Algorithms for Generating Attribute Values for the Classification of Tactical Situations

David Ezra Sidran
University of Iowa
Iowa City, Iowa, USA

Alberto Maria Segre
University of Iowa
Iowa City, Iowa, USA

{dsidran, segre}@cs.uiowa.edu

Keywords:

Tactical Algorithms, Anchored Flanks, Restricted Avenues of Attack, Restricted Avenues of Retreat, Interior Lines, Unsupervised Machine Learning, ClassIT, Computational Military Reasoning

ABSTRACT: In this paper we describe a series of algorithms that generate real-valued attributes used to classify tactical situations using an unsupervised machine learning system. Attributes for the classification of tactical situations include anchored and unanchored flanks, choke points, restricted avenues of attack and retreat, and interior line of support.

1. Introduction.

Our research in Computational Military Tactical Planning (as introduced by Kewley and Embrechts[1]) suggests that, an unsupervised machine learning system (like Gennari and Langley's ClassIT [2]) can make reasoned inferences about previously unseen tactical situations. What is required is an appropriate set of attributes or features that describe significant aspects of a tactical situation and a sufficiently large database of tactical situations. We surveyed 14 Subject Matter Experts (SMEs) and found consistent agreement regarding the presence of, and the ability to identify by SMEs, certain attributes in specific tactical situations, e.g., *Anchored Flanks*, *Interior Lines of Support* and *Restricted Avenues of Attack and Retreat* [3]. We agree with Cheeseman and Stutz that, "a strong interaction between the discovery program and the expert will be the common pattern in Knowledge Discovery in Databases (KDD) for the foreseeable future, because each have complementary strengths." [4] We are also in agreement with Fisher who states that, "conceptual clustering classification schemes can be a basis for effective inference of unseen object properties." [5]

We have previously published a series of algorithms employed in our TIGER (Tactical Inference Generator) test-bed program that are the 'building blocks' of the algorithms presented in this paper [6]. In TIGER, a tactical situation is described as two sets of opposing forces (REDFOR and BLUEFOR) within a set of two dimensional matrices that represent terrain, elevation and topography.[1] Units have a Range of Influence (ROI) that is defined by the unit type, terrain and line of sight [6]. Unit formations are represented by minimum spanning trees (MSTs) that groups units into self-supporting formations [8]. The TIGER function **FindPath**[2], based on A* search, is a least-weighted path algorithm that can incorporate ROI, as well as terrain and slope, into its calculations [9]. TIGER assumes that REDFOR are on the defensive and BLUEFOR is attacking.

The ClassIT algorithm uses category utility to compute the conditional probability (or predictability) of incorporating an instance (or in our implementation, a tactical situation) into a class [2]. The result is a clustering of instances into like classes.

2. The Anchored Flank Algorithm

We use the following generally accepted terms:

- **Flank** - either end of a mobile or fortified military position.
- **Anchored (or Refused) Flank** - a flank that is attached to or protected by terrain, a body of water, or defended fortifications.
- **Unanchored (or Open) Flank** - a flank that is not protected; also said to be "in the air [10]."

There was consistent agreement among our surveyed SMEs [3] that the attribute of 'anchored flanks' was present at certain Civil War battles when they were shown the canonical maps from the West Point Atlas: Antietam (Confederate right and left flanks), Chancellorsville (Union left flank), and Fredericksburg (Confederate left flank) [7].

In TIGER, the flanks of a line (or the flanking units of a line) are defined as the two maximally separated units in

[1] Because the authors did not have access to DTED format terrain and elevation files, we commissioned a third-party to produce elevation, terrain and topography maps from public domain sources including the West Point Atlas of American Wars [7, 15].

[2] **FindPath**(u, G, B, o, W) returns an optimal path from unit u's current position, given G a collection of "gap edges", B a collection of "barrier edges" to o, which is an objective, given as a location in graph coordinates and W is the unit u's world view. A solution path produced by **FindPath** must traverse at least one of the edges given in G and none of the edges in B. The algorithm was previously introduced in [6].

the MST defining a group of units).[3] A prerequisite for a line with anchored flanks (or an *anchored line*) is that there must not be any 'gaps' in the line; that is to say, that an anchored line must consist of a series of supporting units that have overlapping ROI, or fields of fire [6]. Therefore, an attacking (BLUEFOR) unit cannot navigate a path from its current position to an appropriately selected point behind an anchored line without crossing through an ROI of a different color.

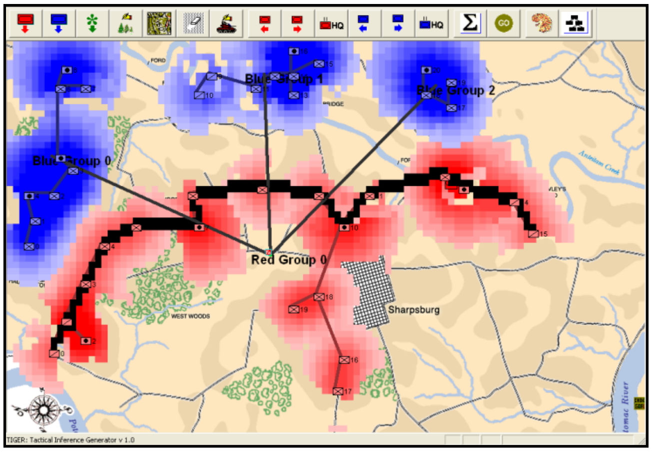

Figure 1. TIGER screen shot of 'flanking attribute' calculations for the battle of Antietam (September 17, 1862, 0600 hours). Note the thick black line that represents the MST spine of REDFOR Group 0, the extended vectors that calculate the Flanking Goal Objective Point and BLUEFOR and REDFOR ROI (red and blue shading). REDFOR (Confederate) has anchored flanks.

In informal terms our algorithm for determining the presence of anchored flanks is as follows. First, determine the *spine* of the MST group (see Figure 1) which is the unique path on the edges of the MST from one flank unit to the other flank unit. Second, determine if the spine traverses an uninterrupted line of ROIs that reach from one flank unit to the opposite flank unit. Next, locate an objective point that places the spine between the objective point and BLUEFOR units (see Figure 1).

If a legal path can be traced from any BLUEFOR unit to the objective point without passing through an ROI then at least one flank is unanchored. If no legal path can be found from any BLUEFOR unit to the objective point without passing through an ROI then both flanks are anchored. The ratio of the number of BLUEFOR units that had to cross through a REDFOR ROI over the total number of BLUEFOR units serves as a measure for the 'flanking attribute' employed by our classification system.

In addition to this 'flanking attribute' metric this algorithm also produces several additional useful attributes, in particular, whether the REDFOR line consists of an unbroken chain of ROI, whether BLUEFOR units must cross REDFOR ROI to reach the selected flanking objective point, and a ratio of compromised to uncompromised BLUEFOR units.

Algorithm for FlankingAttributeValue Function

// Determine if R, a set of REDFOR units, has anchored
// flanks given B, a set of BLUEFOR units. D is a distance
// threshold. W is the 'world view' used by FindPath.
// Matrices that represent terrain and elevation maps are
// global. Returns V, a real-valued attribute suitable for use
// within the ClassIT system.

FlankingAttributeValue (R, B, D, W)
{

    // Calculate the MSTs for REDFOR and BLUEFOR
    B_MST ← **ComputeGroupsByThreshold(B,W,D)**[4]
    R_MST ← **ComputeGroupsByNumber (R, 1)**[5]

    // Calculate ROI for REDFOR
    R_ROI ← **CalculateROI**(R)[6]

    // Find left and right flanks of REDFOR
    l ← **CalculateLeftFlank**(R)[7]
    r ← **CalculateRightFlank(**R)

    // Determine MST spine of R
    R_Spine ← **PlotMSTspine**(R,l, r)

    // Determine the center[8] of REDFOR
    R_Center ← **CalculateCenter**(R)[9]

    // Keep a count of how many BLUEFOR units
    // have a legal path free of R_ROI to their
    // respective objectives; initialize counter

---

[3] Maximally separated can be determined by Euclidean distance, time of travel, or by any other metric that can be used as an edge weight. TIGER uses travel cost as determined by the unit type indexed to terrain type and modified by the cost of the unit type traversing the slope of the elevation. NB: The cost of a flank unit to travel to the opposite flank unit may not be (and rarely is) symmetrical. This is because the two flank units may be of different unit types, may traverse different slopes, terrains, etc.

[4] This function groups units of the same color returning a forest of minimum spanning trees separated at least by distance threshold D and based on the current world view, W, which includes, e.g. line of sight, terrain considerations, etc.(see [6]).

[5] This function groups units of the same color into a fixed number of N subgroups using edge weighting or distance; see [6]. Since we specify only one group here, the function simply computes the REDFOR MST.

[6] Given a set of units and ROI values for each unit type the function maps the ROI into a two-dimensional matrix.

[7] The algorithm for this function was published in [6]. This function returns the respective flanks (maximally separated units) within a group, or MST.

[8] The center of a group of units is the sum of the unit locations weighted by unit's type strength modifier multiplied by the unit's strength.

[9] The algorithm for this function was published in [6].

```
N ← 0

// For each BLUEFOR group, represented by an
// MST in the forest of BLUE MSTs
for each BG_m in B_MST

    // Calculate center of this BLUEFOR group
    B_Center ← CalculateCenter(BG_m)

    // Find first REDFOR ROI-free objective
    // beyond the MST spine of R (R_Spine)
    // along the ray from B_Center to R_Center
    o ← FindOpenPoint(B_Center, R_Center,
                              R_Spine, R_ROI)

    // For each unit in this group see if a legal path
    // exists to the objective point o
    for each u in BG_m
        if (FindPath(u,Ø,R_ROI,o,W))
            N ← N + 1

// Return the ratio of  the number of BLUEFOR
// units with R_ROI clear paths to their respecrive
// objectives to the total number of BLUEFOR unts
return (N / |B| )
}
```

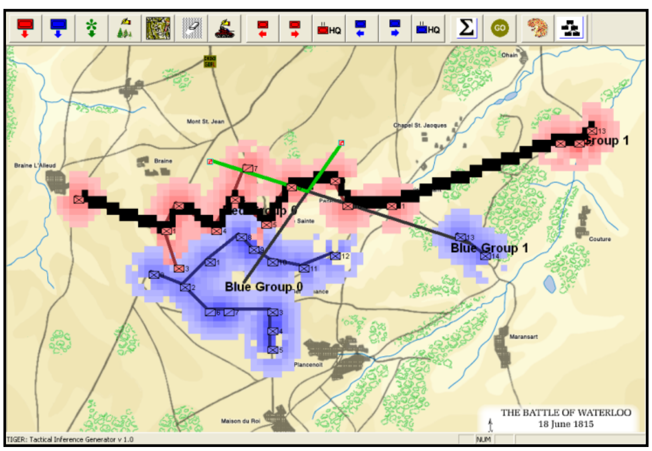

Figure 2. Example of MST Spine being calculated across all REDFOR groups at the battle of Waterloo (June 18, 1815, 1600 hours). Note: REDFOR (Anglo-Allies) have two groups; Group 1 in the east is the Prussian forces. TIGER screen shot.

3. The Interior Line Algorithm

We use the following generally accepted terms:

Interior Lines – The military circumstance of either being able to move over a shorter distance to execute maneuvers and effect reinforcements than the enemy or possessing a more efficient transportation method or faster units than the enemy.

Interior Lines are defined relative to those of the enemy; consequently they can be categorized as either interior (shorter distance between flanks than the distance between the enemy's flanks) or exterior (greater distance between flanks than the enemy's flanks [12].

There was consistent agreement among our surveyed SMEs that the attribute of 'interior lines' was present at certain Civil War battles when they were shown the canonical maps from the West Point Atlas of Chancellorsville (Confederate), Antietam (Confederate), Gettysburg (Union) and Wilderness(Union) [3,7].

In informal terms our algorithm for determining the presence of interior lines is as follows: First, find the left and right flank units for REDFOR and BLUEFOR groups. Next, find the least weighted path between the flank units of each group using **FindPath** (checking in both directions because the costs are not symmetrical due to different unit types, terrains, slopes, etc.). Last, subtract the smallest cost (returned by **FindPath**) for BLUEFOR from the smallest cost (returned by **FindPath**) for REDFOR.

Algorithm for InteriorLinesValue Function

```
// Determine if R, a set of REDFOR units, or B, a set of
// BLUEFOR units, has the attribute of interior lines and
// return a real-valued attribute suitable for use within the
// ClassIT system. W is the 'world view' used by FindPath.

InteriorLinesValue (B, R, W)
{
    // Find left and right flanks of REDFOR
    l ← CalculateLeftFlank(R)
    r ← CalculateRightFlank(R)

    // Calculate ROI for BLUEFOR
    B_ROI ← CalculateROI(B)

    // Find the path with the least cost between l and r
    // (check both directions), store in B_Path
    B_Path ← Min ( FindPath (l, Ø, R_ROI, r, W),
                   FindPath (r, Ø, R_ROI, l, W))

    // Find left and right flanks of BLUEFOR
    l ← CalculateLeftFlank(B)
    r ← CalculateRightFlank(B)

    // Calculate ROI for REDFOR
    R_ROI ← CalculateROI(B)

    // Find the path with the least cost between l and r
    // (check both directions), store in R_PATH
    R_Path ← Min(  FindPath (l, Ø, B_ROI, r, W),
                   FindPath (r, Ø, B_ROI, l, W))

    // Subtract the minimum REDFOR path from the
    // minimum BLUEFOR path and return the Interior
    // Line metric
    return(B_Path - R_Path)
}
```

If the value returned is greater than zero, then BLUEFOR has interior lines. If the value returned is less than zero, then REDFOR has interior lines. The greater the absolute value of the metric, the larger the disparity between interior and exterior lines.

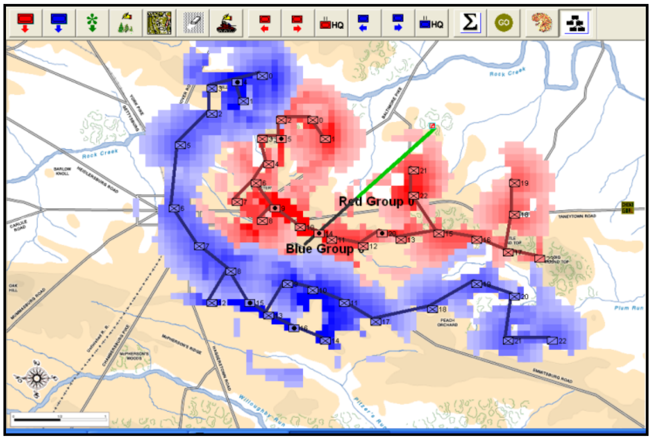

Figure 3. Example of REDFOR (Union) having Interior Lines at the battle of Gettysburg (July 3, 1863, 0600 hours). Note: TIGER displays defending forces as REDFOR. TIGER screen shot.

4. Restricted Avenue of Attack and Restricted Avenue of Retreat Algorithms.

We use the following generally accepted terms:

- **Avenue of Approach** also **Avenue of Attack –** A ground route of an attacking force of a given size leading to its objective or to key terrain in its path [13].
- **Avenue of Retreat** – the ground route of a retreating defending force.
- **Choke Point** - A choke point is a geographical feature on land such as a valley or defile which an armed force is forced to pass, sometimes on a substantially narrower front, and therefore greatly decreasing its combat power, in order to reach its objective [14].

There was consistent agreement among our surveyed SMEs that the attribute of ‘restricted avenues of retreat’ and/or ‘restricted avenues of attack’ were present at certain Civil War battles when they were shown the canonical maps from the West Point Atlas: Chancellorsville (Union), Antietam (Confederate), Fredericksburg (Union) and Kasserine Pass (Allied and Axis) [3, 7, 15].

The point (C) that REDFOR wishes to retreat to requires *a priori* knowledge of the strategic situation which is beyond the scope of the tactical ‘snapshot’ that is presented to TIGER for analysis. Consequently, C, is set by an SME (see Figure 4 for an example where C has been set as a point on the southern bank of the Potomac River that the Confederate army must pass through during a retreat to its strategic base in Virginia).

In informal terms our algorithm for determining the presence of Restricted Avenues of Retreat for REDFOR is as follows: an SME, using TIGER, sets the REDFOR Retreat Choke Point Goal (C) and TIGER calculates the BLUEFOR ROI. Next, for each group in REDFOR, or until **FindPath** returns failure, find a path from the center of the group to C that is completely disjoint from previous REDFOR retreat paths. The value returned, the REDFOR Chokepoint Value (RC), corresponds roughly to a notion of “bandwidth” to a single objective, the Choke Point Goal. Since we are operating on a map with an overlayed square grid, the number of disjoint access paths to a single objective point is an integer between 0 and 8, inclusive. Thus the RC, defined as:

$$RC = \frac{1}{2^{(\text{NumberChokePoints} - 1)}}$$

is a number in the range [0.007813,1.0], where a smaller number means REDFOR has more avenues of retreat, and a value of 1.0 means REDFOR has only a single avenue of retreat (see Figure 4, below, for an example of RC = 1.0).

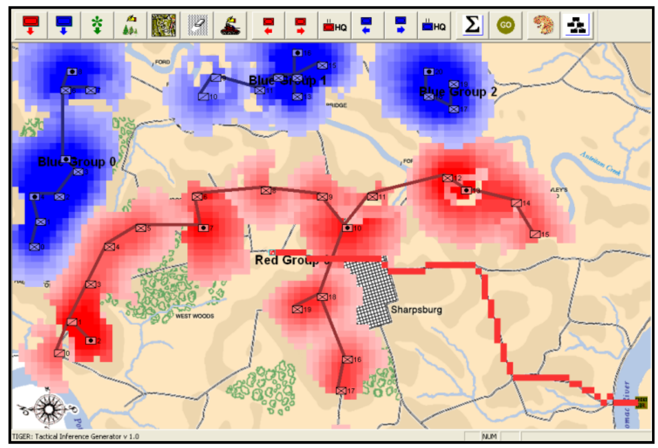

Figure 4. Example of REDFOR (Confederates) having a Restricted Avenue of Retreat (marked by thick red line from Red Group 0 to lower right hand corner of map) at the battle of Antietam (September 17, 1862, 0600 hours). REDFOR has only one avenue of retreat across the Potomac to C, the Choke Point goal. TIGER screen shot.

Algorithm for AvenuesOfRetreatValue Function

```
// Given R, a set of REDFOR units and B, a set of
// BLUEFOR units, calculate the number of choke points
// between R and the Choke Point Goal (C) and
// return a real-valued attribute suitable for use
// within the ClassIT system. W is the ‘world view
// used by FindPath. D is a distance threshold.

AvenuesOfRetreatValue (B, R, C, W, D)
{
    // BARRIER is initialized to hold BLUEFOR ROI
    BARRIER ← CalculateROI(B)

    // Initialize N, the NumberofChokePoints
    N ← 0

    // Calculate the MSTs for REDFOR
    R_MST ← ComputeGroupsByThreshold(R,W, D)
```

```
     // For each REDFOR group, represented by an
     // MST in the forest of REDFOR MSTs
     for each RG_m in R_MST

          // Calculate center of this REDFOR group
          R_Center ← CalculateCenter(RG_m)

          while(P ← FindPath(R_Center,∅,BARRIER,C,
                                                   W))
               N ← N+1

          BARRIER ← BARRIER ∪ P

     // Return the REDFOR Chokepoint Value
     return(1 / Power (2,N-1))
}
```

The algorithm for determining the presence of Restricted Avenues of Attack for BLUEFOR is identical to the algorithm for determining the presence of Restricted Avenues of Retreat except that **FindPath** is called without using enemy ROI and that the Choke Point Goal (C) is not set by an SME; rather C is set as the center of REDFOR.

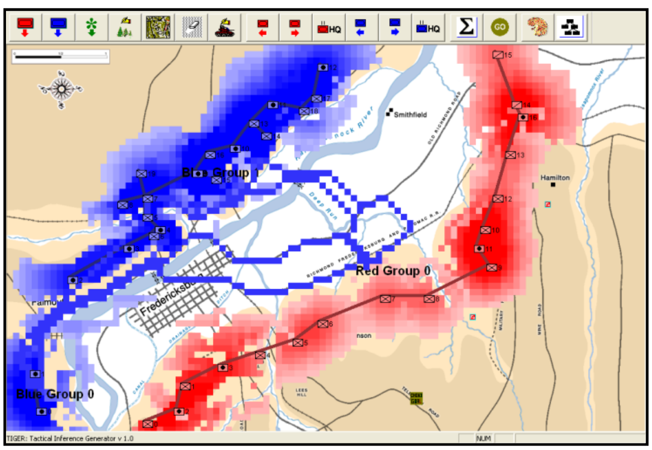

Figure 5. Example of BLUEFOR (Union) having Restricted Avenues of Attack (marked by thick blue lines from Blue Group 0 and Blue Group 1 across the Rappahannock River via pontoon bridges) at the battle of Fredericksburg (December 10, 1862, 0600 hours). TIGER screen shot.

Algorithm for AvenuesOfAttackValue Function

```
// Given R, a set of REDFOR units and B, a set of
// BLUEFOR units, calculate the number of choke points
// between B and the Choke Point Goal (C) and
// return a real-valued attribute suitable for use
// within the ClassIT system. W is the 'world view'
// used by FindPath. D is a distance threshold.

AvenuesOfAttackValue (B, R, C, W, D)
{
     // Determine the center of REDFOR
     R_Center ← CalculateCenter(R)

     // Initialize N, the NumberofChokePoints
     N ← 0

     // BARRIER is initialized empty
     BARRIER ← ∅

     // Calculate the MSTs for BLUEFOR
     B_MST ← ComputeGroupsByThreshold(B,W, D)

     // For each BLUEFOR group, represented by an
     // MST in the forest of BLUEFOR MSTs
     for each BG_m in B_MST

          // Calculate center of this BLUEFOR group
          B_Center ← CalculateCenter(BG_m)

          while(P ← FindPath(B_Center,∅,BARRIER,
                                       R_Center,W))
               N ← N+1

          BARRIER ← BARRIER ∪ P

     // Return the BLUEFOR Chokepoint Value
     return (1 / Power (2,N-1))
}
```

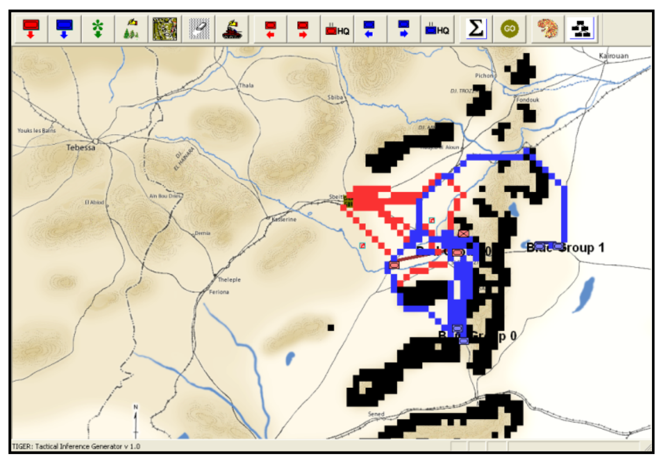

Figure 6. Example of BLUEFOR (Axis) having Restricted Avenues of Attack (marked by thick blue lines) at the battle of Kasserine Pass, February 14, 1943). Note that the black areas have greater slopes than the unit type (armor in this case) allows for transit. Also, note that the slope restrictions are calculated as part of the FindPath function and, consequently are calculated only on an 'as need' basis. Mountain ranges to the west were not considered by the function and, therefore, their slopes are not marked as impassable. TIGER screen shot.

4. Results

We present, below, the results of running the above four algorithms on fifteen historical tactical situations taken from the West Point Atlas. [7, 15]

Table 1 displays the values returned by our Interior Line Value algorithm (3, above). Negative numbers indicate the presence of ‘interior lines’; positive numbers indicate the presence of ‘exterior lines’, or the condition of the attacker having shorter lines. The Union defensive line on the third day of the battle of Gettysburg returned the lowest value (-231,487). This was to be expected as this is the famous ‘fish hook’ defensive line; indeed it is the canonical example of an ‘interior line’. The Confederate defensive line at Antietam, which was also identified by our surveyed SMEs as an example of an interior line, also returns a negative number; though it clearly does not exhibit the condition of ‘interior lines’ as severely as the Union line on the third day of Gettysburg.

Table 2 displays the values returned by our Anchored Flank Algorithm (2, above). The value zero indicates that both flanks are ‘open’ or unanchored and the value one indicates that both flanks are anchored. The Confederate lines at Antietam and Fredericksburg were both identified by our surveyed SMEs as an example of anchored flanks and our results are in agreement with the SME analysis.

Table 3 displays the values returned by our Restricted Avenues of Retreat algorithm (4, above). The Confederate position at Antietam was identified by our surveyed SMEs as an example of a Restricted Avenue of Retreat and our results are in agreement with the SME analysis (Figure 4 clearly shows the one Avenue of Retreat available to the Confederates at Antietam).

Table 4 displays the values returned by our Restricted Avenues of Attack algorithm (4, above). The Union position at Fredericksburg (in which the Union forces had to cross pontoon bridges to attack) and the German positions at Kasserine Pass (where the Germans had to maneuver through mountain passes to attack) were identified by our surveyed SMEs as an example of a Restricted Avenue of Attack or Approach and our results are in agreement with the SME analysis. Figures 5 and 6 clearly show the Restricted Avenues of Attack identified by TIGER.

Table 1. Interior Line Values returned by TIGER analysis of 15 tactical situations from the West Point Atlas.

Tactical Situation:	Value:
Gettysburg, July 3, 1863; 1200 hours	-231497
Lake Trasimene, 217 BCE	-115079
Waterloo June 18; Counterattack 1930 hours	-94884
Gettysburg, July 1, 1863; 14:30 hours	-92279
Gazala, Libya, 27 May 1942	-92273
Antietam, September 17, 1862; 0600 hours	-54097
Kasserine Pass, February 14, 1943	-19347
Shiloh, April 6, 1862; 1200 hours	12418
Shiloh, April 7, 1862 (Counterattack)	34188
Shiloh, April 6, 1862; 0900 hours	34977
Fredericksburg, December 10, 1862; Crossing	35625
Waterloo June 18, 1815; 1000 hours	50007
Fredericksburg, Dec. 13, 1862; 0600 hours	133451
Waterloo June 18, 1815; 1600 hours	139848
Kasserine Pass, February 19, 1943	262357

Table 1. Negative numbers indicate that the defender has interior lines; the lower the number, the greater the advantage.

Table 2. Anchored Flank Values returned by TIGER analysis of 15 tactical situations from the West Point Atlas.

Tactical Situation:	Value:	Overlapping ROIs	NumCrossed / BLUEFOR	Analysis:
Kasserine Pass, February 14, 1943	0	FALSE	0/3	Neither Flank Anchored
Shiloh, April 6, 1862; 0900 hours	0	FALSE	0/9	Neither Flank Anchored
Shiloh, April 7, 1862 (Counterattack)	0	FALSE	0/9	Neither Flank Anchored
Gettysburg, July 1, 1863; 14:30 hours	0	FALSE	0/10	Neither Flank Anchored
Shiloh, April 6, 1862; 1200 hours	0	FALSE	0/9	Neither Flank Anchored
Kasserine Pass, February 19, 1943	0	FALSE	0/3	Neither Flank Anchored
Waterloo June 18, 1815; 1600 hours	0.29	FALSE	4/14	1 Flank Anchored
Gettysburg, July 3, 1863; 1200 hours	0.64	TRUE	14/22	1 Flank Anchored
Gazala, Libya, 27 May 1942	0.67	TRUE	7/9	1 Flank Anchored
Waterloo June 18, 1815; 1000 hours	0.71	FALSE	10/14	1 Flank Anchored
Waterloo June 18; Counterattack 1930 hours	0.77	FALSE	10/13	1 Flank Anchored
Fredericksburg, December 10, 1862; Crossing	1	TRUE	19/19	Both Flanks Anchored
Fredericksburg, December 13, 1862; 0600 hours	1	TRUE	15/15	Both Flanks Anchored
Antietam, September 17, 1862; 0600 hours	1	TRUE	20/20	Both Flanks Anchored
Lake Trasimene, 217 BCE	1	TRUE	10/10	Both Flanks Anchored

Table 2. The relationship between Anchored Line Value and the number of BLUEFOR units that must cross an enemy ROI to reach the Flanking Goal Objective Point.

Table 3. Defender Avenues of Retreat as determined by TIGER analysis of 15 tactical situations from the West Point Atlas.		
Tactical Situation:	Value:	Choke Points:
Gazala, Libya, 27 May 1942	0.007813	8
Waterloo June 18; Counterattack 1930 hours	0.007813	8
Fredericksburg, December 10, 1862; Crossing	0.007813	8
Gettysburg, July 1, 1863; 14:30 hours	0.015625	7
Kasserine Pass, February 19, 1943	0.015625	7
Waterloo June 18, 1815; 1600 hours	0.015625	7
Gettysburg, July 3, 1863; 1200 hours	0.015625	7
Waterloo June 18, 1815; 1000 hours	0.015625	7
Fredericksburg, December 13, 1862	0.015625	7
Kasserine Pass, February 14, 1943	0.0625	5
Shiloh, April 7, 1862 (Counterattack)	0.0625	5
Shiloh, April 6, 1862; 0900 hours	0.5	2
Shiloh, April 6, 1862; 1200 hours	0.5	2
Antietam, September 17, 1862; 0600	1	1
Lake Trasimene, 217 BCE	1	1

Table 3. TIGER analysis of avenues of retreat of 15 historical tactical situations from the West Point Atlas.

Table 4. Attacker Avenues of Attack (or Avenues of Approach) as determined by TIGER analysis of 15 tactical situations from the West Point Atlas.		
Tactical Situation:	Value:	Choke Points:
Waterloo June 18, 1815; 1600 hours	0.007813	8
Gazala, Libya, 27 May 1942	0.007813	8
Fredericksburg, December 13, 1862; 0600 hours	0.007813	8
Antietam, September 17, 1862; 0600 hours	0.007813	8
Gettysburg, July 1, 1863; 14:30 hours	0.015625	7
Waterloo June 18, 1815; 1000 hours	0.015625	7
Waterloo June 18; Counterattack 1930 hours	0.015625	7
Shiloh, April 6, 1862; 0900 hours	0.03125	6
Shiloh, April 7, 1862 (Counterattack)	0.03125	6
Shiloh, April 6, 1862; 1200 hours	0.03125	6
Gettysburg, July 3, 1863; 1200 hours	0.03125	6
Lake Trasimene, 217 BCE	0.03125	6
Kasserine Pass, February 14, 1943	0.0625	5
Kasserine Pass, February 19, 1943	0.0625	5
Fredericksburg, December 10, 1862; Crossing	0.125	4

Table 4. TIGER analysis of avenues of attack of 15 historical tactical situations from the West Point Atlas.

5. Conclusions

While the algorithms presented in this paper play a crucial role in our future research into the applicability of using ClassIT to provide important inferences about similar tactical situations, we suggest that these algorithms (especially when combined with our previously published algorithms in [6]) also represent a more generally useful suite of algorithms for analysis within the field of Computational Military Reasoning or Tactical Planning. To the best of our knowledge algorithms for the determination of anchored flanks, choke points and interior lines have not been previously presented.

These algorithms have been tested on 15 tactical situations drawn from the West Point Atlas series and we are confident that they will be as robust when applied to other tactical situations that are outside the realm of ‘historical’ battles.

We are currently conducting experiments using the output of these algorithms as the attributes used by ClassIT system to classify and evaluate historical situations. We believe that we will need to incorporate more attributes than those produced by the algorithms described in this paper. Because we strongly agree with Cheeseman, Stutz and others of the importance of SME input in the validation of attributes used for classification we intend to conduct other SME surveys that we hope will identify other attributes to be used for the classification of tactical situations.

We also believe that we will need to increase the number of instances (tactical situations) input into the ClassIT system. Currently this in arduous manual process, but we hope to streamline the process by adding the ability to TIGER to read in DTED files and automatically input REDFOR and BLUEFOR unit locations.

Acknowledgements

This research was partially funded by a “seedling” grant from DARPA administered by ISLE (Institute for the Study of Learning and Expertise).

We would like to thank Daniel Shapiro and Frank Bosch at ISLE and Dan Adams at Scitor Corporation.

Ralph Sharp, III of Wizard IT Services produced the topographical, elevation and terrain maps for TIGER.

We would like to thank Colonel John R. Surdu for his consistent support of our research.

References

[1]. Kewley, Robert H. and Embrechts, Mark J. "Computational Military Tactical Planning System." *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 32.2 (May 2002): 161-71.

[2]. Gennari, John H., Langley, Pat and Fisher, Doug. "Models of Incremental Concept Formation." *Artificial Intelligence* 40.1 (September 1989): 11-59.

[3]. Sidran, David E. "Report of First Survey of Subject Matter Experts for Dissertation Research." Dept. of Computer Science, University of Iowa, Summer 2008. http://www.cs.uiowa.edu/~dsidran/DP/Survey1Results.pdf.

[4]. Cheeseman, Peter and Stutz, John. "Bayesian Classification (AutoClass): Theory and Results." 1996. *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: American Association for Artificial Intelligence, 1996. 153-80.

[5]. Fisher, Douglas H. "Knowledge Acquisition Via Incremental Conceptual Clustering." *Machine Learning* 1.2 (July 1987): 139-72.

[6]. Sidran, David E. and Segre, Alberto Maria. "Implementing the Five Canonical Offensive Maneuvers in a CGF Environment." *Workshop Papers*. Simulation Interoperability Workshop 1. Orlando, FL: Simulation Interoperability Standards Organization, 2007. 159 - 166.

[7]. Esposito, Vincent J. (Editor). *The West Point Atlas of American Wars, Volume 1 (1689-1900)*. New York: Frederick A. Praeger, 1964.

[8]. Kruskal, J. B. "On the Shortest Spanning Subtree and the Traveling Saleman Problem." *Proceedings of the American Mathematical Society* 7 (1956): 48-50.

[9]. Hart, P.E., Nilsson, N.J. and Raphael, B. "A Formal Basis for the Heurstic Determination of Minimum Cost Paths." *IEEE Transactions on Systems, Science and Cybernetics* 4.2 (1968).

[10]. *Definition of Civil War Terms*. 11/26/2008 <http://www.civilwarhome.com/terms.htm>.

[11]. Bresenham, Jack E. "Algorithm for Computer Control of a Digital Plotter." *IBM Systems Journal* 4.1 (1965): 25-30.

[12]. Yates, Edward. *Elementary Treatise on Tactics and on Certain Parts of Strategy*. London: Parker, Furnivall & Parker, 1853.

[13]. *Joint Publication 1-02: Department of Defense Dictionary of Military and Associated Terms*. Washington, DC: Department of Defense, 2005.

[14]. Wikipedia contributors. *Choke Point*. 19 November 2008. Wikipedia, The Free Encyclopedia. 26 November 2008 <http://en.wikipedia.org/w/index.php?title=Choke_point&oldid=252857939>.

[15]. Esposito, Vincent J. (Editor). *The West Point Atlas of American Wars, Volume 2 (1900-1953)*. New York: Frederick A. Praeger, 1964.

Authors' Biographies

David Ezra Sidran is a doctoral candidate in the department of computer science at the University of Iowa (Iowa City, IA). He received his Master of Computer Science from the University of Iowa in 2005. He has worked as a computer game designer since 1985, receiving two Codie Awards from the Software Publishers Association and was named Designer of the Year by *Computer Entertainer* magazine. He has authored numerous commercial computer wargames.

Alberto Maria Segre is Professor and Associate Chair of the Computer Science Department at the University of Iowa (Iowa City, IA) and the Gerard P. Weeg Faculty Scholar in Informatics. He holds a B.A. in Music Theory and a B.S., M.S. and Ph.D. in Electrical Engineering, all from the University of Illinois at Urbana-Champaign. His research interests focus on distributed algorithms for discrete optimization problems, with emphasis on algorithmic problems in the biological and health sciences.