer world. Each personality type has a dominant Jungian function (more developed) supported by an auxiliary Jungian function (less developed) as shown in Table 1.

Table 2 defines the Jungian attitude, and perception and judgment functions. Each attitude (extraversion and introversion) is used as a source of information for each function (Thinking, Feeling, Sensing, Intuition), resulting in Jung’s eight psychological types: extraverted Thinking (Te), introverted Thinking (Ti), extraverted Feeling (Fe), introverted Feeling (Fi), extraverted Sensing (Se), introverted Sensing (Si), extraverted iNtuition (Ne), introverted iNtuition (Ni).

The Jungian intuitive functions (Ne and Ni) includes the notion of intuiting solutions, i.e., from sparse data they interpolate missing information, sometimes resulting in remarkable predictions (and sometimes not). This is modelled through a Gaussian process regression function which builds, from the data available to the agent, an internal imaginary view of the solution space for that agent. The agent then samples its imaginary space and is attracted to the area that it “believes” is a maximum. The Gaussian Progress Regressor from the Scikit-learn Python library (implemented based on Algorithm 2.1 in Rasmussen and Williams (1996)) is used with default options.

Acceleration \mathbf{a}^i_t is calculated using Equation 3:

$$\mathbf{a}^i_t = \mathbf{a}_J^i_t + \mathbf{a}_P^i_t \quad (3)$$

where $\mathbf{a}_J^i_t$ is the judging acceleration is calculated using Table 3 and $\mathbf{a}_P^i_t$ is the perceiving acceleration calculated using Equation 4:

$$\mathbf{a}_P^i_t = \sum_{j=1}^3 r_j (\mathbf{c}_j - \mathbf{x}^i_{t-1}) \quad (4)$$

where $r_1 = 0.5$, $r_2 = 0.3$, and $r_3 = 0.2$, and \mathbf{c}_1 , \mathbf{c}_2 and \mathbf{c}_3 are the top 3 candidates derived using Table 4 with $f(\mathbf{c}_1) \geq f(\mathbf{c}_2) \geq f(\mathbf{c}_3)$. In both tables, agent i ’s neighbours are defined as the five nearest agents to agent i measured by Euclidean distance, i.e., the peer group of each agent comprises those who share similar ideas to the agent. To ensure that the auxiliary component plays a lesser role compared to the dominant component, $\mathbf{a}_J^i_t$ is scaled down such that $|\mathbf{a}_J^i_t|^2 = \frac{|\mathbf{a}_P^i_t|^2}{2}$ if $|\mathbf{a}_J^i_t|^2 > \frac{|\mathbf{a}_P^i_t|^2}{2}$ (if \mathbf{P}^i has dominant perception and auxiliary judgment, otherwise vice versa).

Type	ISTJ	ISFJ	INFJ	INTJ
Dominant	Si	Si	Ni	Ni
Auxiliary	Te	Fe	Fe	Te
Type	ISTP	ISFP	INFP	INTP
Dominant	Ti	Fi	Fi	Ti
Auxiliary	Se	Se	Ne	Ne
Type	ESTP	ESFP	ENFP	ENTP
Dominant	Se	Se	Ne	Ne
Auxiliary	Ti	Fi	Fi	Ti
Type	ESTJ	ESFJ	ENFJ	ENTJ
Dominant	Te	Fe	Fe	Te
Auxiliary	Si	Si	Ni	Ni

Table 1: Myers-Briggs Type Table Showing the 16 Personality Types, with Dominant and Auxiliary Functions (Myers, 1962).

Attitude	Extraversion: Directs perception and judgment on outer world of people and things.
	Introversion: Directs perception and judgment on outer world of people and things.
Perception	Sensing: Concrete perception, finds interest in actualities (made aware directly through the senses), prefers to rely on objective, concrete facts.
	Intuition: Abstract perception, finds interest in connecting concepts and drawing parallels (made aware indirectly by way of the unconscious).
Judgment	Thinking: Analyses and determines the truth or falseness of information in an impersonal fashion, comes to conclusions based on a logical process, aimed at an impersonal finding (facts and ideas).
	Feeling: Person-centred assessment, comes to conclusions based on a process of appreciation, giving things a personal, subjective value.

Table 2: Jungian Attitudes, Perceptions, and Judgments (Jung, 1923).

Evaluate

Agent i ’s fitness at timestep t is evaluated as $f^i_t = f(\mathbf{x}^i_t)$. The agent’s personal best (f^i_{best}) and group best at timestep t ($f^g_{best_t}$) are evaluated as described in the next section.

Updates to the Model

In this work, we update the existing model by (1) modelling task types, (2) modelling dynamic problems, and (3) using a genetic algorithm to optimise team personalities.

Task Types

We model different task types by calculating group best $f^g_{best_t}$ at timestep t based on each task type as follows:

- **Disjunctive.** Group performance is the performance of its best member, $f^g_{best_t}$ is the best f^i_{best} , $i \in \{1, \dots, N_{pop}\}$. This is the scenario most commonly used in optimisation algorithms.

Function	Implementation
Te: The agent is influenced by its neighbours' best personal best. It accelerates towards its neighbours' best personal best from the previous timestep.	$\mathbf{a}_{Te}^i = \mathbf{x}_{n_{best_{t-1}}}^i - \mathbf{x}_{t-1}^i$ (5) where $\mathbf{x}_{n_{best_{t-1}}}^i$ is agent i 's neighbours' personal best position in the previous timestep that results in the highest $f(\mathbf{x})$, and \mathbf{x}_{t-1}^i is the agent's position in the previous timestep.
Ti: The agent focusses on its own personal best (the outcome of its own thoughts). It accelerates towards its own personal best, with randomness added to enable exploration.	$\mathbf{a}_{Ti}^i = (\mathbf{x}_{best_{t-1}}^i - \mathbf{x}_{t-1}^i) + \varphi$ (6) where $\mathbf{x}_{best_{t-1}}^i$ is agent i 's personal best position in the previous timestep, \mathbf{x}_{t-1}^i is the agent's position in the previous timestep, and φ is a random float in the interval $[-2.0, 2.0]$.
Fe: The agent "identifies with other agent's feelings" and "seeks harmony" by matching its neighbours' average velocity (direction of thought) from the previous timestep and to a lesser extent accelerates towards its neighbours' best personal best from the previous timestep.	$\mathbf{a}_{Fe}^i = \omega_1 \cdot \bar{\mathbf{v}}_{N_{t-1}}^i + \omega_2 \cdot \mathbf{a}_{Te}^i$ (7) where weights $\omega_1 = 0.8$, $\omega_2 = 0.2$, $\bar{\mathbf{v}}_{N_{t-1}}^i$ is agent i 's neighbours' average velocity in the previous timestep, and \mathbf{a}_{Te}^i is calculated using equation (5).
Fi: The agent "empathises with" its neighbours' ideas by accelerating towards its neighbours' average position from the previous timestep. It also cares about its own personal thoughts, so accelerates towards its own best position.	$\mathbf{a}_{Fi}^i = \omega_1 \cdot (\mathbf{C}_{n_{t-1}}^i - \mathbf{x}_{t-1}^i) + \omega_2 \cdot (\mathbf{x}_{best_{t-1}}^i - \mathbf{x}_{t-1}^i)$ (8) where weights $\omega_1 = 0.8$, $\omega_2 = 0.2$, $\mathbf{C}_{n_{t-1}}^i$ is the centroid (arithmetic mean position) of agent i 's neighbours' positions in the previous timestep.

Table 3: Jungian Judging Functions and Their Use in Calculating Judging Acceleration, \mathbf{a}_j^i (Lim & Bentley, 2018).

Function	Implementation
Se: The agent sees its neighbours' positions and their quality. Candidates are the positions of the agent's nearest neighbours in the previous timestep.	$\mathcal{C}_{Se}^i = \{\mathbf{x}_{n1_{t-1}}^i, \dots, \mathbf{x}_{n5_{t-1}}^i\}$ (9) where $\mathbf{x}_{n1_{t-1}}^i$ is agent i 's first neighbour's position in the previous timestep, and $\mathbf{x}_{n5_{t-1}}^i$ is agent i 's fifth neighbour's position in the previous timestep. The candidates for current and previous timestep \mathcal{C}_{Se}^i and $\mathcal{C}_{Se}^{i_{t-1}}$ are then sorted in the order of decreasing $f(\mathbf{x})$.
Si: The agent remembers all its own previous positions and a few nearby points and their quality. Candidates are the agent's previous path and new points near to their position.	$\mathcal{C}_{Si}^i = \{\mathbf{x}_{0}^i, \dots, \mathbf{x}_{t-1}^i\} \cup \mathbf{P}$ (10) where \mathbf{P} is the set of points near to \mathbf{x}_{t-1}^i . Given $\mathbf{x}_{t-1}^i = (x_1, x_2, \dots, x_n)$, $\mathbf{P} = \{(x_1 + \delta, x_2, \dots, x_n), (x_1 - \delta, x_2, \dots, x_n), (x_1, x_2 + \delta, \dots, x_n), (x_1, x_2 - \delta, \dots, x_n), \dots, (x_1, x_2, \dots, x_n + \delta), (x_1, x_2, \dots, x_n - \delta)\}$ where δ is a random number from a normal distribution $N(\mu, \sigma)$ with $\mu = 1$ and $\sigma = 0.01$. The quality of old solutions is reduced as follows: every solution in the agent's previous path $\{\mathbf{x}_{0}^i, \dots, \mathbf{x}_{t-1}^i\}$ that are more than 10 timesteps old are reduced in quality by decrementing the fitness by 0.001 each timestep. The candidates for current and previous timestep \mathcal{C}_{Si}^i and $\mathcal{C}_{Si}^{i_{t-1}}$ are then sorted in the order of decreasing $f(\mathbf{x})$.
Ne: The agent sees its neighbours' positions and uses them to create an "imaginary solution space". Candidates produced from Se (data from the environment) are used as input to train the Gaussian process regression function. Candidates are then the best quality solutions resulting from sampling this imaginary space.	$f^* = \mathcal{GP}: \text{train}(\mathcal{C}_{Se}, f(\mathcal{C}_{Se})); \text{predict}(\mathcal{C}_{Ne}^i)$ (11) where \mathcal{GP} is the Gaussian process regression function (Williams & Rasmussen, 1996), training on \mathcal{C}_{Se} and $f(\mathcal{C}_{Se})$, and \mathcal{C}_{Ne}^i is a vector of points in \mathbf{D} , sampled every 10 points. The candidates for current and previous timestep \mathcal{C}_{Ne}^i and $\mathcal{C}_{Ne}^{i_{t-1}}$ are then sorted in the order of decreasing f^* .
Ni: The agent sees its own previous positions and a few nearby points and uses them to create an "imaginary solution space". Candidates produced from Si (internal data) are used as input to train the Gaussian process regression function. Candidates are then the best quality solutions resulting from sampling this imaginary space.	$f^* = \mathcal{GP}: \text{train}(\mathcal{C}_{Si}, f(\mathcal{C}_{Si})); \text{predict}(\mathcal{C}_{Ni}^i)$ (12) where \mathcal{GP} is the Gaussian process regression function, training on \mathcal{C}_{Si} and $f(\mathcal{C}_{Si})$, and \mathcal{C}_{Ni}^i is a vector of points in \mathbf{D} , sampled every 10 points. The candidates for current and previous timestep \mathcal{C}_{Ni}^i and $\mathcal{C}_{Ni}^{i_{t-1}}$ are then sorted in the order of decreasing f^* .

Table 4: Jungian Perceiving Functions and Their Use in Getting Candidates (Top Three Candidates Returned as c_1 , c_2 and c_3) (Lim & Bentley, 2018).

- **Conjunctive.** Group performance is the performance of its weakest member, $f^g_{best_t}$ is the worst f^i_{best} , $i \in \{1, \dots, N_{pop}\}$.
- **Additive.** Group performance is a sum of all individual performances, $f^g_{best_t}$ is the sum of all f^i_{best} , $i \in \{1, \dots, N_{pop}\}$.

Each agent’s personal best, f_{best}^i , is calculated as the agent’s best fitness in the last 10 timesteps, i.e., $\max(f_t^i, f_{t-1}^i, \dots, f_{t-9}^i)$. Retaining the memory of all best fitnesses for the entire run is not appropriate for dynamic problems where the optimal solution changes over time.

This work evaluates the performance of groups of agents as they collaborate to solve dynamic problems over time, thus fitness scores are obtained throughout the run. The final group performance f_{best}^{fg} (used as the fitness function for the GA) is calculated using Equation 13:

$$f_{best}^{fg} = \frac{\sum_{t=T_{poll}[0]}^{T_{poll}[4]} f_{best_t}^g}{5} \quad (13)$$

where T_{poll} is a list of timestep values and $f_{best_t}^g$ is the group best at timestep t .

Dynamic Problems

To model dynamic problems, we use a simple two-dimensional problem with a clear gradient as in (Lim & Bentley, 2018). The objective function $f(x, y)$ is described in Equation 14:

$$f(x, y) = -\sqrt{(x - a)^2 + (y - b)^2} \quad (14)$$

where agent i ’s position at time t , $\mathbf{x}_t^i = (x, y)$, the values of a and b are varied over time and $f(x, y)$ is normalised such that $f(x, y) \in [0, 1] : \forall x \in [x_{min}, x_{max}], \forall y \in [y_{min}, y_{max}]$. Figure 2 shows the heatmap and surface plot when $a = 0$ and $b = 0$.

We investigate the following types of change:

- **Static:** This is the baseline scenario with no change, where $a = 0$ and $b = 0$ for the entire duration.
- **Linear:** The position of the maximum moves from left to right on the x -axis. At $t = 0$, $a = -25$ and $b = 0$. The maximum starts at $(-25, 0)$ and a increments by 1 at every timestep, so at $t = T_{max}$, the maximum is at $(25, 0)$, see Figure 3. This models a simple “moving target” problem, for example a design specification that changes over time as the team tries to find the solution.
- **Oscillating:** The position of the maximum moves from left to right on the x -axis and returns to where it started. At $t = 0$, $a = -25$ and $b = 0$, and a increments by 2 at every timestep until $a = 25$ and $b = 0$, then a decreases by 2 at every timestep so at $t = T_{max}$, $a = -25$ and $b = 0$, see Figure 4. This models a problem where best solutions oscillate and repeat, e.g., in trading, sometimes it is good to buy, sometimes it is good to sell.
- **Rotary:** The position of the maximum moves 90° clockwise every 10 timesteps. At $t = 0$, $a = 0$ and $b = 50$, at $t = 11$, $a = 50$ and $b = 0$, at $t = 21$, $a = 0$ and $b = -50$, and so on, see Figure 5. This models a cyclic problem, e.g., designing gifts for different seasons throughout a year.

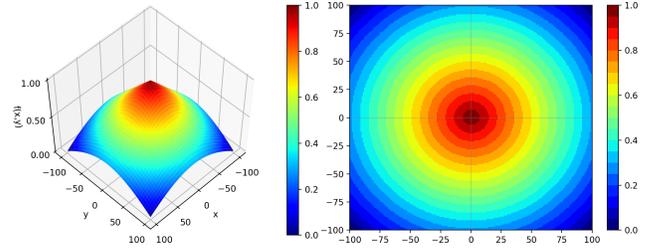


Figure 2: Surface plot (left) and heatmap (right) for normalised Equation 14 with a maximum in $(0, 0)$. Colour ranges from blue (minimum) to red (maximum).

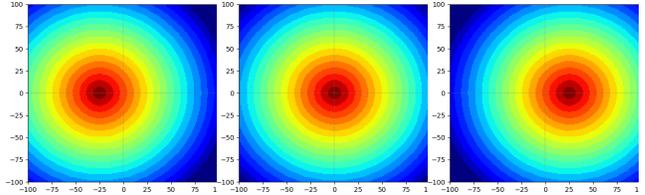


Figure 3: Linear at $t=0, 25$ and 50 .

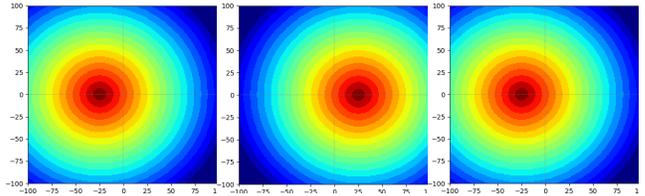


Figure 4: Oscillating at $t=0, 25$ and 50 .

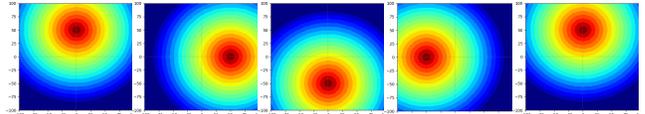


Figure 5: Rotary at $t=0-10, 11-20, 21-30, 31-40,$ and $41-50$.

Genetic Algorithm for Optimising Team Personalities

We used a genetic algorithm to evolve an optimal combination of agent personalities for each dynamic problem. For a team of $n=4$ members as used in the experiments, their personalities are specified by a 16-digit binary genotype. For example, the 16-digit genotype 1100000110001010 corresponds to the phenotype INTJ, ESTP, ISTJ, ISFJ.

A standard canonical GA is used with population size of 20, each individual solution representing the personalities of a team. Each team has four members, for each member of the population, fitness is calculated by decoding the genotype to produce four personalities for the team members. The agent-based model is run 50 times for the team to produce average team performance. The group best at the end of each run is recorded and team performance is measured by their average group best, which is the total group best for all runs divided by total number of runs. Based on fitness, 8 individuals are chosen as parents. 20 child teams are created using single-point crossover from the parents. A single bit flip mutation occurs with a probability of 0.2 per chromosome.

The GA is run for 20 generations. Values were found following preliminary experiments to determine fastest and most effective settings.

Experiments

Our experiments investigate each task type (Disjunctive, Conjunctive, Additive) with each dynamic problem (Static, Linear, Oscillating, Rotary) as illustrated in Table 5.

Static Disjunctive	Static Conjunctive	Static Additive
Linear Disjunctive	Linear Conjunctive	Linear Additive
Oscillating Disjunctive	Oscillating Conjunctive	Oscillating Additive
Rotary Disjunctive	Rotary Conjunctive	Rotary Additive

Table 5: Task types with dynamic problems.

For each experiment, we ran the GA ten times and analysed personalities selected by the GA. The model was initialised with constant settings in Table 6. For each task type and dynamic problem, we measured the number of times each personality is used by the GA to assemble teams. We also measured the average fitness over generations and average fitness over time for each task type and dynamic problem. Finally, we counted the opposing MBTI dichotomies used by the GA to assemble teams, i.e., Extraverts vs. Introverts (E vs. I), Sensors vs. Intuitives (S vs. N), Thinkers vs. Feelers (T vs. F), and Judgers vs. Perceivers (J vs. P).

Constants	Values
T_{max}	50
N_{pop}	4
v_{max}	5.0
x_{min}, y_{min}	-100
x_{max}, y_{max}	100
v_{init}	(1.0, 1.0)
T_{poll}	[10, 20, 30, 40, 50]

Table 6: Constants Settings for The Model

Results

Different personality types are selected at different frequencies as the best team compositions for different tasks and dynamic problems, as illustrated by the heatmap in Figure 6. The GA never chooses teams made from a single personality type, rather, it selects mixtures of different personalities to work in combination. The GA is able to optimise all types of task and dynamic problem, although disjunctive tasks are the easiest to optimise, with the highest average fitness over generations for all types of change, followed by additive tasks and conjunctive tasks (Figure 7). Figure 8 illustrates the average team performance over 50 timesteps for all the ten best teams selected by the GA. Good combinations of personalities are found, resulting in improvement over time by the agents as they solve each type of problem, even the difficult rotary problem, where performance drops when change occurs, but the teams are still able to gradually improve the performance over time (Figure 8).

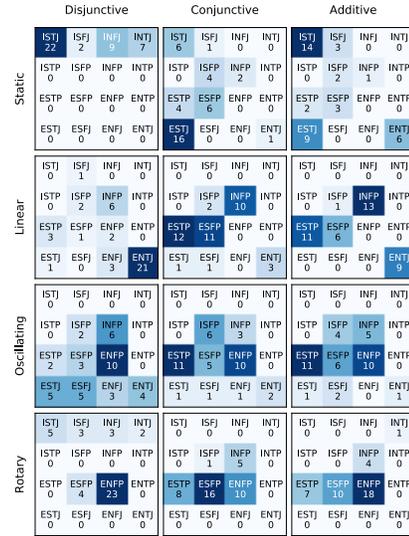


Figure 6: Heatmap of personality type in teams for dynamic problems and task types.

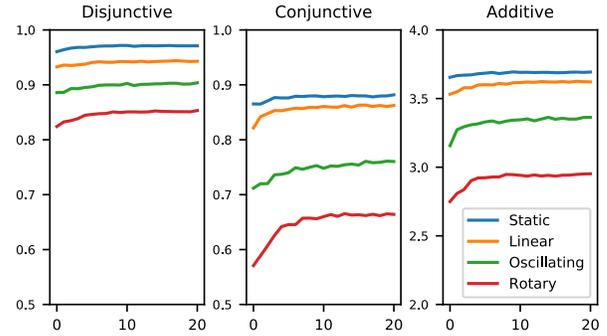


Figure 7: Average fitness over generations for dynamic problems and task types. x-axis is generations, y-axis is ff_{best}^g .

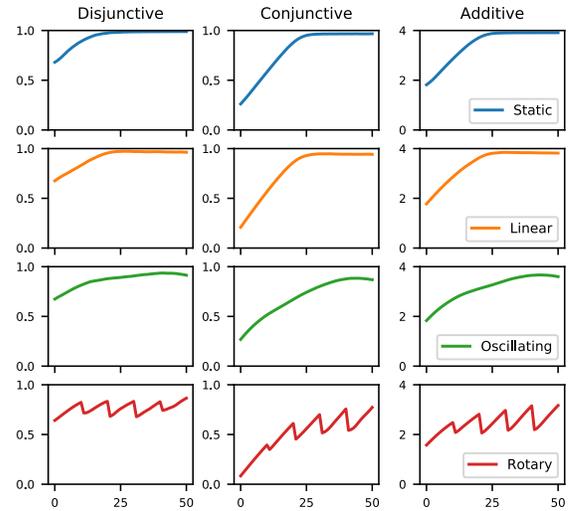


Figure 8: Average team performance over 50 timesteps for all 10 best teams selected by the GA for types of task and change. x-axis is timesteps, y-axis is ff_{best}^g .

Personalities best suited for static problems are different from those suited for dynamic problems (Figure 9). For all static problems, the GA evolved more Introverts compared to Extraverts, with Static Disjunctive exclusively composed of Introverts. For all dynamic problems, the GA evolved more Extraverts compared to Introverts. This is because Extraverted team members communicate more, which is essential when facing a moving target. The finding is supported by literature, where extraverted personalities has been found to be positively correlated to adaptability (Teixeira et al., 2012).

Similarly, for all static problems, the GA evolved teams with more Judgers compared to Perceivers, with Static Disjunctive exclusively composed of Judgers (Figure 9). For all dynamic problems, it evolved more Perceivers compared to Judgers, with Rotary Conjunctive being exclusively composed of Perceivers. Perceivers put more weight on the current state of their environment when making decisions, enabling them to detect and react to change, and management literature has found that Judging individuals prefer to regulate and control, while Perceivers prefer to understand and adapt (Nutt, 1993).

For all static problems, the GA evolved teams with more Thinkers compared to Feelers, but evolved teams with more Feelers compared to Thinkers for all dynamic problems except for Linear Disjunctive (more Thinkers than Feelers) and Linear Additive (equal numbers of Feelers and Thinkers) (Figure 9). This is because Feelers are more influenced by the behaviour of their companions compared to Thinkers. Such personalities have been found in existing research to be more consultative and adaptive to change (Nutt, 1993).

Comparison between Sensors and Intuitives are less clear cut. For all static problems, the GA evolved more Sensors compared to Intuitives, while for dynamic problems, Intuitives are used more frequently than Sensors for all disjunctive tasks and linear and rotary additive tasks (Figure 9). When fitness is determined by using the best solution from any team member, the GA chooses team members that “intuit” the solution space and anticipate where to move. Figure 10 illustrates the solution space as perceived by a team in one run, sampled over time. At $t=50$, team member INTJ (last column row 2) has correctly mapped the solution space, finding four distinct optimal regions corresponding to the locations where the optimal rotates to every 10 timesteps. A study by Allinson et al. (2000) of more than 150 founders of high growth companies found that these founders exhibit higher intuition compared to general population of managers and the intuition has helped them to be quick at identifying and exploiting opportunities. In evolutionary computation literature, gaussian process modelling have been used to improve speed and quality of optimisation (e.g., Büche et al. (2005), Zhang et al. (2010), and Han et al. (2017)). For conjunctive tasks, when fitness of a team depends on the fitness of the worst team member, the GA optimises teams with more Sensors compared to Intuitives.

Finally, in all disjunctive tasks, we found an unusual behaviour by this PSO-based model: instead of team members

converging onto one solution, the team spreads themselves more widely so that members become more likely to catch the moving target as it passes by. This is evident in Figure 11, where all disjunctive tasks have a higher average distance between team members compared to other tasks. In particular, for rotary change, the distance is up to four times more than the other changes, as it is beneficial for team members to be stationed at (0, 50), (-50, 0), (50, 0) and (0, -50).

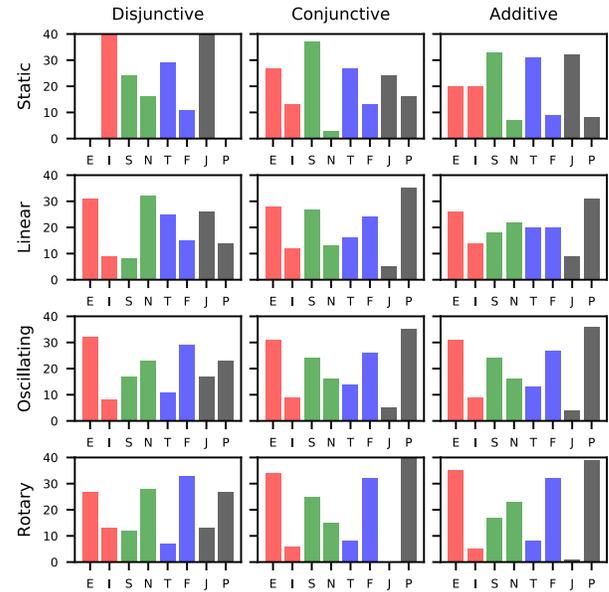


Figure 9: Personality count for task types and dynamic problems in terms of opposing MBTI dichotomy: Extraverts vs. Introverts (E vs. I), Sensors vs. Intuitives (S vs. N), Thinkers vs. Feelers (T vs. F), and Judgers vs. Perceivers (J vs. P). x-axis is the MBTI dichotomies, and y-axis is the personality count.

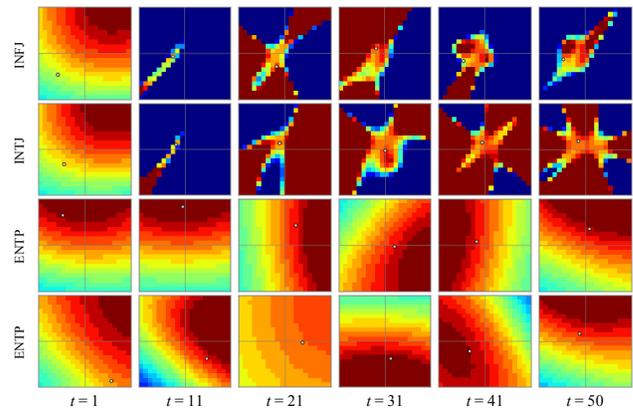


Figure 10: Solution space as perceived by each team member in one of the teams selected by the GA for Rotary Disjunctive, sampled at $t=1, 11, 21, 31, 41, 50$. The circle \circ in each image denotes the agent’s position for that timestep.

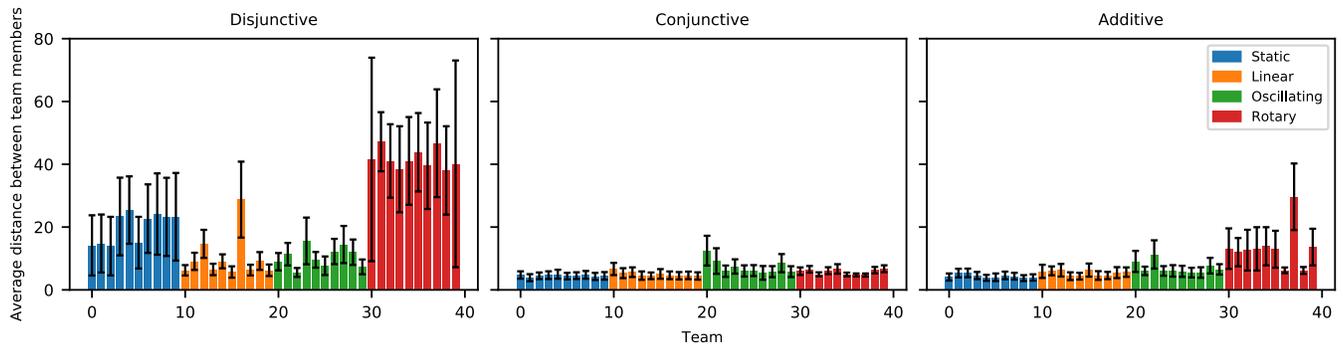


Figure 11: Average distance between team members at $t=50$ for all teams selected by the GA. Error bars represent standard deviation.

Conclusions

Dynamic optimisation is a commonly found class of problem in the real world, and teams of people handle such problems regularly in their working lives. This research used a genetic algorithm to optimise the constituent members of teams as they tackled dynamic problems and task types. The teams were represented by an agent-based model of personality, each corresponding to a different PSO-based behaviour and communication strategy.

We found that different combinations of personalities are selected for different dynamic problems. The GA evolved teams comprising heterogenous personalities, with different combinations of personalities for each type of problem. Introverts, Sensors, Thinkers and Judgers are used frequently by the GA for static problems, while Extraverts, Intuitives, Feelers and Perceivers are used more frequently for dynamic problems. Analysing the movement of the agents over time and their internal representations of the problem, it is clear that Intuitive types learned to “anticipate” repeating solutions, working as a team to spread themselves across the likely good solution areas, rather than behaving in the more typical manner of converging to a single point in the solution space.

References

- Ahrndt, S., Fährndrich, J., & Albayrak, S. (2015). Modelling of personality in agents: from psychology to implementation. *Proceedings of the Workshop on Human-Agent Interaction Design and Models*, pages 1-16.
- Allinson, C. W., Chell, E., & Hayes, J. (2000). Intuition and entrepreneurial behaviour. *European Journal of Work and Organizational Psychology*, 9(1): 31-43.
- Buche, D., Schraudolph, N. N., & Koumoutsakos, P. (2005). Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2): 183-194.
- Durupinar, F., Pelechano, N., Allbeck, J., Gudukbay, U., & Badler, N. I. (2011). How the ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications*, 31(3): 22-31.
- Han, Z.-H., Zhang, Y., Song, C.-X., & Zhang, K.-S. (2017). Weighted gradient-enhanced kriging for high-dimensional surrogate modeling and design optimization. *AIAA Journal*, 55(12): 4330-4346.
- Heppenstall, A. J., Evans, A. J., & Birkin, M. H. (2007). Genetic algorithm optimisation of an agent-based model for simulating a retail market. *Environment and Planning B: Planning and Design*, 34(6): 1051-1070.
- Jin, Y. (2011). Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2): 61-70.
- Jin, Y., Tang, K., Yu, X., Sendhoff, B., & Yao, X. (2013). A framework for finding robust optimal solutions over time. *Memetic Computing*, 5(1): 3-18.
- Jung, C. (1923). *Psychological Types*. Harcourt, Brace, Oxford, England.
- Knoester, D. B., Goldsby, H. J., & McKinley, P. K. (2013). Genetic variation and the evolution of consensus in digital organisms. *IEEE Transactions on Evolutionary Computation*, 17(3): 403-417.
- Lenski, R. E., Ofria, C., Pennock, R. T., & Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936): 139.
- Lim, S. L., & Bentley, P. J. (2018). Coping with Uncertainty: Modelling Personality when Collaborating on Noisy Problems. *Proceedings of the 2018 Conference on Artificial Life*, pages 566-573.
- Lim, S. L., & Bentley, P. J. (2019). Diversity improves teamwork: optimising teams using a genetic algorithm. *Proceedings of the IEEE Congress on Evolutionary Computation*, in press.
- Lim, S. L., & Finkelstein, A. (2011). Anticipating change in requirements engineering. *Relating Software Requirements and Architectures* (pp. 17-34). Springer.
- Myers, I. B. (1962). *The Myers-Briggs Type Indicator: Manual (1962)*. Consulting Psychologists Press, Palo Alto, California.
- Nutt, P. C. (1993). Flexible decision styles and the choices of top executives. *Journal of Management Studies*, 30(5): 695-721.
- Salvit, J., & Sklar, E. (2012). Modulating agent behavior using human personality type. *Proceedings of the Workshop on Human-Agent Interaction Design and Models*, pages 145-160.
- Steiner, I. D. (1972). *Group Processes and Productivity*. Academic Press, New York.
- Teixeira, M. A. P., Bardagi, M. P., Lassance, M. C. P., de Oliveira Magalhães, M., & Duarte, M. E. (2012). Career adapt-abilities scale—Brazilian form: psychometric properties and relationships to personality. *Journal of Vocational Behavior*, 80(3): 680-685.
- Wang, Z.-J., Xu, X.-F., & Zhan, D.-C. (2009). Genetic algorithm for collaboration cost optimization-oriented partner selection in virtual enterprises. *International Journal of Production Research*, 47(4): 859-881.
- Williams, C. K., & Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, pages 514-520.
- Zhang, Q., Liu, W., Tsang, E., & Virginas, B. (2010). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3): 456-474.