社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-B101-1 (5/4)

安全領域の計算アルゴリズムとその守備戦略への適用

An algorithm to compute the safety region and its application to the defense strategy

稲垣多朗、村上和人、成瀬正

Taro INAGAKI, Kazuhito MURAKAMI and Tadashi NARUSE

愛知県立大学 情報科学部

Aichi Prefectural University, Information Science and Technology

is071005@cis.aichi-pu.ac.jp, {naruse,murakami}@ist.aichi-pu.ac.jp

Abstract

We have proposed a new concept of "safety region" which is an index for determining the positions of the defense robots[1]. It is defined as a region that the teammate robot(s) can keep the goal when an opponent robot shoots the ball from the inside of that region if teammates are positioned properly according to their defense strategy. Since it is difficult to obtain the accurate safety region in a short time, we need an algorithm that computes an approximate safety region. We proposed such algorithm in the previous paper[1]. However, the safety region obtained by the algorithm is less accurate than the true one. It is required more accurate algorithm. In this paper, we propose an improved algorithm to compute the approximate safety region. It realizes 95% of accuracy and less than 1 msec of computation time, which is enough for our RoboCup application. We also propose a defense strategy based on the safety region considering the positions of the opponent robots and the pass direction, and show that it works well for determining the positions of the defense robots.

1 はじめに

RoboCup 小型ロボットリーグ (SSL) では，年々攻撃や守備に用いられる戦略が高度化しており，ダイレクトプレイ[2]のような複数台のロボットの連携による攻撃が用いられている．そのような攻撃を防ぐためには，守備ロボットの配置や数を決定するための，リアルタイムにフィールド全体の状況を計算，評価することのできる指標が必要となる．

これまでに，マーク対象を決定するための指標[3]や、連携シュートの際のパス方法を決定するための指標[4]が提案され，試合で用いられている．我々は，ロボットの行動，特に守備ロボットの数を決定するための指標として，"安全領域指標"を提案し，それに基づいた守備ロボットの数の決定方法について議論している[1]．しかしながら，提案した安全領域を計算するアルゴリズムは精度が低く，より精度の高い近似手法が求められている．

本稿では，改良した近似安全領域の計算アルゴリズムを提案する．その計算精度は 95% ，計算時間は 1 ミリ秒未満である．これは，実際の SSL の試合で用いるのに十分な時間である．さらに，相手のロボットの位置やパスの方向を考慮した安全領域指標に基づいた守備戦略を提案し，それが守備ロボットの配置に効果的であることを示す．

2 安全領域

この節では，"安全領域" の定義と，その計算方法について述べる．

2.1 定義

安全領域を，フィールド上の領域であって，その領域内から自チームのゴールへ向けてシュートが行われたときに，守備戦略に従って配置された守備ロボットがそのシュートを防ぐこと（ゴールを守ること）が可能な領域として定義する．安全領域でない領域は非安全領域と呼ぶ．以降の議論では，フィールド上を直線的に転がるシュートのみを考慮し，その他のシュート (いわゆるチップシュートやカーブシュート) については今後の検討に委ねる．

2.2 安全領域の計算

安全領域の計算はどのようなシュート動作 (例えば単独シュートや連携シュート) を行うか，また，どのような守備 (例えば守備戦略や守備ロボットの台数) を行うかによって異なる．真の安全領域を求めるのには非常に時間がかかるので，近似安全領域の計算方法を記述する．以降の議論では，ダイレクトプレイ[2]に対する近似安全領域の計

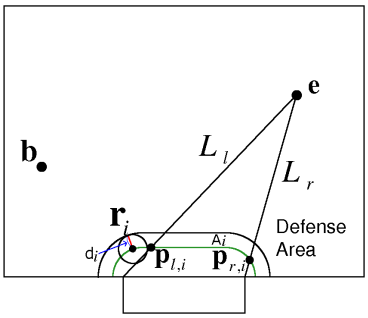

Figure 1: Definition of symbols

算方法について述べる．ダイレクトプレイとは，最初のロボットが 2 台目のロボットへパスを行い，2 台目のロボットがボールを保持せずにゴールへ向けて直接シュートを行う戦略である．最初のロボットによる単独シュートも同様に計算できる．

2.2.1 近似計算方法

守備ロボットは守備戦略に従って適切な座標に移動するものとする．また，ゴールキーパーはディフェンスエリアの内側をその境界線に沿って動き，それ以外の守備ロボットはディフェンスエリアの外側をその境界線に沿って動くものとする．まず，記号を定義する（図 1 参照）．時刻 t におけるボールの座標を $\mathbf{b}$，ロボットがシュートを行う座標を $\mathbf{e}$ とし，$\mathbf{r_i}$ を守備ロボット i の時刻 t における座標とする．L_r, L_l をそれぞれ始点を $\mathbf{e}$，終点をゴールの両ポストとする線分とする．それから，d_i を $\mathbf{r_i}$ とディフェンスエリアの境界線の間の距離 (ロボットの半径に等しい) とし，ディフェンスエリアの境界線と距離 d_i 隔てた曲線を A_i とする．$\mathbf{p_{r,i}}, \mathbf{p_{l,i}}$ をそれぞれ A_i と L_r, L_l の交点とする．$\mathbf{p_{r,i}}, \mathbf{p_{l,i}}$ と $\mathbf{r_i}$ の間の，曲線 A_i 上の長さをそれぞれ $D_{r,i}$,$D_{l,i}$ とする．

パスを行うロボットが時刻 t においてボールを保持し，そしてダイレクトプレイを行うと仮定する．以下の計算式によって安全領域の計算を行う．

まず，

$$t_p = \frac{||\mathbf{e}-\mathbf{b}||}{v_p}, \quad t_s = \frac{||\mathbf{p_{j,i}}-\mathbf{e}||}{v_s} \ (j=r,l), \quad t_i = \frac{v_i}{a_i}, \tag{1}$$

を計算する．t_p と t_s は，それぞれボールが $\mathbf{b}$ から $\mathbf{e}$ へ速度 v_p で，$\mathbf{e}$ から $\mathbf{p_{j,i}}$ へ速度 v_s で移動したときの所要時間である．t_i はロボット i が停止状態から加速度 a_i で最大速度 v_i まで加速する時の所要時間である．

もし $t_i > t_s$ であるなら，

$$D_{j,i} < \frac{1}{2}a_i(t_p+t_s)^2 + R \quad (j=r,l) \tag{2}$$

を，そうでないなら

$$D_{j,i} < \frac{1}{2}a_i {t_i}^2 + v_i(t_p+t_s-t_i) + R \quad (j=r,l) \tag{3}$$

を計算する．ここで，R はロボットの半径とボールの半径の和である．式 (2)，式 (3) はロボットの現在位置と目標位置の間の長さ (左辺) とロボットの移動距離 (右辺) を比較する式である．

式 (2) または式 (3) を満たすか，$\mathbf{b}$ と $\mathbf{e}$ の間にパスコースが存在しない場合，$\mathbf{e}$ は安全慮域内の点とする．守備ロボットが複数である場合，式 (2)，(3) をいずれかのロボットが満たす場合，$\mathbf{e}$ は安全領域内の点とする．また，いずれかのロボットについて $||\mathbf{p_{j,i}} - \mathbf{r_i}|| < \mathbf{R}$ を満たす場合，シュートコースが存在しないと見なし，$\mathbf{e}$ は安全領域内の点とする．

フィールド上のすべての点（格子状の点）について，式 (1)，(2)，(3) の計算を行い，近似安全領域の判定を行う．しかし，すべての点について計算を行うと，実際の SSL の試合で用いる際に与えられる時間を大きく上回ってしまう．

2.3 計算時間の短縮

計算時間の短縮のために，Coarse to Fine 法を用いる．まず，フィールド全体を W×H 個の領域に分割する．通常，W と H は 2 の累乗である．フィールド上の各点に対応する配列 SA$[W][H]$ を用意する．配列 SA$[W][H]$ の (i,j) 要素 SA$[j][i] (0<i<H, 0<j<W)$ はフィールド上の位置を表す．

2.3.1 計算アルゴリズム

Coarse to Fine 法を用いた近似安全領域の計算アルゴリズムを以下に記す．

1. ボールを初期位置 $\mathbf{b}$ に配置し，守備戦略に従って各守備ロボットを配置する．
2. $i=0$，$j=0$ とする．
3. $N=2^n (N<W, N<H)$ とする．(j,i) に対応する領域について節 2.2 の計算方法に従って近似安全領域の計算を行い，その結果が安全領域であるなら SA$[j][i]$= 1，非安全領域であるなら SA$[j][i]$= 0 とする．$(j, i+N-1)$，$(j+N-1, i)$，そして $(j+N-1, i+N-1)$ それぞれに対応する領域についても近似安全領域の判定を行い，その結果を対応する SA の要素に代入する．
4. 計算した 4 点の SA の値がすべて一致するなら，$j \le k < j+N$，$i \le l < i+N$ であるすべての SA$[k][l]$ に一致した値を代入する．一致しなければ，$N=N/2$ として，ステップ 3 と 4 を $N=1$ となるまで分割された 4 つの部分について計算する．
5. $i=i+N$ とする．$i<H$ であるなら，ステップ 3 へ移る．
6. $i=0$，$j=j+N$ とする．$j<W$ であるなら，ステップ 3 へ移る，そうでなければ計算を終了する．

3 実験

ここでは，ダイレクトプレイに対する安全領域の実験結果を示す．実験には RoboDragons の守備戦略[5]を用いる．

その理由は，戦略の細部を把握しているので解析が容易だからである．

3.1 実験方法

安全領域はフィールド上のすべての点について求めるべきであるが，それは非常に困難であるので，フィールド全体を 40mm 間隔の格子状の領域に分割し，各分割領域の中心点に対して計算を行い，その結果を各格子内の領域の結果とする．2 節で述べた計算アルゴリズムを用いて近似安全領域を求める．

安全領域は実際のロボットを動かして求めるべきであるが，実際にロボットを用いて連携シュートを行って安全領域を求めるとロボットに非常に負荷がかかる．また，実際のロボットには動作の不確定性がある．そのため，RoboDragons システムのシミュレータ[1] を用いて連携シュートのシミュレーションを行う．各ロボットやボールが物理モデルに沿って動けばシミュレーション結果と実際の安全領域は一致するはずであるので，シミュレーション結果から導出した安全領域を正答とし，提案手法とシミュレーションの結果を比較する．以下に，RoboDragons システムのシミュレータを用いたダイレクトプレイによる連携シュートシミュレーション方法を示す．

1. フィールドを n 個の格子状の領域に分割し，ボールの初期位置 $\mathbf{b}$ を決める．本実験では $n = 14888$ である．ボールを初期位置 $\mathbf{b}$ に置く．守備ロボットを RoboDragons の守備戦略に従って配置する．以降，守備ロボットは戦略システムに従って動作させる．シュートを行う位置 $\mathbf{e_i}$ はすべての分割した領域の中央点のうちの一つとする．
2. 時刻 t において，$\mathbf{b}$ からシュート位置 $\mathbf{e_i}$ に向けてパス速度 v_p でボールを移動させる．
3. シュート位置 $\mathbf{e_i}$ にボールが到達した時刻 t_e において，$\mathbf{e_i}$ から，各守備ロボットから最も遠いシュートコース（図 1 の L_r または L_l）上をシュート速度 v_s でボールを移動させる．(ただし，シュートコースは時刻 t で決める．)
4. ボールがゴール内に入った時，$\mathbf{e_i}$ を非安全領域内の点とし，そうでないとき $\mathbf{e_i}$ を安全領域内の点とする．
5. 2〜4 の操作をフィールド上のすべての分割した領域の中央点 $\mathbf{e_1}...\mathbf{e_n}$ に対して行う．

本実験において用いたパス速度 v_s，シュート速度 v_p は，それぞれ，小型リーグで一般的な値である $4.0m/sec$，$8.0m/sec$ とした．提案手法による安全領域計算，シミュレーション時に用いたボールの初期位置は，JapanOpen2009 の 4 試合，RoboCup2009 の 4 試合，計 8 試合のログファイルから，*Kick Off*，*Direct Freekick*，*Indirect FreeKick* が開始された時のボール座標を候補点として抽出し，これらからランダムに選んだ．候補点の総数は 458 である．

[1] 計算機上で，ボールや任意の数のロボットを物理法則に則って動作させる機能，任意の場所へボールとロボットを移動 (ワープ) させる機能，ロボットを戦略システムに従って動作させる機能を持つ．

3.2 実験結果

3.2.1 適合率

ロボット 2 台による RoboDragons の守備戦略に対し，シミュレーション結果による安全領域と提案手法による安全領域を導出し比較した．結果の一例を図 2 と図 3 に示す．提案手法による安全領域計算に用いたロボットの加速度および速度の値は，それぞれ，$a_i = 2.0m/sec^2$，$v_i = 0.6m/sec$ とした．使用した加速度と速度の値は，シュートシミュレーション時における測定値である．

これらの図で，赤色の領域はシミュレーションでは非安全領域であるが近似計算では安全領域である領域，緑色の領域は近似計算では非安全領域であるがシミュレーションでは安全領域である領域，そして青色の領域はシミュレーションと近似計算ともに非安全領域である領域であり，白色の領域はシミュレーションと近似計算ともに安全領域である領域である．フェール・セーフの考え方からいえば，実験結果の図において，赤の領域の面積が 0 になり，緑の領域の面積ができる限り小さくなることが望ましい．近似精度を評価するために，次の値を定義する．

$$R_c = \frac{A_b + A_w}{A_b + A_w + A_g + A_r} \tag{4}$$

$$R_a = \frac{A_g}{A_b + A_w + A_g + A_r} \tag{5}$$

$$R_s = \frac{A_r}{A_b + A_w + A_g + A_r} \tag{6}$$

式中の A_b，A_w，A_g，A_r は，それぞれ青色，白色，緑色，そして赤色の領域の面積を表す．R_c を適合率と呼ぶことにする．R_a は緑色の領域の割合，R_s は赤色の領域の割合である．10 通りのボールの初期座標に対して，提案手法で得た近似安全領域と，シミュレーション結果から得た安全領域を比較し，式 (4) から (6) により得られた各値の平均を表 1 に示す．表 1 を見ると，提案手法は十分高い近似精度を持つことが分かる．しかしながら，この手法には，図 3 の左上に見られる赤色の領域のように，まだ改善の余地が残されている．

Table 1: Rate of areas

R_c	R_a	R_s
0.951	0.013	0.036

3.2.2 計算時間

2 種類のコンピュータを用いて，提案手法の計算にかかる時間の測定を行った．表 2 に，10 通りのボールの初期座標に対する，守備ロボットが 2 台の時の提案手法による近似安全領域の計算時間と適合率 R_c の平均値を記す．

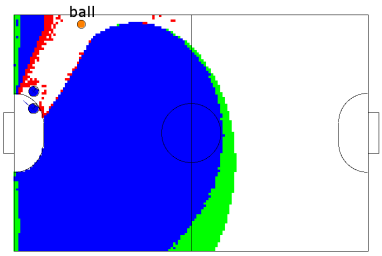

Figure 2: Safety region: case 1

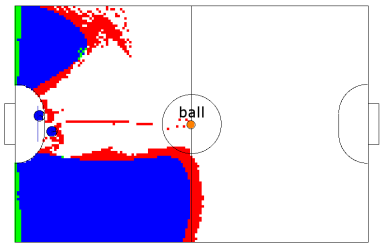

Figure 3: Safety region: case 2

Table 2: Computation time and coincidence rate : Coarse-to-fine method

N	computation time(msec)		coincidence
	Athron64 X2	Xeon 3.3GHz	rate R_c
don't use	10.8	6.6	0.950
1	8.5	4.5	0.951
2	3.1	1.4	0.951
4	1.4	0.56	0.951
8	0.94	0.37	0.951
16	0.84	0.32	0.951
32	0.78	0.30	0.951

3.3 考察

RoboCup 小型リーグにおいて，RoboDragons システムを用いてリアルタイムに安全領域指標に基づく戦略を用いる場合，安全領域の近似計算に与られる時間は 1 ミリ秒が限度であると考えられる．表 2 を見ると，Coarse-to-fine を用いることにより大幅に計算時間が短縮していることが分かる．$N \geq 8$ とすることで適合率を減らすことなく計算時間を 1 ミリ秒以下にすることができる．

表 2 の結果では，N の値を大きくしても適合率は変化していない．これは，ボールの初期位置が，3.1 節で記したように *Kick Off*，*Freekick* の開始位置から選ばれているからである．ボールの初期位置がディフェンスエリアのすぐ近くにある場合は N の値に注意しなければならない．図 4 と 5 は，N の値によって適合率が大きく変化する場合の一例である．この例では，N が 8 以上の時正しい安全領域が得られなくなっている．この事実は，安全領域指標を実際の守備戦略に組み込む際に注意を払わなくてはいけないということを示している．

4 安全領域指標に基づく戦略

4.1 相手のロボットの位置を考慮した守備戦略

文献[1]では，安全領域を用いた守備戦略（あるいは守備ロボットの配置アルゴリズム）を提案している．そのアルゴリズムでは，非安全領域に相手のロボットの位置を考慮して重みをつけている．その一例を図 6 から 8 に示す．図 6 は非安全領域（灰色の領域）と，3 台の相手ロボット（黄色の円），そして 2 台の守備ロボット（青色の円）を示している．図 7 は重み付けを行った非安全領域を示している．暗い領域であるほど重みが大きい領域となっている．そして，図 8 は図 7 の重みを付けた非安全領域を基に，文献[1]で提案したアルゴリズムに従って新たな守備ロボットを追加し，改めて安全領域を求めた結果である．図 8 を見ると，確かに非安全領域が大幅に減少していることが分かる．しかしながら，もしパスを受けてシュートを行うロボットが左上のコーナーにいるロボットだった場合，そのロボットは非安全領域内に存在するのでそのシュートを防ぐことができない．よって，最初にパスが行われる方向，つまりパスを行うロボットの向きも考慮する必要がある．次の節では，そのようなアルゴリズムを提案する．

4.2 相手のロボットの位置とパスの方向を考慮した守備戦略

実際の試合では，相手ロボットの位置だけでなく，最初にパスが行われる方向も重要な評価対象である．そのため，相手ロボットの位置に加えてパスが行われる方向の評価も考慮した，新たな重み付け非安全領域を提案する．以下では，パスを行うロボットの中心の座標からボールの座標へ引いた直線を “パスライン” と呼ぶ．

まず，安全領域/非安全領域の計算を行い，それから文献[1]と似た方法を使って重み付け非安全領域を計算する．以下の重み付け関数 $w(\mathbf{e})$ に従って，非安全領域内のすべての点 $\mathbf{e}$ に重みを付ける．

$$w(\mathbf{e}) \quad = \quad max(1, 100 \times (1 - \frac{||\mathbf{r_e} - \mathbf{e}||}{max(M, t_p \times v_r)})) \quad (7)$$

式中の $\mathbf{r_e}$ は相手ロボットの位置を，v_r は相手ロボットの移動速度を，そして t_p は式 (1) によって求められる時間である．M は, 相手ロボットの位置がボールに近い場合，重みを付与する範囲が狭くなる状況を回避するための値である．本実験においては，$M = 270$ とした．$w(\mathbf{e})$ は 1 から 100 の値を持つ．

次に，パスラインを考慮した重みを付ける．L をボールの位置と $\mathbf{e}$ を結ぶ直線とする．それから，パスラインと L の間の角度 θ を求める．θ が小さいほど，$w(\mathbf{e})$ の値

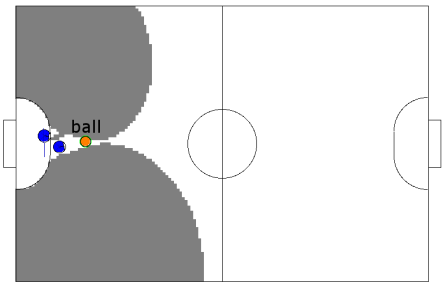

Figure 4: Coarse-to-fine method: N = 2

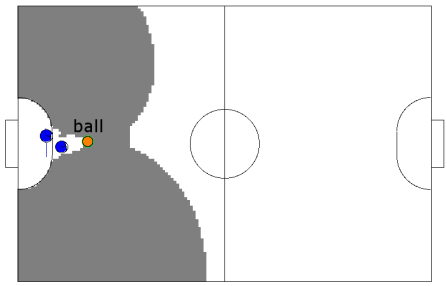

Figure 5: Coarse-to-fine method: N = 8

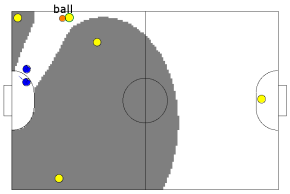

Figure 6: Safety/Unsafety region

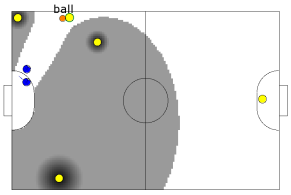

Figure 7: Weighted unsafety region

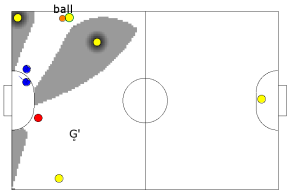

Figure 8: Safety region: After 3rd robot is placed

を重くするべきである．そこで，次の式のように改良した重み付け関数 $w'(\mathbf{e})$ を定義する．

$$w'(\mathbf{e}) = \begin{cases} w(\mathbf{e}) \times (max(1, 10 \times \cos(3 \times \theta))) & \\ & (|\theta| \leq \pi/6) \\ w(\mathbf{e}) & \\ & (|\theta| > \pi/6) \end{cases} \quad (8)$$

式 (8) によって，$|\theta|$ が $\pi/6$ ラジアン以下である $\mathbf{e}$ の重みをさらに大きくしている．$w'(\mathbf{e})$ は 1 から 1000 の値を持つ．この値は，$\mathbf{e}$ が $\mathbf{r_e}$ またはパスラインに近いほど大きな値となる．

$w'(\mathbf{e})$ によって重みを付けた非安全領域を基に，以下のアルゴリズムを用いて，$n+1$ 台目の守備ロボットの目標位置を求める．

1. 守備ロボットの座標を $\mathbf{r_i}(\mathbf{i} = \mathbf{1}...\mathbf{n})$ とし，安全領域と非安全領域を求める．非安全領域のまとまりごとに番号を付け，$N'_k(k = 1...)$ とする．
2. 全ての非安全領域内の各点に式 (8) によって重みを付与する．
3. 非安全領域のまとまり $N'_k(k = 1...)$ それぞれについて，各まとまり内のすべての点の重みの総和を求める．
4. 重みの総和が最大である非安全領域 N'_m の重心 $\mathbf{G'}$ を求める．
5. $\mathbf{G'}$ から守備するゴールの空き角度を求め，その角を二等分する線分上，かつディフェンスエリアラインに沿った位置を $n+1$ 台目の守備ロボットの目標位置 $\mathbf{r'_{n+1}}$ とする．

4.3 考察

図 6 と同じロボット及びボールの初期配置について上記の改良アルゴリズムを適用する．その結果，図 9 のような重み付け非安全領域が得られた．図 9 では，パスラインは左上の相手ロボットの方向へ向かっている．図 9 の重み付け非安全領域を基に 3 台目の守備ロボット（赤色の円）を先ほどの追加位置決定アルゴリズムに従って追加し，改めて非安全領域を計算すると図 10 のような結果が得られた．この図を見ると，提案戦略に従って追加された守備ロボットによって，連携シュートを行うであろう相手ロボットが安全領域内に包含されたことが分かる．これは，連携シュートを行う相手ロボットがシュートを決めることができないことを示しており，提案戦略が以前のものよりも有効な守備を行っていることを示している．

図 6 において，パスを行うロボットのみを動かして左上以外の相手ロボットにパスを行うようにしても，図 11 と 12 のようにより効率的な位置に守備ロボットを追加することができる．

新たに追加する守備ロボットの位置は **G'** によって決定しているということは重要である．例えば，もし，最初に 3 台目の守備ロボットが図 11 の位置に配置されていて，それから，パスを行うロボットが向きを変えて図 12 のように別のロボットに向けてパスを行おうとした場合，3 台目の守備ロボットは短い時間で図 12 で示される位置まで移動することができる．これは，図 11 及び 12 でパスラインの先にいる相手の 2 台のロボットがともに同じ非安全領域のまとまりの中に配置されており，**G'** がその 2 台のロボットの中央に存在することが理由である．これは

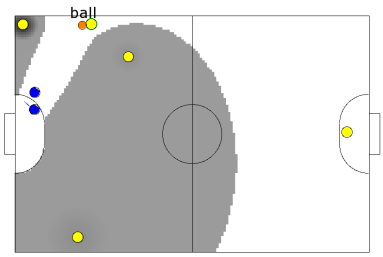

Figure 9: Weighted unsafety region using the proposed algorithm

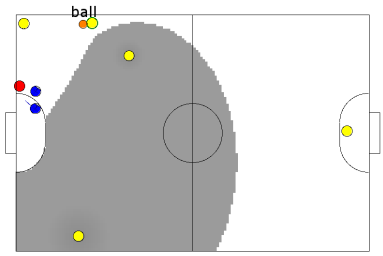

Figure 10: Safety region: After 3rd robot is placed by the proposed algorithm (case1)

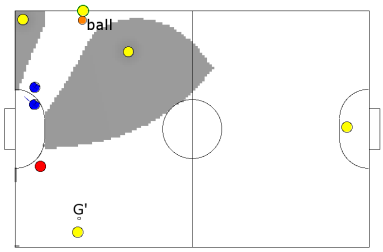

Figure 11: Safety region: case2

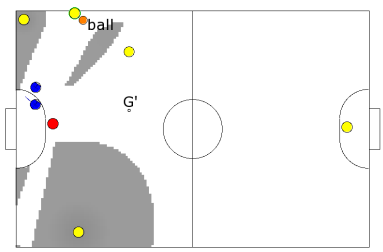

Figure 12: Safety region: case3

この配置アルゴリズムの大きな利点である．

5 おわりに

本論文では，RoboCup 小型ロボットリーグにおける局面評価のための指標として，安全領域を提案した．安全領域とは，その領域内から行われたシュートを，守備ロボットが守備戦略に従って動くことにより防ぐことが可能な領域として定義される．さらに，改良した近似安全領域の計算アルゴリズムを提案した．このアルゴリズムは 95% の計算精度を持ち，その計算時間は 1 ミリ秒以下である．これは RoboDragons のシステムで用いるのに十分なものである．さらに，安全領域を用いた，相手ロボットの位置及びパスの方向を考慮した守備戦略を提案し，それが守備ロボットの追加に有効であることを確かめた．

今後の課題として，提案手法の近似精度の更なる改善，計算時間の更なる短縮の検討，安全領域を用いた戦略の実装があげられる．

謝辞

この研究の一部は，愛知県立大学理事長特別研究費，学長特別研究費の支援を受けている．記して感謝する．

参考文献

[1] J. Maeno, A. Ishikawa, K. Murakami and T. Naruse “ Safety region: an index for evaluating the situation of RoboCup Soccer game”, 第 31 回人工知能学会 AI チャレンジ研究会資料, SIG-Challenge-B001-2, 2010, http://winnie.kuis.kyoto-u.ac.jp/sig-challenge/SIG-Challenge-B001/SIG-Challenge-B001.html

[2] R. Nakanishi, J. Bruce, K. Murakami, T. Naruse and M. Veloso, ”Cooperative 3-robot passing and shooting in the RoboCup Small Size League”, RoboCup 2006: Robot Soccer World Cup X, LNCS 4434 pp.418-425

[3] T. Laue, A. Burchardt, S. Fritsch, S. Hinz, K. Huhn, T. Kirilov, A. Martens, M. Miezal, U. Nehmiz, M. Schwarting and A. Seekircher “B-Smart (Bremen Small Multi Agent Robot Team) Extended Team Description for RoboCup 2009”, http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_b-smart.pdf

[4] S. Zickler, J. Bruce, J. Biswas, M. Licitra and M. Veloso“CMDragons 2009 Extended Team Description”, http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_cmdragons.pdf

[5] H. Achiwa, J. Maeno, J. Tamaki, S. Suzuki, T. Moribayasi, K. Murakami and T. Naruse “RoboDragons 2009 Extended Team Description”, http://small-size.informatik.uni-bremen.de/tdp/etdp2009/small_robodragons.pdf