22nd International Congress on Modelling and Simulation, Hobart, Tasmania, Australia, 3 to 8 December 2017
mssanz.org.au/modsim2017

An optimal recruitment algorithm based on an efficient tree search policy

P. Lalbakhsh [a], **A. Novak** [b] **and T. Caelli** [c]

[a] *La Trobe University, Melbourne, VIC 3086, Australia*
[b] *Defence Science and Technology Group, Melbourne, VIC 3207, Australia*
[c] *Royal Melbourne Institute of Technology, Melbourne, VIC 3001, Australia*
Email: plalbakhsh@students.latrobe.edu.au

Abstract: In the area of highly specialized skills training where the cost of training is high and available infrastructure is limited and limiting, recruitment and manpower training scheduling can be quite complex. For Royal Australian Navy (RAN) pilot training, there are some additional unique features such as feedback loops that are generated by the requirement for graduated pilots to return into the training continuum as instructors. Furthermore, the trainee numbers are relatively small, the failure rates are high and highly variable. In this paper, we consider a simplified RAN training scheduler solution as an optimal control problem having feedback loops and cost functions that penalize prolonged waiting periods between courses (buffers) with an overall objective of minimizing the total training time. The solution algorithm converges on an optimal recruitment strategy through which the training continuum maintains a functional squadron over a specific timeframe, while imposing the least possible cost to the organisation. Solutions also take into account course pass rates, squadron wastage, the number of trainees in each course or the number of trainees waiting in the intermediate buffers. The proposed algorithm uses states and actions as used in Markov Decision Processes (MDP) where current states and actions are used to predict new states to minimize cost. The algorithm differs from MDPs in so far as the MDP "optimal policy" for prediction future states and associated optimal actions is replaced by an optimal tree search process where traversing a level in the tree is interpreted as taking an action resulting in a transition from one state to another. The algorithm uses a recruitment-wastage near-equilibrium condition to prune the tree avoiding suboptimal solutions. To select the best recruitment strategy, the combined cost from root-to-leaf is considered as the final merit thus replacing the *stochastic MDP* policy approach with a *deterministic optimal tree search* strategy. The algorithm benefits from a solution archive that maintains a sorted list of the best *n* created solutions. The result of the experiments show that the algorithm can efficiently perform tree search in order to rapidly find feasible recruitment policies with optimal costs.

Keywords: *Decision Support System (DSS), dynamic systems, Markov decision process (MDP), optimal recruitment, optimization*

1. INTRODUCTION

The algorithm investigated in this paper is a part of a larger decision support system for manpower planning and tracking in Royal Australian Navy (RAN) aviation. It provides an optimal recruitment policy such that squadron functionality is sustained consistent with the aim of optimal workforce planning: devising strategies that provide the required number of qualified personnel at the specified time at minimal cost (Wang, 2005). Given variable failure rates, squadron attrition rate, course limitations, limited number of instructors and different types of costs in the continuum, this problem can be addressed as a dynamic multi-objective optimization problem.

Workforce planning (WP) and optimized recruitment is a common problem in different sectors, from health (Auerbach, Buerhaus, & Staiger, 2007) to education (Howson & McNamara, 2012) and from corporate workforce planning (Parrish, Freudenberg, Scsigulinsky, & Cho, 1995) to military (Wang, 2005). Different solutions have been proposed that are typically customized for each sector according to its characteristics and the predefined goals of clients and decision makers. Markov models (Zais & Zhang, 2016), simulations (Hertz, Cavalieri, Finke, Duchi, & Schönsleben, 2014), solutions based on system dynamics (Cave & Willis, 2016) and optimization strategies (Harper, Kleinman, Gallagher, & Knight, 2013) are the dominant types of formulations used to address the WP problem. However, translating these formulations into realistic and situated optimization problems requires the articulation of domain constraints, choice of appropriate algorithms and their scalability (Wang, 2005).

Here we present an algorithm that generates an optimal recruitment policy such that squadron functionality is sustained. This algorithm is scalable for this case of Royal Australia Navy aircrew training. As for implementation, the algorithm has been developed in Java to allow easy integration with decision support systems in Australian Navy such as Timetabling for Australian Navy Training Authority Aviation (TA-AVN) (Bayliss, Novak, Nguyen, Caelli, & Harrison, 2016) and the Agent-Based Discrete Event Simulator for Aircrew Training (ATHENA) (Nguyen, Shokr, Novak, & Caelli, 2016).

The algorithm extends standard tree search backtracking to include MDP state and action variables. Starting from an initial state which represents the initial student numbers waiting in buffers and the size of the squadron, the algorithm creates potential subtrees (while also pruning those that cannot provide cost effective solutions) for each term of the training program. The objective is to find actions as recruitment policies that create branches in the search tree which guarantee and maintain squadron workforce pool at minimal cost. The advantage of this algorithm is that it provides stable and scalable solutions as it emerges over time (terms) and has an inherent backtracking property for more costly branches. For performance evaluation, the algorithm was run on a simplified model of the Royal Australian Navy pilot training continuum and the results show that our strategy is able to find low cost solutions to stabilize the squadron through the problem search space over a reasonable time.

2. RAN PILOT TRAINING RECRUITMENT PROBLEM

The RAN pilot training continuum can be represented as a linear sequence of courses providing specialised training to prepare students to serve in respective squadrons upon graduation and Figure 1 shows the simplified version considered in this paper. This model has enough in common with the real-world operational structure to serve as an exemplar of the optimisation methodology ultimately to be implemented in the full continuum. As shown in Figure 1, this simplified continuum has two different courses, three student waiting buffers and one squadron. *Course 1* is assumed as the perquisite course for *Course 2* and only students who have passed C*ourse 1* can be enrolled in *Course 2*. The course structure and dynamics are investigated with the aim of optimising the recruitment for achieving the squadron capability at a minimum cost. Each course has its own associated cost and its own infrastructure limits which dictate the maximum number of students who can enroll in the courses during a term. Students who cannot enroll in a course have to wait in buffers which have associated waiting costs. Finally, students who have successfully passed both courses, typically wait in the squadron buffer to be recruited by the squadron. The instructor feedback loop shown in Figure 1 refers to the fact that graduated pilots can be posted to *Course 1* and *Course 2* as instructors. Student failure rates, squadron wastage ratio, and the feedback loop are some of the characteristics that makes the prediction challenging since the objective is to estimate how many students should be recruited to supply the right number of graduates to the squadron. An insufficient number of graduates compromises the squadron and imposes an additional cost to the system for hiring external instructors that the squadron cannot provide. On the other hand, having too many graduates is also a problem, since the organization has to support personnel who are not operational.

3. TERMINOLOGY AND FORMALISM

The terminology and formulation presented in this paper is based on the simplified training continuum model presented in Figure 1. Each course c_i is defined by three parameters: N_{c_i}, C_{c_i} and f_{c_i}, where N_{c_i} is the infrastructure limit referring to the maximum numbers of students that can be enrolled in c_i per term, C_{c_i} is the per-student training cost and f_{c_i} refers to the failure rate at c_i. The course failure distribution has been modelled with a beta-binomial enabling us to control the location and spread independently. This allows us to explore the effects of different failure rates on policies, costs, and finally the sustainability of the squadron (Rosner, 2005).

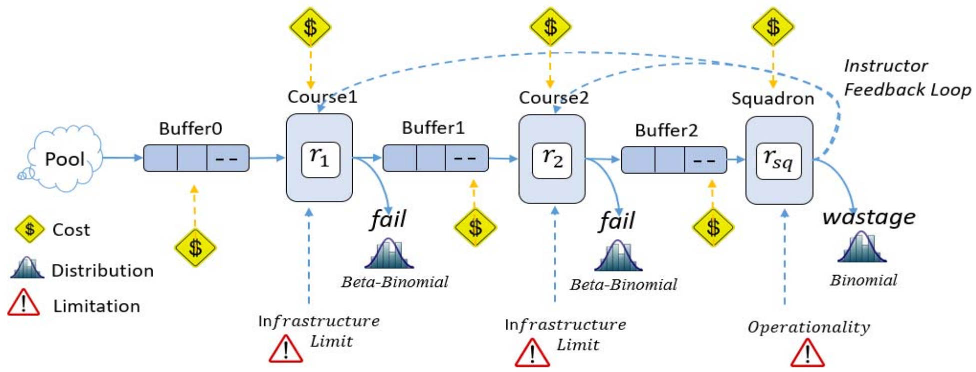

Figure 1. A simplified model of the Royal Australian Navy pilot training continuum.

Each c_i is deterministically connected to its input buffer b_{i-1} and to its output buffer b_i. The recruitment policy for each c_i is denoted as r_i. At the start of each term, *t*, r_i students from buffer b_{i-1} enroll into course c_i. The upper bound for the number of enrolments for c_i per term is computed as:

$$r_i < \min\left(N_{b_{i-1}}, N_{c_i}\right), i \in \{1,2\}, \quad (1)$$

where $N_{b_{i-1}}$ is the number of students waiting in buffer b_{i-1} and N_{c_i} is the maximum number of students that can enroll in course c_i each term. Buffers work in a first-in-first-out (FIFO) fashion and there is no maximum size for the number of students each buffer can maintain. However, there is a positive cost associated with each student waiting in buffer *i* which is denoted as C_{b_i}.

Finally, the squadron is represented as $sq(\omega, C_{sq}, f_{sq})$, where ω refers to the minimum number of personnel required to sustain the squadron functionality, C_{sq} is the per-pilot cost and f_{sq} is a percentage of squadron population that is lost due to attrition each term.

We will use N_{sq}^t in the formulation to refer to the size of the squadron at term *t*. It must be noted that all costs for buffers, courses and the squadron are computed as per-term costs. There are other parameters that contribute to the functionality of the continuum such as the number of instructors N_{int} hired from the squadron (instructor feedback), and the number of external instructors N_{ext}. External instructors are hired when there are not enough pilots to serve as instructors. They add to the total cost which is to be minimized and are therefore used in the process to select the optimal solution path through the training continuum search tree structure.

4. SYSTEM MODELLING

In order to have a realistic and easily understandable model, we borrowed three key variables from Markov Decision Process (MDP) models: *states, actions and rewards (costs) for a selected action at a given state - leading to an optimal recruitment policy.* This enables us to describe the condition of the training continuum via state/action diagrams and evaluate the aggregated consequences of different recruitment policies over the complete continuum. The details of this simplified training continuum is depicted in Figure 2a. The recruitment policy, adopted by courses and the squadron at term, *t,* can be represented by an action at a given time $a_t(r_1^t, r_2^t, r_{sq}^t)$ that has an associated cost. Here r_1^t and r_2^t refer to the number of recruits to course c_1 and c_2, respectively, and r_{sq}^t is the number of recruits to the squadron at term *t*. Similarly, state $s_t(N_{b_0}^t, N_{b_1}^t, N_{b_2}^t, N_{sq}^t)$ refers to the status of the buffers, trainees enrolled in courses and the size of the squadron at term *t*. $N_{b_0}^t$, $N_{b_1}^t$ and $N_{b_2}^t$ are students waiting in b_0, b_1 and b_2 respectively. N_{sq}^t is the number of pilots available in the squadron.

Given s_t and a_t, we can compute the cost of the training continuum for term *t*, the number of students which passed courses successfully, the number of students waiting in buffers, the next state of the continuum and the size of the squadron. We refer to the combination of s_t and a_t using operator ⊗ as formulated in (2):

$$\mathrm{X}_{s_t}^{a_t} = s_t \otimes a_t \tag{2}$$

$\mathrm{X}_{s_t}^{a_t}$ is referred to as a composite of s_t and a_t, which represents the transition of the continuum from term t to term *t+1*. Given state s_t and action a_t, composite $\mathrm{X}_{s_t}^{a_t}$determines the cost of the continuum, updates buffer sizes, computes the size of the squadron and finally determines the next state at term *t+1*. Fig. 2(b) diagrammatically shows the transition resulted by the ⊗ operator. Calculation of the total cost for $\mathrm{X}_{s_t}^{a_t}$ is performed using (3):

$$Cost\left(\mathrm{X}_{s_t}^{a_t}\right) = Cost\ (s_t) + Cost\ (a_t) + ExcessCost(N_{int}) + Cost(N_{ext}) + \textstyle\sum_{i=1}^{2} ExcessCost(b_i) \tag{3}$$

where $Cost\ (s_t)$ refers to the cost for state s_t and is computed as:

$$Cost\ (s_t) = Cost\left(N_{b_1}^t\right) + Cost\left(N_{b_2}^t\right) + Cost\left(N_{sq}^t\right) \tag{4}$$

$Cost\left(N_{b_1}^t\right)$ and $Cost\left(N_{b_2}^t\right)$ refer to the cost for students waiting in buffers $b_1$and b_1 computed as:

$$Cost\left(N_{b_i}^t\right) = N_{b_i}^t \times \left(C_{c_i}^t \times \beta_i\right) \tag{5}$$

Where $N_{b_i}^t$ is the number of students waiting in b_i, $C_{c_i}^t$is the per-trainee cost for c_i and β_i is the cost increase factor for trainees held in b_i.

$Cost\left(N_{sq}^t\right)$ is the cost of the squadron at term *t* computed as:

$$Cost\left(N_{sq}^t\right) = N_{sq}^t \times C_{sq}^t \tag{6}$$

where N_{sq}^t is the size of the squadron at term *t* and C_{sq}^t is the per-pilot cost.

$Cost\ (a_t)$ in (3) refers to the cost of action a_t adopted at term *t* which is computed as:

$$Cost\ (a_t) = r_{sq}^t \times C_{sq}^t + \textstyle\sum_{i=1}^{2}\left(r_i \times C_{c_i}^t\right) - \left(r_2 \times \left(C_{c_1}^t \times \beta_1\right) + r_{sq}^t \times \left(C_{c_2}^t \times \beta_2\right)\right) \tag{7}$$

which is the cost of recruiting pilots to the squadron and trainees to courses, minus the waiting cost of recruits who leave buffers b_1 and b_2.

$ExcessCost(N_{int})$ in (3) is the cost of the available internal instructors, beyond the requirements of the given action a_t and $Cost(N_{ext})$ is the cost of hiring external instructors, if required.

Finally, $\sum_{i=1}^{2} ExcessCost(b_i)$ in (3) computes the excess cost of buffers b_1 and b_2.

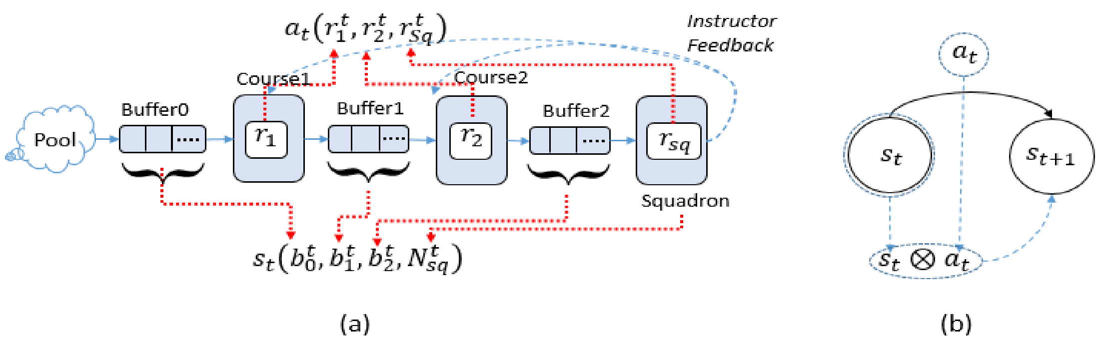

Figure 2. (a) Representation of the anatomy of the training continuum in *state* and *action* format. **(b)** a sample state diagram for the transition of s_t to s_{t+1} resulted by the ⊗ operator.

Although the cost of a composite is an important factor in finding feasible solutions for RAN pilot training continuum, the main goal of the algorithm is to maintain squadron sustainability through a finite number of state transitions with an optimum cost. given $\mathrm{X}_{s_t}^{a_t}$ as the transition enabler from s_t to $s_{t+1}$by action a_t, we define a solution Γ as a finite series of valid composites that results in a sustainable squadron. Each solution $\Gamma = \{\mathrm{X}_{s_0}^{a_0}, \ldots, \mathrm{X}_{s_k}^{a_k}\}$ is refered to as a trail and the sustainability of the squadron is met if Γ contains a sub-set of stabilizing composites. A stabilizing composite creates a feedback that directs a trail towards a circular state/action selection which in-turn guarantees that the squadron maintains enough personnel at each term. Given state s_k and action a_k, a stabilizing composite in Γ satisfies the following condition:

$$s_k \otimes a_k = s_{k-h} \,, 0 \leq h \leq k \tag{8}$$

where k is the length of Γ.

For every scenario, a set of solutions might be created. Let Γ^* be a set of all possible trails, the optimal trail Γ^+ is defined as a trail that involves the minimum cost calculated according to (3). Therefore, the goal of our algorithm is to find the optimal trail Γ^+ as:

$$\Gamma^+ = arg_{min}\big(Cost(\Gamma_i)\big),\ \Gamma_i \in \Gamma^* \,, 0 < i \leq |\Gamma^*| \tag{9}$$

where $\Gamma^+ \in \Gamma^*$.
$Cost(\Gamma_i)$ in (9) refers to the cost of the i^{th} trail as the combined cost of its composites which is computed as:

$$Cost(\Gamma_i) = \sum_{j=1}^{k} Cost\left(X_{s_j}^{a_j}\right), k = length\ (\Gamma_i) \tag{10}$$

Fig. 3 shows two possible types of trails that can be encountered as valid solutions according to (8). Fig 3.a shows how a single state can lead to a sustainable squadron by choosing action a_k when the continuum is at state s_k. Fig. 3.b shows another possibility, when in state s_k, action a_k creates a feedback loop that sustains the squadron functionality over a cycle of state transitions.
Instead of using policy or value iteration to derive an optimal policy as it typically performed in MDPs here we replace Bellman's algorithm (Bertsekas, 1995) by a scalable deterministic backtracking tree search approach as follows. However, optimal convergence is guaranteed via the backtracking feature which selects the "best of the best" branches - similar to Belman's algorithm.

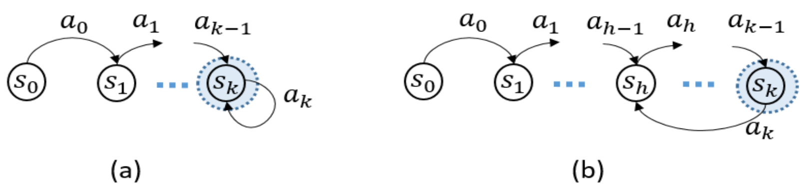

Figure 3. Two possible representations of valid solutions to sustain squadron functionality. In both trails, action a_k is selected in state s_k to create a feedback loop that guarantees the squadron sustainability.

IMPLEMENTATION OF THE TREE SEARCH ALGORITHM

Figure 4 provides a flowchart of the algorithm. It starts from an initial state s_0, for which all possible recruitment actions are generated and stored in an action list l_{s_0}. The algorithm then recursively creates valid composites with all actions in l_{s_0} as $\bar{X}_{s_0}^{a_i}$, $a_i \in l_{s_0}$ and each created composite directs the trail to a next state with its associated costs. Figure 5 shows how the tree search is directed over the problem search space during the recursion. As shown in this figure, only the branches with admissible states can grow down to the adjusted threshold depth. The admissibility of a given state s_i is checked by the following condition:

$$Admissibility\ (s_i) = \begin{cases} N_{b_2} + N_{sq} - N_{int} < \omega & false \\ \quad otherwise & true \end{cases} \tag{11}$$

The condition mentioned in (11) assures that there are enough personnel to maintain squadron functionality. The algorithm retains the top n solutions in a solution archive that makes it easier for the human decision makers to select a final solution or use the archive's information to elaborate their own solution.

5. EXAMPLE SCENARIO

This section involves an experimental scenario which reveals how the algorithm generates solutions for the simplified model of the training continuum. This version was implemented in NetBeans 8.0.2 using JDK 1.8. The initial state for this scenario was determined as $s(\infty, 2,5,22)$, therefore b_0 is representing a pool with infinite people ready to be recruited. Table 1 presents the parameter setting of the algorithm for this scenario. The algorithm covered the whole search space considering branches with the maximum depth of 11 in 19 seconds on a Dell Precision Tower 3420 with an Intel® Core i7-6700 CPU and 32 GB of RAM. The sorted solution archive showed that the best solution is a trail with the length of 4 and the cost of 3777.5 and the 20th solution is a trail with the length of 11 and the cost of 71432.5. Figure 6 shows two sample solutions found by the algorithm. Figure 6.a presents the generated top solution, while Figure 6.b shows another solution (ranked 8 in the archive) with a different structure and a cost of 70916.5.

Table 1. Parameter setting for running the experiment

Parameter	Description	Value
maxDepth	The maximum length for the generated trail. This parameter stops the tree to grow more than a desired length.	11
sq_{lkr}	Squadron wastage ration per term.	0.1
ω	Minimum number of pilots that sustains squadron functionality.	20 ∓ 3
C_{sq}	Per-pilot cost	3
N_{c_1}, N_{c_2}	Infrastructure limits for courses or the maximum number of seats for *Course 1* and *Course 2*.	10
C_{c_1}, C_{c_2}	Course per-trainee cost.	1,2
f_{c_1}, f_{c_2}	Failure rate for courses. These values are derived from a beta-binomial distribution with parameters $a_1 = 1, b_1 = 4$ and $a_2 = 1, b_2 = 2$. a_1, b_1 are associated with c_1 and a_2, b_2 are associated with c_2.	0.2, 0.33
$\beta_0, \beta_1, \beta_2$	Waiting cost coefficients for buffers per-trainee, per term.	0,1.5, 1.5
archiveSize	Size of the solution archive	20

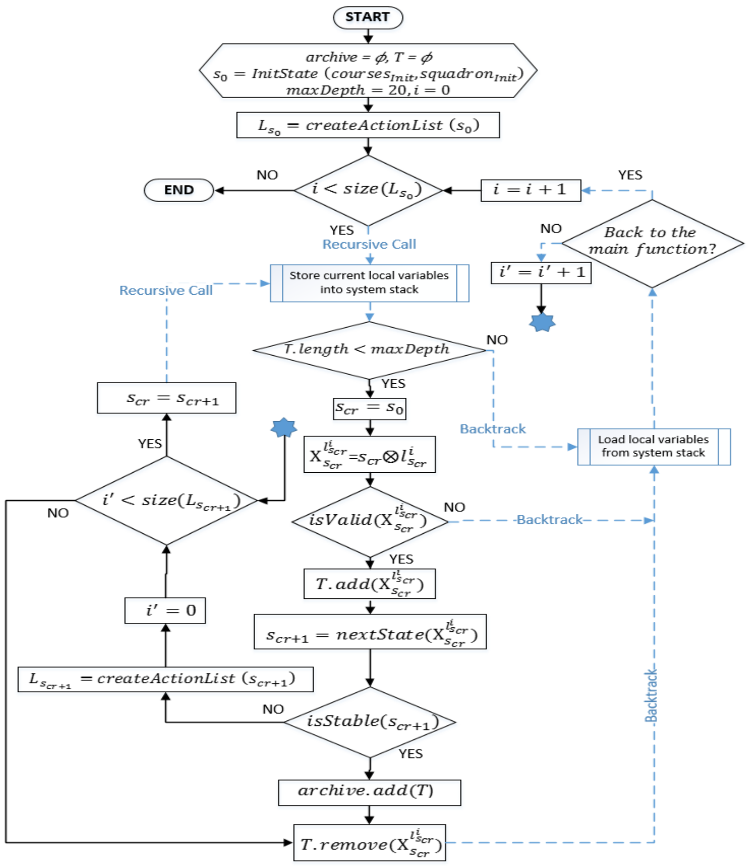

Figure 4. Flowchart of the algorithm for RAN pilot training continuum recruitment problem.

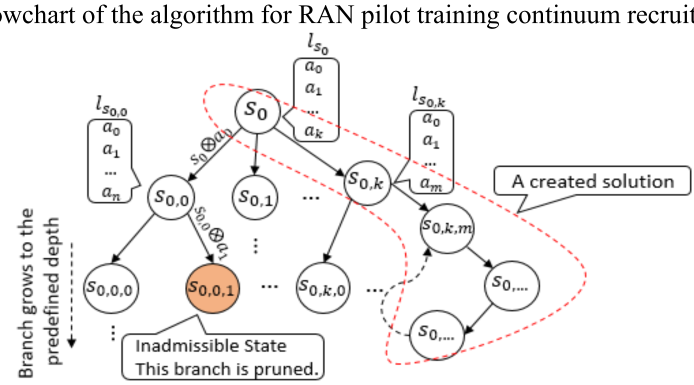

Figure 5. Depth first tree-search for creating solutions through the search space.

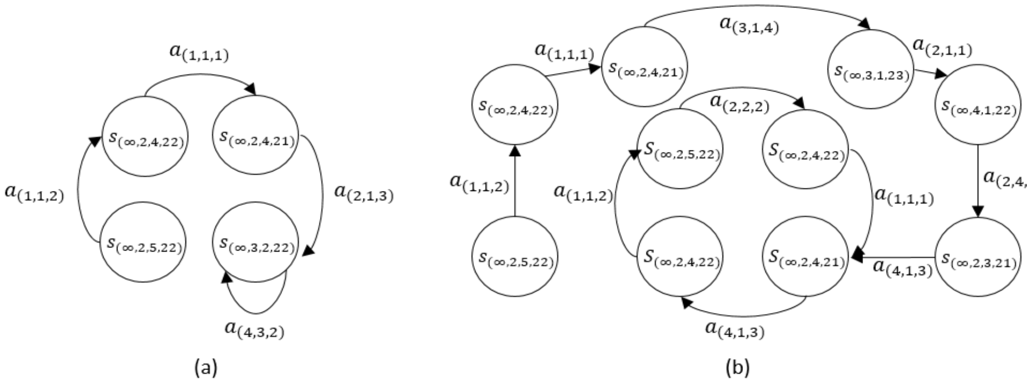

Figure 6. (a) optimum solution found by the proposed algorithm with four states and one stabilizing composite **(b)** rank 8th solution with ten states and a stabilizing circle with four composites.

6. CONCLUSIONS

The proposed algorithm is a preliminary version of the final tree-search algorithm which will be developed to computerize the workforce planning and optimal recruitment of the real-world model of the Royal Australian Navy pilot training. Through a deterministic tree search, this algorithm generates the optimal recruitment policy that guarantees a low-cost squadron functionality. Our strategy fits well with the medium scale WP problems, and it can be easily improved to cope with larger problem search spaces. This comprehensibility of the algorithm makes the final product more transportable and popular. It also provides an appropriate workflow for utilizing methodologies like agile development method (Ramos, Ferreira, & Barcel, 2013) that speeds up the project implementation, decreases the cost of the project and guarantees meeting objectives set by the clients.

REFERENCES

Auerbach, D. I., Buerhaus, P. I., & Staiger, D. O. (2007). Better late than never: Workforce supply implications of later entry into nursing. *Health Affairs, 26*(1), 178-185.

Bayliss, C., Novak, A., Nguyen, V., Caelli, T., & Harrison, S. (2016, 26-29 Sep 2016). *Optimizing Aircrew Training Schedules using Tabu Search.* Paper presented at the SimTecT 2016, Melbourne, Australia.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control* (Vol. 1): Athena scientific Belmont, MA.

Cave, S., & Willis, G. (2016). Applying system dynamics and systems thinking methods to the development of robust workforce policies for the health and social care system in England. *International Journal of Applied Systemic Studies, 6*(3), 239-257.

Harper, P., Kleinman, E., Gallagher, J., & Knight, V. (2013). Cost-effective workforce planning: optimising the dental team skill-mix for England. *Journal of Enterprise Information Management, 26*(1/2), 91-108.

Hertz, P., Cavalieri, S., Finke, G. R., Duchi, A., & Schönsleben, P. (2014). A simulation-based decision support system for industrial field service network planning. *Simulation, 90*(1), 69-84.

Howson, J., & McNamara, O. (2012). Teacher workforce planning: the interplay of market forces and government polices during a period of economic uncertainty. *Educational Research, 54*(2), 173-185.

Nguyen, V., Shokr, M., Novak, A., & Caelli, T. (2016). *A Reconfigurable agent-based discrete event simulator for helicopter aircrew training.* Paper presented at the ISMOR Conference.

Parrish, E. J., Freudenberg, R. P., Scsigulinsky, A., & Cho, R. (1995). Computer-based data integration and management process for workforce planning and occupational readjustment: Google Patents.

Ramos, A. L., Ferreira, J. V., & Barcel, J. (2013). LITHE: an agile methodology for human-centric model-based systems engineering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(3), 504-521.

Rosner, B. (2005). Beta-Binomial Distribution. *Encyclopedia of Biostatistics*.

Wang, J. (2005). A review of operations research applications in workforce planning and potential modeling of military training. Australia: Defence Science and Technology Organisation, Salisbury, (Australia) Systems Science Lab.

Zais, M., & Zhang, D. (2016). A Markov chain model of military personnel dynamics. *International Journal of Production Research, 54*(6), 1863-1885.