Extending the Hint Factory: Towards Modelling Productivity for Open-ended Problem-solving

Mehak Maniktala*
North Carolina State University
mmanikt@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

Min Chi
North Carolina State University
mchi@ncsu.edu

ABSTRACT
Determining when and whether to provide personalized support is a well-known challenge called the assistance dilemma. A core problem in solving the assistance dilemma is the need to discover when students are off-track or unproductive, so that the tutor can intervene. Such a task is particularly challenging for open-ended domains such as logic proofs, and programming. In this paper, we present a data-driven method to determine *step-level productivity* in a logic proof tutor. This approach leverages and modifies the Markov decision processes in the Hint Factory, a data-driven hint generator, to develop four productivity metrics. Our results provide evidence suggesting that, for each productivity metric, students' training productivity significantly correlates to their posttest performance. We conclude with a discussion outlining challenges posed when comparing these productivity metrics to a ground truth, and propose a preliminary approach to address them.

Keywords
productivity, open-ended, hint timing, assistance dilemma, logic proofs

1. INTRODUCTION
Intelligent Tutoring Systems (ITSs) provide individuals with adaptive feedback and hints, improving learning [25]. Studies suggest that hints, when provided appropriately, can augment students' learning experience [10, 22] and improve their performance [7]. However, researchers often find that students display poor help-seeking behavior [2, 21]; some abuse hints to expedite problem completion, and some avoid seeking help when they are in need [1, 20].

To deal with non-optimal help-seeking behavior, several ITSs provide unsolicited assistance [3, 18, 13]. However, determining *when* to provide proactive assistance, i.e., unsolicited assistance in anticipation of future struggle, is particularly challenging in open-ended domains where there are many possible correct solutions. The assistance dilemma is a well-recognized challenge in the domain of ITSs, where a trade-off exists between giving and withholding information to achieve optimal learning [14]. On one hand, providing assistance may reduce frustration and save students' time, but may lead to shallow learning or a lack of motivation to learn by oneself. On the other hand, withholding information can encourage students to learn by themselves, but may lead to frustration and wasted time [14, 16]. A core problem of the assistance dilemma is the need to discover when students are off-track or unproductive so that the tutor can intervene. We hypothesize that developing a proactive hint policy for a logic tutor, where tutor interventions are delivered upon predictions of unproductivity in training, can improve students' logic proof strategies in a posttest without hints.

Contributions. In this paper, we present our novel, data-driven approach to measure productivity in an open-ended logic ITS. We extend the Hint Factory [24], a generalizable data-driven method for hint generation, to define four metrics of productivity. We then present our preliminary analysis on evaluating these metrics in the logic tutor.

2. RELATED WORK
Several studies have used the term "unproductive" to refer to undesirable behavior during training [12, 9, 19]. For example, Beck and Gong [8] define unproductive persistence or "wheel-spinning" based on whether or not a student achieved mastery (three correct problems) in 10 problem attempts. Their definition of unproductivity has been used in recent studies to predict when an intervention can help students by distinguishing between productive and unproductive behavior using decision trees [12] and Recurrent Neural Networks [9]. However, this definition of problem-level and problem-completeness based productivity is not suitable for our objective of guiding students toward efficient problem-solving strategies at a finer step-level granularity, specifically in open-ended domains.

McLaren et al. in a study on an open-ended inquiry-learning program defined unproductive events as the actions taken by the student that do not help them achieve the goal of a particular level, i.e., the steps that students take that are unlikely to advance their understanding of the concepts being taught [16]. Similar to this study, we define productivity on a step level to identify student steps that are not likely

*This reseach was supported by the NSF grants 1726550 and 1651909.

Mehak Maniktala, Tiffany Barnes and Min Chi "Extending the Hint Factory: Towards Modelling Productivity for Open-ended Problem-solving" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020),* Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 796 - 801

to advance their problem-solving strategies. However, our definition is different in that we did not use a pre-defined domain-specific metric for efficient and inefficient strategies; rather we focused on solution length or optimality, which is valued across problem-solving domains.

In many multi-step open-ended but well-structured problem-solving domains, shorter solutions are considered to be more optimal than longer ones, and solving problems in less time reflects both learning and fluency [15, 23, 26, 11]. We use these basic assumptions about time and solution length to design a data-driven, domain-agnostic approach to model productivity on problem-solving steps. The Hint Factory is a data-driven approach for hint generation. The approach uses prior students' transaction log data to form an interaction network and assign scores to problem-solving states (snapshots of an on-going or completed proof) [24, 5]. A core insight of the work in this paper is that we can similarly use interaction networks to score productive problem-solving steps without the need to model the domain.

3. METHOD & PRELIMINARY RESULTS

In this section, we define *productivity*, a data-driven measure quantifying how much a student's most recent step contributes to an efficient solution. We explore four new productivity metrics defined using the combination of the *quality* of each state (local or global), and the amount of *progress* made in a step (absolute or relative). First, we present the Hint Factory. Next, we present our extension of the Hint Factory, and how we use it to define productivity.

3.1 Hint Factory

Hint Factory is a method for generating hints in multi-step open-ended domains [6]. In the Hint Factory, historical student solutions are used to form Markov decision processes (MDP) from interaction networks, where vertices are observed student problem-solving states (snapshots of their on-going or completed proof), and edges are problem-solving steps, i.e, a transition between states. The Hint Factory uses value iteration, a reinforcement learning technique, given in Equation 1, to assign an *expected value* $LQV(s)$ to each state s, where $LR(s)$ is the state's reward, γ is the discount factor, and $P_a(s, s')$ is the proportion of the observed solutions in state s that lead to state s' using the action (i.e. step) a. In the Hint Factory, a large reward is set for the problem-completion or *goal states* (100), penalties for incorrect states (10), and a cost for taking each action (1) [5]. A non-zero cost on actions causes the MDP to penalize longer solutions.

$$LQV(s) := LR(s) + \gamma \max_a \sum_{s'} P_a(s, s') LQV(s') \quad (1)$$

3.2 State Quality - Extending the Hint Factory metrics

In this section, we leverage the Hint Factory approach to generate two quality metrics that determine the expected values for each observed problem-solving state. The first metric of state quality was defined as part our prior work on the Hint Factory, which we label as *local quality.* Local quality provides insights about how far a state is from the closest goal state, weighted by the probabilities of transitions, but it cannot provide information about whether the state is on an efficient path to a solution.

$$GQV(s) := GR(s) + \gamma \sum_{s'} P_a(s, s') GQV(s') \quad (2)$$

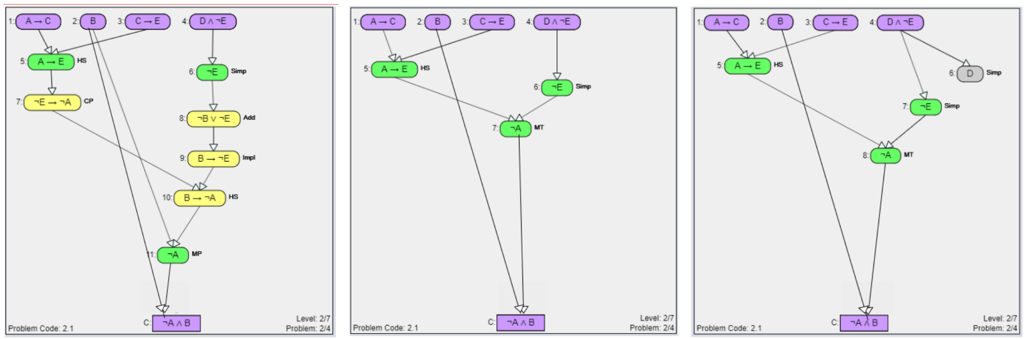
Global Quality. We devised a novel, data-driven global quality value function, GQV in Equation 2, to give higher values to states on efficient solution paths. Equation 2 sums $GQV(s')$ over all states s' reachable from s, weighted by $P_a(s, s')$, taking into account all future actions from a current state, rather than just the one with the max expected value. The global rewards GR are identical to LR for errors and actions, but are different for goals, giving shorter, more efficient solutions higher rewards. The global reward $GR(g)$ for each goal state g on a problem is $GR(g) = 100 - p * \delta(g)$ where $\delta(g)$ is the difference between the solution length of g and that of the shortest solution, and p is a penalty for longer solutions. We set $p = (100 - 80)/\delta_{median})$ where δ_{median} is the difference between the median and shortest solution lengths for each problem because median student solution lengths are assigned a global reward of 80. Meaning, the student's performance with a median solution length represents a low B grade. The proof of convergence for the modified value iteration equation 2 is given in appendix A.

We now demonstrate the differences between local and global quality metrics using solution trajectories (series of steps) of varying solution lengths: T_{short}, T_{medium}, and T_{long} in Figures 1 and 2. T_{short} is the shortest solution (four steps), with all nodes (logic statements) used. Note that a node is said to be *used* if it contributes towards deriving the conclusion of the problem. T_{medium} has five steps with one unused node D; and T_{long} has eight steps, and all nodes used.

We generated interaction networks to determine the quality values for each problem using our historical data for N = 796 students. Figure 2 shows the quality values for the three trajectories in Figure 1. The start state in Figure 2 consists of the three given logic statements (the topmost state). Arrows between states represent steps, i.e, transition between states by logic rules applications. Non-start states are represented by a +(XYZ), where XYZ is the new logic statement derived in a step. The start state has a high global quality, but low local quality. The start state's global quality is high because all efficient paths contain it, but its local quality is low because it is probabilistically farther away from goals than any other state in the figure. The local quality for states that are only found in incomplete attempts is lower than that for the start state. The local quality of the goal states on all three trajectories is 100. The global quality value for the goal state in each solution trajectory differs, with 100 for the T_{short} goal (since it's the most efficient), 95 for T_{medium}, and 80 for T_{long} goal states.

From the start state to the goal in T_{short}, both local and global quality state values increase monotonically since it is the most efficient solution. Note that not all quality values increase over every trajectory. For example, step $T_{medium} -$ 2's pre-state $(+A \rightarrow E)$ global quality is higher than that for its post-state $(+D)$ since the pre-state is on a more efficient path, but the local quality increases from pre- to post-state. Step $T_{long} - 3$'s pre-state $(+\neg E)$ has higher local and global quality values than its post-state $(+\neg E \rightarrow \neg A)$ since the post-state is farther from and less likely to reach T_{long}'s

Figure 1: Three Solutions with Varying Number of Steps for a logic problem in Deep Thought



(a) Trajectory T_{long}: Eight Steps and all used nodes

(b) Trajectory T_{short}: Four Steps and all used nodes

(c) Trajectory T_{medium}: Five Steps and four used nodes

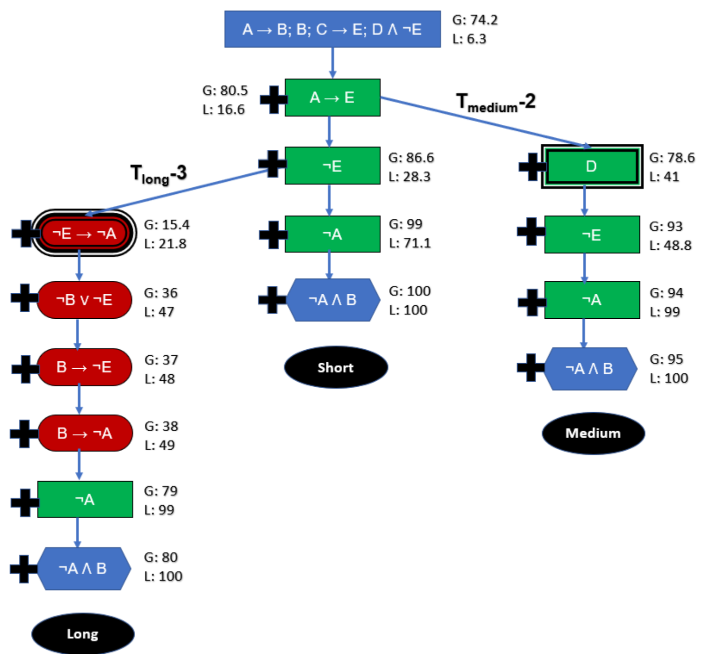


Figure 2: Illustration for the concepts of State Quality and Productive steps in 3 trajectories T_{short}, T_{medium}, and T_{long}

goal than the pre-state is to T_{short}'s closer goal. The global quality decreases for two reasons: (1) T_{long} goal is on a less efficient path, and (2) global quality performs a weighted sum over all the subsequent, previously-observed states in the larger (unshown) interaction network, many of which lead to incomplete attempts.

These examples demonstrate the differences between local and global state quality metrics. The main strength of generating these quality values is the MDP approach which ensures that each state quality value is based not only on the distance from a solution but also on the probability of transition at each of the successive steps. This allows us to rate steps in a more probabilistic manner than a simplistic comparison based on the distance from the most efficient expert solution.

3.3 Progress - Change in Quality

Since state quality is a measure of relative "goodness", we *compare* the quality of the current state with that of the previous and start states to evaluate the productivity in a step. In this section, we define two measures for *progress*: *relative*, the change in state quality from the *previous* problem-state, and *absolute*, the change in state quality from the *start*.

Relative progress is the difference between the quality values of the current and previous states. Relative progress with local quality identifies whether the previous or current state is closer to the goal. When using the global quality values, the relative progress identifies the state closer to a goal, and provides additional information detailing which one of the two is on a more efficient solution path.

Consider a valid, but long solution attempt. A relative progress measure reveals whether a student is progressing toward a solution in a step, but not whether their trajec-

Quality-Progress	Productivity Formula
Global-Absolute	$(\text{GQV}_{post-state} - GQV_{start-state}) \geq 0$
Global-Relative	$(\text{GQV}_{post-state} - GQV_{pre-state}) \geq 0$
Local-Absolute	$(\text{LQV}_{post-state} - LQV_{start-state}) \geq 0$
Local-Relative	$(\text{LQV}_{post-state} - LQV_{pre-state}) \geq 0$

Table 1: Step Productivity formula based on state Quality values and step Progress

tory is efficient. Therefore, we define *absolute progress* as the difference between the current and start states' quality, using either quality measure. Absolute progress using local quality reveals whether a student's current state is farther or closer from any goal states than when they began working on the proof. Global quality based absolute progress reveals the amount of efficient progress a student has made between the start and the goal. For example, if we compare the absolute progress on two consecutive steps of a solution attempt, if a student is taking efficient steps, then the absolute progress will increase on every step.

3.4 Productivity - Quality & Progress

We define four kinds of productivity measures based on quality {Local, Global} and progress {Relative, Absolute}. A step is considered *productive* if the progress of its post-state using either quality measure is a non-negative number, and *unproductive* otherwise, as shown in Table 1.

In consultation with an expert who has more than twenty years of experience teaching discrete math, we labeled the steps as productive or not in each of the three trajectories $(T_{short}, T_{medium}, T_{long})$ shown in Figure 1. Expert-assigned unproductive steps in Figure 2 are displayed in red and others are in green. According to the local quality and absolute progress (local-absolute) productivity metric, all the steps are productive because they eventually lead to a solution. However, this metric is not sensitive to variations in the solution lengths. When we use local-relative productivity, only the $T_{long} - 3$ step is unproductive, as it is the only step where a post-state is probabilistically farther from a solution than the pre-state. Using the global-relative measure, steps $T_{medium} - 2$ and $T_{long} - 3$ are unproductive because they have a pre-state on a more efficient path to the solution than the post-state. The global-absolute metric is the only measure that labels the four expert-identified unproductive steps correctly. [1]. Note that each type of productivity captures a different perspective on a step towards the solution. Overall, the global-absolute productivity metric aligns perfectly with the expert's labels for the sample trajectories. However, using a panel of experts to rate each step would provide a more robust assessment of the ground truth. A major challenge in a manual inspection by a panel of experts is our vast state-space (N = 72,560).

3.5 Selecting a Productivity Metric

Productivity Using	Corr
Global-Absolute	0.328
Global-Relative	0.261
Local-Absolute	0.294
Local-Relative	0.236

Table 2: Correlation between students' Training Productivity and Posttest Optimality (all correlations are significant with $p < 0.01$)

To understand which one of the four productivity metrics is most indicative of how students' work in the tutor's training section affects their posttest solution optimality, we conducted a correlation test. Note, we evaluate students on an *optimality* score, which is as an exponential decay function on normalized steps e^{-steps} to account for the small variance in the number of steps. *Steps* are normalized to the interquartile range for each specific problem to account for varying lengths/difficulties. Very short solutions with step count less than or equal to Q1 (first quartile) have, $optimality = 1$, and those with step count greater than Q3 (third quartile) have an optimality score of 0.36 or less based on the exponential decay curve.

For each student in our dataset ($N = 437$), we computed their posttest optimality and the proportion of training steps that are productive using each productivity metric. We then calculated the correlation between each type of training productivity with posttest optimality using Pearson's coefficient. Table 2 shows that higher productivity in training is significantly correlated to better posttest optimality for all the productivity metrics. Among them, the global-absolute metric is the most correlated with posttest optimality.

4. CONCLUSION & ADVICE SOUGHT

In this paper, we provide a unique extension of the Hint Factory to determine productivity on a step-level in a logic ITS. Outside the scope of this paper, we assessed the impact of intervening with hints using a predictor of unproductivity (global-absolute) in a controlled study with two conditions: control and adaptive. Students in both conditions could request hints, but interventions using the predictor were given only in the adaptive condition. We found that the adaptive condition students had significantly better posttest optimality and time than their control peers. Our long term aim is to assess the generalizabitily of this approach in other open-ended domains such as programming, and to address the assistance dilemma for open-ended problem-solving.

For this doctoral consortium, we would like advice on how to further assess and compare the productivity metrics against a ground truth. We plan to form a panel of experts to rate a larger number of steps, but it is infeasible for them to rate each step in our vast state-space. Do we employ data-driven methods to determine the ground truth? I would like to discuss data-driven ways to evaluate the productivity metrics such as using inferring rewards from Gaussian processes [4] or using procedural solution generators [17].

Our preliminary results are promising, and through this consortium, we seek to determine a method to assess the ground truth of step-level productivity in the logic tutor.

[1]Note that these four unproductive states also correspond to the four infrequently used (yellow) nodes in the student solution shown in Figure 1a. However, some unproductive nodes have been observed to be frequently used, and some productive nodes to be infrequently used in our tutor, suggesting that the use-frequency alone cannot determine productivity

5. REFERENCES

[1] V. Aleven and K. R. Koedinger. Limitations of student control: Do students know when they need help? In *International Conference on Intelligent Tutoring Systems*, pages 292–303. Springer, 2000.

[2] V. Aleven, B. Mclaren, I. Roll, and K. Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, 2006.

[3] I. Arroyo, J. E. Beck, C. R. Beal, R. Wing, and B. P. Woolf. Analyzing students' response to help provision in an elementary mathematics intelligent tutoring system. In *Papers of the AIED-2001 workshop on help provision and help seeking in interactive learning environments*, pages 34–46. Citeseer, 2001.

[4] H. Azizsoltani, Y. J. Kim, M. S. Ausin, T. Barnes, and M. Chi. Unobserved is not equal to non-existent: using gaussian processes to infer immediate rewards across contexts. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1974–1980. AAAI Press, 2019.

[5] T. Barnes, J. Stamper, and M. Croy. „using markov decision processes for automatic hint generation ". *Handbook of Educational Data Mining*, 467, 2011.

[6] T. Barnes, J. C. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *EDM*, pages 197–201, 2008.

[7] T. Bartholomé, E. Stahl, S. Pieschl, and R. Bromme. What matters in help-seeking? a study of help effectiveness and learner-related factors. *Computers in Human Behavior*, 22(1):113–129, 2006.

[8] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International conference on artificial intelligence in education*, pages 431–440. Springer, 2013.

[9] A. Botelho, A. Varatharaj, T. Patikorn, D. Doherty, S. Adjei, and J. Beck. Developing early detectors of student attrition and wheel spinning using deep learning. *IEEE Transactions on Learning Technologies*, 2019.

[10] A. Bunt, C. Conati, and K. Muldner. Scaffolding self-explanation to improve learning in exploratory learning environments. In *International Conference on Intelligent Tutoring Systems*, pages 656–667. Springer, 2004.

[11] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *Frontiers in artificial intelligence and applications*, 158:511, 2007.

[12] S. Kai, M. V. Almeda, R. S. Baker, C. Heffernan, and N. Heffernan. Decision tree modeling of wheel-spinning and productive persistence in skill builders. *JEDM| Journal of Educational Data Mining*, 10(1):36–71, 2018.

[13] S. Kardan and C. Conati. Providing adaptive support in an interactive simulation for learning: An experimental evaluation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3671–3680. ACM, 2015.

[14] K. R. Koedinger and V. Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

[15] R. E. Mayer. *Thinking, problem solving, cognition.* WH Freeman/Times Books/Henry Holt & Co, 1992.

[16] B. M. McLaren, M. Timms, D. Weihnacht, D. Brenner, K. Luttgen, A. Grillo-Hill, and D. H. Brown. A web-based system to support inquiry learning. In *Proceedings of the 6th International Conference on Computer Supported Education-Volume 1*, pages 43–52. SCITEPRESS-Science and Technology Publications, Lda, 2014.

[17] K. L. McMillan and A. Rybalchenko. Solving constrained horn clauses using interpolation. *Tech. Rep. MSR-TR-2013-6*, 2013.

[18] R. C. Murray and K. VanLehn. A comparison of decision-theoretic, fixed-policy and random tutorial action selection. In *International Conference on Intelligent Tutoring Systems*, pages 114–123. Springer, 2006.

[19] S. Park and N. Matsuda. Predicting students' unproductive failure on intelligent tutors in adaptive online courseware. In *Proceedings of the Sixth Annual GIFT Users Symposium*, volume 6, page 131. US Army Research Laboratory, 2018.

[20] T. W. Price, Z. Liu, V. Cateté, and T. Barnes. Factors influencing students' help-seeking behavior while programming with human and computer tutors. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 127–135. ACM, 2017.

[21] T. W. Price, R. Zhi, and T. Barnes. Hint generation under uncertainty: The effect of hint quality on help-seeking behavior. In *International Conference on Artificial Intelligence in Education*, pages 311–322. Springer, 2017.

[22] M. Puustinen. Help-seeking behavior in a problem-solving situation: Development of self-regulation. *European Journal of Psychology of education*, 13(2):271, 1998.

[23] M. U. Smith. *Toward a unified theory of problem solving: Views from the content domains.* Routledge, 2012.

[24] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.

[25] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[26] K. Yacef. The logic-ita in the classroom: a medium scale experiment. *International Journal of Artificial Intelligence in Education*, 15(1):41–62, 2005.

APPENDIX

A. PROOF OF CONVERGENCE FOR THE MODIFIED BELLMAN BACKUP FUNCTION

Theorem: The modified Value iteration (Eqn 2) converges to GQV^* for any initial estimate GQV, i.e.,

$$\lim_{k\to\infty} GQV_k = GQV^* \quad \forall GQV$$

For any estimate of the value function GQV, we define the modified Bellman backup operator $\hat{B} : R^{|S|} \to R^{|S|}$

$$\hat{B}GQV(s) = GR(s) + \gamma \sum_{s'\in S} P_a(s,s')GQV(s')$$

Before we provide the proof of convergence, we provide the proof of contraction, i.e, for any two value functions GQV and GQV':

$$||\hat{B}GQV_k - \hat{B}GQV'_k|| \leq \gamma ||GQV_k - GQV'_k||$$

where the max norm:

$$||GQV|| = \max_{s\in S} |GQV(s)|$$

$||v - v'||$ = Infinity norm (max difference over all states)

Proof of contraction:

$$||\hat{B}GQV - \hat{B}GQV'||$$

$$= \left|\left|\left[GR(s) + \gamma \sum_{s'\in S} P_a(s,s')GQV(s')\right] - \left[GR(s) + \gamma \sum_{s'\in S} P_{a'}(s,s')GQV'(s')\right]\right|\right|$$

$$= \gamma \left|\left|\left[\sum_{s'\in S} P_a(s,s')GQV(s') - \sum_{s'\in S} P_{a'}(s,s')GQV'(s')\right]\right|\right|$$

$$= \gamma \left|\left|\left[\sum_{s'\in S} P_a(s,s')(GQV(s') - GQV'(s'))\right]\right|\right|$$

$$\leq \gamma \max_{s} \sum_{s'\in S} P_a(s,s')|GQV(s') - GQV'(s')|$$

$$\leq \gamma \sum_{s'\in S} P_a(s,s')||GQV - GQV'||$$

$$= \gamma ||GQV - GQV'||$$

since $P_a(s,s')$ are non-negative and sum to one

Proof of Convergence:

$$||GQV_{k+1} - GQV^*||_\infty$$
$$= \left|\left|\hat{B}GQV_k - GQV^*\right|\right|_\infty$$
$$\leq \gamma\, ||GQV_k = GQV^*||_\infty \leq ...$$
$$\leq \gamma^{k+1}\, ||GQV_0 - GQV^*||_\infty \longrightarrow 0$$