CHEMICAL ENGINEERING TRANSACTIONS

VOL.

A publication of

AIDIC

The Italian Association
of Chemical Engineering
Online at www.cetjournal.it

Guest Editors:
Copyright © AIDIC Servizi S.r.l.
ISBN 978-88-95608-xx-x **ISSN** 2283-9216

Optimal Strategies to Control Particle Size and Variance in Antisolvent Crystallization Operations Using Deep RL

Vidhyadhar Manee*, Roberto Baratti**, José A. Romagnoli*

***** Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA, 70809
**Dipartimento di Ingegneria Meccanica, Chimica e dei Materiali, University of Cagliari, Cagliari

Solution crystallization operations have complex dynamics that are typically lumped into two competing processes namely nucleation and growth. Mathematical models can be used to describe these two processes and their effect on the crystal population when subject to variables like temperature, addition of anti-solvent, etc. To ensure that the crystals meet specific performance objectives, the models need to be solved and the control variables need to be optimized. This has largely been done until now using algorithms from dynamic programming or optimal control theory. Recently, however, it has been shown that learning frameworks like Reinforcement Learning can solve large optimization problems efficiently while offering distinct advantages. In this work, we explore the possibility of computing the optimal profiles of a semi-batch crystallizer to control the mean size and variance using four different deep RL algorithms. The performance on one of the tasks is evaluated experimentally on the anti-solvent crystallization of NaCl in a water-ethanol system.

1. INTRODUCTION:

Solution crystallization is employed in various industries to produce crystals with desired properties(Black, 2019; Hartel, 2019; Vicum, Mazzotti and Iggland, 2019). The process of crystal growth from solution can be attributed to two competing mechanisms namely nucleation and growth. Either one of these mechanisms can be preferentially favored to produce crystals with interesting properties. This is typically done by altering the supersaturation of the solution by way of cooling or anti-solvent addition(Ghadipasha *et al.*, 2018).

When crystallization is employed to grow crystals, the target is usually to grow crystals that satisfy certain product objectives such as size, shape, purity, etc. The pioneers in the field of crystallization largely discovered optimal recipes for these objectives by trial and error. Eventually, mathematical models were used to describe the kinetics of crystallization and techniques like Pontryagin's maximum principle were used to compute the optimal cooling and anti-solvent addition rates. The current approach to this problem is to use dynamic optimization to find the temperature and anti-solvent flow rate trajectories that produce crystals of a desired mean size and variance. This produces a stationary trajectory that can be implemented without any supervision. However, mathematical models are always imperfect and these stationary trajectories need some external control (feedback) to produce the desired product objectives. Model Predictive Controllers(Mesbah, 2010), solve this problem, by solving the optimization problem at every sampling interval using the most recent measurements from the experiment. This solves the issue of model inaccuracy and produces more reliable product quality. But the cost of optimization at each sampling interval can be expensive and time consuming depending on the complexity of the model. While this bottleneck can be alleviated by solving the model offline, this can only be done for a discrete number of cases and developing a general control law using this approach will be inefficient and resource hungry.

Here, we attempt to use reinforcement learning (RL) to learn the optimal response for all possible states in the model off-line. The RL framework is equipped to do this efficiently because it learns from past experience which greatly reduces the learning time. In addition, the use of non-linear function approximators, enables the generalization of learned experience. We describe a general RL framework that is model agnostic but we restrict our results to the Stochastic Modeling Framework (Baratti, Tronci and Romagnoli, 2017). Two possible scenarios are described. In the first scenario, we try to optimize only the mean size of the crystals. In the second, we optimize both the mean size and variance of the crystals. We refer to the first task as MO (Mean-Only) while we call the second task MV (Mean & Variance) in the rest of the paper.

2. STOCHASTIC MODELING FRAMEWORK:

In crystallization, the crystal size distribution can be controlled by altering the supersaturation of the solution. There are many techniques to do this such as evaporation, cooling, anti-solvent addition, etc. In non-isothermal anti-solvent crystallization specifically, the addition of an anti-solvent and the temperature of the solution are simultaneously manipulated to control the supersaturation of the solution. The effect of these variables on the growth of crystals can be described by the stochastic modeling approach (Baratti, Tronci and Romagnoli, 2017). According to this approach, the variation of the crystal size distribution ψ is described by the Fokker Planck Equation (Eq 1).

$$\frac{\partial \psi(L,t)}{\partial t} = \frac{\partial}{\partial L}\left\{-\left[rL\left(1-\frac{L}{K}\right)+DL\right]\psi(L,t)+D\frac{\partial}{\partial L}\left[L^2\psi(L,t)\right]\right\} \tag{1}$$

with a log-normal initial condition and boundary conditions given by Eq (2-3).

$$\psi(L,0) = exp\left\{-\frac{[\ln(L) - \mu_0]}{2\sigma_0^2}\right\}/\left(\sigma_0 L\sqrt{2\pi}\right) \tag{2}$$

$$D\left\{g(L)\frac{\partial}{\partial L}[g(L)\psi(L,t)]\right\} = h(L)\psi(L,t) \text{ at L = 0 and } \forall t \tag{3a}$$

$$\frac{\partial \psi(L,t)}{\partial L} = 0 \; at \; L \to \infty \; and \; \forall t \tag{3b}$$

The parameters in the equation r, K and D can be expressed as functions of the anti-solvent flow rate (q) and the temperature (T) (Ghadipasha *et al.*, 2018).

3. REINFORCEMENT LEARNING:

In standard reinforcement learning (Sutton and Barto, 1998), the optimization problem is set up to be solved by an agent who interacts with the environment by picking actions over a sequence of time-steps with the goal of maximizing a cumulative reward. The problem is framed as a Markov Decision Process with a state space, an action space, rewards and a stationary transition dynamics distribution with conditional density $p(s_{t+1}|s_t,a_t)$ following the Markov property $p(s_{t+1}|s_t,a_t) = p(s_{t+1}|s_1,a_1,\dots,s_t,a_t)$ (i.e. past history does not affect future dynamics). The agent samples actions in the environment by following a policy $\pi(a|s,\theta)$, where $\theta \in \mathbb{R}^n$ is a vector of n parameters, and generates a trajectory of states, actions and rewards. The rewards accumulated by the agent is measured by the return r_t^γ which is defined as the discounted sum of rewards from time step t onwards $r_t^\gamma = \sum \gamma^{k-t} r(s_k,\ a_k)$. Since the returns are not accessible until after a trajectory is completed, value functions are used to predict the returns. The value function is defined as the expected total discounted reward (i.e. $V^\pi = E(r_t^\gamma \mid s_t,\pi\ ), Q^\pi = \ E(r_t^\gamma \mid s_t,a_t,\pi)$). The agent's goal, now, in this setting, is to obtain the policy that maximizes a performance objective such as the cumulative discounted reward, for instance ($J(\pi) = E(r_1^\gamma|s_t)$).

There are three major types of approaches to solve the RL problem namely Q-learning, policy gradient and actor-critic. Since the state space and action space of the Fokker Planck Equation is continuous, Q-learning algorithms like DQN are inadequate because they can currently only solve problems with a discrete action space. Therefore, we restrict our focus to the policy gradient and actor-critic algorithms. Algorithms belonging to both these approaches have their own sets of advantages and disadvantages. It is unclear which algorithm would be most efficient at solving this task. In this work, we experiment with two on-policy algorithms A2C (Mnih *et al.*, 2016) & PPO (Schulman *et al.*, 2017) in addition to two off-policy algorithms SAC (Haarnoja *et al.*, 2018) & TD3 (Fujimoto, Van Hoof and Meger, 2018) to analyze their performance in terms of average returns and training time.

4. SETUP OF THE ENVIRONMENT:

As described before, the reinforcement learning setup requires an environment, an agent to pick actions, a critic to evaluate the actions and a rewarding scheme to guide the agent. In the section below, the reinforcement learning setup for this problem will be described. This applies to all the algorithms we have tested in this study.

State: The state of the system encapsulates information about the crystal size distribution. There are multiple ways to describe the state in a way that the Markov property is satisfied. The simplest case would contain the mean size of the crystals and the variance in the state definition. This has the advantage of being easy to interpret and also satisfies the Markov property. But in order to guide the agent towards the desired crystal size distribution, we also include the target mean size and target variance in our state definition. These targets can alternatively be included in the reward formulation, but we do it here in order to learn a general optimal control law. The range of mean sizes permitted by the model is between 120 microns and 165 microns while the range of the standard deviation is between 48 to 64 microns. The two cannot be chosen independently (Cogoni *et al.*, 2014). At training time, we randomly sample the target mean size and target variance from a uniform distribution of possible values. The state values are normalized between -1 and 1 before they are passed to the actor and critic networks for training.

$$State = [Mean, Target\ Mean, Variance, Target\ Variance] \tag{4}$$

Action: The actions that can influence the state are the temperature and the anti-solvent flow rates. Both these actions are constrained with the temperature having an upper and lower bound of 30°C and 10 °C respectively while the anti-solvent flow rate is bounded between 3 $^{mL}/_{min}$ and 0.7 $^{mL}/_{min}$. Similar to the state definition, the action values are normalized between -1 and 1 before they are passed to the actor neural network for training.

Reward: The reward signal is an important piece of the RL framework because it gives the RL agent an estimate of the quality of its actions. We only reward the agent (+1 for mean size and +0.1 for variance) when the state values are sufficiently close to the target values. This was arbitrarily set to be 0.5 microns for the mean size and 0.3 units for the standard deviation. For all other scenarios, the agent is not rewarded.

Environment: The environment takes in the value of the current state and the actions executed by the agent in that state to generate the subsequent state. This is done by solving Equation 1 using a finite volume method. We use the FiPy library in python to perform this numerical approximation. The state values (mean and variance) are used to generate the corresponding log-normal distribution, and this is passed as the initial condition to equation 1. The FiPy code then solves the equation for one period of sampling time and returns the next state.

Table 1- Algorithm to teach the agent to respond to disturbances.

Algorithm
1. Pick an initial start state (mean value and variance) from a uniform distribution.
2. Run the policy (i.e. execute the action selected by the agent in the environment)
3. Get the reward and the next state.
4. Introduce a disturbance with probability ϵ.
5. Repeat steps 2-4 until the target state is reached.
6. Update the agent and the critic using the experience gained thus far.

In this section, we show the performance of the algorithms on the two tasks described above. In the MO task, the agent is given 30 timesteps to reach the desired mean size. Since the reward is +1 for every successful timestep, the agent can score a maximum of 30 on this task. Table 12 displays the average reward and learning time for the algorithms. Both the on-policy and the off-policy algorithms learn to solve this task with different degrees of accuracy. A common pattern among the policies learned by the algorithms is that they maximize the rewards for the intermediate values of target mean-sizes (125-160 microns) while they perform relatively poorly in the extremes (120-125 & 160-165 microns). In the case of the PPO, the relatively poor average reward can mostly be attributed to poor performance on the extreme target mean-sizes.

The learning time of the algorithms, on the other hand, clearly distinguishes the on-policy algorithms from the off-policy ones. A2C & PPO take significantly longer to learn the optimal policy. But this problem is marginally offset by the fact that multiple environments can be run in parallel with on-policy algorithms. This is not the case with off-policy algorithms. Despite this, the off-policy algorithms still learn considerably faster.

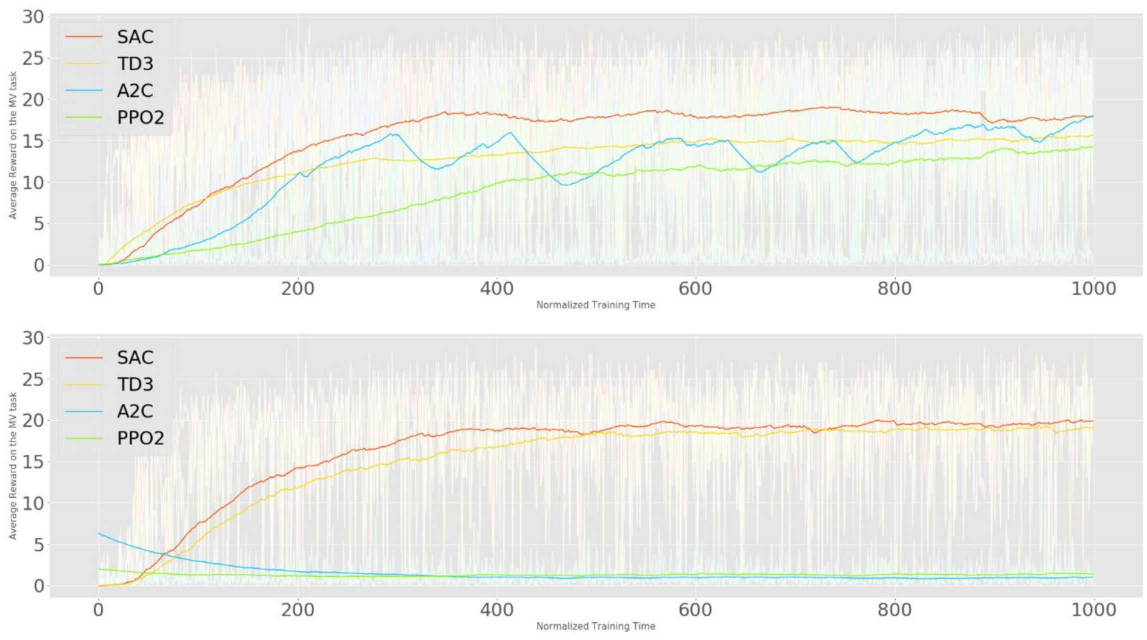

Figure 1: The plot on the top and bottom shows the progress of the training reward in the MO and MV task respectively. The training time has been normalized for the sake of comparison. The true training times are provided in Tables 2 and 3.

Figure 2 shows the performance of the algorithms on the MO task for a target mean size of 140 microns. All four algorithms manage to reach the target quickly but with slightly different optimal trajectories. This can be attributed to the presence of input multiplicities (Cogoni et al., 2014) in the model due to two competing inputs. This means that an asymptotic CSD, characterized by its mean and variance can be obtained using two sets of input values. The trajectories generated by the SAC algorithm was implemented experimentally and the results are shown in Figure 3.

Table 2: Performance of the algorithms on the MO and MV tasks

Algorithm	Average Reward – MO task	Time Steps to Learn – MO task	Average Reward – MV task	Time Steps to Learn – MV task
A2C – On-policy	14.45	2.5M	1.8	4M
PPO – On-policy	11.31	3.5M	2.3	3M
SAC – Off-policy	**17.56**	**0.333M**	**18.8**	**0.4M**
TD3 – Off-policy	14.56	0.2M	17.69	0.4M

In the MV task, the agent again has 30 steps to maximize the rewards. It can score a maximum of +33 now with +1 per timestep for reaching the desired mean size and +0.1 per timestep for the target variance. These rewards were chosen by grid search method and other rewarding schemes could be equally successful.

It is seen from Table 2 that the off-policy agents are more successful at this task than the on-policy agents. They reach a high average reward with fewer steps. The on-policy agents struggle to learn in the environment despite being given ample time (3-4M steps). Figure 4 shows the performance of the off-policy agents on the MV task for a target mean size of 140 microns. It is seen that both SAC and TD3 get close to the target variance.

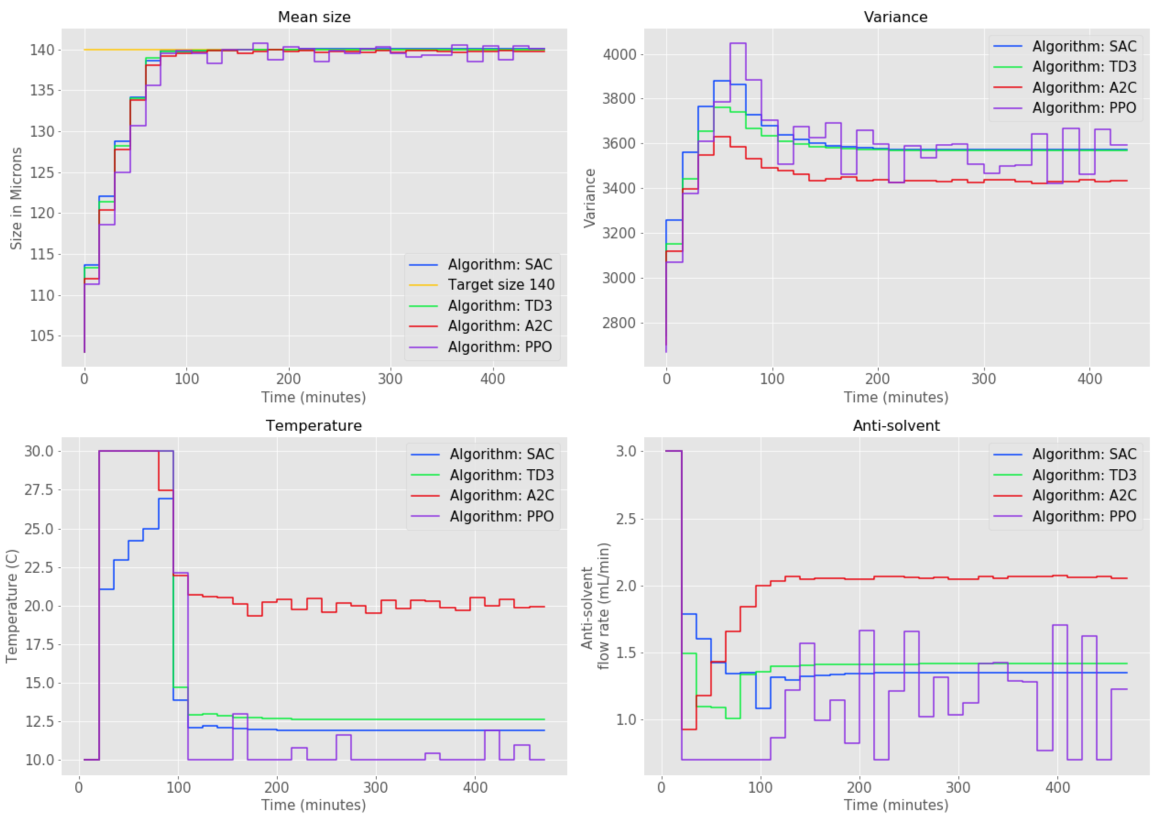

Figure 2 Optimal Response to achieve a desired target size of 140 microns.

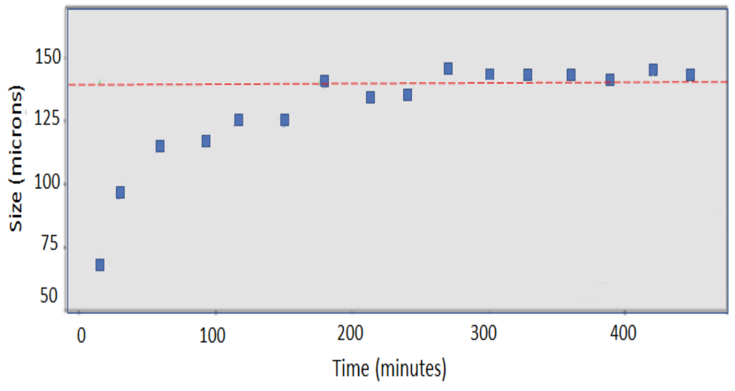

Figure 3: Experiment to control the mean size of the crystals using the variable trajectories generated by SAC in Figure 2.

5. CONCLUSION:

This work shows that it is possible to develop a general control law for a crystallization model using RL agents. While this framework is more resource intensive than other frameworks during the training phase, it enables the user to train the model offline and minimize computation time during the run. The results of simulation show that both off-policy and on-policy agents can control the mean-size of the crystals while the off-policy agents can control both the mean-size and the variance. The profiles developed by the RL agent on the MO task was implemented experimentally and the results indicate that it works well in practice.

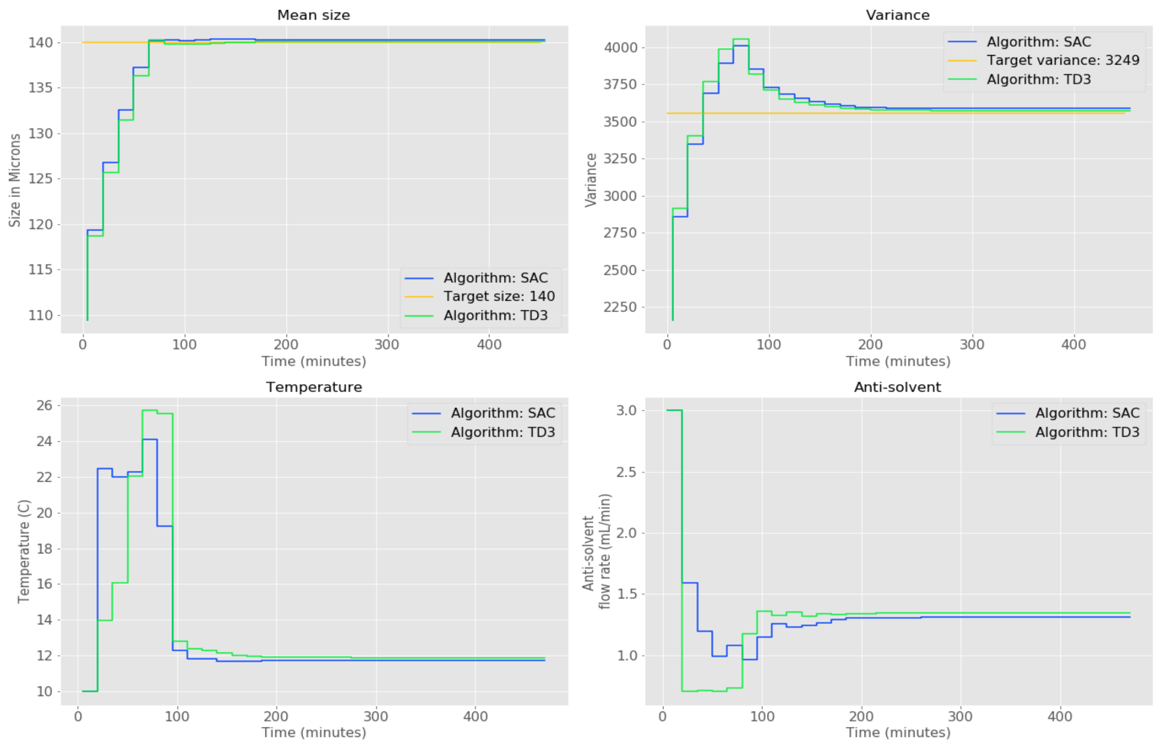

Figure 4: Optimal Response to achieve a desired target size of 140 microns and the lowest possible variance.

6. REFERENCES:

Baratti, R., Tronci, S. and Romagnoli, J. A. (2017) 'A generalized stochastic modelling approach for crystal size distribution in antisolvent crystallization operations', *AIChE Journal*, 63(2). doi: 10.1002/aic.15372.

Black, S. N. (2019) 'Crystallization in the Pharmaceutical Industry', in *Handbook of Industrial Crystallization*. Cambridge University Press, pp. 380–413. doi: 10.1017/9781139026949.013.

Cogoni, G. *et al.* (2014) 'Controllability of semibatch nonisothermal antisolvent crystallization processes', *Industrial and Engineering Chemistry Research*, 53(17). doi: 10.1021/ie404003j.

Fujimoto, S., Van Hoof, H. and Meger, D. (2018) 'Addressing Function Approximation Error in Actor-Critic Methods', in *35th International Conference on Machine Learning, ICML 2018*.

Ghadipasha, N. *et al.* (2018) 'A model-based approach for controlling particle size distribution in combined cooling-antisolvent crystallization processes', *Chemical Engineering Science*, 190. doi: 10.1016/j.ces.2018.06.032.

Haarnoja, T. *et al.* (2018) 'Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor', in *35th International Conference on Machine Learning, ICML 2018*.

Hartel, R. W. (2019) 'Crystallization in Foods', in *Handbook of Industrial Crystallization*. Cambridge University Press, pp. 460–478. doi: 10.1017/9781139026949.015.

Mesbah, A. (2010) *Optimal Operation of Industrial Batch Crystallizers A Nonlinear Model-based Control Approach*. Available at: https://repository.tudelft.nl/islandora/object/uuid%3Aced4380a-0808-4398-9fe7-5cc0601b0449 (Accessed: 11 April 2021).

Mnih, V. *et al.* (2016) 'Asynchronous methods for deep reinforcement learning', *33rd International Conference on Machine Learning, ICML 2016*, 4.

Schulman, J. *et al.* (2017) 'Proximal policy optimization algorithms', *arXiv*. arXiv. Available at: https://arxiv.org/abs/1707.06347v2 (Accessed: 3 December 2020).

Sutton, R. S. and Barto, A. G. (1998) 'Reinforcement Learning: An Introduction', *IEEE Transactions on Neural Networks*, 9(5). doi: 10.1109/tnn.1998.712192.

Vicum, L., Mazzotti, M. and Iggland, M. (2019) 'Precipitation and Crystallization of Pigments', in *Handbook of Industrial Crystallization*. Cambridge University Press, pp. 479–512. doi: 10.1017/9781139026949.016.